



UNIVERSITY OF MISKOLC

MIKOVINY SÁMUEL DOCTORAL SCHOOL OF SCIENCES

Head of the Doctoral School:

Prof. Dr. Peter Szücs

**Estimation of Hydrogeological and Geotechnical Soil Parameters Using
Artificial Neural Networks**

PhD Thesis

Author:

Hasan Eteraf

Geotechnical Engineer

Supervisors:

Dr. Balázs Kovács

Dr. Viktória Mikita

Miskolc, 2024

DECLARATION

I, the undersigned.... Hasan Eteraf...., aware of my criminal responsibility hereby declare that the submitted PhD dissertation is the result of my own work, the literary references are clear and complete.

I uploaded my scientific publications to MTMT repository (MTMT ID: 10068351) I enclose the list of publications including the references in printed format.

Date: 29 August 2024

.....

signature

I, the undersigned ...Hasan Eteraf....,

hereby confirm that in Earth Science and Engineering field, I have no doctoral procedure in process, respectively my application for doctoral procedure had not been rejected within two years, furthermore I had no unsuccessful doctoral thesis defense within two years.

Furthermore, I declare that there is no procedure in process related to the withdrawal of my doctoral degree, respectively my doctoral degree had not been taken back from me within five years.

Date: 29 August 2024

.....

signature

ACKNOWLEDGEMENT

I have had the great good fortune to have received invaluable support and assistance from different people during my research, all of whom have made an important contribution to the successful completion of this research.

First and foremost, I want to say deep appreciation to my two supervisors. Aside from their role as scientific leaders, they have been extremely supportive in many aspects of my life. Their generous investment of time, valuable knowledge, and consistent dedication to my academic and personal development were critical in bringing my project to fruition. I owe them a debt of gratitude that I will never be able to fully compensate it.

I would also like to thank all of my colleagues and administrative staff at the institute. Their help, inspiration, and support have been invaluable throughout the process, providing the basis for my work.

Finally, my sincerest gratitude goes to my devoted family and friends, whose encouragement and spirit have been the foundation of my efforts throughout this study. Their efforts have been critical in overcoming the different obstacles encountered during this investigation.

Hasan Eteraf

Table of Contents

1- Introduction	1
1.1. Background	1
1.2. Problem statement.....	1
1.3. Research aims and significance	2
2- Literature Review	5
2.1. Geotechnical and hydrogeological concepts.....	5
2.1.1. Hydraulic conductivity.....	5
2.1.2. Soil classification	8
2.1.3. Shear strength parameters	11
2.2. Empirical correlations for soil parameter estimations	13
2.3. Regression analysis	15
2.4. Artificial Neural Network in Geotechnical Engineering	16
3- Methodology.....	21
3.1. Data Collection and Preparation	21
3.2. Experiments	22
3.2.1. Grain size distribution.....	22
3.2.2. Hydraulic conductivity.....	24
3.2.3. Shear strength parameters	24
3.3. Overview of data trend.....	24
3.4. Multiple linear regression	28
3.5. Artificial Neural Networks (ANNs).....	29
3.5.1. Concept of ANNs.....	29
3.5.2. Types of neural network.....	36
3.5.3. Implementation of artificial neural network to estimate soil parameters	41
4- Data analysis and results.....	43
4.1. Hydraulic conductivity values obtained by the experiments	43
4.2. Estimation of hydraulic conductivity by indirect method.....	44
4.3. Estimation of hydraulic conductivity by Multiple linear regression analysis.....	45
4.4. Artificial Neural Network model for hydraulic conductivity parameter	48

4.5. Artificial Neural Network model for cohesion parameter	60
4.6. Artificial Neural Network model for internal friction angle parameter	66
4.7. Simultaneous modeling of shear strength parameters	68
5- Theses	74
Thesis 1	74
Thesis 2	75
Thesis 3	76
Thesis 4	78
Thesis 5	78
6- Summary.....	80
List of research related to the dissertation	82
Bibliography	83
Annex 1. The percentage of soil components in each sample	I
Annex 2. Grain size distributions for all samples	III
Annex 3. Regression analysis results for purpose of choosing best ANN model for K	XXIV

LIST OF FIGURES

FIGURE 1. PARTICLE SIZE DISTRIBUTION CURVE (SOURCE: WWW.ELEMENTARYENGINEERINGLIBRARY.COM)	8
FIGURE 2. SOIL TEXTURE TRIANGLE (SOURCE: SOIL SURVEY DIVISION STAFF, 1993)	9
FIGURE 3. SHEPARD'S CLASSIFICATION SYSTEM (SOURCE: SHEPARD, 1954).	10
FIGURE 4. TRIANGULAR DIAGRAM OF WITHDRAWN STANDARD (SOURCE: EN ISO 14688-2:2004)	11
FIGURE 5. THE MOHR-COULOMB FAILURE CRITERIA AND MOHR'S CIRCLE (SOURCE: WWW. KHAAKPEY.IR).	12
FIGURE 6. COLLECTION SITE OF SOIL SAMPLE	21
FIGURE 7. LABORATORY TESTS CONDUCTED IN THE LAB.....	22
FIGURE 8. LOCATION OF A POINT IN PYRAMID (SOURCE: OWN COMPILATION)	25
FIGURE 9. THE SCHEMATIC PICTURE OF PYRAMID MODEL (SOURCE: OWN COMPILATION)	25
FIGURE 10. LOCATION OF ALL POINTS IN PYRAMID (SOURCE: OWN COMPILATION)	26
FIGURE 11. HYDRAULIC CONDUCTIVITY VALUES DISTRIBUTION (SOURCE: OWN COMPILATION).....	26
FIGURE 12. DIFFERENT PLANES OF PYRAMID BY CHANGING IN GRAVEL CONTENTS (SOURCE: OWN COMPILATION)	27
FIGURE 13. A SIMPLE NEURAL NETWORK STRUCTURE (SOURCE: WWW.MEDIASOFT.IR).....	29
FIGURE 14. A BIOLOGICAL NEURON (WWW. HAMRUYESH.COM).....	30
FIGURE 15. A CONVENTIONAL MATHEMATICAL MODEL OF NEURONS (SOURCE: SUBHASHINI, 2020)	31
FIGURE 16. THE BINARY STEP FUNCTION (SOURCE: WWW.ENCORD.COM).	32
FIGURE 17. LINEAR ACTIVATION FUNCTION (SOURCE: WWW.ENCORD.COM).....	33
FIGURE 18. SIGMOID, LOGISTIC ACTIVATION FUNCTIONS (SOURCE: WWW.ENCORD.COM).....	33
FIGURE 19. HYPERBOLIC TANGENT FUNCTION (SOURCE: WWW.ASHUTOSHTRIPATHI.COM)	34
FIGURE 20. RELU ACTIVATION FUNCTION (SOURCE: WWW.ENCORD.COM)	34
FIGURE 21. NON-LINEAR ACTIVATION FUNCTIONS (SOURCE: WWW.ENCORD.COM)	35
FIGURE 22. DERIVATIVE OF ACTIVATION FUNCTIONS (SOURCE: WWW.TOWARDSDATASCIENCE.COM)	36
FIGURE 23. SCHEMATIC DIAGRAM OF A PERCEPTRON (SOURCE: WWW.MEDIUM.COM)	36
FIGURE 24. SCHEMATIC DIAGRAM OF FEEDFORWARD NEURAL NETWORKS (SOURCE: WWW.OKSIM.UA)	37
FIGURE 25. SIMPLE CNN ARCHITECTURE (SOURCE: WWW.GEEKSFORGEESKS.ORG)	38
FIGURE 26. RECURRENT NEURAL NETWORK	38
FIGURE 27. MODULAR NEURAL NETWORK (SOURCE: WWW.WIKIDOCX.NET)	39
FIGURE 28. A MOSTLY COMPLETE CHART OF NEURAL NETWORKS (SOURCE: WWW.ASIMOVINSTITUTE.ORG)	40
FIGURE 29. (-LOGK) VALUES OBTAINED BY LABORATORY TESTS (SOURCE: OWN COMPILATION)	43
FIGURE 30. COMPARISON OF (-LOGK) VALUES OBTAINED BY EMPIRICAL FORMULAE AND LABORATORY TESTS (SOURCE: OWN COMPILATION)	45
FIGURE 31. RELATIONSHIPS BETWEEN THE INDEPENDENT VARIABLES AND THE DEPENDENT VARIABLE (SOURCE: OWN COMPILATION)....	46
FIGURE 32. MULTIPLE LINEAR REGRESSION ANALYSIS RESULT (SOURCE: OWN COMPILATION).....	48
FIGURE 33. REGRESSION PLOTS OF THE BEST-PERFORMING MODEL FOR (-LOG K) PREDICTION (SOURCE: OWN COMPILATION)	50
FIGURE 34. COEFFICIENT OF DETERMINATION VALUE FOR THE BEST MODEL OF ANN (SOURCE: OWN COMPILATION)	51
FIGURE 35. MEAN SQUARED ERROR FOR ALL MODELS (SOURCE: OWN COMPILATION).....	51
FIGURE 36. DISTRIBUTION OF RESIDUALS (ERRORS) (SOURCE: OWN COMPILATION)	52
FIGURE 37. MATLAB CODE FOR IMPORTING EXPERIMENTAL DATA FROM AN EXCEL FILE (SOURCE: OWN COMPILATION).....	52
FIGURE 38. NEURAL NETWORK STRUCTURE FOR (-LOG K) PREDICTION (SOURCE: OWN COMPILATION).....	53
FIGURE 39. MATLAB CODE FOR DEFINING AND TRAINING THE NEURAL NETWORK USING THE FEEDFORWARD FUNCTION (SOURCE: OWN COMPILATION)	53
FIGURE 40. MATLAB CODE FOR DERIVING WEIGHTS, BIASES AND ACTIVATION FUNCTIONS. (SOURCE: OWN COMPILATION).....	55
FIGURE 41. MATLAB CODE FOR DEFINING USER INPUT AND ESTIMATING THE OUTPUT PARAMETER. (SOURCE: OWN COMPILATION) ...	56
FIGURE 42. MATLAB CODE FOR VISUALIZING THE RESULTS OF THE ANN MODEL (SOURCE: OWN COMPILATION)	56
FIGURE 43. COMPARISON OF (- LOG K) VALUES OBTAINED BY ANN AND EXPERIMENTS (SOURCE: OWN COMPILATION)	57
FIGURE 44. MATLAB CODE TO CALCULATE THE ROOT MEAN SQUARED ERROR (RMSE). (SOURCE: OWN COMPILATION)	57
FIGURE 45. RELATIONSHIP BETWEEN THE WEIGHT PERCENTAGE OF SAND AND HYDRAULIC CONDUCTIVITY VALUE (SOURCE: OWN COMPILATION).	58

FIGURE 46. RELATIONSHIP BETWEEN THE WEIGHT PERCENTAGE OF SILT AND HYDRAULIC CONDUCTIVITY VALUE (SOURCE: OWN COMPILATION).	58
FIGURE 47. RELATIONSHIP BETWEEN WEIGHT PERCENTAGE OF CLAY AND GRAVEL AND HYDRAULIC CONDUCTIVITY VALUE (SOURCE: OWN COMPILATION).	58
FIGURE 48. EXAMPLE OUTPUT OF THE MATLAB PROGRAM FOR (-LOG K) PREDICTION (SOURCE: OWN COMPILATION).	59
FIGURE 49. COHESION VALUES OBTAINED BY LABORATORY TESTS (SOURCE: OWN COMPILATION).	61
FIGURE 50. INTERNAL FRICTION ANGLE VALUES OBTAINED BY LABORATORY TESTS (SOURCE: OWN COMPILATION).	61
FIGURE 51. NEURAL NETWORK STRUCTURE FOR COHESION PREDICTION (SOURCE: OWN COMPILATION).	62
FIGURE 52. REGRESSION PLOTS OF THE BEST-PERFORMING MODEL FOR COHESION PREDICTION (SOURCE: OWN COMPILATION).	62
FIGURE 53. COMPARISON OF COHESION VALUES OBTAINED BY ANN AND EXPERIMENTS (SOURCE: OWN COMPILATION).	65
FIGURE 54. EXAMPLE OUTPUT OF THE MATLAB PROGRAM FOR COHESION PREDICTION (SOURCE: OWN COMPILATION).	65
FIGURE 55. NEURAL NETWORK STRUCTURE FOR PREDICTION OF INTERNAL FRICTION ANGLE (SOURCE: OWN COMPILATION).	66
FIGURE 56. REGRESSION PLOTS OF THE BEST-PERFORMING MODEL FOR PREDICTION OF INTERNAL FRICTION ANGLE (SOURCE: OWN COMPILATION).	66
FIGURE 57. COMPARISON OF INTERNAL FRICTION ANGLE VALUES OBTAINED BY ANN AND EXPERIMENTS (SOURCE: OWN COMPILATION).	67
FIGURE 58. EXAMPLE OUTPUT OF THE MATLAB PROGRAM FOR PREDICTION OF INTERNAL FRICTION ANGLE (SOURCE: OWN COMPILATION).	68
FIGURE 59. NEURAL NETWORK STRUCTURE FOR SIMULTANEOUS MODEL OF SHEAR STRENGTH PARAMETERS (SOURCE: OWN COMPILATION).	69
FIGURE 60. PERFORMANCE EVALUATION OF ANN MODEL (SOURCE: OWN COMPILATION).	70
FIGURE 61. COMPARISON OF COHESION VALUES OBTAINED BY ANN AND LABORATORY TESTS (SOURCE: OWN COMPILATION).	71
FIGURE 62. COMPARISON OF FRICTION ANGLE VALUES OBTAINED BY ANN AND LABORATORY TESTS (SOURCE: OWN COMPILATION).	72
FIGURE 63. COMPARISON OF RESULTS OBTAINED BY ANN AND LABORATORY TESTS (SOURCE: OWN COMPILATION).	74
FIGURE 64. SIMPLIFIED INPUT AND OUTPUT BOX OF THE PROGRAM (SOURCE: OWN COMPILATION).	75
FIGURE 65. PLOT OF RESIDUALS VS. PREDICTED VALUES (SOURCE: OWN COMPILATION).	76
FIGURE 66. ESTIMATION ACCURACY OF DIFFERENT MODELS (SOURCE: OWN COMPILATION).	77
FIGURE 67. EFFECT OF EACH COMPONENT ON OUTPUT (SOURCE: OWN COMPILATION).	78
FIGURE 68. NEURAL NETWORK STRUCTURE FOR SIMULTANEOUS MODEL OF SHEAR STRENGTH PARAMETERS (SOURCE: OWN COMPILATION).	79
LIST OF TABLES	
TABLE 1. CLASSIFICATION OF SOIL SAMPLES.	23
TABLE 2. DESCRIPTIVE STATISTICS.	45
TABLE 3. VARIABLES CORRELATIONS OF MLR MODEL.	47
TABLE 4. MLR MODEL SUMMARY.	47
TABLE 5. ANALYSIS OF VARIANCE OF MLR MODEL.	47
TABLE 6. ANN MODELS FOR HYDRAULIC CONDUCTIVITY PREDICTION.	50
TABLE 7. COMPARISON BETWEEN SEVERAL STUDIES FOR OBTAINING HYDRAULIC CONDUCTIVITY.	59
TABLE 8. COMPARISON BETWEEN SEVERAL STUDIES FOR OBTAINING SHEAR STRENGTH PARAMETERS.	72
TABLE 9. COMPARISON OF SIMULTANEOUS MODEL AND SEPARATE MODELS.	79

1- Introduction

1.1. Background

The complex characteristics of the ground's soil layers are fundamental to the safety and success of many geotechnical engineering projects (Das, 2017). To do this, precise measurements of the properties of the soil that govern the behavior of the ground beneath our feet are crucial (Ameratunga, 2016). In the field of geotechnical practice, soil permeability—which is commonly evaluated as hydraulic conductivity—and strength characteristics are key components that have a major influence on the stability and operation of structures (Han, 2020).

Critical to many engineering applications, hydraulic conductivity is especially important for hydrogeology, geotechnical engineering, and groundwater management. Engineers need to understand hydraulic conductivity in order to evaluate the flow of water through soil, which is necessary for managing groundwater resources, building efficient drainage systems, and assessing the possibility of pollutant transfer (Gao, 2024). For instance, accurate measurement of hydraulic conductivity aids in assessing the viability and effectiveness of different remediation methods, such as groundwater extraction and treatment, in groundwater remediation projects. Hydraulic conductivity is also useful in irrigation management for streamlining water distribution networks and guaranteeing correct irrigation of crops while reducing water waste (Gupta, 2024).

Apart from that, a key soil characteristic that controls the safety and stability of civil engineering structures, such as retaining walls, slopes, and foundations, is shear strength (Hu, 2020). Shear strength is a crucial factor in the design of geotechnical engineering since it indicates the soil's resistance to deformation and failure under applied loads (Vanapalli, 2009). For example, in foundation engineering, understanding the parameters of shear strength is crucial to ascertaining the soil's carrying capability, which in turn affects the design of both shallow and deep foundations. Similar to this, in slope stability analysis, shear strength knowledge aids engineers in determining whether naturally occurring and artificially constructed slopes are stable, reducing the possibility of landslides and other slope collapses that could jeopardize infrastructure and human life.

1.2. Problem statement

Several direct methods exist for measuring hydraulic conductivity and shear strength parameters of soil. For hydraulic conductivity, these include the constant head permeability test, falling head permeability test, packer test, slug test, and pumping test. Also, for shear strength, the methods include the direct shear test, triaxial compression test, vane shear test, and unconfined compression test. However, due to the significant dependence on in situ and laboratory testing techniques, conventional methods for measuring these parameters frequently prove to be laborious and resource-intensive (Hicher, 1996). The installation of monitoring wells for hydraulic conductivity testing and the gathering of undisturbed soil samples for shear strength testing are two common fieldwork requirements for these technologies (Craig, 2004). While laboratory and in-situ measurements provide high safety and accuracy, they can be time-consuming and resource-intensive. Finding a way to estimate these parameters with an acceptable safety index and reduced time requirements would be advantageous, as it would help mitigate delays and lower project costs. Furthermore, it can be difficult to get representative samples due to the inherent regional heterogeneity in soil properties, which can result in inaccuracies in the measured data (Dołęgowska, 2016). Because of this, engineers frequently struggle to precisely describe the

behavior of soil and forecast how it will react to various loading scenarios. The challenges in obtaining accurate measurements are made more difficult by the dependence on sophisticated equipment and specialist knowledge. Accurate performance and interpretation of test findings are crucial for in situ methods like cone penetration test and borehole permeability test, which call for specialized equipment and experts with advanced training (Elhakim, 2016). Similar to this, in order to guarantee accurate results, laboratory tests for determining shear strength—such as direct shear tests and triaxial compression tests—require specific equipment and skilled personnel. The expense and complexity of testing are increased by the requirement for specialized tools and knowledge, which also restricts the methods' applicability in distant or resource-constrained places where these resources can be hard to come by or unavailable (Nam, 2011). Additionally, the time-consuming nature of traditional testing methods poses significant challenges in project planning and execution. Delays in obtaining test results can impede decision-making processes, leading to uncertainties in design parameters and construction schedules. In fast-paced construction environments, where timely decisions are crucial to project success, the prolonged testing procedures associated with traditional methods can hinder progress and increase project risks (Viana da Fonseca, 2015). As a result, there is a growing demand for alternative approaches that offer faster, more cost-effective, and reliable solutions for estimating soil parameters, such as hydraulic conductivity and shear strength.

It is noteworthy to mention that getting soil samples for lab testing might be especially difficult in some situations, such as historical sites or places with restricted access. Strict preservation guidelines are frequently in place for historical sites, limiting the amount of infrastructure and soil disturbance (Prieto-Taboada, 2014). In these situations, typical sample techniques cannot be practical, forcing engineers to use non-intrusive approaches to indirectly infer soil parameters, including geophysical surveys or remote sensing technology (Cozzolino, 2018). Although these techniques provide insightful information on subsurface conditions, they might not be as accurate or comprehensive as direct sampling and testing. Moreover, in the preliminary design phase of engineering projects, there is often a need for quick and reliable data to inform decision-making and design optimization. Laboratory testing methods, which involve sample collection, transportation, preparation, and analysis, can be time-consuming and may not align with the fast-paced nature of preliminary design processes (Sharma, 2021). Engineers face the challenge of balancing the need for comprehensive soil characterization with the time constraints of project schedules. Consequently, there is a growing demand for innovative techniques that can rapidly estimate soil parameters with minimal time and resource requirements, allowing for more efficient and informed decision-making during the early stages of project development.

1.3. Research aims and significance

To address the limitations of conventional approaches for measuring soil parameters, several scientists and researchers have attempted to create new strategies that use easily obtainable soil properties for estimation. Grain size distribution is one such feature that is frequently easy to find in any engineering project. Grain size distribution is a key indicator of soil's hydraulic conductivity and shear strength, offering important details about the composition and structure of the material (Belkhatir, 2013). Because of this, scientists have looked into a number of methods for estimating these characteristics using information on grain size.

For estimating soil characteristics from grain size distribution, empirical approaches have been frequently used. Based on trends seen in empirical data, these methods create connections between hydraulic conductivity or shear strength and grain size measurements (Meskini-Vishkaee, 2018). Although empirical methods are straightforward and simple to apply, they might not be as robust and broadly applicable as models built on larger datasets.

In addition to empirical approaches, researchers have employed statistical methods such as regression analysis to estimate soil parameters from grain size data. Regression models are developed by fitting mathematical equations to empirical data, allowing for the quantification of relationships between independent variables and dependent variables (Nemes, 2004). By analyzing large datasets of soil samples with known parameters, regression models can identify statistically significant correlations and derive predictive equations for parameter estimation. However, the accuracy of regression-based models depends heavily on the quality and representativeness of the training data, as well as the appropriateness of the chosen model structure (Klein, 1999).

The introduction of artificial intelligence (AI) has transformed the estimation of soil parameters based on grain size distribution, providing sophisticated methodologies that meet the issues associated with older methods (Khalili-Maleki, 2022). Among the many AI techniques, artificial neural networks (ANNs) have received a lot of attention due to their capacity to manage the complex and nonlinear correlations that exist in soil data. ANNs are computational models based on the structure and function of the human brain, with interconnected nodes (neurons) structured in layers (Citakoglu, 2017). These networks excel in extracting patterns and correlations from huge datasets, making them ideal for predicting soil characteristics based on grain size distributions. One of the primary advantages of utilizing ANNs for soil parameter estimation is their ability to capture complex correlations between input variables and output parameters. This versatility allows ANNs to account for complex soil behavior and environmental influences that simpler models may not fully capture. Moreover, ANNs demonstrate resilience and flexibility in the face of noisy or imperfect datasets, which are frequent problems in geotechnical engineering research (Park, 2011). ANNs are able to produce reliable predictions even in the face of uncertainty by adapting over time and learning from past events, which allows them to overcome data constraints (Prieto, 2016). Additionally, ANNs are scalable, which makes it possible to include a variety of input factors and ambient variables in the modeling process. This adaptability improves soil parameter estimations' accuracy and comprehensiveness, enabling researchers and engineers to make well-informed decisions in a range of geotechnical applications (Bolón-Canedo, 2011). It is noteworthy to mention that the faster estimates provided by ANNs also enhance risk management by allowing for earlier identification of potential issues, enabling more proactive and effective mitigation strategies, and ultimately supporting project safety and success.

In this study, I attempted to estimate the hydraulic conductivity of soil and shear strength parameters based on grain size distribution using artificial neural networks. Recognizing the challenges associated with traditional methods and the potential of ANNs to overcome them, a comprehensive dataset comprising laboratory-tested soil samples was collected. Subsequently, the collected data were utilized to train and validate ANN models for predicting hydraulic conductivity and shear strength parameters. Also, the performance of the ANN models was compared to other

indirect methods. By evaluating the accuracy and reliability of ANN-based predictions compared to conventional methods, this study aims to highlight the effectiveness of artificial neural networks in soil parameter estimation. Additionally, it seeks to demonstrate how ANNs can advance geotechnical engineering practices by enhancing safety management and reducing risk levels.

2- Literature Review

2.1. Geotechnical and hydrogeological concepts

This section emphasizes the fundamental geotechnical and hydrogeological concepts needed for this study.

2.1.1. Hydraulic conductivity

The concept that controls the flow of fluid in a particular substance is stated in Darcy's law. Darcy's law equation explains how a liquid can pass through any type of porous material, such as a rock. The flow between two points is exactly proportional to the pressure differential, distance, and connectivity of flow within rocks between the points, which forms the basis of the law. Permeability is the measurement of interconnectivity. Equation 1, Darcy's law, describes the flow of a fluid in a porous material (Darcy, 1856).

$$Q = -K \cdot A \cdot \frac{dh}{dl} \quad (\text{Eq.1})$$

Wherein: Q is the rate of water flow [m³/s], K is the hydraulic conductivity [m/s], A is cross-sectional area perpendicular the flow direction [m²], and dh/dl indicates a hydraulic gradient [-].

The first and most important parameter chosen for investigation in this study is the hydraulic conductivity. The ease with which water can permeate rock or soil is known as hydraulic conductivity. Hydraulic conductivity is also called coefficient of permeability in geotechnical terms which is reduced sometimes to permeability. Water flow across soils can be best understood by measuring hydraulic conductivity. A precise determination of hydraulic conductivity is necessary for numerous applications in either steady-state or transient scenarios. For example, evaluating the spatial and temporal variations in groundwater in response to changes in stream elevation is crucial for groundwater storage and recharge (Kamp, 2001). Also, in order to better understand slope stability, slopes are being evaluated more often in response to rain events or shifting water levels (Dou, 2014). In addition, hydraulic conductivity is necessary for the environmental analysis of the movement and removal of hazardous viruses or contaminants (Gupta, 2024). The porosity of soil, or the percentage of void space between the particles, is directly associated with its permeability. The soil's permeability and porosity increase with increasing particle size (Aimrun, 2004). As a result, soil that has a higher percentage of coarse particles is more permeable than soil that contains a higher percentage of tiny particles.

The process of measuring hydraulic conductivity is sometimes complex, leading to its frequent estimation through various methods. These methods include grain size analysis techniques or experimental procedures, both in situ and in laboratory settings. Direct laboratory measurement involves tests like the constant head permeability and falling head permeability methods, each suited for different soil types. Constant head approach is recommended when there is a large amount of liquid discharge through the sample and a coarse granular soil, like sand. Equation 2 is used to compute hydraulic conductivity via constant head method (European Committee for Standardization, 2007).

$$K = \frac{Q.L}{A.\Delta h.\Delta t} \quad (\text{Eq.2})$$

Where:

- K = Hydraulic conductivity (cm/s or m/s)
- Q = Discharge flow rate (cm³/s or m³/s)
- L = Length of the soil specimen (cm or m)
- A = Cross-sectional area of the soil specimen (cm² or m²)
- Δh = Head loss (cm or m)
- t = Time duration (s)

The falling head permeability test is a common laboratory method used to determine the hydraulic conductivity of soils, particularly those with low to intermediate permeability, such as fine-grained soils including clays and silts. In this test, the flow of water through a soil sample is measured as the water level within a standpipe installed on top of the sample gradually decreases over time. By monitoring the rate of water flow and the change in water level, the hydraulic conductivity of the soil can be calculated using established formulas. This method provides valuable insights into the soil's ability to transmit water and is essential for various geotechnical and hydrogeological applications. Equation 3 is used for the calculation of hydraulic conductivity through falling head method (European Committee for Standardization, 2007).

$$K = \frac{\alpha.L}{A.t} \ln\left(\frac{h_1}{h_2}\right) \quad (\text{Eq.3})$$

Where:

- K = Hydraulic conductivity (cm/s or m/s)
- L = Length of the soil specimen (cm or m)
- α = Cross-sectional area of standpipe
- h_1 = Initial head (cm or m)
- h_2 = Final head (cm or m)
- A = Cross-sectional area of the soil specimen (cm² or m²)
- t = Time duration (s)

Additionally, indirect methods also are available to estimate hydraulic conductivity. For instance, there is the possibility to calculate hydraulic conductivity through consolidation tests. These tests are used, where factors like consolidation coefficient (C_v) are crucial in predicting settlement time and soil strength. The e-log t curve method proposed by Casagrande and Fadum

is commonly employed to determine C_v as shown in equation 4, which in turn can be used to calculate the permeability coefficient numerically at each stage of the consolidation test (Zeng, 2020).

$$K = c_v \cdot \gamma_w \cdot mv \quad (\text{Eq.4})$$

where c_v is the consolidation coefficient (m^2/s), mv is the coefficient of compressibility (kPa^{-1}), γ_w is the unit weight of water (kN/m^3).

Following laboratory methods, in situ tests like the pumping test and slug test are essential for measuring hydraulic conductivity directly in the field. These methods capture the natural soil conditions, accounting for heterogeneity and anisotropy that lab tests might miss. The pumping test, which involves water extraction from a well and monitoring surrounding water levels, is useful for large-scale aquifers. In contrast, the slug test provides a quicker, simpler approach by observing water level recovery after a sudden water addition or removal in a well, making it ideal for coarse-grained soils and fractured rock. Other in situ methods include the packer test (Lugeon test), which is effective in fractured rock environments, and permeameter tests that measure permeability in situ, particularly for sands and gravels. The Cone Penetration Test (CPT) with Piezocone offers high-resolution profiles of hydraulic conductivity by analyzing pore water pressure dissipation during soil penetration, though it requires expert interpretation. In situ infiltration tests, such as the double-ring infiltrometer, assess near-surface soils' ability to transmit water, although these are generally limited to surface layers.

Despite their advantages, in situ tests for hydraulic conductivity come with several limitations that can impact their applicability and accuracy. One significant challenge is the need for specialized equipment and highly trained personnel, which can drive up costs and make these methods less accessible, particularly in remote or resource-constrained locations. The logistics of setting up and maintaining test sites can also be complex and time-consuming, often requiring careful site preparation and monitoring over extended periods. Additionally, in situ tests are sensitive to external factors, such as variations in groundwater levels, the influence of nearby wells, and boundary conditions that can introduce uncertainties into the results. For example, in pumping tests, the presence of impermeable boundaries or other nearby pumping activities can skew the observed drawdown, leading to inaccurate estimates of hydraulic conductivity. Similarly, slug tests, while quick and convenient, may not provide sufficiently detailed data, particularly in heterogeneous or low-permeability soils, where the recovery response might be too slow or complex to interpret accurately. Furthermore, environmental factors such as temperature fluctuations, seasonal variations, and soil moisture content can affect the test results, making it difficult to obtain consistent and reliable data. In some cases, the physical disturbance caused by the test itself, such as borehole drilling or water injection, can alter the natural soil structure and properties, leading to misleading results.

So, developing a model that can be continuously updated and refined with new data would significantly enhance the ability to obtain critical design parameters for geotechnical projects. Such a dynamic model would drastically reduce the time required to acquire accurate soil parameters, enabling faster decision-making and project progression. As the model learns from an increasing amount of data, its predictions would become more accurate, thereby reducing the

uncertainties typically associated with traditional methods. Moreover, by minimizing the need for extensive field testing and reducing reliance on costly and time-consuming laboratory tests, this approach would lower the overall costs of geotechnical investigations. Ultimately, a model that can be continuously reconstructed with new data would streamline the process of obtaining design data, leading to safer, more economical, and faster project execution.

2.1.2. Soil classification

A complex natural material, soil is made up of varied sized and shaped mineral and organic particles. A variety of techniques, such as grain size analysis, can be used to ascertain the content, structure, and texture of soil in order to establish its physical and chemical qualities. The distribution of particle sizes can have a big impact on how soils behave in certain applications (Santamarina, 2004). Mineral particles make up soils, and how these particles are arranged and sized greatly affects the soil's characteristics. A single soil sample can contain particles that vary in size, from very coarse, exceeding 100 millimeters, to very small, less than 2 microns (Gee, 2002). The permeability, porosity, compressibility, shear strength, and settling velocity of soil are among the other qualities that are greatly influenced by the grain size distribution of soil particles. Soils comprising larger particles exhibit greater strength attributed to heightened inter-particle friction, whereas finer soils demonstrate increased sensitivity to variations in water content (Eteraf, 2023). Understanding the particle size distribution of a soil enables us to predict its geotechnical and hydrogeological properties to make more informed decisions regarding its utilization. As depicted in figure 1, Soils can be classified by reference to particle size as follows (Opsal, 2018).

- Clay – less than 0.002mm
- Silt – 0.002mm to 0.063mm
- Sand – 0.063mm to 2mm
- Gravel – 2mm to 63mm
- Cobbles - 63mm to 200mm
- Boulders – greater than 200mm

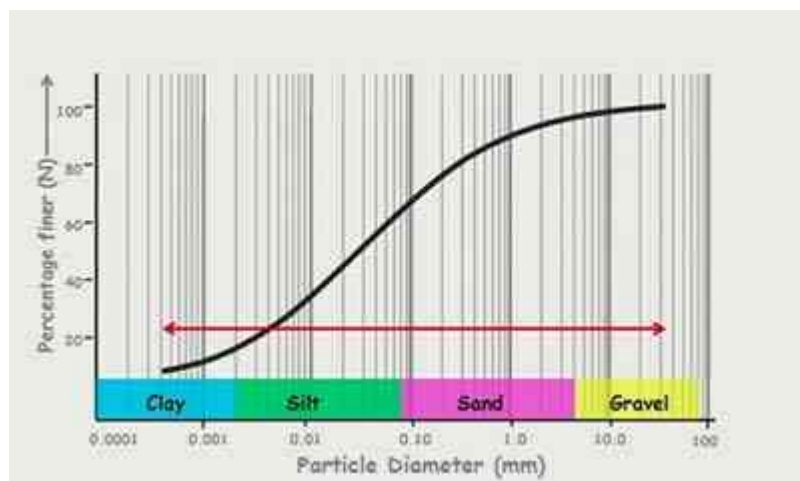


Figure 1. Particle Size Distribution Curve (Source: www.elementaryengineeringlibrary.com)

Soil texture was considered the most significant soil attribute in establishing hydraulic parameters in the majority of earlier research. But one thing that has been considered recently more than it used to be, is the disregard for gravel particles. Many building projects employ gravel soils as their basis, particularly fine gravel. Consequently, it is impossible to overlook the significance of gravel influence in defining soil qualities (Lu, 2021).

One of the most common methods for characterizing soils is through their textural characteristics. It is common practice to describe the fractional amount of different grain size particles in a soil using the relative fractions of sand, silt, and clay content. A texture-based classification technique is used to group soils into classes based on their soil texture (Twarakavi, 2010). Probably the most well-known is the soil texture classification scheme recommended by the USDA (Abdelfattah, 2007). The USDA has classified soil textures into 12 categories. A helpful tool for categorizing soil texture is the ternary diagram. Figure 2 depicts a ternary diagram of the soil texture triangle using soil texture classifications based on the USDA. The ternary diagram illustrates the different types of soil texture in relation to the percentages of sand, silt, and clay (Barman, 2020).

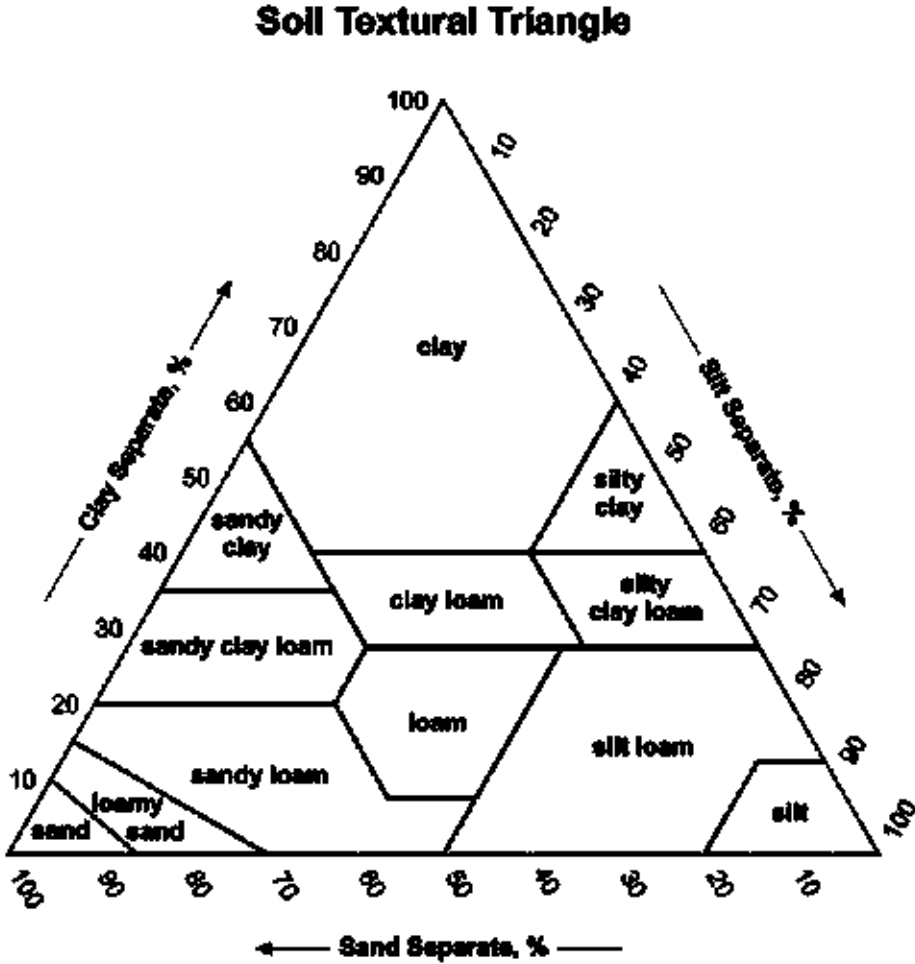


Figure 2. Soil Texture Triangle (Source: Soil Survey Division Staff, 1993)

Another ternary diagram that was presented in 1954 is based on the ratios of silt, sand, and particles the size of clay (Shepard, 1954). Ternary diagrams, like the one created by Shepard, are useful tools for illustrating a three-component system whose total sum equals 100 percent. According to Figure 3, the components stand for the proportions of clay, silt, and sand in a sediment sample. Each sediment sample's particle size distribution determines whether it is plotted as a point inside the diagram or along its edges. In order to characterize sediment samples, Shepard divided the ternary diagram into ten pieces. According to Shepard's standards, 'Silty Sands' and 'Sandy Silts' have less than 20% clay-sized particles, whereas 'Clays' have at least 75% clay-sized particles. 'Sand-Silt-Clays' has a minimum of 20% of each constituent. For every one of the ten categories, there are detailed descriptions available in the metadata linked to the dataset that was used to create the sediment distribution map.

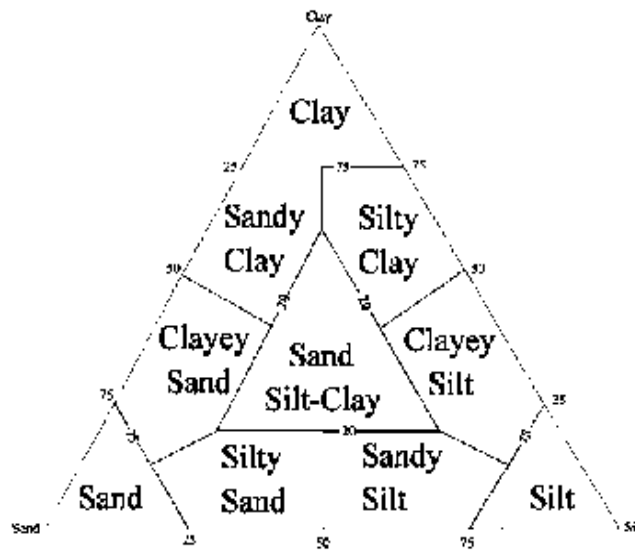


Figure 3. Shepard's Classification System (Source: Shepard, 1954).

The USDA and Shepard soil classification methods, both widely used in soil analysis, each offer distinct approaches but also share common limitations. The USDA method can oversimplify soil textures and fail to account for variations in soil properties at different depths or locations, and it does not consider gravel content. The Shepard method, which also uses a triangular diagram but includes additional categories for specific soil textures, provides a more detailed classification. Yet, its complexity and the increased number of categories can make it less intuitive and more difficult to apply in field settings. Overall, while both methods offer valuable insights into soil texture, their limitations in addressing engineering parameters and their complexity in practical applications can restrict their suitability for detailed engineering purposes.

Additional methods that incorporate soil behavior and engineering properties is the Eurocode 7 method. The Eurocode 7 standard method for soil classification, outlined in EN 1997-2:2007, provides a comprehensive framework for assessing soil properties relevant to geotechnical engineering. According to figure 4, this method integrates soil texture classification with

engineering properties, incorporating not only the fractions of sand, silt, and clay but also explicitly accounting for the gravel fraction.

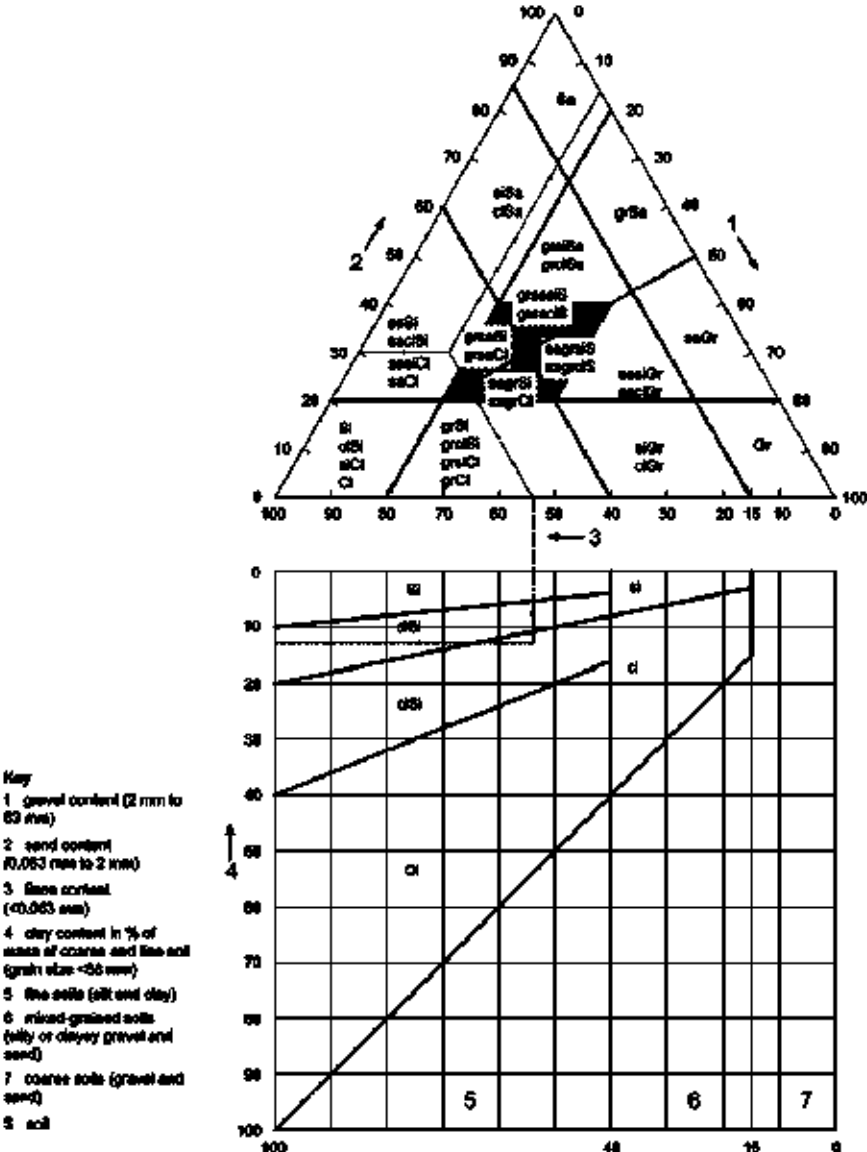


Figure 4. Triangular Diagram of Withdrawn Standard (Source: EN ISO 14688-2:2004)

2.1.3. Shear strength parameters

Understanding the shear strength of soil is paramount in geotechnical engineering, as it directly influences the stability and performance of various structures and infrastructures (Vanapalli, 2009). Shear strength represents the soil's resistance to internal friction and cohesion, both critical parameters in determining its stability under different loading conditions. Cohesion refers to the soil's ability to resist shear stress even in the absence of normal stress, while the internal friction angle characterizes the soil's resistance to shear deformation under applied

normal stress. These parameters are crucial in assessing the bearing capacity of foundations, the stability of slopes and embankments, and the design of retaining structures (Terzaghi, 1943).

Grain size distribution plays a significant role in determining soil shear strength. Finer-grained soils tend to exhibit higher cohesion due to increased surface area and particle bonding, while coarser-grained soils typically have higher internal friction angles due to interlocking particle arrangements. Therefore, understanding the grain size distribution of soil is essential in predicting its shear strength behavior accurately (Saravanan, 2020).

The Mohr-Coulomb failure criterion, as shown in figure 5, is a fundamental concept in geotechnical engineering used to describe the failure behavior of soils and rocks under shear stress. It defines the conditions under which a material will fail, based on the relationship between shear stress and normal stress. According to the Mohr-Coulomb theory, failure occurs when the shear stress on a plane reaches a critical value determined by the cohesion and internal friction angle of soil. The criterion is expressed by the equation shown in picture.

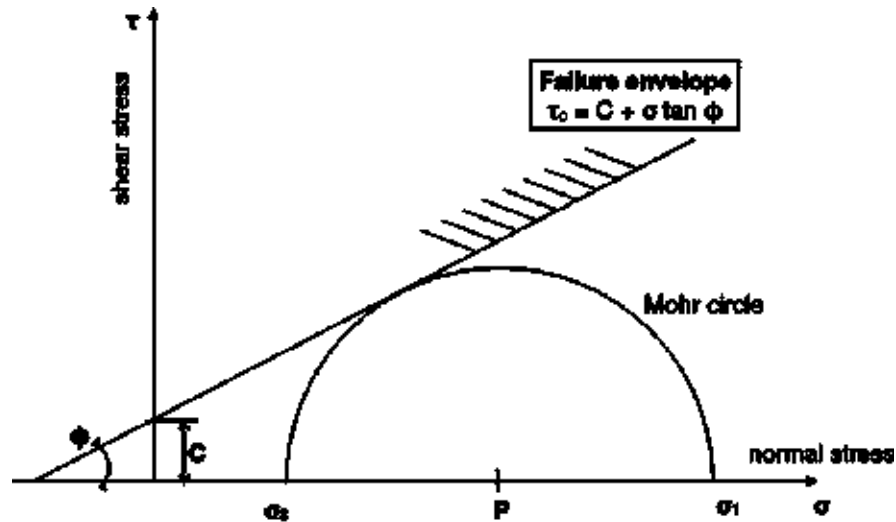


Figure 5. The Mohr-Coulomb Failure Criteria and Mohr's Circle (Source: www. khaakpey.ir).

where τ is the shear stress, c is the cohesion, σ is the normal stress and ϕ is the angle of internal friction. This linear relationship is graphically represented by a failure envelope on a Mohr's circle diagram, which plots the shear and normal stresses. The Mohr-Coulomb failure envelope is a straight line that intersects the shear stress axis at the cohesion value and has a slope equal to the tangent of the internal friction angle.

Two main laboratory tests are commonly used to measure soil shear strength parameters: the direct shear test and the triaxial test. In the direct shear test, a soil sample is subjected to shear stress along a predefined plane, allowing for the measurement of both cohesion and the internal friction angle. Equation 5 is used to calculate shear strength in the direct shear test (European Committee for Standardization, 2007).

$$\tau = \frac{F}{A} \quad (\text{Eq.5})$$

where:

τ is the shear stress (kPa)

F is the force required to shear the soil sample (N)

A is the cross-sectional area of the soil sample (mm²)

In the triaxial test, the soil sample is confined radially and axially while subjected to axial stress, allowing for more realistic simulation of field conditions. The triaxial test provides additional insights into the stress-strain behavior of soil and is particularly useful for cohesive soils. The equation 6 and 7 are used in the triaxial test to calculate cohesion (c) and the internal friction angle (ϕ) (European Committee for Standardization, 2007).

$$c = \frac{\sigma'_3 \cdot \tan\left(\frac{\phi}{2}\right)}{3} \quad (\text{Eq.6})$$

$$\tan\phi = \frac{\sigma'_1 - \sigma'_3}{2 \cdot \sigma'_3} \quad (\text{Eq.7})$$

where:

σ'_1 is the major principal stress (kPa)

σ'_3 is the minor principal stress (kPa)

Numerous studies have been conducted to investigate the shear strength parameters of soil and the factors influencing them. Researchers have explored various aspects such as grain size distribution, mineral composition, soil structure, and environmental conditions to better understand soil behavior under different loading conditions. However, despite the wealth of research in this field, there remains a gap in studies focusing on estimating shear strength parameters quickly and efficiently, especially in situations where experiments are time-consuming or when rapid data is needed for preliminary design purposes. Thus, in recent years, researchers have made attempts to estimate this behavior of soil based on its basic characteristics by using AI approaches such as artificial neural networks which will be discussed in next sections (Jasim, 2019).

2.2. Empirical correlations for soil parameter estimations

In many engineering fields, such as hydrogeology, geotechnical, agricultural, and environmental engineering, it is essential to understand and estimate soil properties. To effectively handle a

variety of issues, including evaluating groundwater flow, controlling pollutant transport, guaranteeing soil stability, and streamlining irrigation techniques, it is imperative to have a precise understanding of soil parameters. Nonetheless, there are a number of difficulties in using in situ and laboratory procedures to measure soil characteristics using traditional methods. Installing monitoring wells and conducting significant fieldwork are necessary for in situ measurements, which can be expensive and time-consuming (Kuang, 2012). Besides, the long procedure of in situ measurements becomes unfeasible when fast data is required for early design or cost estimation. In the meantime, variations in soil moisture content frequently impede laboratory investigations and make it difficult to collect representative soil samples, particularly for heterogeneous soil formations. This results in extended testing times (Meyer, 2019). These difficulties highlight the need for creating substitute approaches, such as empirical correlations, which provide a quicker and more economical way to estimate soil parameters using easily accessible soil variables. These creative ideas allow academics and engineers to handle important engineering difficulties more effectively and make well-informed judgments by getting beyond the constraints of conventional methodologies (Brinkgreve, 2010). Hence, scientists have created empirical correlations that use information on grain size distribution to estimate hydraulic conductivity. Correlations provide a practical and effective substitute for direct measurements by utilizing the connection between hydraulic conductivity and soil particle size. A plethora of empirical formulas, each customized to certain soil types and conditions, have been offered. So, some formulas are commonly used for coarse-grained soils, while the others are suitable for fine-grained soils (Eteraf, 2020). In the following, some empirical formulae for estimating hydraulic conductivity based on grain size distribution will be discussed, categorized into coarse and fine grain analyses.

For coarse-grained soils, empirical correlations such as the Hazen formula, Zamarin equation, United States Bureau of Reclamation (USBR) method, and Chapuis equation are commonly utilized (Říha, 2018). The Hazen formula, one of the earliest empirical correlations, relates hydraulic conductivity to the effective size of soil particles, providing a straightforward estimation approach (Hazen, 1911). The Zamarin equation considers both effective size and uniformity coefficient to predict hydraulic conductivity, offering enhanced accuracy for coarse-grained soils (Singh, 2023). The USBR method incorporates particle size distribution parameters to estimate hydraulic conductivity, particularly suited for soils with diverse grain sizes (USBR, 1990). Additionally, the Chapuis equation integrates grain size distribution and porosity to predict hydraulic conductivity, offering a comprehensive approach for coarse-grained soil analysis (Chapuis, 2004).

For fine-grained soils, empirical correlations such as the Seelheim equation, Sivappulaiah method, Carrier equation, and Beckman model are commonly employed. The Seelheim equation utilizes soil particle size parameters to estimate hydraulic conductivity, focusing on the unique characteristics of fine-grained soils (Doury, 2010). The Sivappulaiah method considers both particle size distribution and void ratio to predict hydraulic conductivity, offering improved accuracy for fine-grained soils with varying degrees of compaction (Sivappulaiah, 1985). The Carrier equation incorporates soil particle size distribution and specific surface area to estimate hydraulic conductivity, providing a comprehensive approach for analyzing fine-grained soils

(Carrier, 2003). Additionally, the Beckman model utilizes soil grain size parameters to predict hydraulic conductivity, particularly suited for cohesive soils with complex particle arrangements (Bilardi, 2020). It's noteworthy that all of these empirical formulae are fully or partially reliant on grain size distribution, highlighting its crucial role in estimating hydraulic conductivity across a spectrum of soil types and conditions in geotechnical engineering applications.

As technology has developed, there has been a noticeable change from using traditional empirical and analytical methodologies for estimating soil properties to using artificial intelligence (AI) methods. Artificial intelligence (AI) techniques, such as machine learning algorithms and neural networks, provide clear benefits over traditional methods by utilizing extensive datasets and recognizing complex patterns in soil parameters (Singh, 2020). More precise and trustworthy estimates are produced as a result of these AI models' ability to represent intricate nonlinear interactions between input variables and soil characteristics. Furthermore, AI techniques can learn from fresh data continuously, improving their predicting power over time. As a result, in order to address geotechnical difficulties, engineers and researchers are increasingly turning to AI-driven solutions, which provide unmatched efficiency and precision in calculating soil characteristics for a variety of engineering applications (Onyelowe, 2023).

2.3. Regression analysis

Regression analysis has been used extensively in soil research, in particular to estimate soil properties such as hydraulic conductivity (Yoon, 2015). The use of regression methods in this context extends back several decades and demonstrates a progression from simple linear models to complicated machine learning algorithms, reflecting advances in statistical approaches and processing capacity.

The use of regression analysis in soil science began with simple linear regression models that attempted to demonstrate direct correlations between soil characteristics and grain size distribution. A linear model was first introduced to predict soil moisture characteristics based on particle-size distribution and bulk density data, providing a solid understanding of how soil texture influences hydraulic properties (Arya, 1981). Similarly, linear regression models were developed to estimate soil water retention properties based on particle size distribution, organic matter percentage, and bulk density (Gupta, 1979). These early attempts helped to demonstrate the power of regression analysis in quantifying soil hydraulic characteristics using easily available soil texture data. In order to address particular issues with soil property estimate, recent works have further developed linear regression approaches. For a more accurate assessment of soil hydraulic characteristics, linear regression was employed, demonstrating the importance of incorporating soil structure and organic matter content into regression models to improve the precision of hydraulic conductivity forecasts (Vereecken, 2010).

As it became clear that linear regression could not capture the complex, researchers began to investigate non-linear regression models. For instance, a nonlinear regression approach was used to model hydraulic conductivity, recognizing the curvilinear link between grain size distribution and soil hydraulic behavior. Nonlinear models provided more accurate predictions

by incorporating logarithmic and exponential functions, which better described the complex interactions between soil particles and water flow (Ahuja, 1984).

Multiple regression techniques were established to improve the predictive ability of regression models by allowing several independent variables to be considered simultaneously. This method allows for the inclusion of additional soil properties, such as bulk density, organic matter content, and porosity, in addition to grain size distribution. A comprehensive multiple regression model was used to improve estimates of hydraulic conductivity by including several soil factors. By incorporating a broader range of influencing factors, multiple regression models provided a more holistic and accurate representation of soil hydraulic parameters (Saxton, 1986).

The concept of pedotransfer functions (PTFs) represents a significant development in the use of regression models in soil research. PTFs employ easily available soil data to forecast hydraulic parameters, frequently using regression techniques (Zhang, 2019). Wösten was among the pioneers in this field, using texture and other soil parameters to forecast unsaturated soil hydraulic functions. Their findings demonstrated the effectiveness of PTFs in making hydraulic property calculations more accessible and practical for large-scale applications (Wösten, 1988). This approach was expanded by predicting the soil moisture retention characteristic from texture, bulk density, and carbon content (Vereecken, 1989).

A notable trend in this century has been the incorporation of machine learning methodologies alongside conventional PTFs to improve their precision and relevance. Advanced techniques like support vector regression (SVR) and random forest regression have been used to predict the intricate interactions between grain size distribution and hydraulic conductivity. Machine learning-based regression models have been shown to outperform traditional regression models. Studies demonstrated that contemporary algorithms utilize computing power to identify patterns and relationships that traditional techniques may miss. For instance, the use of neural network-based approaches, such as the neuro-m model for fitting parametric pedotransfer functions, has significantly enhanced prediction accuracy. Although these methods are grounded in the fundamentals of regression, they offer a range of advantages and complexities that extend beyond traditional techniques (Schaap, 2000; Minasny, 2002).

2.4. Artificial Neural Network in Geotechnical Engineering

The exceptional problem-solving abilities and high performance of the human brain have inspired hardware and software architects to aspire to simulate its capabilities (Park, 2011). For the past few decades, computers have enabled the development of computational algorithms aimed at mimicking human brain function. This effort has spurred extensive research by computer scientists, engineers, and mathematicians, resulting in significant advancements in the field of artificial intelligence (AI) (Dong, 2020). A key branch of AI is computational intelligence, which includes “artificial neural networks” (ANNs) (Hanne, 2017). In this field, numerous mathematical and software models have been developed to emulate the human brain. These models are employed to tackle a wide range of scientific, engineering, and practical problems across various domains (Zheng, 2023). The use of intelligent systems, particularly ANNs, has become so widespread that they are considered fundamental mathematical tools.

They are commonly used in academic and professional settings for tasks requiring analysis, decision-making, estimation, prediction, design, and construction (Gohary, 2017).

The development of artificial intelligence (AI) and human cognition is reflected in the long and complex history of artificial neural networks (ANNs). Over the course of seven decades, they have undergone periods of intensive research, notable advancements, and sporadic setbacks (Capra, 2020). The 1940s are when the idea for ANNs was first developed. A simplified computational model of the neuron was introduced in 1943 by (Pitts, 1943), who proposed that networks of these neurons could theoretically execute any calculation that a Turing machine could. The McCulloch-Pitts neuron model served as the inspiration for the first artificial neural network. In 1949, psychologist Donald Hebb proposed a theory of learning that became a cornerstone for neural network learning algorithms. Hebb's rule, often summarized as “cells that fire together wire together,” suggested that the synaptic strength between neurons increases when they are activated simultaneously (Alghafri, 2021). This idea of synaptic plasticity influenced later learning algorithms in neural networks. The 1950s and 1960s were characterized by optimism and the development of early neural network models. The perceptron, an early type of artificial neural network capable of learning and recognizing patterns, was introduced by Rosenblatt (1958). Inspired by biological processes in the human brain, the perceptron aimed to mimic the learning abilities of the human mind. Rosenblatt's work attracted significant attention and funding, notably from the U.S. Navy, which recognized its potential for military applications. The perceptron consisted of a single layer of artificial neurons, called “units,” which processed inputs and produced an output. The perceptron learning rule adjusted the weights of the inputs based on the error between the predicted and actual outputs. Despite its simplicity, the perceptron could solve only linearly separable problems, which limited its practical applications (Glorot, 2010). Due to unfulfilled expectations, the first “AI winter,” marked by a drop in funding and interest in AI research, started in the 1970s. Research on perceptron by (Minsky, 1969), highlighted the drawbacks of single-layer perceptron and caused this downturn. They showed that the XOR problem, a simple but non-linearly separable problem, was beyond the capabilities of perceptron. This discovery led to a loss of faith in neural networks and a notable decline in the amount of research being done in the area. Neural networks noticed an increase in attention in the 1980s, partly because of the invention of the backpropagation algorithm. Backpropagation offered a way to train “deep” neural networks, or multi-layer neural networks. The networks were able to learn intricate, non-linear patterns which made it possible to modify the weights in hidden layers. Research demonstrated the practical viability of backpropagation and sparked renewed enthusiasm and research in neural networks (Rumelhart, 1986). The development of techniques for training deep networks expanded possibilities for applications such as pattern recognition, speech processing, and image analysis.

With the development of backpropagation and increased computational resources, ANNs became practical for engineering applications during the 1990s. These networks began to excel in tasks such as pattern recognition, signal processing, and control systems (LeCun, 1998). Engineers leveraged ANNs to optimize design processes, improve fault detection systems, and enhance predictive maintenance. The ability of ANNs to learn from data and adapt to new information without explicit programming opened new frontiers in automation and artificial

intelligence, embedding themselves as a crucial tool in the engineer's toolkit. This evolution not only improved efficiency and innovation within engineering disciplines but also spurred further research and development, leading to the sophisticated deep learning models that drive modern technological advancements (Abioye, 2021).

Despite the fact that artificial neural networks (ANNs) were introduced decades ago, they have become indispensable tools in estimating soil parameters recently (Pham, 2020). This is due to a criticism that has existed in the last ten years that empirical methods are limited to their respective domains because they were developed under specific boundary conditions and may result in random errors in the hydraulic conductivity values (Chandel, 2022). The capacity of this method to capture intricate relationships within datasets makes it more dependable. With their networked nodes (neurons) arranged in layers, artificial neural networks (ANNs) imitate the composition and operations of the human brain by processing information through mathematical calculations. Artificial neural networks (ANNs) have been widely used in the field of geotechnical engineering to forecast several soil properties, such as compressibility, shear strength, and hydraulic conductivity. These networks are especially well-suited to capture complex patterns and correlations associated with soil attributes because they are adept at managing nonlinear and multidimensional facts (Bahmed, 2019). In addition, ANNs demonstrate resilience while processing noisy or sparse information, which allows them to generate trustworthy estimates even when uncertainties are present. ANNs are able to adjust their predicting powers over time by learning from prior experiences and utilizing big datasets with a variety of soil qualities and environmental circumstances (Jiadong, 2024). Furthermore, the adaptability of ANNs permits the incorporation of diverse input parameters, from fundamental soil characteristics to intricate environmental elements, consequently augmenting the precision and comprehensiveness of soil parameter estimations (Jalal, 2021).

Considering the hydraulic conductivity parameter, artificial neural networks (ANN), fuzzy logic, and adaptive-neuro fuzzy inference system (ANFIS) were utilized to predict hydraulic conductivity (K) values of soil samples using measurable parameters like specific gravity, porosity, and percentages of gravel, sand, and silt. Statistical analysis demonstrated that the ANFIS method exhibited superior performance compared to ANN and fuzzy logic (More, 2018). The effectiveness of using artificial neural network (ANN) methods to calculate the hydraulic conductivity (K) of ninety-five samples of coarse-porous media was investigated by (Akbulut, 2005). The K values obtained by ANN, multiple linear regression (MLR), and two empirical approaches were compared. The results showed that ANN outperformed the other methods in terms of prediction when it came to estimating K values. Three data-driven techniques—multiple linear regression (MLR), adaptive-neuro fuzzy inference system (ANFIS), and artificial neural network (ANN)—were compared to predict hydraulic conductivity (K) in porous media. The study revealed that the ANFIS model outperformed both MLR and ANN in terms of predictive accuracy (Arshad, 2013). In a separate study, neural computing models were devised to forecast the hydraulic conductivity (K) of coarser porous media, incorporating both artificial neural networks (ANN) and adaptive-neuro fuzzy inference system (ANFIS). By including various grain sizes (d₁₀, d₃₀, and d₆₀) in the model development process, a more robust correlation with K was achieved. The ANFIS model notably

demonstrated superior reliability compared to the ANN model in predicting K values (Yilmaz, 2012). Another study assessed the effectiveness of artificial neural network (ANN), adaptive-neuro fuzzy inference system (ANFIS), and support vector machine (SVM) techniques in estimating the hydraulic conductivity (K) values of porous media. The study concluded that both ANN and ANFIS models yielded satisfactory results in predicting K values (More, 2022). In another study, the accuracy of multiple linear regression (MLR) and feed-forward neural network (FFNN) in calculating the hydraulic conductivity of porous media was compared. Based on six input variables for model building, the investigation demonstrated that FFNN outperformed MLR in terms of predictive power (Williams, 2021). Despite the remarkable prediction capabilities of supervised algorithms, limitations in their application persist.

Regarding other soil parameters, there are a variety of studies which used artificial neural networks for estimation. Artificial neural networks (ANN) were employed to forecast the settlement of shallow foundations on coarse-grained soils. A dataset of 79 samples—69 for training and 10 for testing—was used to develop the ANN model, which incorporated five input parameters: applied net pressure, average standard penetration test (SPT) values, foundation width, foundation shape, and foundation depth (Sivakugan, 1998). An artificial neural network (ANN) model was used to study the relationship between relative density and average effective stress (σ') in typically loaded and over-consolidated sands. The model aimed to estimate Cone Penetration Test (CPT) and cone resistance (q_c) as output variables. For training and testing, the study utilized 93 and 74 data points, respectively. The nonlinear connection demonstrated a high correlation coefficient of 0.91 for testing data and 0.97 for training data (Goh, 1995). Neural networks were investigated to anticipate displacement in deep foundation pit retaining systems. The research demonstrated that employing soft computing techniques, such as neural networks, proved to be a valuable and reliable tool for predicting deformation in foundation pits (Wei, 2023). In another study, shallow settlement near a foundation pit excavation was computed using artificial neural networks (ANNs) in an AI prediction model (Chen, R. P., 2019). The majority of real-world uses demonstrate that it is possible to estimate slope stability analysis using ANN. In this context, a few studies looked into the use of ANNs in slope stability (Lu, 2003). For instance, used ANNs and Grey systems to forecast slope stability. Liquefaction resistance, considered a crucial component of geotechnical engineering, was modeled using artificial neural networks (ANNs) (Abbaszadeh, 2016). ANNs were also used by a number of researchers to enhance the site characterization estimate. For instance, a fuzzy set-based model was created to infer the subsurface profile and describe data from drilling operations (Huang, 1997).

While artificial neural networks (ANNs) have garnered significant attention and application in geotechnical engineering for estimation and modeling purposes, there remains a considerable gap in research when it comes to utilizing soil particles, such as silt, clay, sand, and gravel, as input data to estimate crucial soil parameters like hydraulic conductivity. In practical scenarios, the available input often boils down to the soil component, or grain size distribution, making it imperative to develop models that rely solely on this parameter. Consequently, in this study, the focus was precisely on estimating hydraulic conductivity and shear strength parameters exclusively through grain size distribution. The aim was to overcome the practical constraints encountered in real-world applications, where complete soil component data may not always be

easily accessible, by highlighting this technique. This study emphasizes how crucial it is to create reliable and useful models that make use of easily available data in order to improve the precision and application of soil parameter predictions in geotechnical engineering settings.

In this chapter, fundamental geotechnical and hydrogeological concepts were reviewed, establishing the basis for the estimation of hydraulic conductivity and shear strength parameters. Empirical correlations used for parameter estimation were examined, along with their practical applications and limitations. The methodology of regression analysis was discussed as a means to develop predictive models for soil parameters. Additionally, the application of artificial neural networks (ANNs) in geotechnical engineering was explored, highlighting their potential advantages over traditional methods in enhancing estimation accuracy. This comprehensive review provides the foundation for utilizing advanced techniques, such as ANNs, to improve the estimation of geotechnical parameters.

3- Methodology

3.1. Data Collection and Preparation

The soil samples and materials used in this study were collected from the Hejőpapi area shown in Figure 6, situated in Borsod-Abaúj-Zemplén County, Hungary. The geographical coordinates of the collection site are 47°54'00" North latitude and 20°55'00" East longitude. In the laboratory, the collected soil samples were systematically separated into their constituent particles: gravel, sand, silt, and clay. This separation process ensured that each type of particle could be individually analyzed and combined in precise proportions to create new soil samples for experimentation.

The new soil samples were prepared by carefully combining varying quantities of each type of particle. The goal of this procedure was to reproduce different soil compositions and investigate their characteristics in a controlled environment. To ensure that the experimental results were free of biases or mistakes, great care was taken to preserve consistency across all samples, including maintaining uniform water content and unit weight. The precise ratios of gravel, sand, silt, and clay were adjusted to see how different soil compositions affected the characteristics under study. It is noteworthy that for the scope of this study, which focused on specific soil parameters, only small-sized gravel particles were used. All prepared samples were subjected to identical environmental conditions and handling procedures to ensure uniformity. This standardization was crucial for minimizing variability and enhancing the reliability of the experimental results. By maintaining these stringent preparation protocols, including consistent water content and unit weight, the study aimed to produce accurate and reproducible findings that could contribute valuable insights into soil behavior and characteristics.



Figure 6. Collection Site of Soil Sample

In general, 205 soil compositions were reconstructed in the laboratory. Depending on the type of parameter under investigation, all or some of these samples were investigated. The percentages of each component were variable and included 0, 10, 15, 20, 25, 30, 40, 50, 75 and 100 percent (Annex 1). It should be noted that the naming order of the samples in this research was determined after obtaining the results and performing the final analysis, which will be discussed in detail in

the next chapter. The desired tests were performed on each reconstructed sample according to the standard to obtain the investigated parameters.

It should be mentioned that the laboratory work for this study was carried out across two institutions. The initial phase of the experiments was conducted in the soil mechanics laboratory at the University of Miskolc. The subsequent phase of the laboratory work was carried out at the Budapest University of Technology. Some laboratory tests conducted in lab is depicted in figure 7



Figure 7. Laboratory Tests Conducted in the Lab

3.2. Experiments

3.2.1. Grain size distribution

Following the creation of the desired soil combinations, the grain size distribution for each new soil sample was determined (Annex 2). To achieve this, both the sieve analysis test and the hydrometer test were conducted in accordance with the Eurocode 7 standard. The grain size distribution analysis was crucial for providing a comprehensive understanding of the sample characteristics before conducting further experiments. It allowed for the categorization of samples, ensuring that each one was appropriate for specific experimental procedures. Additionally, it facilitated the separation of samples into coarse and fine grains, which is essential for tailoring experiments to different soil types. The analysis also enabled the determination of key parameters such as D_{10} , D_{30} and D_{60} . These parameters are critical for various aspects of this investigation, including the assessment of soil permeability and the mechanical behavior of the soil samples.

The sieve analysis was employed to determine the particle size distribution of the coarser fractions of the soil. According to Eurocode 7, the soil sample was dried and then passed through a series of standard sieves arranged in descending order of size. Each sieve's aperture size was selected to capture specific ranges of particle sizes. The soil retained on each sieve was weighed, and these

weights were used to calculate the percentage of the total sample. This process resulted in a detailed profile of the soil's grain size distribution for particles larger than 0.063 mm, providing essential data for understanding the soil's physical properties.

For the finer particles, a hydrometer test was conducted following the Eurocode 7 guidelines. This test involved dispersing a known quantity of soil in a liquid and measuring the relative density of the suspension at various time intervals using a hydrometer. The hydrometer readings allowed for the calculation of particle sizes smaller than 0.063 mm based on the sedimentation rate of the particles, as described by Stokes' law. The hydrometer test was essential for accurately characterizing the silt and clay fractions of the soil samples, complementing the data obtained from the sieve analysis.

By integrating the results of both the sieve analysis and the hydrometer test, a complete grain size distribution curve was obtained for each soil sample. These curves were instrumental in identifying the uniformity and gradation of the soil, which in turn informed the design and interpretation of subsequent experiments. After drawing the curves, soils were classified according to (European Committee for Standardization, 2007). This classification identified a variety of soil types from the samples. Table 1 shows the various types of soils identified, as well as the quantity of samples for each type. This precise categorization was helpful for understanding the unique properties and behaviors of each soil type, resulting in more accurate analysis and implementation of the experimental data.

Table 1. Classification of Soil Samples

Soil name	Sample count	Soil Name	Sample count
Cl	13	grclSa	16
siCl	4	grsaSi	5
Si	1	grSi	3
grCl	20	siSa	3
sasiCl	12	clGr	6
saCl	18	saclGr	11
sagrCl	11	sasiGr	7
grsiCl	15	grsiSa	4
grsaCl	16	Sa	2
clSi	2	sagrSi	1
clSa	6	siGr	4
saSi	3	grSa	5
saclSi	7	saGr	2
grclSi	7	Gr	1

3.2.2. Hydraulic conductivity

Hydraulic conductivity, a crucial parameter for understanding water movement in soil, was measured for 205 samples based on Eurocode 7 criteria in this study. It was tried to have a similar initial condition for all specimens before conducting the tests. Following the concepts outlined in Darcy's law in previous chapter, hydraulic conductivity, also known as coefficient of permeability or permeability, refers to the ease with which water permeates rock or soil. To capture soil type heterogeneity, two different laboratory procedures were used: the constant head permeability test for coarse-grained soils and the falling head permeability test for fine-grained soils. The constant head permeability test indicates soils with high permeability, such as sand. It involves measuring the discharge flow rate (Q) through the soil sample under a constant hydraulic head (Δh) during a set duration (t). Conversely, the falling head permeability test, suitable for soils with low to intermediate permeability like clays and silts, monitored the gradual decrease in water level (h_1 to h_2) within a standpipe installed on top of the soil sample. By recording the change in water level over time (t) and considering sample dimensions, hydraulic conductivity was computed using the pertinent equations.

3.2.3. Shear strength parameters

The soil's shear strength parameters were determined using direct shear tests on a total of 95 samples, in accordance with the Eurocode 7 criteria. These tests were critical in determining the soil's resistance to internal friction and cohesion, especially for coarse and fine-grained soils. Following Eurocode 7 protocols enhanced scientific accuracy and complying with international standards, hence increasing the reliability and credibility of the findings. Each direct shear test was methodically carried out, with special attention given to sample preparation, loading conditions, and testing techniques. To account for variability and assure the correctness of the data, each test was performed three times, as is standard practice in geotechnical engineering experiments.

The direct shear test involves exposing soil samples to controlled shear stress along a designated plane to determine cohesion (c) and internal friction angle (ϕ). These characteristics were determined after a careful study of the test findings, which included calculations based on the measured shear stress and normal stress applied to the samples. The cohesion (c) was calculated using the intercept of the shear stress-normal stress plot, while the internal friction angle (ϕ) was found using the slope of the linear component of the curve. The complete investigation included differences in soil composition, grain size distribution, and other relevant parameters, resulting in a comprehensive understanding of the soil's shear strength behavior.

3.3. Overview of data trend

In this study, high numbers of laboratory tests were conducted to see the effect of different combinations of gravel, sand, silt and clay on hydraulic conductivity, in particular hydraulic conductivity. In order to avoid the possibility of error in estimating soil parameters, the traditional triangular system for determining soil texture and subsequently using it to determine the hydraulic gradient has been upgraded to a 3D tetrahedral system. So, the measured parameters have been plotted and charted in a 3D tetrahedral system (gravel, sand, silt and clay content). Based on this system the factors affecting the different geotechnical and hydrogeological parameters of soil based on particle size distribution can and will be investigate while the parameter chosen for investigation in my study was hydraulic conductivity.

As the first step, only the soil compositions with a difference of 25% were chosen for this purpose. Figure 8 shows the result for one point in pyramid as an example which belongs to bottom level where the gravel percentage is zero. The same procedure was done for all chosen samples and the results were gathered to transfer to pyramid model. It is noteworthy to take into account that in this work, I employed $(-\log K)$, the negative logarithm of hydraulic conductivity, rather than its precise value for each composition. The usage of $(-\log K)$ is based on the lognormal distribution character of K data, and therefore $\log K$ values became normally distributed.

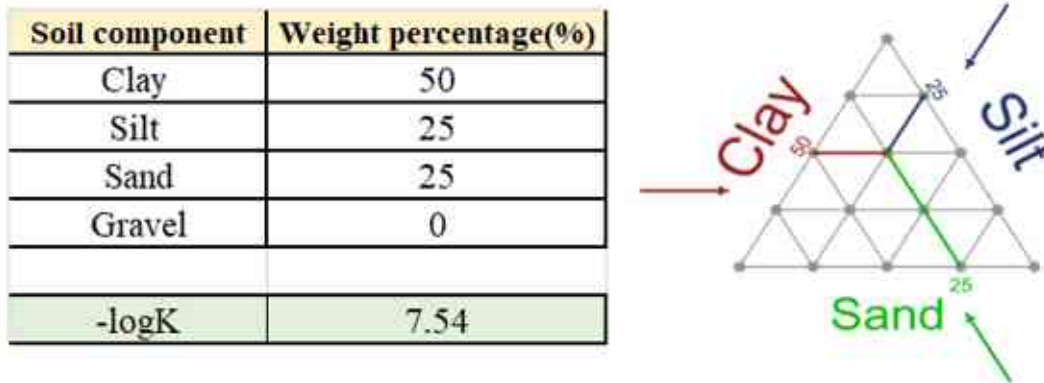


Figure 8. Location of a Point in Pyramid (Source: Own Compilation)

Figure 9 schematically shows the pyramid model considered in this study. Also changes in composition are shown on each plate of the pyramid.



Figure 9. The Schematic Picture of Pyramid Model (Source: Own Compilation)

The main goal of this task is to visualize the model in 3D. So, a tool to do a 3D model was required for geological modeling and interpretation. Voxler software was chosen to create the 3D model in this investigation. To draw points into the model in Voxler, it was necessary to transfer the coordinates in the Cartesian coordinate system. For this purpose, assuming a constant value for one soil component (gravel) and considering the sum of 100% of all components, I was able to

draw the corresponding points of each type of mixture in Voxler coordinates. Finally, all the points in a pyramid were demonstrated which can be seen in figure 10.

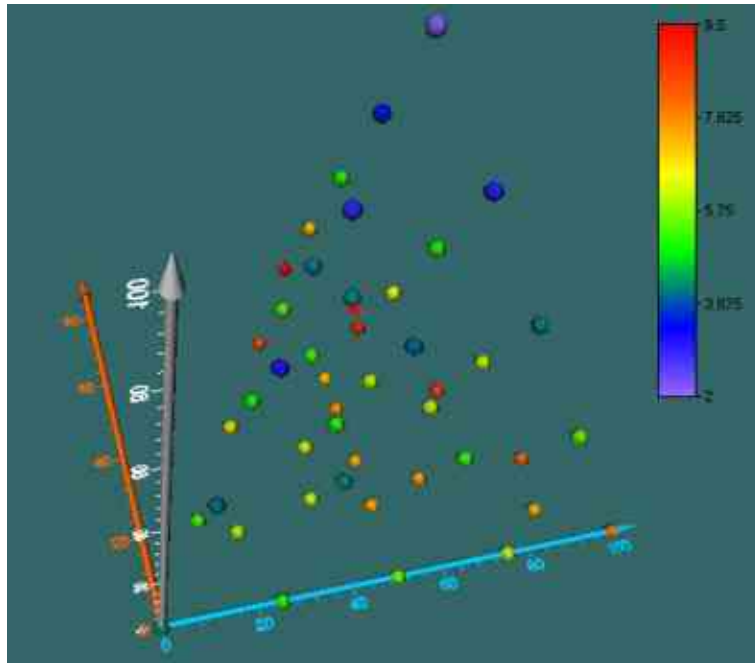


Figure 10. Location of all Points in Pyramid (Source: Own Compilation)

To render 3D volumetric data, Voxler uses a direct volume rendering approach. It's a terrific approach to view Earth and atmospheric sciences data, as it allows us to see all elements of the data in a 3D volume rather than just the surface (Golden Software, 2015). The VolRender is a very useful tool for illustrating the hydraulic conductivity distribution that I wanted to look into for my research. Figure 11 shows the volume of the data from gridded data which was made before in surfer software.

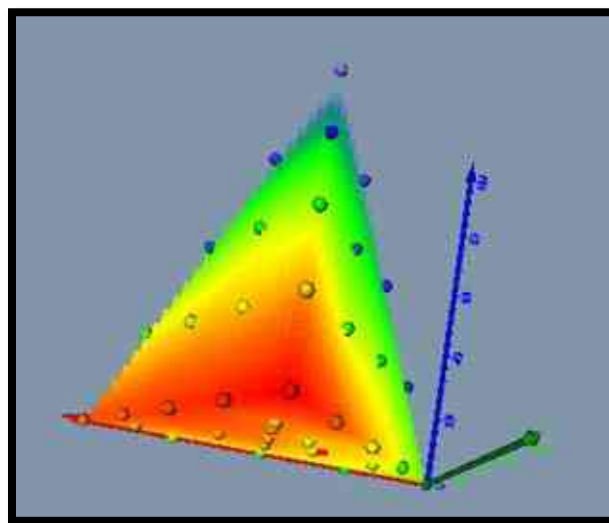


Figure 11. Hydraulic Conductivity Values Distribution (Source: Own Compilation)

By this method I can examine the pyramid from different angles. With the help of it, I am able to draw plates in the desired directions and see the distribution of the considered parameter. Therefore, by moving the plates in the direction of increasing each soil component, its effect on the hydraulic gradient can be observed. A comparison between them will lead to a better understanding of soil conditions. For instance, figure 12 shows the different planes of pyramid concerning variation in gravel contents.

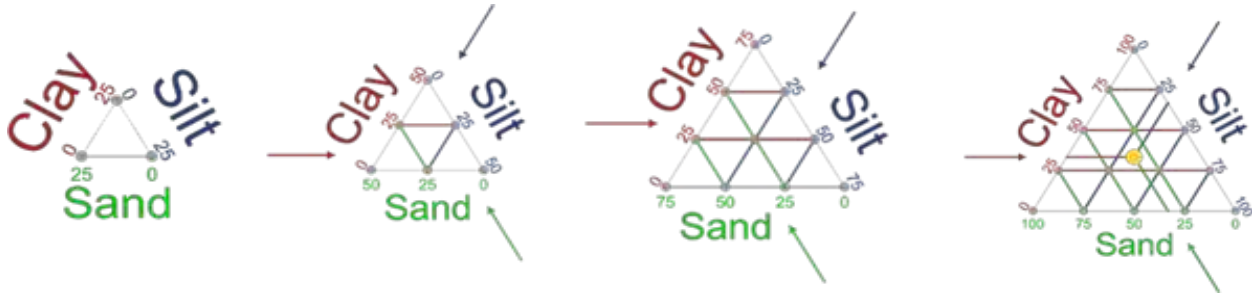


Figure 12. Different Planes of Pyramid by changing in gravel contents (Source: Own Compilation)

It was tried to draw contour map for all these sections and analysis the relation between gravel content and hydraulic conductivity. The same procedure has been done for the other directions regarding increasing in clay, silt and sand percentages in combination. For this purpose, the unit vector and the normal vector of the plates were required. The normal vector was determined by using 3 points. For example, the following 3 points were used for the normal vector of a plate with constant silt.

$$P = (0,0,0), \quad Q = (50,86.6,0), \quad R = (50, 75.7,25)$$

By finding the PQ and PR vectors and taking their cross product, the vector perpendicular to them was obtained and then by doing some calculation, the equation for a plane was determined. Having the equations of the desired plates, it was possible to study the changes of the desired parameter from different angles, which means that by doing this, the effect of each variable on the desired parameter can be examined separately. Considering the triangle at the base of the pyramid and the planes parallel to it, each side of the triangle represents a component (clay, sand or silt). But in other sections that are not in the mentioned direction, the sides do not represent the mentioned parts. The point to be considered is that at these points the height will indicate the amount of gravel.

Finally, the desired distributions of hydraulic conductivity changes in different planes of the pyramid were extracted from the software. Utilizing the program features, I am able to obtain the estimated range of the desired parameter throughout the entire pyramid. This indicates that for each soil composition with varying percentages of different soil types, an estimated range for the hydraulic conductivity can be considered. Additionally, another noteworthy point that can be achieved using this pyramid is that the extent of the impact of each variable on the desired parameter can be examined. As mentioned earlier, the Volrender feature shows us the volumetric distribution of the desired parameter. According to the following figure, it can be observed that the effect of clay on permeability changes is much higher than that of other parameters.

3.4. Multiple linear regression

In this study, Multiple Linear Regression (MLR) is initially utilized to estimate hydraulic conductivity based on the percentages of clay, silt, sand, and gravel in soil samples. Based on regression analysis, a linear relationship is established between the “response variable” and one or more “explanatory variables.” In multiple linear regression, the parameters of a linear model are estimated using an objective function and the values of the variables. Thus, if there are n observations from the P dimensional independent variable X and it is aimed to establish a relationship with the response variable Y , the multiple linear regression model can be used as shown in equation 8 (Uyanik, 2013).

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon \quad (8)$$

Where Y_i is dependent variable, x_i is explanatory variables, β_0 is y-intercept (constant term), β_p is the slope coefficients for each explanatory, and variable ϵ is the model’s error term (also known as the residuals)

The coefficients in an MLR model are estimated using the method of least squares. This method aims to minimize the sum of the squared differences between the observed values and the values predicted by the model. The performance of the Multiple Linear Regression (MLR) model in this work was evaluated using several key metrics. One important metric was R-squared (R^2), which represents the proportion of variance in the dependent variable explained by the independent variables. It is calculated as the ratio of the sum of squares due to regression (SSR) to the total sum of squares (SST). The coefficient of determination, R^2 , is interpreted as the proportion of the variability in the dependent variable accounted for by the regression model. This value is frequently expressed as a percentage, indicating the additional explanation of variability provided by the model among the total variability (Nathans, 2012). According to the classification proposed by Chin (1998), the range of R-squared values is a significant indicator of a regression model’s explanatory power. An R-squared value of 0.67 or higher is considered substantial, meaning that the independent variables account for a significant portion of the variability in the dependent variable. Values between 0.33 and 0.67 are categorized as moderate, indicating a moderate level of explanatory power. R-squared values from 0.19 to just below 0.33 are classified as weak, reflecting a relatively lower level of explanatory power. This classification provides valuable insight into the strength of the relationship between the independent and dependent variables captured by the regression model.

Another crucial metric is the adjusted R-squared, which adjusts for the number of predictors in the model, providing a more accurate measure for models with multiple variables. This adjustment ensures that the model’s explanatory power is not overstated due to the inclusion of more predictors. Additionally, I examine the p-values of the coefficients to determine whether each independent variable significantly contributes to the model. Variables with low p-values (typically less than 0.05) are considered significant predictors, indicating their strong influence on the dependent variable. Furthermore, I utilize the Mean Squared Error (MSE) to assess the average of the squared differences between observed and predicted values. MSE is a measure of the model’s accuracy; lower MSE values indicate better fit, signifying that the predicted values are close to the actual values. By evaluating these metrics collectively—R-squared, adjusted R-squared, p-values

of the coefficients, and MSE—it is possible to validate the robustness and reliability of the MLR model for estimating hydraulic conductivity based on grain size distribution.

To ensure I developed a reliable model, I divided the data into two parts. I used 80% of the observations to train the Multiple Linear Regression (MLR) model and kept the remaining 20% for validation. This method helps us rigorously assess how well the model performs and how well it can generalize to new data. By training the model on the larger portion of the data, I make sure it effectively learns the underlying patterns and relationships between the independent variables (percentages of clay, silt, sand, and gravel) and the dependent variable (hydraulic conductivity). This way, I can be confident that my model isn't just fitting the specific data I have, but can also predict outcomes accurately for new, unseen data.

3.5. Artificial Neural Networks (ANNs)

3.5.1. Concept of ANNs

Artificial neural networks contain artificial neurons known as units. These units are organized into a succession of layers that collectively form the entire artificial neural network in a system. A layer can have a dozen or millions of units, depending on how complicated neural networks are necessary to understand the dataset's underlying patterns. An artificial neural network typically consists of three layers: input, output, and hidden. The input layer collects inputs from the outside world that the neural network must interpret or learn about. The data is then passed through one or more hidden layers, which turns it into useful data for the output layer. Lastly, the artificial neural networks' reaction to the supplied input data is presented as an output by the output layer. Units are connected from one layer to another in most neural networks. The weights assigned to each of these relationships indicate how much effect one unit has upon the others (Abdolrasol, 2021). The neural network gains more and more knowledge about the data as it moves from one unit to the next, ultimately producing an output from the output layer. Figure 13 shows a simple neural network architecture.

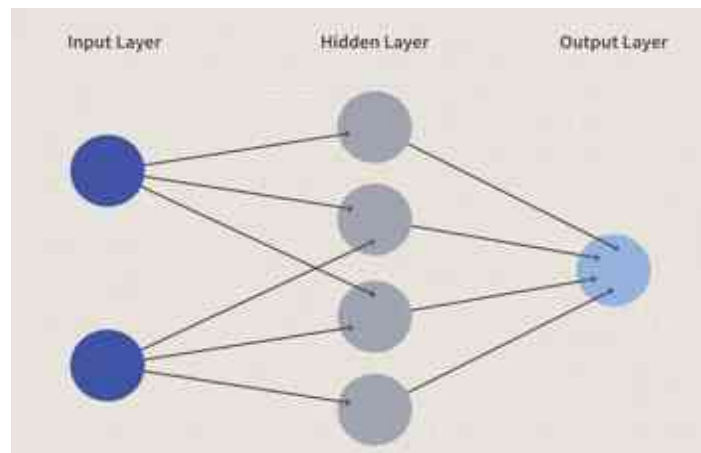


Figure 13. A Simple Neural Network Structure (Source: www.mediasoft.ir)

An artificial neural network's first layer, known as the input layer, transfers data from outside sources to the second layer, known as the hidden layer. Each neuron in the hidden layer takes in information from the neurons in the layer above, calculates the weighted sum, and then relays it to

the neurons in the layer below. Because these connections are weighted, the effects of the inputs from the preceding layer are essentially maximized by giving each input a unique weight, which is then modified during training to improve model performance. These systems are designed to recognize patterns, process data, and learn from experience. An ANN consists of interconnected groups of artificial neurons (nodes), which work collaboratively to solve specific problems. Each neuron processes inputs and generates an output that is transmitted to other neurons in the network. Figures 14 and 15 show a biological neuron and a conventional mathematical model of neurons. The main components of an artificial neural network are neurons, layers, weights (W), biases (b), and activation functions. Neurons (N) are the basic units of an ANN, analogous to biological neurons, receiving inputs (x), processing them, and producing outputs (y) using activation functions (f). A biological neuron consists of a cell body or soma for processing impulses, dendrites for receiving them, and an axon for transmitting them to other neurons. Artificial neural networks' input nodes receive input signals; the hidden layer nodes compute these input signals; and the output layer nodes compute the final output by processing the hidden layer's results with activation functions. Synapses connect biological neurons and allow impulses to be transmitted from dendrites to the cell body. Synapses are the weights that connect one-layer nodes to next-layer nodes in artificial neurons. Weight determines the strength of the linkages. Learning in biological neurons occurs in the cell body nucleus, also known as the soma, which contains a nucleus that aids in impulse processing. If the impulses are strong enough to cross the threshold, an action potential is generated and propagates via the axons. This is made possible by synaptic plasticity, which is the ability of synapses to strengthen or weaken over time in response to changes in their activity. Backpropagation is a learning approach in artificial neural networks that modifies node weights based on errors or disparities between expected and actual outcomes. When an impulse is strong enough to cross the threshold and cause a neuron to fire, this is known as activation in biological neurons. An activation function is a mathematical function that maps input to output and performs activations in artificial neural networks (Zou, 2009).

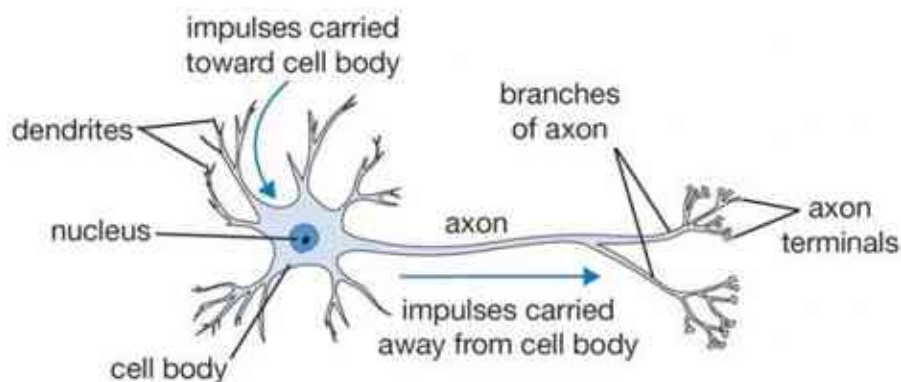


Figure 14. A Biological Neuron ([www. hamruiyesh.com](http://www.hamruiyesh.com))

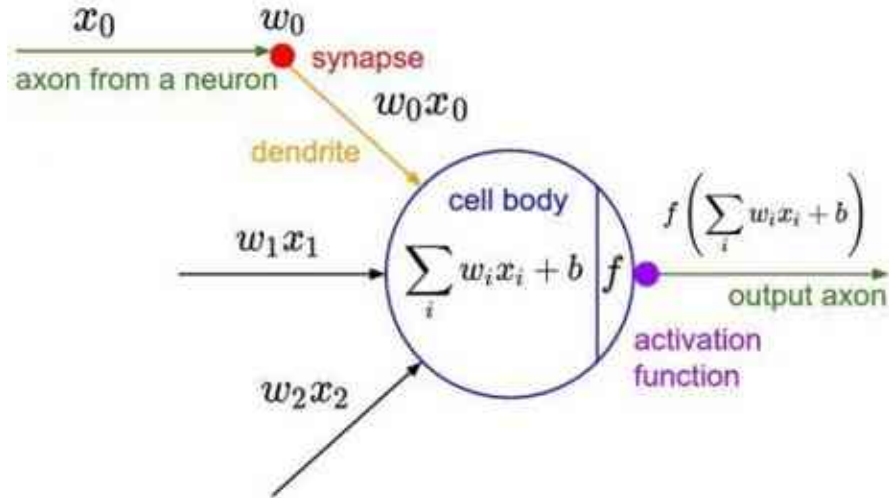


Figure 15. A Conventional Mathematical Model of Neurons (Source: Subhashini, 2020)

The output layer generates the final set of computations, estimates, or classifications based on the input data and outcomes processed by the hidden layers. Mathematically, the output of a neuron can be represented as equation 9.

$$Y = \Sigma (\text{weights} \times \text{input}) + \text{bias} \quad (\text{Eq.9})$$

Weights determine the signal (or strength of the link) between two neurons. In other words, the weight determines how much influence the input has on the output. Constant biases are an additional input into the following layer with a value of 1. Bias units are unaffected by the previous layer (there are no incoming connections), but they do have outbound connections with their own weights. The bias unit ensures that even if all inputs are zeros, the neuron will still be activated.

Activation functions are applied to the weighted sum of inputs to determine the output of a neuron, allowing the network to handle non-linear transformations and complex patterns in data. Choosing the activation function for the hidden and output layers is an important decision when developing a neural network. It means that the activation functions play an important part in determining whether a neuron should be activated or not. Neural networks cannot function without activation functions; otherwise, the model's output would just be a linear function of the input. Stated differently, it would not be able to manage substantial amounts of intricate data. In every forward propagation layer, activation functions are an extra yet important step. Even if the network had numerous layers, neurons, or nodes, problems between layers could not be analyzed without activation functions. By introducing nonlinearity via activation functions, neural networks may mimic more complex functions within each node, allowing the neural network to learn more efficiently. Activation functions can be mainly classed into three types: binary step, linear, and non-linear, with several subcategories, derivatives, variants, and other calculations being employed in neural networks. The simplest sort of activation function is the binary step, which produces a binary output based on whether the input is greater than or less than a specific threshold. Linear

functions are also generally simple, with the output proportionate to the input. Non-linear functions, like sigmoid and tanh functions, are more complex and bring nonlinearity into the model. In each situation, the activation function is chosen based on the individual problem and challenge to be solved (Rasamoelina, 2020). It's not always clear which one data scientists and machine learning engineers should employ, so trial and error is sometimes necessary. However, that is always the beginning point for selecting the appropriate activation function for a neural network or any other complex algorithmic-based model that requires activation functions. In the following, some important activation functions of their concepts will be discussed.

a) The binary step function acts as a door that only opens upon the satisfaction of a predetermined threshold. The neuron is active when an input exceeds that threshold and is inhibited otherwise. The output from the preceding layer is transferred to the subsequent stage of the neural network's hidden layers as soon as a neuron is triggered. The binary step is only based on thresholds, and naturally, it has drawbacks such as not being differentiable and not being able to backpropagate signals. When there are numerous outputs, it is unable to give multi-value outputs or multi-class classification difficulties. Nonetheless, the binary step is a practical and straightforward activation function to include for very modest neural networks (Qin, 2020). The mathematical explanation is shown in figure 16.

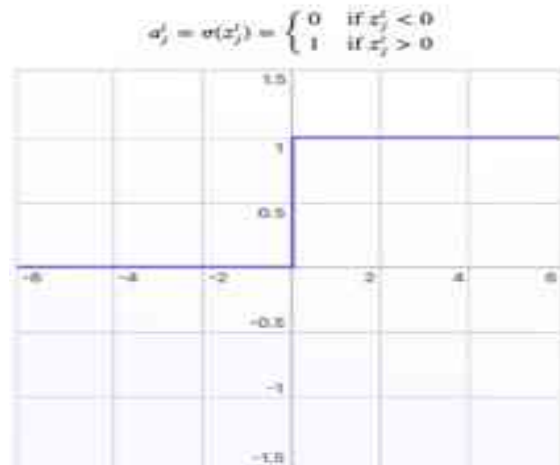


Figure 16. The Binary Step Function (Source: www.encord.com).

b) **Linear Activation Function:** When the output signal is intended to be identical to the input signal, linear activation functions are applied. Identity is differentiable, and this activation function doesn't alter the signal in any way. In most situations, this may not seem particularly helpful, but it is when it is aimed neural network's outputs to be continuous as opposed to discrete or changed. Both a reduction and a convergence of data are absent. With this activation function applied to each layer, a neural network's layers would combine into one. Therefore, not very helpful unless that's precisely what you require or if the hidden layers that follow have distinct activation functions (Rasamoelina, 2020). Figure 17 shows the mathematical explanation.

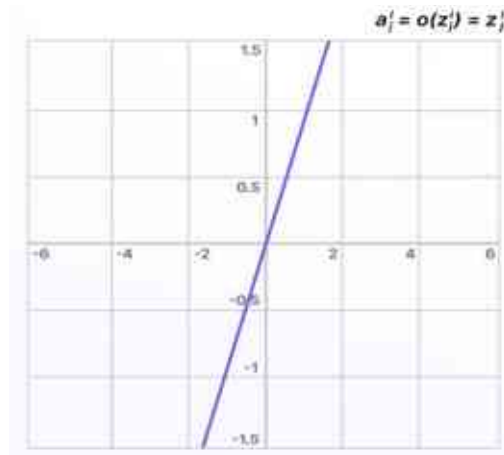


Figure 17. Linear Activation Function (Source: www.encord.com).

c) Sigmoid, Logistic Activation Functions: The sigmoid activation function, also known as the logistic activation function, converts inputs into outputs ranging from 0 to 1. Because of this, sigmoid is known as the "squashing function" and is differentiable. Larger, more positive inputs should result in output values close to 1.0, whereas smaller, more negative inputs should yield outputs closer to 0.0. It is particularly useful for classification or probability prediction tasks, making it suitable for usage in computer vision and deep learning network training. However, using vanishing gradients in hidden layers might be troublesome, causing problems while training a model (Pratiwi, 2020). Figure 18 shows the mathematical concept.

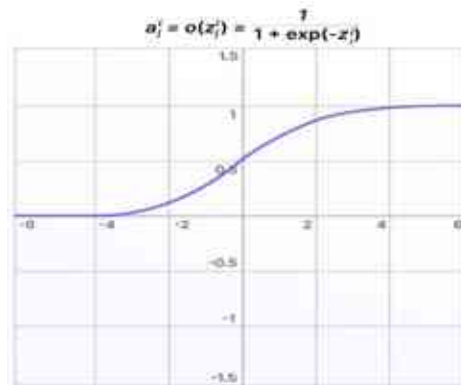


Figure 18. Sigmoid, Logistic Activation Functions (Source: www.encord.com).

d) Hyperbolic Tangent Function: The hyperbolic tangent function (tanh) is a widely used activation function. It converts input numbers to a range of -1 to 1. The tanh activation function can be beneficial in artificial neural networks because it is zero-centered, which helps to mitigate the vanishing gradient problem. Also, the values are more easily transferred to a scale of extremely negative, neutral, or positive. Adding to a neuron's output introduces nonlinearity, allowing the network to acquire complicated representations. However, it is important to note that Tanh suffers from the same saturation problem as the sigmoid function, in which gradients become extremely small for large input values. Despite this drawback, tanh is still a good choice for hidden layers in neural networks because of its balanced behavior near zero. Additionally, tanh is sigmoidal (s-

shaped), as shown in figure 19. It is worth noting that feed-forward networks, which will be discussed later, use both the tanh and logistic sigmoid activation functions (Shakiba, 2020).

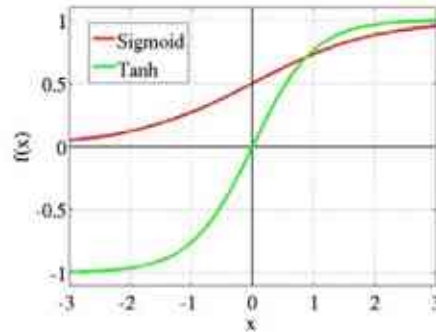


Figure 19. Hyperbolic Tangent Function (Source: www.ashutoshtripathi.com)

e) ReLU (Rectified Linear Unit) Activation Function: One of the most frequent activation functions used in artificial neural networks is the Rectified Linear Unit. ReLU is a piecewise linear function that outputs the input if it is positive; else, it returns zero. However, compared to linear functions, the rectified linear unit (ReLU) is more computationally efficient. When applied to a neuron's output, ReLU creates nonlinearity, allowing neural networks to learn complex representations. ReLU is now the standard option for hidden layers in convolutional neural networks (CNNs) and multilayer perceptrons (MLPs), despite its ease of use. It's important to be mindful of the “dying ReLU” issue, though, as sluggish convergence can result from some neurons going dormant during training. This problem is addressed by variants such as Leaky ReLU and Parametric ReLU, which accept small negative values and enhance training stability and convergence (Agarap, 2018). The ReLU is partially rectified, as seen in figure 20. When z is less than zero, $f(z)$ equals zero; when z is more than or equal to zero, $f(z)$ equals z . Both the derivative and the function are monotonic. However, the problem is that all of the negative values instantly become zero, which makes it harder for the model to correctly fit or train on the data. This indicates that any negative input to the ReLU activation function immediately causes the value to become zero in the graph, which in turn has an impact on the final graph by improperly mapping the negative values.



Figure 20. ReLU Activation Function (Source: www.encord.com)

Nonlinear activation functions address the constraints and challenges of simpler activation functions, such as the vanishing gradient problem. Non-linear activation functions have various advantages, including the ability to assist backpropagation and stacking. Non-linear combinations and functions are employed across a network, which implies that while developing and training a model, just the weights and biases need to be adjusted, and the outputs are represented as functional computations. In other words, when non-linear activation functions are utilized, everything that enters, passes through, and exits a neural network can be measured more efficiently, thus the equations are changed until the desired outputs are obtained. There are several other non-linear activation functions, as shown in figure 21 and 22.

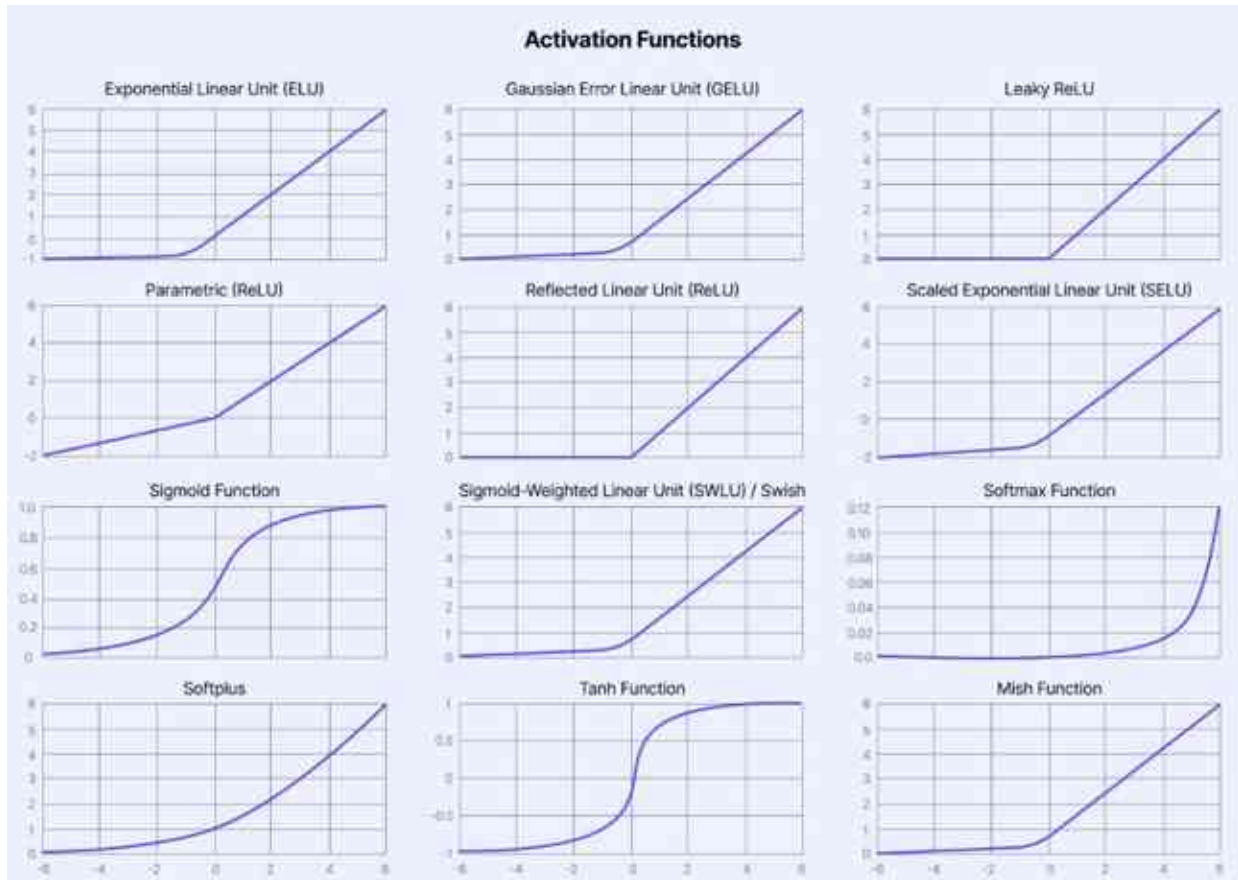


Figure 21. Non-linear Activation Functions (Source: www.encyord.com)

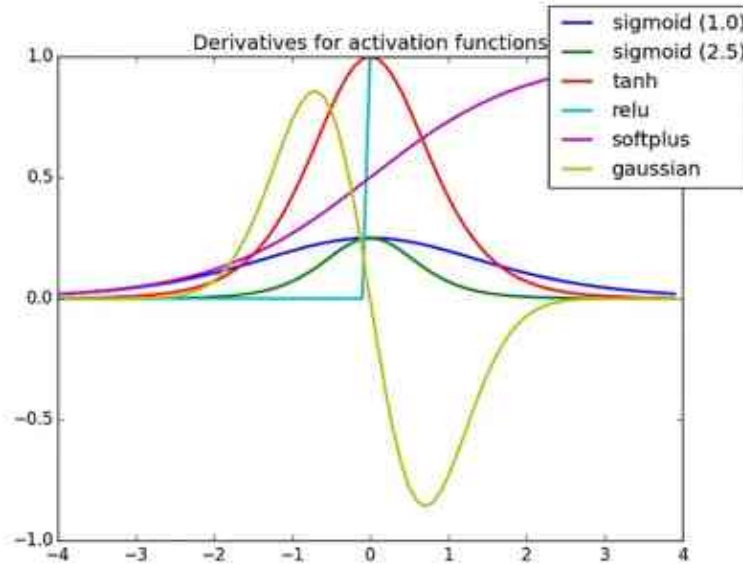


Figure 22. Derivative of Activation Functions (Source: www.towardsdatascience.com)

3.5.2. Types of neural network

The most crucial stage in creating a network is selecting its structure. In this context, the first step in identifying network structure is choosing the type of network. Each type of neural network has its strengths and weaknesses, making them suitable for different applications and types of data. In the following, some of the most dominant artificial neural networks will be discussed briefly.

a) Perceptron: Perceptron is among the earliest and most basic artificial neural network types. In terms of modern deep learning model complexity, the perceptron is extremely simple. However, the techniques used in its design are more applicable to advanced deep network topologies. It is a supervised learning binary classification method invented by Frank Rosenblatt (1958). It categorizes input data into one of two distinct states using a training technique on previous input data. The perceptron attempts to partition the input data using a linear decision boundary. Support vector classifiers, another type of supervised learning method, perform a similar technique. It assigns weights to a set of scalar input features as well as a constant “bias” term. Figure 23 shows a schematic diagram of a perceptron.

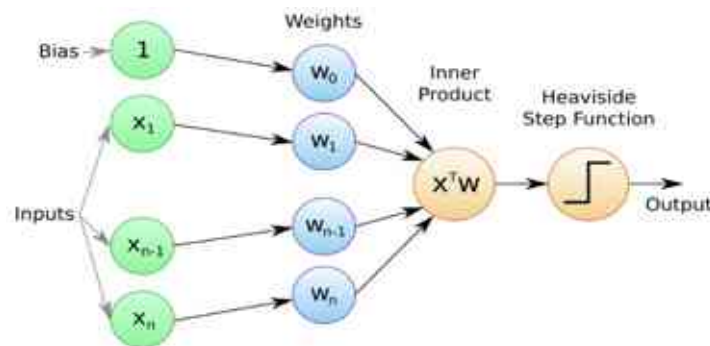


Figure 23. Schematic Diagram of a Perceptron (Source: www.medium.com)

b) Feedforward Neural Networks (FNNs): One of the most fundamental kinds of artificial neural networks developed to date is a feedforward neural network. Information transfers in a single path in this network: forward from the input nodes to the output nodes, passing through any hidden nodes that may exist. The network is free of loops and cycles. Compared to their more complex cousins, such as recurrent and convolutional neural networks, feedforward neural networks are the original type of artificial neural network to be created. A feedforward neural network contains two phases including feedforward and backpropagation. In the feedforward phase, input data is fed into the network and propagated forward through it. At each hidden layer, the weighted total of the inputs is computed and processed through an activation function, introducing nonlinearity into the model. This procedure will continue until the output layer is reached and a forecast is formed. In the Backpropagation Phase, once a prediction is produced, the error (the difference between the predicted and actual outputs) is determined. This error is then propagated back through the network, with weights changed to reduce it. Weights are frequently adjusted using a gradient descent optimization technique (Eldan, 2016). Figure 24 shows a schematic diagram of feedforward neural networks

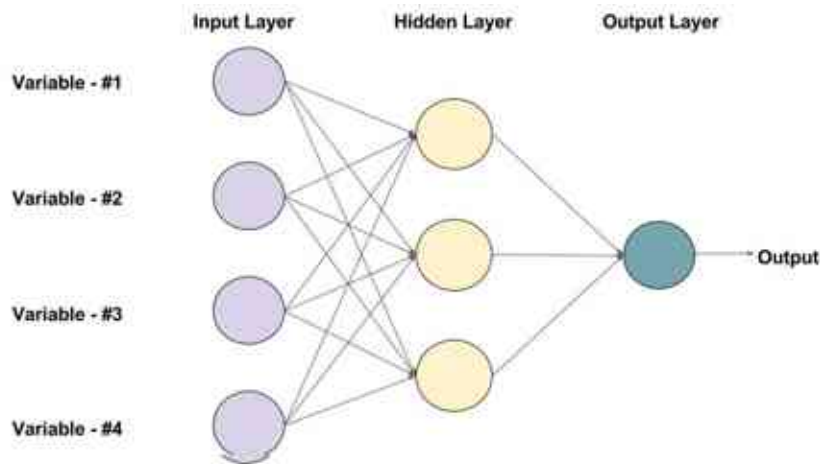


Figure 24. Schematic Diagram of Feedforward Neural Networks (Source: www.oksim.ua)

c) Convolutional Neural Network (CNNs): The developed form of artificial neural networks, known as convolutional neural networks (CNNs), is primarily used to extract features from grid-like matrix datasets. For instance, visual datasets with a lot of data patterns, such pictures or movies. Tens or even hundreds of layers can be found in a convolutional neural network, each of which is trained to recognize a unique characteristic of an image. Every training image is subjected to various resolutions of filters, and the result of every convolved image serves as the input for the subsequent layer. The filters can begin with relatively basic criteria, such edges and brightness, and progress in sophistication to include features that specifically identify the object (O'shea, 2015). Figure 25 shows simple CNN architecture.

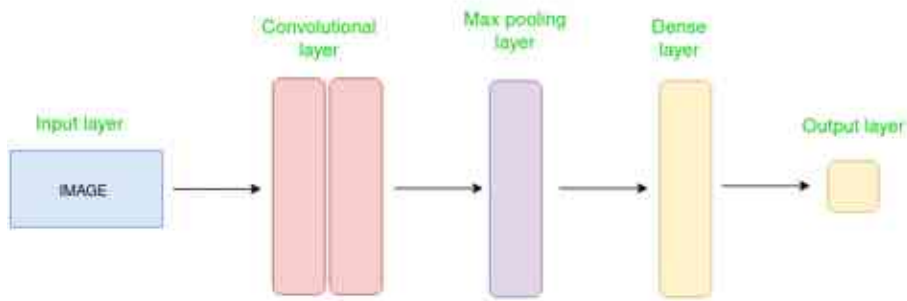


Figure 25. Simple CNN Architecture (Source: www.geeksforgeeks.org)

d) Recurrent Neural Networks (RNNs): One kind of artificial neural network that makes use of sequential or time series data is the recurrent neural network (RNN). These deep learning algorithms are integrated into well-known programs like Siri, voice search, and Google Translate. They are frequently utilized for ordinal or temporal problems like language translation, natural language processing (nlp), speech recognition, and picture captioning. Recurrent neural networks (RNNs), like feedforward and convolutional neural networks (CNNs), learn from training data. Their ability to use information from previous inputs to affect the present input and output sets them apart. Recurrent neural networks rely on the previous parts in the sequence to determine their output, in contrast to typical deep neural networks, which presume that inputs and outputs are independent of one another (Schmidt, 2019).

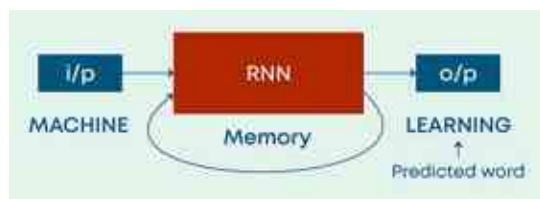


Figure 26. Recurrent Neural Network

e) Modular Neural Network (MNNs): An advanced method of creating neural networks is the modular neural network (MNN), which breaks down a large, difficult problem into smaller, easier-to-manage subproblems. Every one of these sub-issues is managed by an independent module, or sub-network, that is task specific. The ultimate output can be formed by combining the outputs of these modules, each of which can function independently. Better scalability, simpler training, and enhanced interpretability are just a few benefits of this modularity. The architecture of an MNN varies greatly depending on the problem being solved and the design decisions taken by the network's creators. Common architectural patterns include hierarchical systems, in which higher-level modules integrate the outputs of lower-level modules, much like the human brain processes information in stages. Parallel structures enable modules to function individually, with outputs aggregated later, which is beneficial for jobs that can be broken down into discrete sub-tasks. Some MNNs may incorporate recurrent connections, in which the output of one module is sent back into another, allowing for dynamic information interchange and temporal processing (Chen K. , 2015). Figure 27 shows a modular neural network.

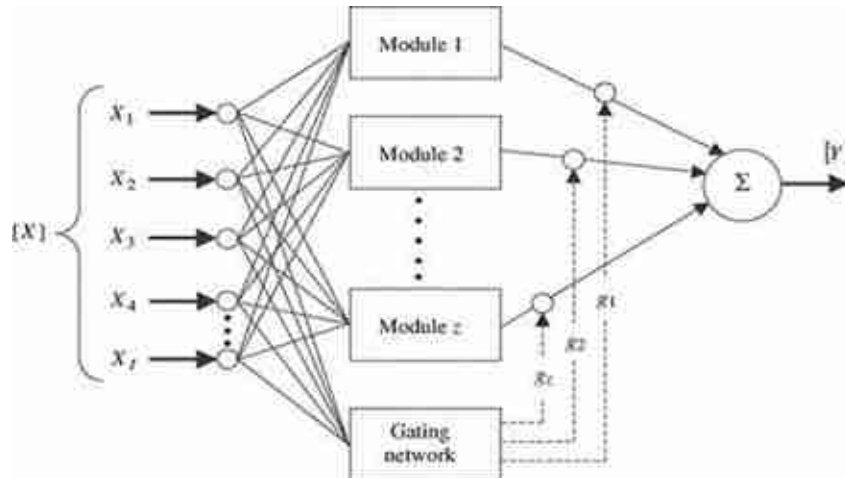


Figure 27. Modular Neural Network (Source: www.wikidocs.net)

There are several other types of neural networks, each with its own architecture designed for certain tasks and applications. Each type of neural network has distinct benefits in data processing and learning, making it appropriate for a variety of applications such as image recognition, natural language processing, and time-series prediction. Figure 28 depicts a nearly complete chart of different neural network types, displaying their designs and emphasizing the variety of structural distinctions that enable their specific functionality.

A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

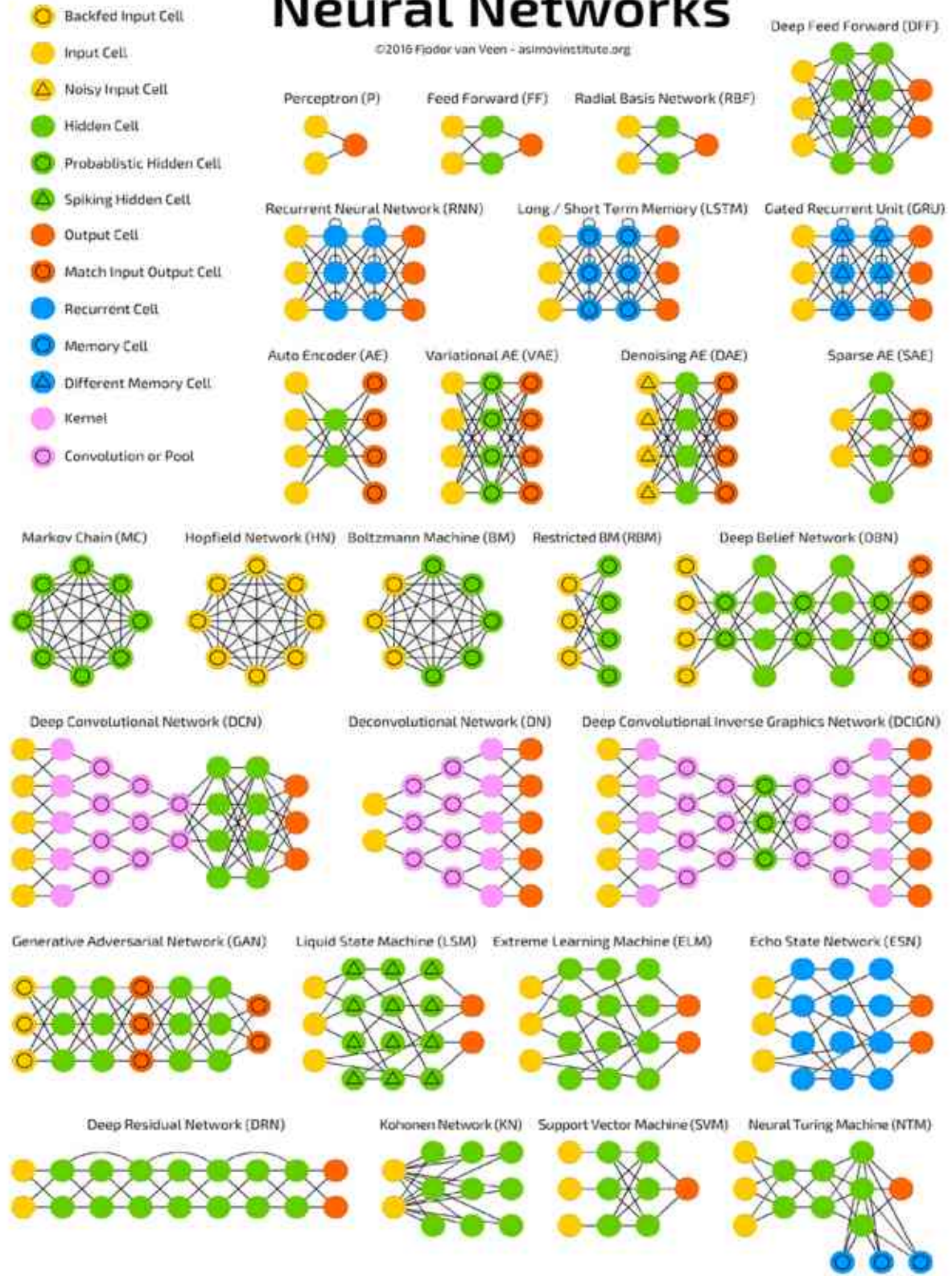


Figure 28. A Mostly Complete Chart of Neural Networks (Source: www.asimovinstitute.org)

3.5.3. Implementation of artificial neural network to estimate soil parameters

The goal of this study was to use artificial neural networks (ANNs) to estimate hydraulic conductivity based on soil component weight percentages. A total of 205 soil samples were collected, with each sample's hydraulic conductivity and weight percentages of various soil components (such as sand, silt, clay, and gravel) determined. The dataset was divided into two subsets: 70% (144 samples) was utilized to train the neural network, and the remaining 30% (61 samples) was set aside for testing and validation. The data was divided at random to achieve a representative distribution of samples across both subsets. This split ratio seeks to offer enough data for training while keeping enough samples to assess the network's performance.

A feedforward neural network was selected because of its simple architecture and its efficacy in related applications. FFNs' flexibility enables the integration of many input variables as well as model architecture optimization to obtain greater predictive performance when compared to classic regression techniques. This neural network is best suited for tabular data, as each input feature (soil component) helps estimate a continuous output (hydraulic conductivity). FFNs are also well-suited to this task because they can capture complex nonlinear relationships between input variables (such as soil components) and output variables (such as hydraulic conductivity) without requiring prior assumptions about the underlying data distribution. Furthermore, FFNs are very versatile and can handle big datasets with a variety of input features, making them ideal for dealing with the multidimensional nature of soil properties and interactions. It also was taken into account that; previously conducted studies have shown that FFNs have been widely used in soil science and hydrology research because of their capacity to accurately predict complicated and nonlinear interactions between soil parameters.

As mentioned earlier, the network's design consists of three layers: input, hidden, and output. MATLAB software, notably the Neural Network Toolbox, was utilized in this study to code and build the neural network. It offers a user-friendly environment for developing, training, and evaluating neural networks, as well as support for function approximation and nonlinear regression problems. The input layer contained nodes representing the number of soil components considered (e.g., sand, silt, clay, gravel). The output layer contained a single node indicating the estimated hydraulic conductivity. The neural network was trained using the Levenberg-Marquardt backpropagation algorithm which is a popular method for training FFNNs by MATLAB because of its efficiency and speed in convergence to a solution. This method combines the benefits of gradient descent and Gauss-Newton methods, making it ideal for training networks on the dataset. After choosing the feedforward method for simulating soil hydraulic conductivity based on soil components, the investigation progressed to optimizing the neural network architecture by testing with different numbers of layers and neurons. This strategy is based on the observation that neural network performance can be highly sensitive to the architecture used, and determining the ideal configuration is critical for making correct predictions. Each architecture's performance was evaluated iteratively using criteria like prediction accuracy. The goal of carefully evaluating a variety of designs was to find the configuration that produced the optimal balance of model complexity and predictive performance.

The same approach was used in other neural network models to predict the soil's shear strength characteristics, namely cohesion and internal friction angle. For this, values from experimental

procedures were applied to a dataset consisting of 95 soil samples for each parameter. An input layer representing several soil components (sand, silt, clay, and gravel) and a single output node corresponding to the expected Cohesion or internal friction angle were used in the creation of the neural networks. To ensure a representative distribution across both subsets, the datasets for each parameter were divided into two groups: 70% for training and 30% for testing and validation. The feedforward neural network (FFN) technique, which had previously been validated for estimating hydraulic conductivity, was used separately to predict cohesion and internal friction angle. Each FFN's architecture was adjusted using iterative testing with varying numbers of layers and neurons, with the goal of balancing model complexity and prediction accuracy. The construction, training, and evaluation procedures were facilitated by MATLAB's Neural Network Toolbox, which utilized the Levenberg-Marquardt backpropagation algorithm for its convergence efficiency.

In this chapter, the methodology employed for the research was outlined. Data collection and preparation processes were described, followed by the laboratory experiments conducted to obtain relevant soil parameters. An overview of data trends was provided to highlight key patterns observed in the dataset. Multiple linear regression was utilized to develop predictive models, while artificial neural networks (ANNs) were introduced as an advanced technique. The concept of ANNs was explained, along with a discussion of different types of neural networks. Finally, the implementation of ANNs for estimating soil parameters was detailed, demonstrating their application in improving the accuracy of predictions.

4- Data analysis and results

I used several approaches, such as indirect methods, the multiple linear regression (MLR) method, and the artificial neural networks method, to create a comprehensive comparison. Empirical calculations based on easily measured soil parameters are used in indirect approaches. The MLR approach entails creating a statistical model that links several soil properties to hydraulic conductivity, improving forecast accuracy by taking into account the combined impact of numerous variables. The primary objective was to identify the optimal ANN model architecture that provides the most accurate and reliable predictions. The ANNs method is a complex strategy that may increase prediction precision. It models complex, nonlinear relationships among soil parameters and hydraulic conductivity by utilizing machine learning methods. This section compares the ways in which these approaches work in order to assess how well they predict hydraulic conductivity. The task of selecting the optimal ANN architecture for predicting hydraulic conductivity is inherently challenging due to the numerous possible configurations and the need to balance model complexity with generalization ability. Below is a detailed comparison and analysis of the results obtained from each method.

4.1. Hydraulic conductivity values obtained by the experiments

Figure 29 illustrates the $(-\text{LogK})$ values for 205 soil compositions, sorted in descending order. These values were obtained through laboratory experiments as discussed in the previous chapter. Each soil composition varied in component percentages, specifically including 0, 10, 15, 20, 25, 30, 40, 50, 75, and 100 percent, as detailed in Annex 1. The naming order of the samples in the result tables was determined post-analysis, ensuring that they are presented in a systematic manner based on their $(-\text{LogK})$ values for clearer interpretation and comparison.

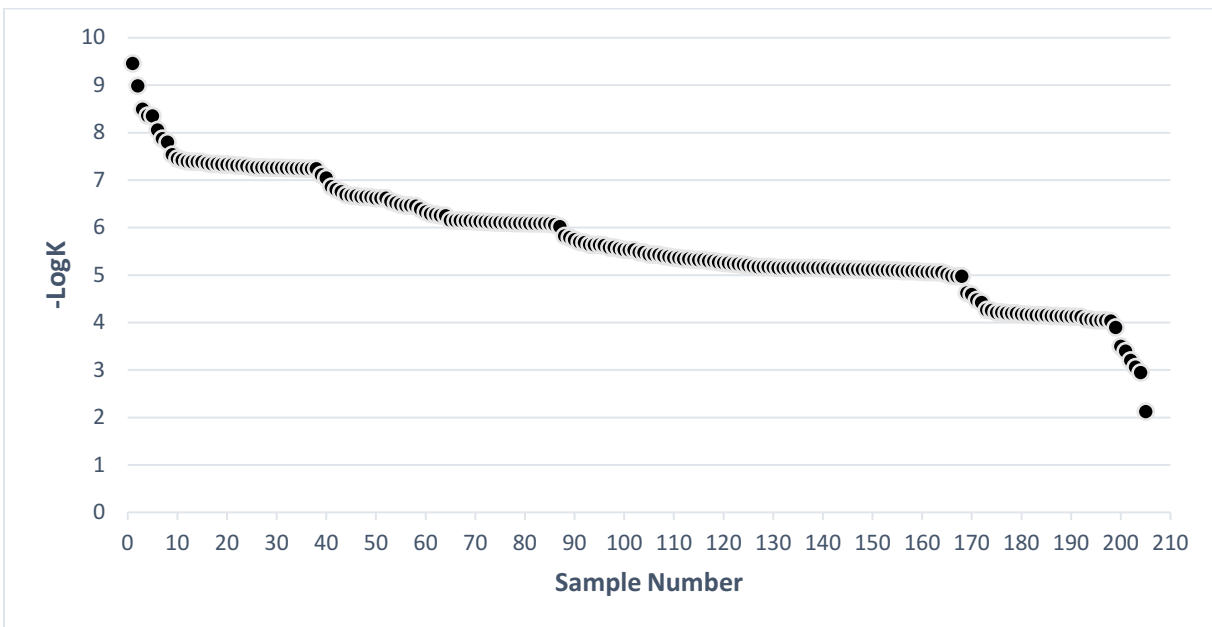


Figure 29. $(-\text{LogK})$ Values Obtained by Laboratory Tests (Source: Own Compilation)

4.2. Estimation of hydraulic conductivity by indirect method

As it was mentioned in second chapter, Numerous empirical formulas have been proposed by scientists to predict hydraulic conductivity for both fine-grained and coarse-grained soils. These formulas typically utilize soil properties such as grain size distribution, porosity, void ratio, and plasticity characteristics. For coarse-grained soils, formulas often rely on parameters like effective grain size and uniformity coefficient, while for fine-grained soils, plasticity index and liquid limit are more commonly used. The diversity in empirical approaches reflects the complex nature of soil properties and their impact on hydraulic conductivity. To check the validation of the empirical formula, I used the Carrier and Beckman method to predict hydraulic conductivity for a range of fine-grained soil samples with different compositions. This method was chosen due to its close alignment with the initial properties of the samples, ensuring a relevant comparison. This method is particularly suitable for remolded or disturbed samples, making it highly applicable in practical scenarios where obtaining undisturbed samples is challenging. Its reliance on easily measurable soil properties like plasticity index and liquid limit makes it versatile for various types of clays, providing a reliable estimate of hydraulic conductivity under different conditions. The Carrier and Beckman method, developed in 1984, offers an empirical formula which is mentioned as equation 10 specifically tailored for fine-grained soils, particularly clays.

$$K = 0.174 \times \frac{[e - 0.027 (PL - 0.242PI) / (PI)]^{4.29}}{(1+e)} \quad (m/s) \quad (\text{Eq.10})$$

where k is the hydraulic conductivity, PL is the liquid limit, PI is the plasticity index, and e is the void ratio.

Figure 30 presents a comparison between the measured hydraulic conductivity data and the values predicted by the Carrier and Beckman method. The plot shows the measured data on the x-axis and the predicted data on the y-axis, with a trendline indicating the correlation between the two datasets. The coefficient of determination (R^2) is 0.5187, suggesting a moderate correlation. While an R^2 value of 0.51 indicates some level of predictive capability, it is not particularly strong, suggesting that the empirical formula may not fully capture the variability in the hydraulic conductivity of the soil samples. This discrepancy highlights the need for further refinement of empirical models or the use of complementary methods to improve the accuracy of hydraulic conductivity predictions for fine-grained soils. Furthermore, a significant portion of the data points, as shown in Figure 30, fell below the bisector line, demonstrating a persistent underestimation of the values predicted by the Carrier and Beckman approach in comparison to the measured data. So, the empirical formula might not be completely reliable for some factors impacting hydraulic conductivity in the samples, as suggested by this consistent underestimating.

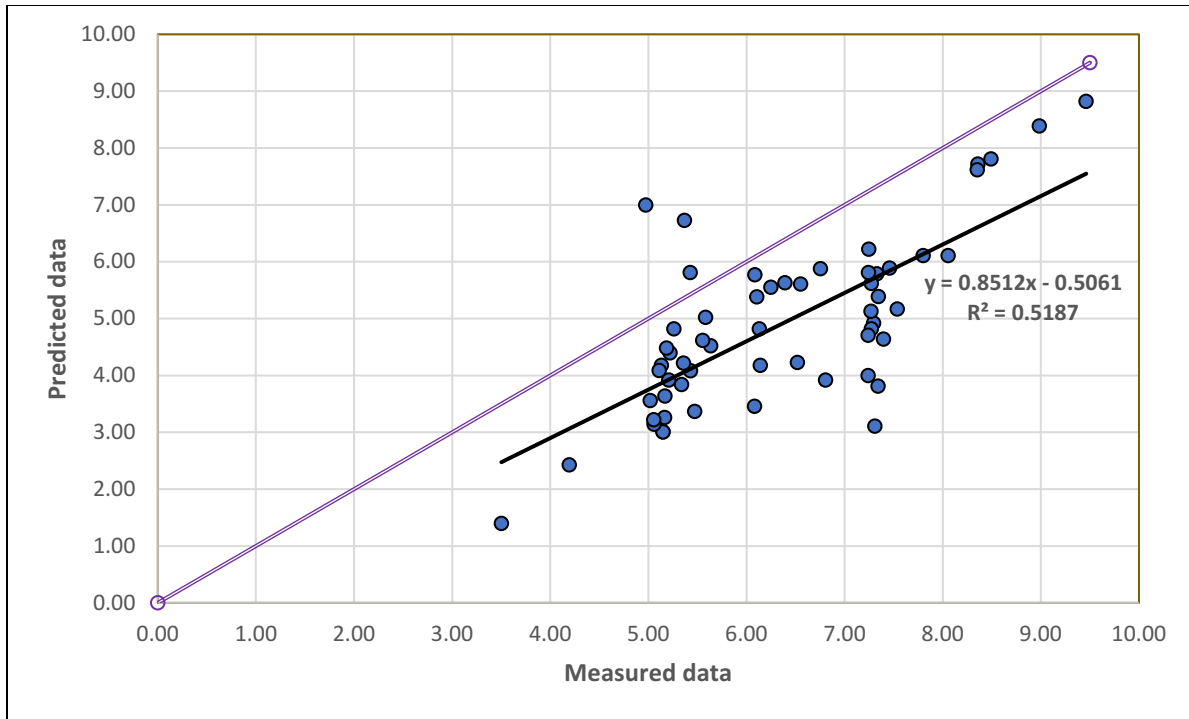


Figure 30. Comparison of (-LogK) Values Obtained by Empirical Formulae and Laboratory Tests (Source: Own Compilation)

4.3. Estimation of hydraulic conductivity by Multiple linear regression analysis

To conduct the Multiple linear regression analysis, I used SPSS software. Initially descriptive statistics obtained to gain insight into the central tendency and dispersion of the variables, providing a comprehensive overview of the data's characteristics. According to Table2, the standard deviation for normalized hydraulic conductivity is approximately 1.212.

Table 2. Descriptive statistics

Variable	Observations	Obs. with missing data	Obs. without missing data	Minimum	Maximum	Mean	Std. deviation
-Log K	205	0	205	2.127	9.462	5.725	1.212
Clay (%)	205	0	205	0.000	100.000	24.317	17.553
Gravel (%)	205	0	205	0.000	100.000	24.951	17.836
Sand (%)	205	0	205	0.000	100.000	25.439	18.486
Silt (%)	205	0	205	0.000	100.000	25.293	18.522

Before performing the regression analysis, the existence of a linear relationship between each of the independent variables and the dependent variable was investigated by using the "Scatter Plot" and also calculating the correlation coefficient. At figure 31, a linear relationship can be seen

between each of the independent variables and the dependent variable. Notably, there is a positive correlation observed for clay and silt, whereas an inverse relationship is evident in the case of sand and gravel.

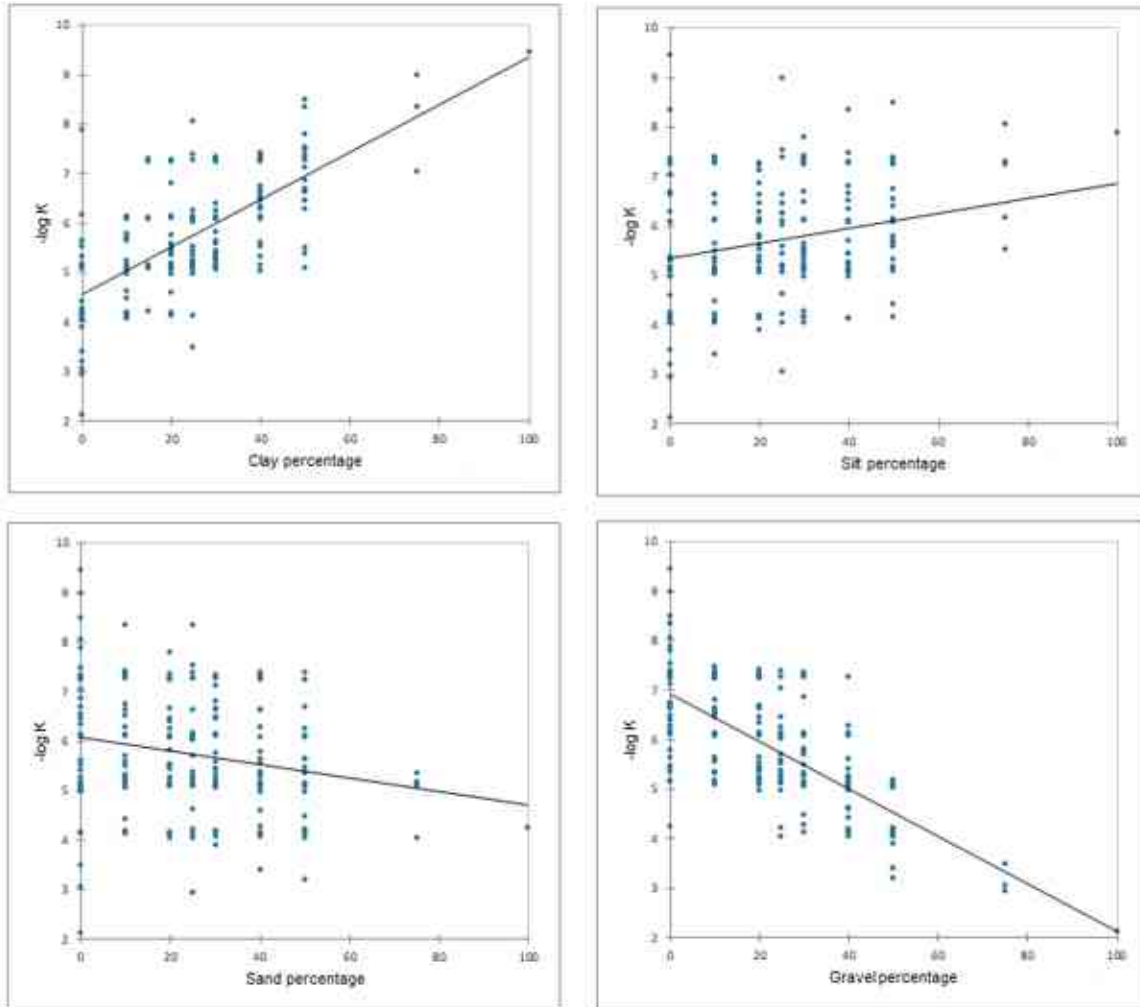


Figure 31. Relationships Between the Independent Variables and the Dependent Variable (Source: Own Compilation)

In the next step, the value of Pearson correlation coefficient between independent and dependent variables was examined. The correlation coefficient was calculated according to the “Pearson’s correlation coefficient” formula. If the correlation coefficient value between both variables is significant in the two-tailed test, they are marked with *. The output can be seen in Table 3. In the subtitle of this table, the sign ** indicates the significance of the statistical test (rejecting the null hypothesis or the correlation coefficient being meaningless) at the error level of 0.01 or the test level of 0.99. According to the table, there is a good linear relationship between the dependent variable and all other variables. Additionally, the strong correlations between Clay, Gravel, and ($-\log K$), suggest that these variables have more weight importance than the others in terms of their influence on the hydraulic conductivity parameter.

Table 3. Variables Correlations of MLR Model

Variable		Clay	Silt	Sand	Gravel	-log K
-log K	Pearson Correlation	.695**	.232**	-.208**	-.709**	1
	Sig. (2-tailed)	0.000	0.001	0.003	0.000	
	N	205	205	205	205	205

** . Correlation is significant at the 0.01 level (2-tailed).

After performing the multiple regression, the desired results were extracted, which I will continue to interpret. According to Table 4, since the correlation coefficient ($R=0.906$) and determination coefficient (R Square= 0.822) as well as adjusted determination coefficient (Adjusted R Square= 0.819) were calculated, it seems that the regression model is appropriate. The closer these values are to 1, the more the model expresses the relationship between the dependent and independent variables. In other words, the regression model was able to cover or express a greater percentage of changes in the dependent variable.

Table 4. MLR Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	DW	MSE
1	0.906	0.822	0.819	0.515	1.797	0.266

In Table 6, the variance analysis for the regression model has been done. Considering the size of F and the value of $Sig < 0.0001$, I conclude that the regression model will be appropriate. Because most of the changes in the dependent variable have been seen in the regression model. This means that the contribution of the model (Regression) in the total changes that can be seen in the last row (Total) of the column (Sum of Squares) is much higher than the contribution of error or residuals.

Table 5. Analysis of variance of MLR Model.

Source	DF	Sum of squares	Mean squares	F	Sig	p-values signification codes
Model	3.000	246.050	82.017	308.691	<0.0001	***
Error	201.000	53.404	0.266			
Corrected Total	204.000	299.454				
<i>Computed against model $Y=Mean(Y)$</i>						
<i>Signification codes: $0 < *** < 0.001 < ** < 0.01 < * < 0.05 < . < 0.1 < \circ < 1$</i>						

Finally, the scatter plot depicted in figure 32 derived from the multiple linear regression (MLR) analysis which represents the relationship between predicted and real data. The coefficient of determination (R^2) is 0.81, indicating that 82% of the variability in the real data is explained by the model. This value suggests that the MLR model has a relatively good fit and effectively captures the relationship between the independent variables and the dependent variable.

Upon closer inspection of the data distribution, both the learning and validation samples align closely with the regression line, though some scatter is evident, particularly for higher values of the dependent variable. The consistent spread of points around the regression line across both samples implies that the model generalizes well to unseen data, as evidenced by the validation sample. While the model demonstrates a good fit, I desired a more accurate estimation for hydraulic conductivity. Therefore, to improve the predictive performance, I conducted the estimation using an artificial neural network (ANN) approach, which will be discussed in the next section.

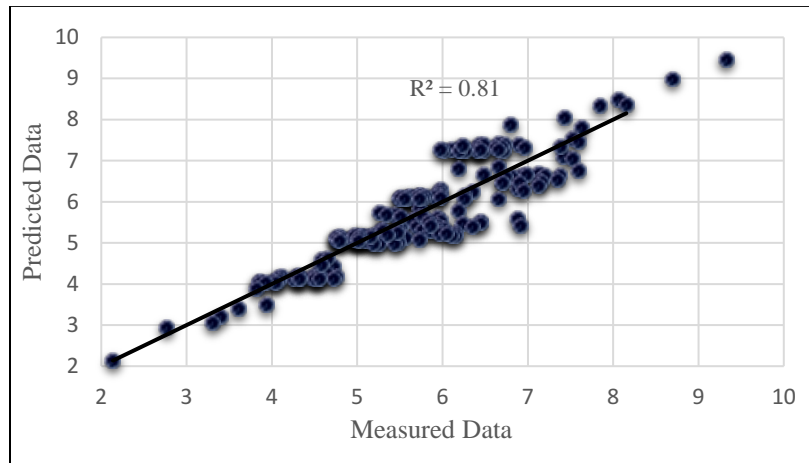


Figure 32. Multiple Linear Regression Analysis Result (Source: Own Compilation)

4.4. Artificial Neural Network model for hydraulic conductivity parameter

In this section, I present the results of the analysis for estimating hydraulic conductivity based on the weight percentages of soil components (clay, silt, sand, and gravel) using artificial neural networks. The primary goal was to identify the optimal ANN model architecture that provides the most accurate and reliable predictions. The task of selecting the optimal ANN architecture for predicting hydraulic conductivity is inherently challenging due to the numerous possible configurations and the need to balance model complexity with generalization ability. I evaluated multiple models with varying configurations of hidden layers and neurons.

I began my exploration with two neurons in a single layer. Subsequently, I expanded my investigation by varying the number of neurons within this initial layer. Following this, I introduced an additional layer, increasing the depth of the network to two hidden layers. I conducted several iterations, adjusting the number of neurons in each layer to discern their impact on performance. In pursuit of further insights and more confident outcomes, I conducted a test by

introducing a third hidden layer. Despite altering the number of neurons across these layers, my efforts did not yield any noticeable improvements in performance.

All in all, I tested a total of ten different ANN models, each with a unique architecture. The architectures varied in terms of the number of hidden layers and the number of neurons within those layers. The selection of the optimal artificial neural network structure was guided by key performance metrics including the coefficient of determination (R^2), Root Mean Square Error (RMSE), and Mean Squared Error (MSE). These metrics were employed as indicators of the model's predictive accuracy and goodness of fit. R^2 , also known as the coefficient of determination, measures the proportion of the variance in the dependent variable that is predictable from the independent variables, thus assessing the model's explanatory power. RMSE represents the square root of the average squared differences between predicted and observed values, providing a measure of the model's prediction error. Lastly, MSE calculates the average of the squared differences between predicted and observed values, offering insight into the variance of the prediction errors. In the context of ANN modeling, these metrics serve as crucial evaluation tools to iteratively refine and optimize the network architecture, ensuring robust and reliable predictions. The equations 11, 12 and 13 which are mentioned below correspond to these metrics.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (\text{Eq.11})$$

$$MSE = \frac{\sum_{i=1}^n (y_i - x_i)^2}{n} \quad (\text{Eq.12})$$

$$R^2 = \frac{\sum_{i=1}^n (y_i - x_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (\text{Eq.13})$$

Which, n represents the number of measurements, y_i and x_i are the observed and predicted values of the dependent variable, respectively, and \bar{y} is the mean of the observed values of the dependent variable.

Table 6 shows the Regression analysis for all ten models across training, validation, testing, and overall datasets. Model number 4, highlighted in the table, performed the best across all phases, with highest R values. Figure 33 shows regression plots for the best-performing model (Model 4), with separate plots for training, validation, testing, and all data combined. The other model's plot is mentioned in Annex3. The (R^2) value for the best model is shown in figure 34 which equals 0.92. This high value of (R^2) indicates that this architecture is particularly well-suited for this project.

Table 6. ANN Models for Hydraulic Conductivity Prediction.

Model no	output	Number of hidden layers	Number of neurons	R Values			
				Training	Validation	Testing	All
1	-Log K	1	2	0.67245	0.7122	0.69542	0.6829
2	-Log K	1	4	0.7157	0.72214	0.6519	0.70154
3	-Log K	2	2,2	0.82521	0.8021	0.78025	0.8102
4	-Log K	2	1,3	0.94939	0.96013	0.9373	0.95471
5	-Log K	2	4,1	0.77591	0.7928	0.84146	0.79013
6	-Log K	2	3,4	0.86813	0.84292	0.803	0.8562
7	-Log K	3	2,5,1	0.88235	0.7452	0.80187	0.84094
8	-Log K	3	4,1,6	0.8125	0.72491	0.8232	0.80191
9	-Log K	3	3,2,2	0.87513	0.81793	0.7029	0.85517
10	-Log K	3	1,4,1	0.88922	0.9032	0.87961	0.89183

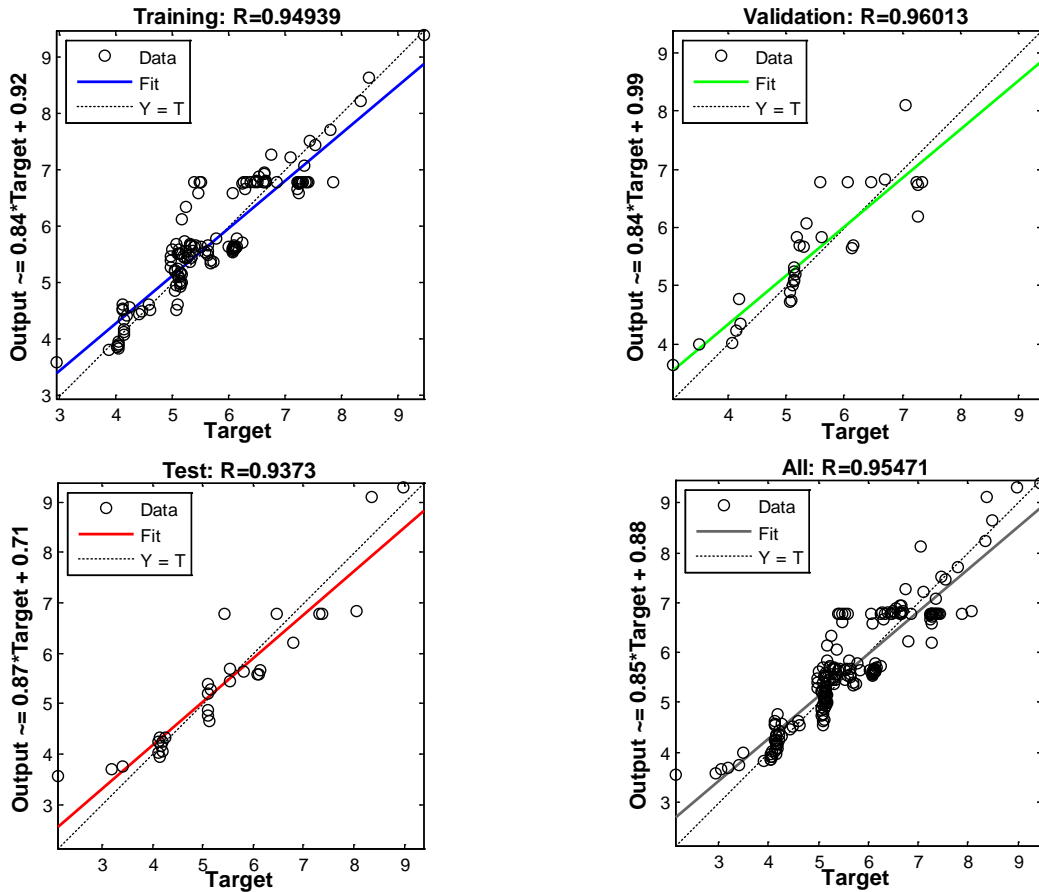


Figure 33. Regression Plots of the Best-Performing Model for (-Log K) Prediction (Source: Own Compilation)

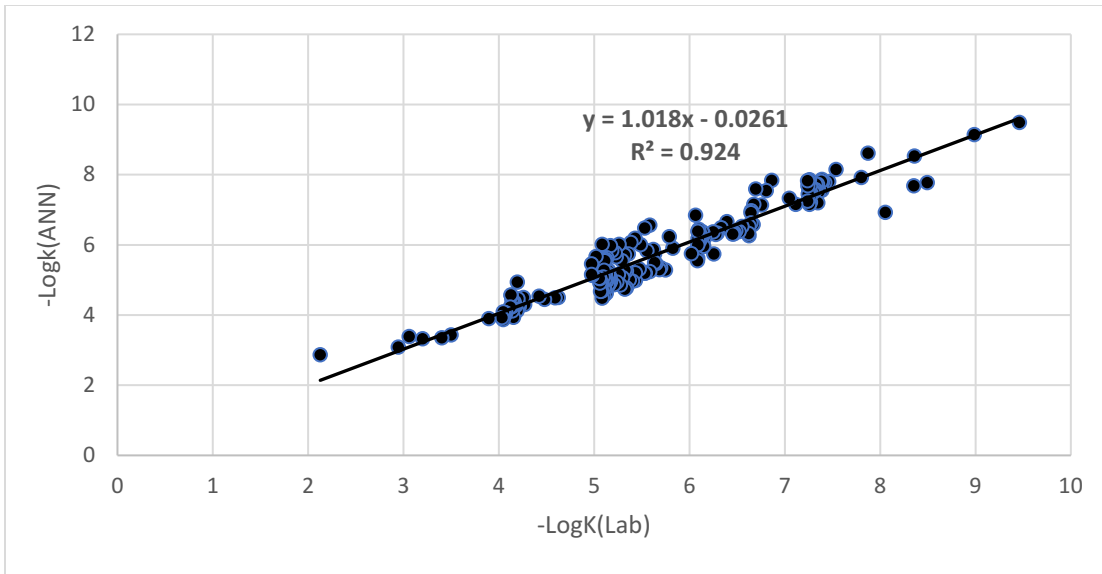


Figure 34. Coefficient of Determination Value for the Best Model of ANN (Source: Own Compilation)

In another point of view, the results emphasized that the best validation performance was achieved at epoch 4, with a Mean Squared Error (MSE) of 0.27001. Figure 35 illustrates the MSE against the number of epochs, ranging from 0 to 10. The MSE for the training, validation, and test sets decreased rapidly in the initial epochs and then stabilized, demonstrating that the network quickly learned and subsequently fine-tuned its parameters. The validation error, which is crucial for assessing the network's generalization ability, was lowest at epoch 4, indicating the optimal performance of the network on unseen data at this point.

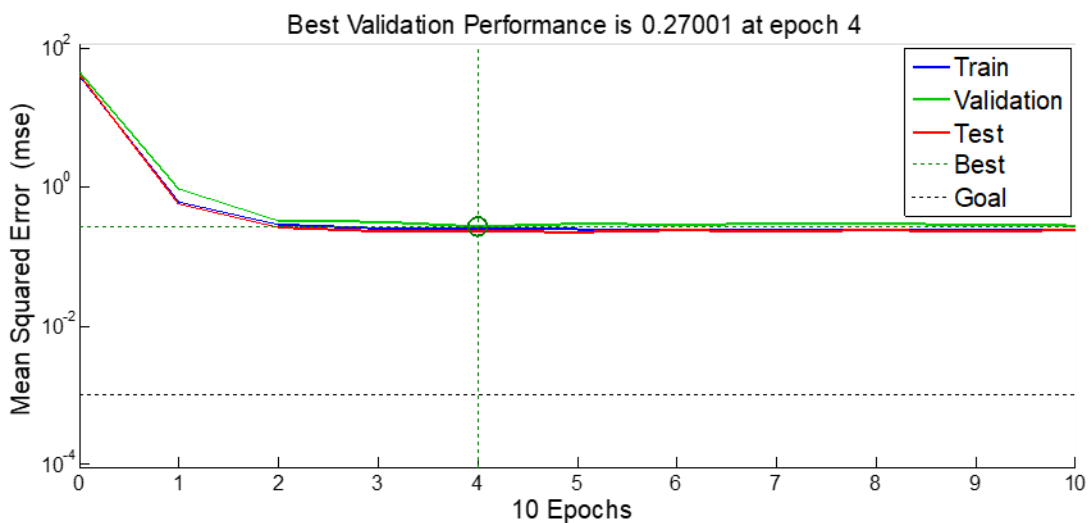


Figure 35. Mean Squared Error for all Models (Source: Own Compilation)

Figure 36 shows the error histogram, displaying the distribution of residuals (errors) across the training, validation, and testing datasets. The residuals represent the differences between the actual (-log K) values and the predicted values from the neural network model. Most of the residuals are

centered around zero, indicating that the model's predictions are, on average, close to the actual values. The presence of a small number of larger residuals suggests occasional deviations, but these are relatively infrequent. The histogram is color-coded, with blue, green, and red bars representing training, validation, and testing errors, respectively, and the presence of zero error instances is also highlighted.

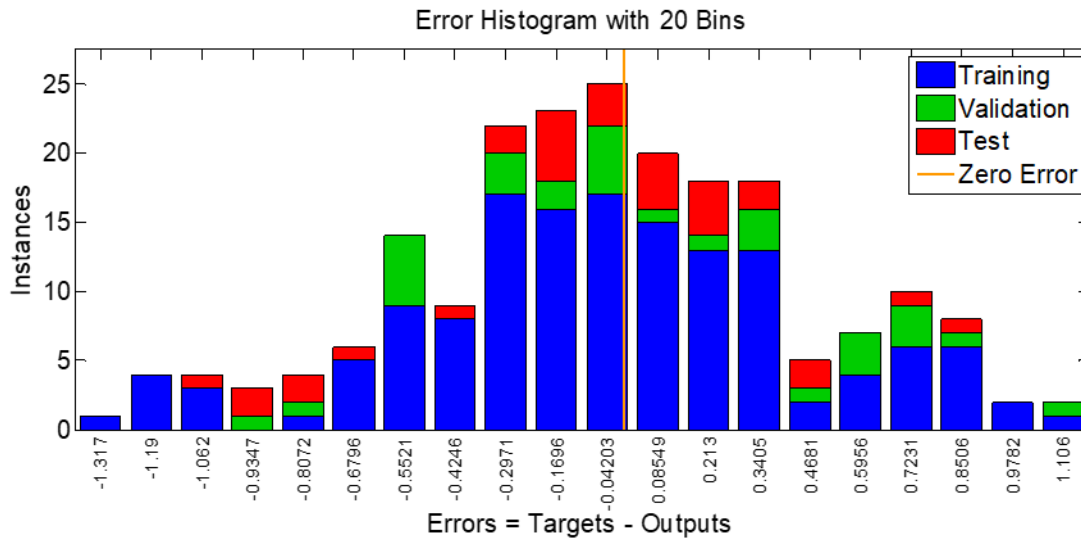


Figure 36. Distribution of residuals (errors) (Source: Own Compilation)

Thus, the analysis shows that ANN model number 4 with two hidden layers correspondence number of neurons is the most effective for predicting hydraulic conductivity based on soil component weight percentages.

To begin the coding, I called measured data from an Excel file containing the experimental data according to figure 37. I specified the first sheet and the range 'A2' to import the data into MATLAB using the 'xlsread' function. The inputs, which are the soil component percentages, were stored in the variable 'inputs' by selecting the first four columns of the data. The target output parameter (-Log K) was stored in the variable 'targets' by selecting the fifth column.

```

9      %%%%***** Extracting measured data *****%%%%%%%%
10     filename = 'data.xlsx';
11     sheet = 1;
12     range = 'A2:F206';
13     data = xlsread(filename, sheet, range);
14
15     inputs = data(:, 1:4);
16     targets = data(:, 5);
17

```

Figure 37. MATLAB Code for Importing Experimental Data from an Excel File (Source: Own Compilation)

The neural network architecture used for estimating the hydraulic conductivity of soil is depicted in figure 38. The network consists of an input layer with four nodes representing the weight

percentages of clay, silt, sand, and gravel, followed by two hidden layers, each with different numbers of neurons, and a single-node output layer that estimates $-\text{Log } K$. As it was mentioned earlier, the design with [1 3] neurons achieved the optimal balance of underfitting and overfitting for the data.

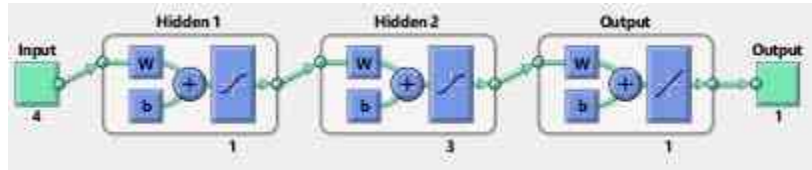


Figure 38. Neural Network Structure for $(-\text{Log } K)$ Prediction (Source: Own Compilation)

Hence, I defined the structure of the neural network using the ‘feedforwardnet’ function. According to figure 39, I set the training parameters to display the training window ‘(showWindow = true)’, with a maximum of 1000 epochs and a performance goal of 0.001. I then trained the network using the ‘train’ function, passing the network, inputs, and targets as arguments.

```

Hasan.m* +
23 %%% %***** The Neural Network Structure *****%%
24 %*** layers & Neurons ***%
25 - net = feedforwardnet([1 3]);
26 - net.trainParam.showWindow = true;
27 - net.trainParam.epochs = 1000;
28 - net.trainParam.goal = 0.001;
29
30 - net = train(net, inputs', targets');
31

```

Figure 39. MATLAB Code for Defining and Training the Neural Network Using the Feedforward Function (Source: Own Compilation)

The neural network model includes specific weights and biases between layers, with activation functions applied at each layer. biases, and activation functions used in the neural network to predict hydraulic conductivity were derived by the code as show in figure37. These elements are essential because they describe how the network makes predictions based on incoming data processing. Through training, the network learns parameters called weights and biases that enable efficient mappings of inputs to outputs. The model gains non-linearity from the activation functions, which allows it to recognize intricate patterns in the data. Understanding and interpreting the behavior of the model, improving debugging, optimizing model performance, and ensuring that the network's predictions are founded on significant modifications of the input data are all made possible by being aware of these components. Additionally, reconstructing the program in the future depends on having an extensive understanding of the weights, biases, and activation functions. Mathematical explanations will be discussed in the following.

Starting from the input layer to the first hidden layer, the weights and bias were:

$$W_1 = [-1.6529 \quad -0.38346 \quad 0.27885 \quad 1.3238]$$

$$b_1 = [0.33056]$$

The input to the first hidden layer, Z_1 , was computed using the equation 14.

$$z_1 = W_1 \cdot x + b_1 \quad (\text{Eq.14})$$

Where W_1 is the weight matrix and b_1 is the bias vector. The used activation function for the first hidden layer was ‘tansig’ which is hyperbolic tangent sigmoid. It applied to z_1 according to equation 15.

$$h_1 = \text{tansig}(z_1) = \frac{2}{1+e^{-2z_1}} - 1 \quad (\text{Eq.15})$$

Next, for the second hidden layer, the weights and biases were:

$$W_2 = \begin{bmatrix} 4.3343 \\ 9.2543 \\ -6.8083 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -4.9807 \\ -2.3088 \\ -5.8666 \end{bmatrix}$$

The input to the second hidden layer, $Z_{2,i}$, was computed using the equation 16.

$$z_{2,i} = W_{2,i} \cdot h_1 + b_{2,i} \quad (\text{Eq.16})$$

Where $W_{2,i}$ and $b_{2,i}$ are the weights and biases for the i-th neuron in the second hidden layer. According to equation 17, the tansig activation function was again applied to each $Z_{2,i}$ to obtain the outputs $h_{2,i}$.

$$h_{2,i} = \text{tansig}(z_{2,i}) = \frac{2}{1+e^{-2z_{2,i}}} - 1 \quad (\text{Eq.17})$$

Finally, in the output layer, the weights and bias were:

$$W_3 = [-1.5828 \quad -0.16039 \quad 0.39496]$$

$$b_3 = [-1.0544]$$

The input to the output layer, Z_3 , was computed using the equation 18.

$$z_3 = W_3 \cdot h_2 + b_3 \quad (\text{Eq.18})$$

where W_3 is the weight matrix and b_3 is the bias vector. The used activation function to obtain the final output (y) was ‘purelin’ which is the identity function and applied to z_3 according to equation 19.

$$y = \text{purelin}(z_3) = z_3 \quad (\text{Eq.19})$$

This process involves sequential matrix multiplications and the application of activation functions to propagate inputs through the network layers, transforming them into the final predicted output.

```
33
34 %% %***** Extracting Weights, Biases, and Activation Functions *****%%
35 % Weights and biases from input to first hidden layer
36 - weights_input_to_hidden1 = net.IW{1,1};
37 - biases_hidden1 = net.b{1};
38
39 % Weights and biases from first hidden layer to second hidden layer
40 - weights_hidden1_to_hidden2 = net.LW{2,1};
41 - biases_hidden2 = net.b{2};
42
43 % Weights and biases from second hidden layer to output layer
44 - weights_hidden2_to_output = net.LW{3,2};
45 - biases_output = net.b{3};
46
47 % Display weights and biases
48 - disp('Weights from input to first hidden layer:');
49 - disp(weights_input_to_hidden1);
50
51 - disp('Biases of first hidden layer:');
52 - disp(biases_hidden1);
53
54 - disp('Weights from first hidden layer to second hidden layer:');
55 - disp(weights_hidden1_to_hidden2);
56
57 - disp('Biases of second hidden layer:');
58 - disp(biases_hidden2);
59
60 - disp('Weights from second hidden layer to output layer:');
61 - disp(weights_hidden2_to_output);
62
63 - disp('Biases of output layer:');
64 - disp(biases_output);
65
66 % Display activation functions
67 - disp('Activation function of first hidden layer:');
68 - disp(net.layers{1}.transferFcn);
69
70 - disp('Activation function of second hidden layer:');
71 - disp(net.layers{2}.transferFcn);
72
73 - disp('Activation function of output layer:');
74 - disp(net.layers{3}.transferFcn);
```

Figure 40. MATLAB Code for Deriving Weights, Biases and Activation Functions. (Source: Own Compilation)

The program continued with defining the output as it is shown in figure 41. I wrote a section of the script that prompts the user to input the percentages of clay, silt, sand, and gravel. I included a check to ensure the sum of these percentages equals 100. If the condition is met, the script forms an array 'Percentage_composition' from the input values. The trained neural network then estimates the output parameter (-log K) based on these input percentages, and the estimated value is displayed. If the sum of the percentages is not 100, an error message is shown.

```

76 % ***** Output Definition *****
77
78 - data1 = input('Weight percentage of Clay: ');
79 - data2 = input('Weight percentage of Silt: ');
80 - data3 = input('Weight percentage of Sand: ');
81 - data4 = input('Weight percentage of Gravel: ');
82
83 - if (data1 + data2 + data3 + data4) == 100
84 -     disp('Percentage composition:');
85 -     Percentage_composition = [data1 data2 data3 data4];
86 -     outputs = abs(net(Percentage_composition));
87 -     disp('Estimation:');
88 -     disp(outputs);
89 - else
90 -     disp('Error: The sum of percentages should be equal to 100');
91 - end

```

Figure 41. MATLAB Code for Defining User Input and Estimating the Output Parameter. (Source: Own Compilation)

As it was mentioned earlier, the model was executed using a dataset split where 70% of the data was allocated for training and the remaining 30% was reserved for validation and testing. Following this, estimations were conducted to evaluate the predictive accuracy and generalization capability of the ANN model, confirming its efficacy in practical applications. For visualizing the results, a new code was written. According to figure 42, I assigned each soil component percentage column from the data to separate variables (C1, C2, C3, C4). I plotted the actual and predicted values of the output parameter (-log K) in different figures using the 'plot' function. The 'hold on' command was used to overlay multiple plots on the same figure for easy comparison.

```

13 % ***** Displaying Of Output Graphs *****
14
15 - C1=data(1:end, 1);
16 - C2=data(1:end, 2);
17 - C3=data(1:end, 3);
18 - C4=data(1:end, 4);
19
20 - figure(1);plot(Y_actual,'b');
21 - hold on ; plot(Y_fm,'r');
22 - figure(2);plot(C1,'b');
23 - hold on; plot(C4,'g');
24 - figure(3) ; plot(C2,'r');
25 - figure(4) ; plot(C3,'y');
26 - figure(5); plot(C4,'g');
27

```

Figure 42. MATLAB Code for Visualizing the Results of the ANN Model (Source: Own Compilation)

Figure 43 compares the actual (-log K) values with the neural network's predictions for the testing dataset. To better visualize the model's performance, the data was sorted in descending order. This sorting allows for a clearer comparison and helps highlight the extent of the fitting between the predicted and actual values. According to the figure, except for a few outliers, the neural network's predictions are very close to the actual values. The close alignment of the predicted values (red line) with the actual values (blue line) demonstrates the model's ability to accurately predict hydraulic conductivity based on soil composition. It indicates that the model has learned the complex relationships between soil components (clay, silt, sand, and gravel) and hydraulic conductivity. This high level of agreement between predicted and actual values confirms the accuracy of the ANN model.

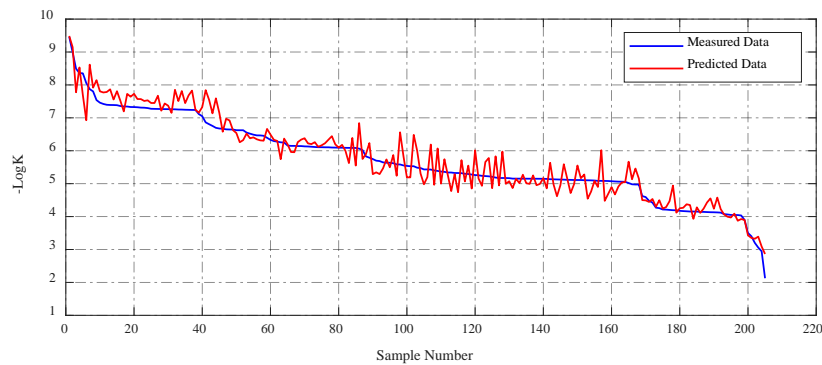


Figure 43. Comparison of (- Log K) Values Obtained by ANN and Experiments (Source: Own Compilation)

As a next step of coding, I evaluated the performance of the neural network by calculating the Root Mean Squared Error (RMSE). I computed the 'RMSE' value by taking the square root of the mean of the squared differences between the predicted values and the measured values which is shown in figure 44. It is noteworthy to mention that the RMSE value obtained by the model was 0.0768, indicating a high level of accuracy in the model's predictions. This low root mean square error suggests that the differences between the predicted values and the actual values are minimal, thereby validating the model's effectiveness in capturing the underlying patterns of the dataset.

```

28 %%% ***** RMSE Value ***** %%%
29
30 rmse=sqrt(mean(Y_nm-Y_actual).^2)
31

```

Figure 44. MATLAB Code to Calculate the Root Mean Squared Error (RMSE). (Source: Own Compilation)

Another important point obtained from the model was the importance of each parameter in estimation. It was found that the importance of the clay and gravel components is evident from the importance weight assigned to each variable. As mentioned earlier, the data was sorted in descending order to help understand the performance of the model; this means that as the number of tests increases, the values of -Log K drop. According to figures 45 to 47, there is a visible trend between these parameters and the output parameter, and they have less deviation of data in comparison to the silt and sand. So, the results indicate the weight percentages of clay and gravel have a significant and meaningful effect on hydraulic conductivity, more so than silt and sand. It means that they are critical factors influencing the model's predictions. The high importance

weight of these variables underscores their significant role in the underlying processes modeled by the ANN, demonstrating that accurate measurement and inclusion of these components are essential for reliable model performance.

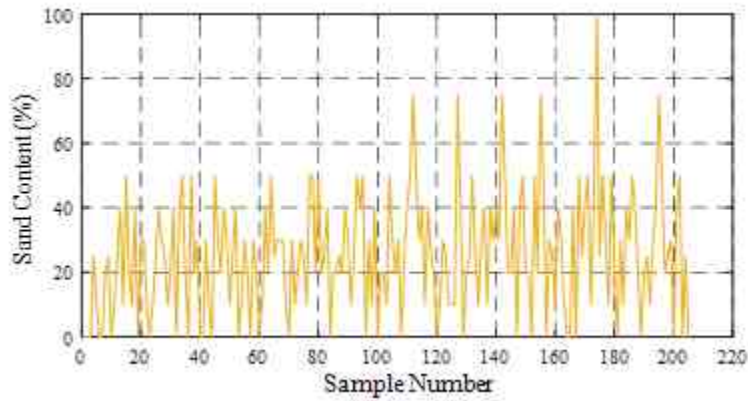


Figure 45. Relationship Between the Weight Percentage of Sand and Hydraulic Conductivity Value (Source: Own Compilation).

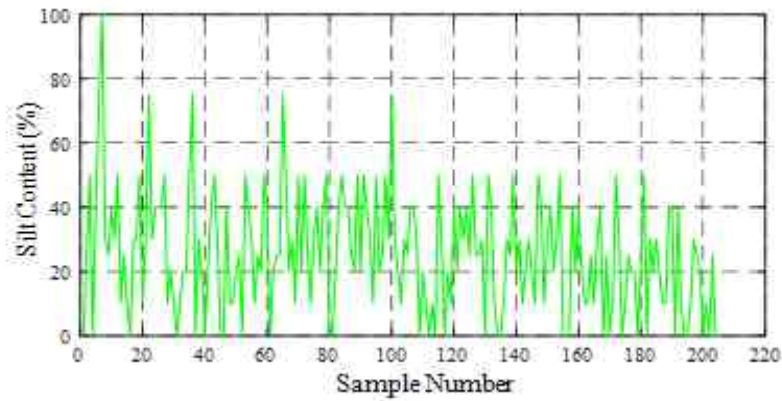


Figure 46. Relationship Between the Weight Percentage of Silt and Hydraulic Conductivity Value (Source: Own Compilation).

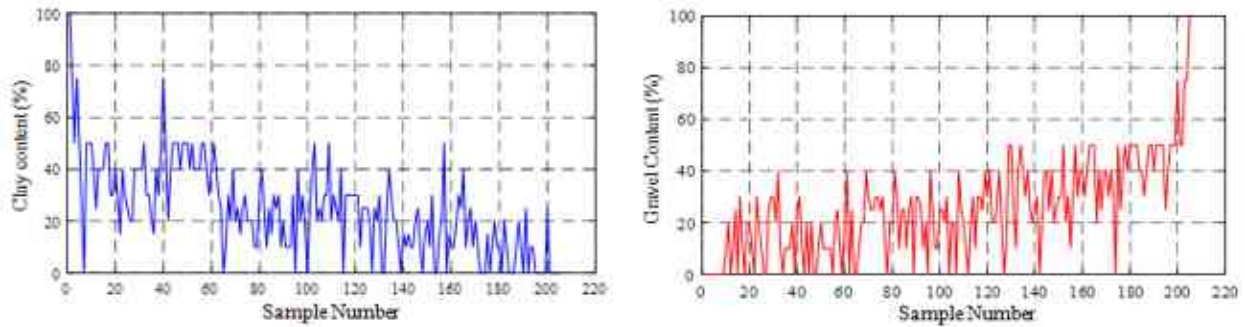


Figure 47. Relationship Between Weight Percentage of clay and Gravel and Hydraulic Conductivity Value (Source: Own Compilation).

These steps illustrate the process of using MATLAB to model an artificial neural network for estimating an output parameter from soil component percentages, including data extraction, network training, input handling, output visualization, and performance evaluation. Figure 48 shows the program with a specific test where the input percentages were 50% clay, 30% silt, 10% sand, and 10% gravel. After running the program, the estimated value of (-logK) was 7.2607e+00. This demonstrated the capability of the neural network model to accurately predict the output parameter based on the given soil composition.

```

Command Window
Weight percentage of Clay: 50
Weight percentage of Silt: 30
Weight percentage of Sand: 10
Weight percentage of Gravel: 10
Percentage composition:

Percentage_composition =

    50    30    10    10

estimation:
    7.2607e+00
fe >>

```

Figure 48. Example output of the MATLAB program for (-Log K) Prediction (Source: Own Compilation).

Notably, it should be mentioned that I could not find any similar work in the past where the input data to predict the K value was the same as what I chose for this study. However, there were a variety of modeling approaches that predicted K from different sets of input data. Table 7 shows the comparison of the results for estimation of K which were obtained from some previous research using various ANN model and findings from this work. According to the table, the results of this study are among the few ones with $R^2 > 0.90$, demonstrating superior predictive accuracy with higher R^2 values compared to those reported in previous studies. This underscores the effectiveness of the optimized ANN model developed in this research, highlighting its reliability in accurately modeling hydraulic conductivity.

Table 7. Comparison Between Several Studies for Obtaining Hydraulic Conductivity.

Model	Year	Method	R^2
This study	2024	ANN	0.92
More et al	2018	ANN	0.85
Zheng et al	2021	ANN	0.60
Asif Khaja et al	2022	ANN	0.78
Mozaffari et al	2024	ANN	0.80
Yilmaz et al	2012	ANN	0.89
Williams et al	2021	ANN	0.95
Merdun et al.	2006	ANN	0.52
Azarhoosh et al	2023	ANN	0.95
Azadmard et al	2020	ANN	0.40
Cahyadi et al	2021	ANN	0.86
Faloye et al	2022	ANN	0.92
Zuo et al	2021	ANN	0.76
Arshad et al	2013	ANN	0.68

In summary, I presented an analysis of estimating hydraulic conductivity based on the weight percentages of soil components (clay, silt, sand, and gravel) using artificial neural networks. I aimed to identify the optimal ANN model architecture that provides the most accurate and reliable predictions. I evaluated multiple models and found out the most effective, achieving R^2 values of 0.92. This high level of accuracy, with an RMSE value of 0.0768, underscores the model's effectiveness in capturing the underlying patterns in the data. Notably, I could not find any previous studies using the same input data to predict the K value, though various other modeling approaches exist. Comparison of these results with those from similar studies showed that my model achieved superior predictive accuracy with R^2 values consistently above 0.90, positioning it among the best-performing models. The analysis also highlighted the significant impact of clay and gravel on hydraulic conductivity, more so than silt and sand highlighting their essential role in the model's predictions.

Regarding complexity, the MLR method is the least complex and easy to interpret, but it is limited in handling non-linear relationships. The empirical formula is simple to use but offers limited accuracy and flexibility. On the other hand, the ANN model is the most complex and computationally intensive, but it is highly flexible and capable of capturing intricate patterns in the data. The ANN approach has the best accuracy R^2 value of 0.92, indicating better predictive accuracy. With an R^2 value of 0.81, the MLR approach demonstrated moderate accuracy; this is a better fit but less precise than the ANN model. With an R^2 value of 0.52 and the lowest accuracy of all investigated methods, the empirical formula proposed by Carrier and Beckman clearly needed improvement.

4.5. Artificial Neural Network model for cohesion parameter

While extensive research has been conducted on estimating hydraulic conductivity, studies focusing on predicting the mechanical behavior of soil based on its characteristics are relatively rare. This scarcity highlights the significance and value of this work, which aims to fill this gap by leveraging artificial neural networks to estimate soil's shear strength parameters, namely cohesion and internal friction angle.

In this section, I extend the application of artificial neural networks to predict the shear strength characteristics of soil, specifically cohesion and internal friction angle. Following the methodology used in previous neural network models for hydraulic conductivity, I employed a dataset comprising 95 soil samples for each shear strength parameter. Each sample included the weight percentages of soil components (sand, silt, clay, and gravel) as input features. The datasets for both cohesion and internal friction angle were divided into training (70%) and testing/validation (30%) groups to ensure a representative distribution across both phases. Figure.49 and 50 show the experimental results for cohesion and internal friction angle. Similar to previous sections, the naming order of the samples in graphs and tables was chosen after the results were obtained and sorted in the order of descending.

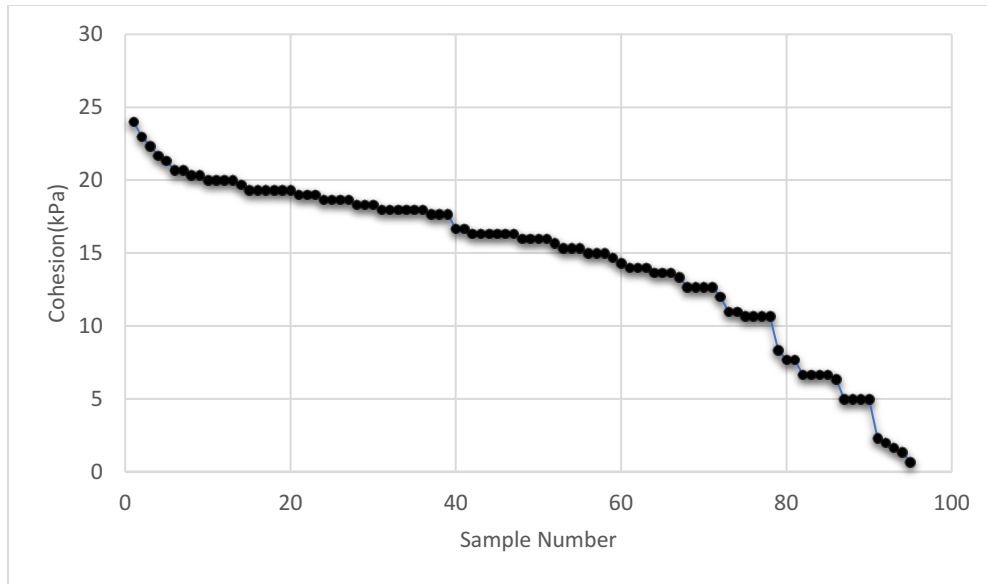


Figure 49. Cohesion Values Obtained by Laboratory Tests (Source: Own Compilation).

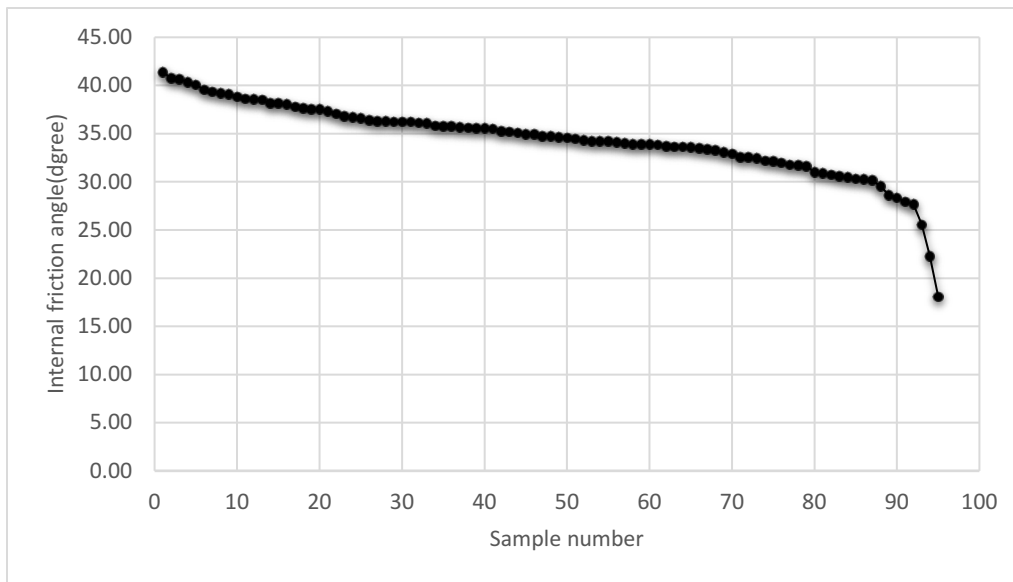


Figure 50. Internal Friction Angle Values Obtained by Laboratory Tests (Source: Own Compilation).

The Feedforward Neural Network (FFN) technique, validated in the hydraulic conductivity study, was applied to predict cohesion and internal friction angle independently. Iterative testing with varying architectures was conducted to identify the optimal balance between model complexity and prediction accuracy. The construction, training, and evaluation of the neural networks were performed using MATLAB's Neural Network Toolbox, utilizing the Levenberg-Marquardt backpropagation algorithm for its efficient convergence properties.

The neural network architecture for estimating soil cohesion is illustrated in figure 51. It comprises an input layer with four nodes corresponding to the weight percentages of clay, silt, sand, and gravel. This is followed by three hidden layers, each with 3, 1 and 4 neurons respectively, and a

single-node output layer that estimates cohesion. Through iterative testing, the mentioned structure was identified as optimal, achieving a balance between underfitting and overfitting. The model's performance was evaluated using key metrics. According to figure 52, the correlation coefficient (R) for cohesion was approximately 0.90 which indicates a relatively good correlation between the predicted and actual values. Additionally, the Root Mean Square Error (RMSE) value for cohesion was found to be 0.53, presenting the model's predictive accuracy.

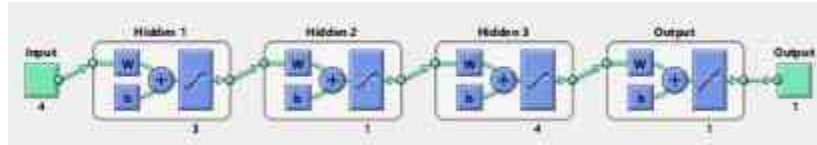


Figure 51. Neural Network Structure for cohesion Prediction (Source: Own Compilation).

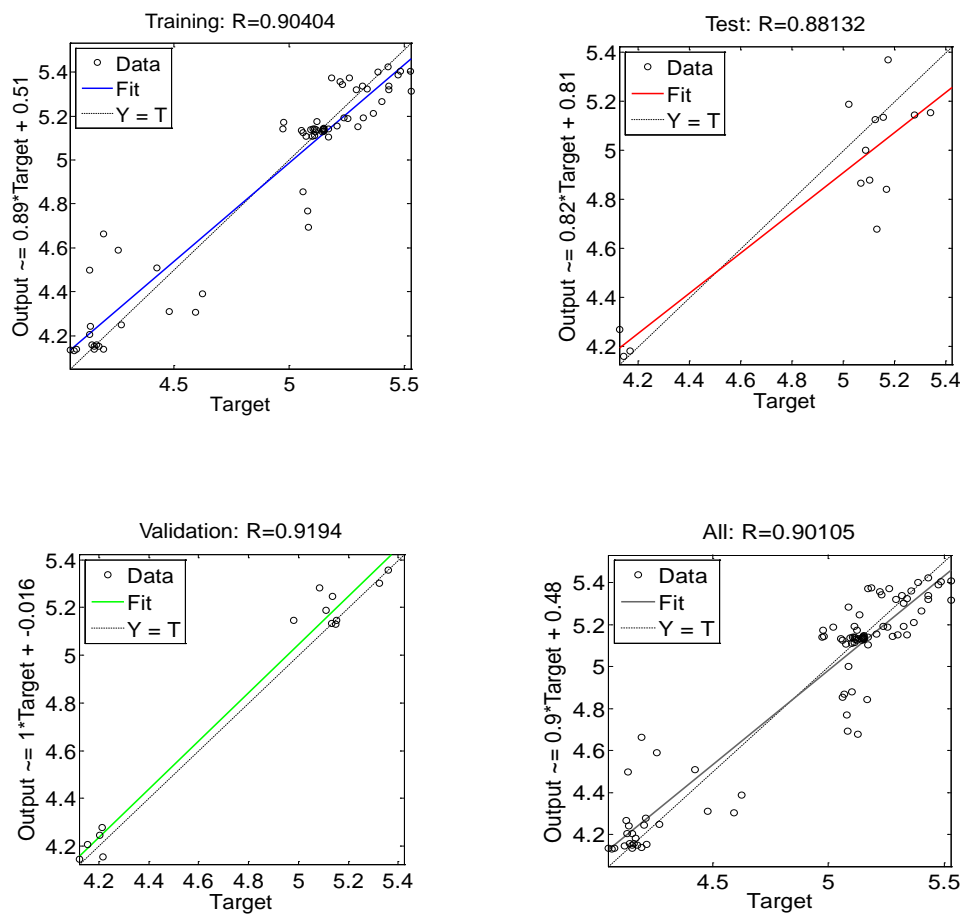


Figure 52. Regression Plots of the Best-Performing Model for Cohesion Prediction (Source: Own Compilation).

Similar to the work done for the prediction of hydraulic conductivity, the same procedure was followed for extracting the weights, biases, and activation functions used in the neural network model. As a result, the mathematical statements were defined. For the first hidden layer z_1 can be calculated as equation 20. The obtained weights and bias were as follows.

$$z_1 = W_1 \cdot x + b_1 \quad (\text{Eq.20})$$

$$W_1 = \begin{bmatrix} 1.2559 & -0.12284 & 0.098879 & 0.34655 \\ 1.6120 & -1.3428 & -1.6726 & -1.8099 \\ -1.6767 & 0.29718 & 0.35914 & 0.20731 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} -1.1025 \\ 1.2018 \\ -1.5530 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Substituting the values:

$$z_1 = \begin{bmatrix} 1.2559 \cdot x_1 + (-0.12284) \cdot x_2 + 0.098879 \cdot x_3 + 0.34655 \cdot x_4 - 1.1025 \\ 1.6120 \cdot x_1 + (-1.3428) \cdot x_2 + (-1.6726) \cdot x_3 + (-1.8099) \cdot x_4 + 1.2018 \\ -1.6767 \cdot x_1 + 0.29718 \cdot x_2 + 0.35914 \cdot x_3 + 0.20731 \cdot x_4 - 1.5530 \end{bmatrix}$$

Then, hyperbolic tangent sigmoid activation function applied to z_1 and the output named h_1 . For the second hidden layer, z_2 was computed based on equation 21.

$$z_2 = W_2 \cdot h_1 + b_2 \quad (\text{Eq.21})$$

$$W_2 = [-1.2909 \quad -1.1013 \quad 0.27013]$$

$$b_2 = [0.31217]$$

Substituting the values:

$$z_2 = [-1.2909 \cdot h_{1,1} + (-1.1013) \cdot h_{1,2} + 0.27013 \cdot h_{1,3} + 0.31217]$$

The same activation function applied to z_2 and the output named h_2 . For the third hidden layer, z_3 was calculated based on equation 22.

$$z_3 = W_3 \cdot h_2 + b_3 \quad (\text{Eq.22})$$

$$W_3 = \begin{bmatrix} -6.5676 \\ -5.7014 \\ 5.6391 \\ -5.5933 \end{bmatrix}$$

$$b_3 = \begin{bmatrix} 4.7544 \\ 1.4064 \\ 1.6923 \\ -5.6070 \end{bmatrix}$$

Substituting the values:

$$z_3 = \begin{bmatrix} -6.5676 \cdot h_2 + 4.7544 \\ -5.7014 \cdot h_2 + 1.4064 \\ 5.6391 \cdot h_2 + 1.6923 \\ -5.5933 \cdot h_2 - 5.6070 \end{bmatrix}$$

The tansig activation function applied to z_3 and the output named h_3 . Ultimately, For the final output layer, z_4 was determined through equation 23.

$$z_4 = W_4 \cdot h_3 + b_3 \quad (\text{Eq.23})$$

$$W_4 = [1.5689 \quad 0.53742 \quad -0.12731 \quad -0.29467]$$

$$b_4 = -1.5038$$

Substituting the values:

$$z_4 = 1.5689 \cdot h_{3,1} + 0.53742 \cdot h_{3,2} + (-0.12731) \cdot h_{3,3} + (-0.29467) \cdot h_{3,4} - 1.5038$$

As a final step, the purelin activation function applied to z_4 .

The model performance is evaluated by comparing the ANN's predictions with the actual values as shown in figure 53. The provided graph illustrates the results, where the blue line represents the actual cohesion values, and the red line represents the predicted values by the ANN. Both series data show a general downward trend, indicating that the ANN model captures the overall trend of the actual cohesion values effectively. However, the ANN predictions show more short-term fluctuations compared to the actual values, suggesting the model's sensitivity to minor variations in the input data, which may lead to overfitting. In the first 35 samples, the predicted values closely follow the actual values, demonstrating a good fit. This indicates that the ANN model performs better with samples that have a high amount of clay, where the soil cohesion is more consistent and less variable. However, from Test Numbers 36 to 100, the model's performance starts to diverge. This divergence can be attributed to the increased presence of coarse particles in the soil samples during this segment, which introduces additional variability and complexity that the ANN model may not fully capture.

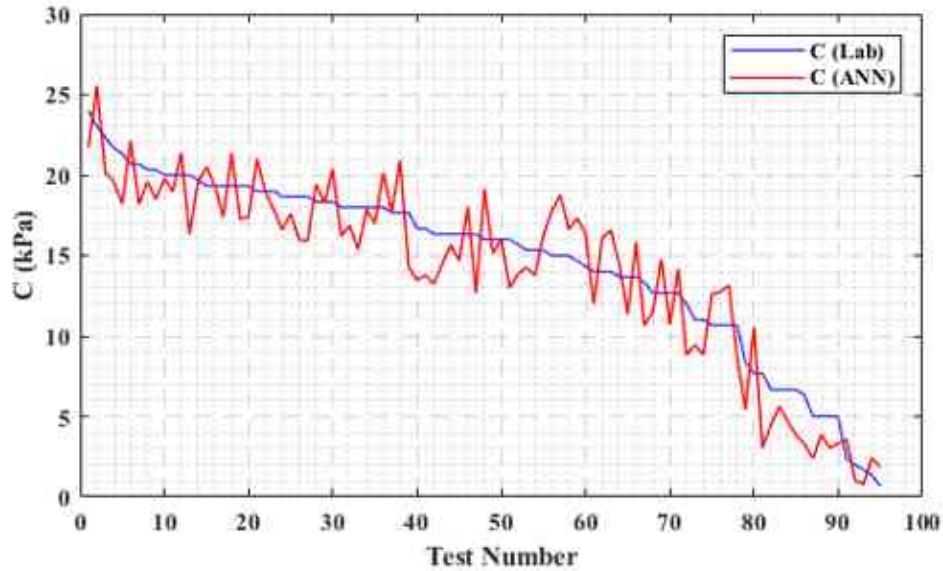


Figure 53. Comparison of Cohesion Values Obtained by ANN and Experiments (Source: Own Compilation).

I conducted the ANN modeling using MATLAB, which allowed us to efficiently code and implement the model. Figure 54 shows the Command Window of a MATLAB program designed to estimate soil cohesion using an artificial neural network. This example demonstrates how the program accepts user inputs and processes them to generate an output estimation. In this instance, the user inputs the weight percentages of various soil components, specifically: 50% Clay, 30% Silt, 10% Sand, and 10% Gravel. These inputs represent the composition of the soil sample in terms of its constituent particles and should sum up to 100% to accurately describe the soil's composition. After entering the input components, the program calculates the percentage composition of the soil sample and confirms the correct processing of the inputs. Subsequently, the ANN model processes these inputs to estimate the soil cohesion. The output represents the estimated cohesion value, in kilopascals (kPa), based on the given soil composition.

```

Command Window

Weight percentage of Clay: 50
Weight percentage of Silt: 30
Weight percentage of Sand: 10
Weight percentage of Gravel: 10
Percentage composition:

Percentage_composition =

    50    30    10    10

estimation:
    1.9499e+01

fx >> |

```

Figure 54. Example output of the MATLAB program for Cohesion Prediction (Source: Own Compilation).

4.6. Artificial Neural Network model for internal friction angle parameter

The same procedure was followed to estimate the other shear strength parameter, the internal friction angle, using the ANN model. The neural network architecture for predicting the internal friction angle is depicted in figure 55. It consists of an input layer with four nodes representing the weight percentages of clay, silt, sand, and gravel. This is followed by two hidden layers, each with different numbers of neurons, and a single-node output layer that estimates the internal friction angle. The optimal network architecture was determined to have [3 2] neurons, which provided a suitable balance between underfitting and overfitting for the data. The model's effectiveness in predicting the internal friction angle was similarly assessed using performance metrics. The correlation coefficient (R) for the internal friction angle was approximately 0.91 that shows a good correlation between the predicted and actual values as shown in figure 56. Furthermore, the obtained RMSE value for the internal friction angle was 0.39, demonstrating the model's accuracy in prediction.

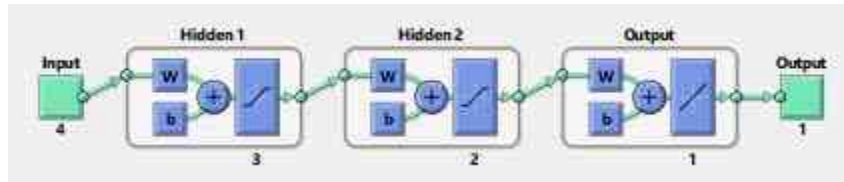


Figure 55. Neural Network Structure for Prediction of Internal Friction Angle (Source: Own Compilation).

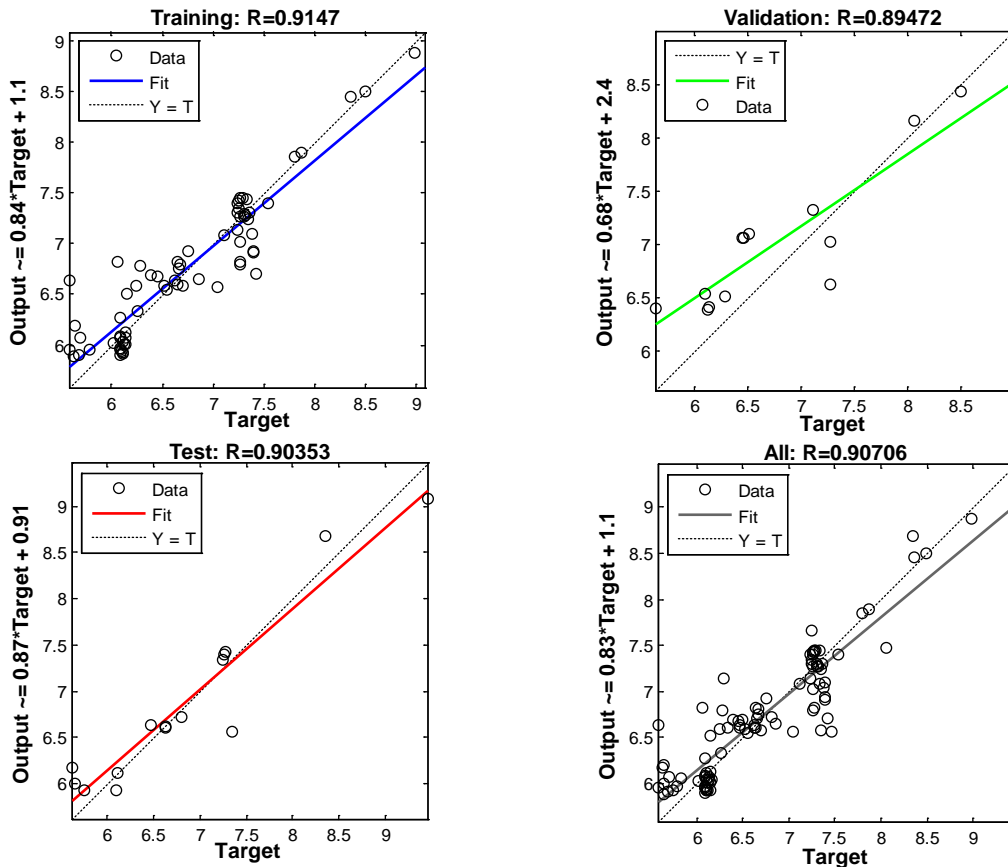


Figure 56. Regression Plots of the Best-Performing Model for Prediction of Internal friction Angle (Source: Own Compilation).

The process of extracting weights and mathematical relationships for the internal friction angle parameter is exactly the same as the same process done for cohesion. However, it should be kept in mind that the structure of the artificial neural network was different for this parameter. Also, the chosen activation function was similar to ones used for cohesion. Therefore, here only the weights and biases related to the first, second and final layers are mentioned.

$$W_1 = \begin{bmatrix} -1.1545 & -0.87914 & -0.086417 & 0.81092 \\ -0.46195 & -0.35259 & -0.21482 & 0.13076 \\ -1.5385 & 0.81031 & -0.82653 & -0.35283 \end{bmatrix} \quad b_1 = \begin{bmatrix} 2.1058 \\ -0.1074 \\ -1.4499 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} -1.9914 & -0.88889 & 0.014035 \\ 0.87937 & -0.80969 & -0.66601 \end{bmatrix} \quad b_2 = \begin{bmatrix} 1.3215 \\ 1.8575 \end{bmatrix}$$

$$W_3 = [-2.8072 \quad -0.71874] \quad b_3 = [-0.76291]$$

Like the procedure for cohesion estimation, the model performance for estimating the internal friction angle is evaluated by comparing the ANN's predictions with the actual values as shown in figure 57. In the provided graph, the blue line represents the actual internal friction angle values measured in the lab, while the red line represents the values predicted by the ANN. It can be found out that the ANN model effectively captures the overall trend of the actual internal friction angle values. Between Test Numbers 30 and 60, the predicted values closely follow the actual values, demonstrating a good fit. This indicates that the ANN model performs better with samples that have a well-distributed mixture of all soil components (e.g., sand, silt, clay, and gravel), where the internal friction angle is more consistent and less variable. Moreover, the model's performance starts to diverge after Test Number 80. This divergence can be attributed to the lack of coarse material and the predominance of fine grains in the soil samples during this segment.

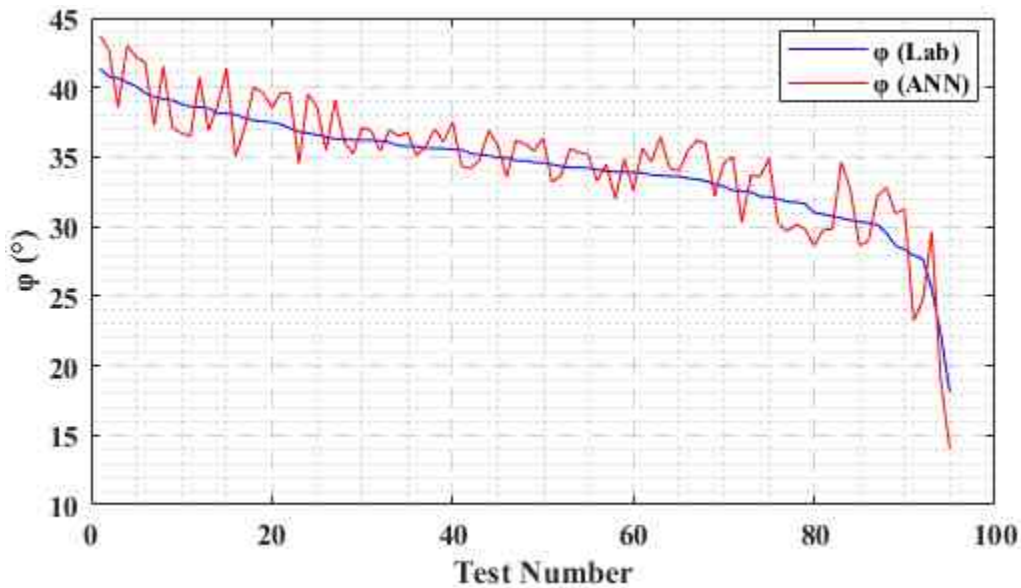
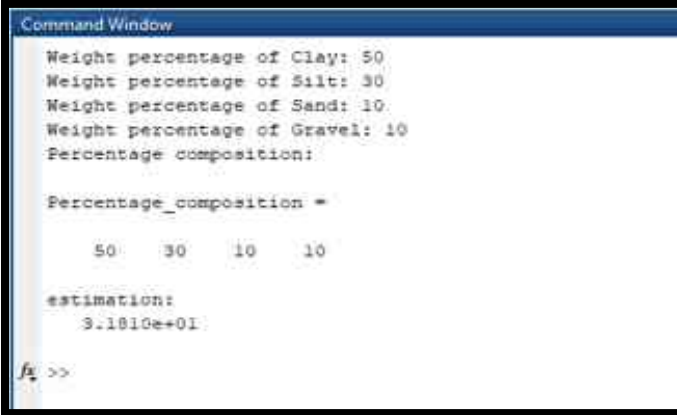


Figure 57. Comparison of Internal friction angle Values Obtained by ANN and Experiments (Source: Own Compilation).

Similar to the work done for cohesion, I conducted ANN modeling for the internal friction angle using MATLAB. This allowed us to efficiently code and implement the model. The figure 58 shows the MATLAB Command Window estimating the internal friction angle of soil using an ANN. The program accepts user inputs for soil composition, such as 50% Clay, 30% Silt, 10% Sand, and 10% Gravel, which should total 100%. After confirming the correct processing of these inputs, the ANN model estimates the soil's internal friction angle in degrees ($^{\circ}$) based on the given composition.



```
Command Window
Weight percentage of Clay: 50
Weight percentage of Silt: 30
Weight percentage of Sand: 10
Weight percentage of Gravel: 10
Percentage composition:

Percentage_composition =

    50    30    10    10

estimation:
    3.1810e+01
f1 >>
```

Figure 58. Example output of the MATLAB program for Prediction of Internal Friction Angle (Source: Own Compilation).

4.7. Simultaneous modeling of shear strength parameters

One powerful application of ANNs is the creation of joint models. Joint models in artificial neural networks represent a sophisticated approach to predictive modeling by simultaneously predicting multiple output variables from a shared set of input features. Unlike traditional models that handle each output independently, joint models capture interdependencies and correlations between outputs, enhancing predictive accuracy and computational efficiency. This approach is particularly advantageous in fields where outputs are inherently related, such as geotechnical engineering, where predicting both cohesion and friction angle together improves the understanding of soil behavior. The design of a joint model involves configuring the neural network architecture with multiple output neurons corresponding to each variable of interest and utilizing a multi-output loss function during training to optimize predictions across all outputs simultaneously.

The decision to opt for a joint model was primarily driven by the understanding that cohesion and friction angle exhibit correlated behavior in soil mechanics. As friction angle increases, cohesion typically decreases, and vice versa. By modeling cohesion and friction angle together, rather than separately, the jointly model can capture synergistic effects and dependencies that affect both parameters simultaneously. This approach leads to more accurate predictions of soil behavior across a wide range of scenarios. Engineers and researchers can optimize geotechnical designs and analyses more effectively when they have a comprehensive understanding of both these parameters. This includes designing foundations, slopes, excavations, and other structures that rely on accurate soil parameter predictions.

In traditional neural networks, each output corresponds to a single neuron in the output layer, predicting a single target variable. However, to model the parameters simultaneously, the output layer was configured to have multiple neurons, each representing a different output (cohesion and friction angle). By having multiple neurons in the output layer, the neural network learns to optimize weights and biases across all outputs simultaneously. The shared input features (Weight percentage of clay, silt, sand and gravel) are processed through hidden layers that extract relevant patterns and relationships. These hidden layers act as shared representations, contributing to the prediction of both shear parameters and leveraging correlations between them. The network architecture comprises hidden layers with sizes [3, 2, 4] as shown in figure 59, utilizing hyperbolic tangent sigmoid ('tansig') functions for activation in the hidden layers and a linear ('purelin') function for the output layer.

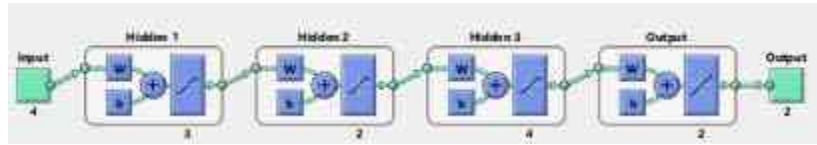


Figure 59. Neural Network Structure for Simultaneous Model of shear strength parameters (Source: Own Compilation).

The process of extracting weights and mathematical relationships of this model is the same as the process done for previous models. This involves accessing the neural network's weights and biases, which are critical for understanding the transformations applied to the input data at each layer of the network. These weights and biases are determined during the training phase and are used to predict the output values from the given input features. By analyzing these parameters, I can gain insights into how the neural network processes the data. Below, I present the weights and biases related to the first, second, third, and final layers of the network. Additionally, the activation functions applied at each layer were consistent with those used in previous models, ensuring a comparable non-linear transformation of the data through the network.

$$W_1 = \begin{bmatrix} 0.9277 & -0.6135 & -0.7876 & -0.8224 \\ -4.5183 & 1.5298 & 2.1891 & -0.5534 \\ 7.8806 & -2.1196 & -1.0415 & -2.4748 \end{bmatrix} \quad b_1 = \begin{bmatrix} 1.5801 \\ -0.1071 \\ 1.1257 \\ -4.5023 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 3.3580 & -0.1524 & 3.6949 \\ 3.2142 & 1.8756 & 0.0967 \end{bmatrix} \quad b_2 = \begin{bmatrix} 0.1190 \\ 4.6239 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 1.2571 & -2.4997 \\ -0.5505 & -2.5127 \\ -2.7794 & -2.4002 \\ -1.1793 & -1.3582 \end{bmatrix} \quad b_3 = \begin{bmatrix} -2.7906 \\ -3.4382 \\ 0.3887 \\ -3.2169 \end{bmatrix}$$

$$W_4 = \begin{bmatrix} -0.5483 & -0.0175 & -1.7251 & 1.1896 \\ 1.2425 & -2.9448 & -1.1047 & 0.9930 \end{bmatrix} \quad b_4 = \begin{bmatrix} -0.3728 \\ -1.0683 \end{bmatrix}$$

It was aimed to assess the predictive performance of the new joint model for the shear strength parameters, cohesion, and friction angle. To evaluate the effectiveness of the new model and to make comparisons with the previous separate models, the Root Mean Squared Error (RMSE) and the Coefficient of Determination (R^2) were calculated. The RMSE quantifies the average magnitude of the errors between the predicted values and the actual values of cohesion and friction angle. Figure shows the parts of code related to this matter. Additionally, I plotted regression curves to visually depict the relationship between the actual and predicted values for both cohesion and friction angle. These plots include trendlines that highlight the degree of alignment between the predicted and actual values. Figure 60 illustrates the evaluation of the ANN model's predictive performance.

```

79
80 % Evaluate the network on the entire dataset
81 - y_pred_all = net(X');
82
83 % Transpose predictions to match the shape of y
84 - y_pred_all = y_pred_all';
85
86 % Calculate performance metrics (RMSE)
87 - rmse_c = sqrt(mean((y(:, 1) - y_pred_all(:, 1)).^2));
88 - rmse_phi = sqrt(mean((y(:, 2) - y_pred_all(:, 2)).^2));
89
90 % Calculate R squared
91 - SS_tot_c = sum((y(:, 1) - mean(y(:, 1))).^2);
92 - SS_res_c = sum((y(:, 1) - y_pred_all(:, 1)).^2);
93 - R2_c = 1 - (SS_res_c / SS_tot_c);
94
95 - SS_tot_phi = sum((y(:, 2) - mean(y(:, 2))).^2);
96 - SS_res_phi = sum((y(:, 2) - y_pred_all(:, 2)).^2);
97 - R2_phi = 1 - (SS_res_phi / SS_tot_phi);
98
99 % Display performance metrics
100 - disp(['RMSE for Cohesion: ', num2str(rmse_c, '%.2f')]);
101 - disp(['RMSE for Friction Angle: ', num2str(rmse_phi, '%.2f')]);
102 - disp(['R^2 for Cohesion: ', num2str(R2_c, '%.2f')]);
103 - disp(['R^2 for Friction Angle: ', num2str(R2_phi, '%.2f')]);
104

```

Figure 60. Performance Evaluation of ANN Model (Source: Own Compilation).

Upon implementation, the joint neural network achieved significant improvements in predictive performance compared to previous independent modeling efforts. The new Root Mean Squared Error (RMSE) values were notably reduced: RMSE for cohesion was 0.41 and for friction angle was 0.25, indicating enhanced accuracy in predicting these soil parameters based on the input features. Moreover, the new R^2 values exhibited substantial increases, with R^2 value of 0.97 for cohesion and 0.90 for friction angle. These R^2 values signify a marked improvement over the previous independent models, where R^2 for cohesion was 0.81 and for friction angle was 0.82.

To visualize the model's performance, the regression curves were plotted for both cohesion and friction angle by comparing actual values against predicted values which are shown in figure 61 and 62. This is done using scatter plots, where each point represents an actual-predicted pair. Additionally, I incorporate trendlines to better illustrate the correlation between actual and predicted values. Specifically, for each subplot, I use the 'polyfit' function to compute the coefficients of a linear fit for the data points, and the 'polyval' function to evaluate this linear fit across the range of actual values.

The reduction in RMSE and increase in R^2 values directly contribute to the reduction in the standard deviation of the errors. RMSE quantifies the average magnitude of prediction errors, where lower values indicate that predictions are closer to actual values, thus reducing the variability in prediction errors. Similarly, R^2 measures the proportion of variance in the dependent variable that is predictable from the independent variables; higher R^2 values indicate a better fit of the model to the data, thereby reducing the overall spread of errors around the fitted line. By achieving lower RMSE and higher R^2 values, the joint neural network effectively minimizes the variability and standard deviation of prediction errors. This enhancement underscores the model's capability to leverage shared representations and correlations between outputs, thereby refining predictions and supporting more accurate and reliable decision-making in geotechnical engineering applications. Consequently, this joint neural network model stands out as the preferred choice for predicting both cohesion and friction angle due to its superior performance metrics and comprehensive approach to capturing soil behavior characteristics.

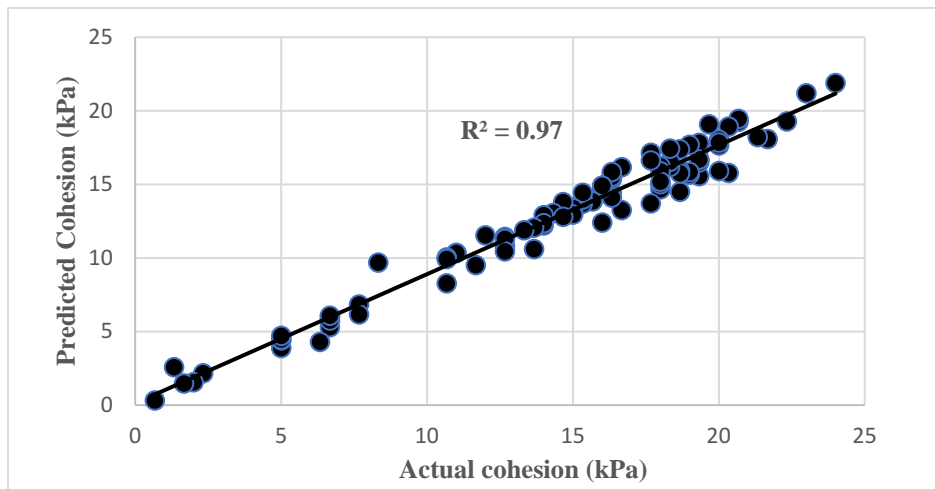


Figure 61. Comparison of Cohesion Values Obtained by ANN and Laboratory Tests (Source: Own Compilation).

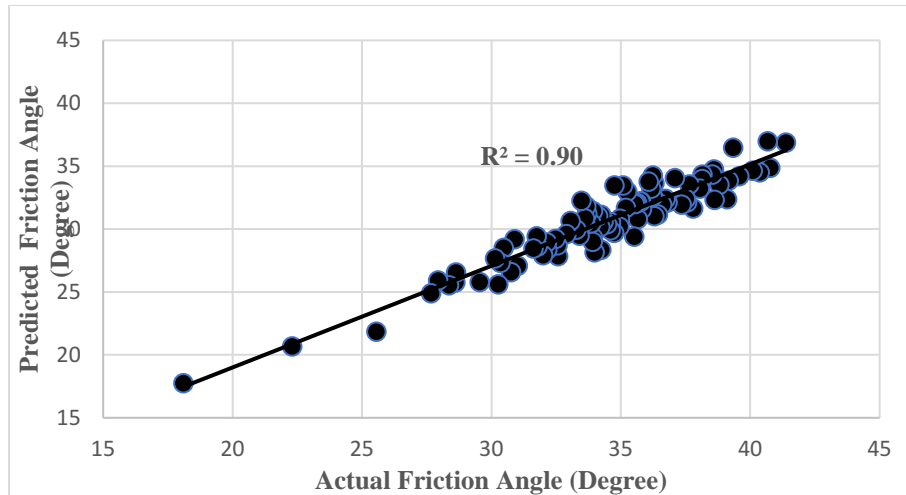


Figure 62. Comparison of Friction Angle Values Obtained by ANN and Laboratory Tests (Source: Own Compilation).

I specifically chose to compare this study with previous research that estimated shear strength parameters based on easily measurable soil properties, particularly grain size distribution. It is noteworthy that no prior research used the same input variables as this study. A key feature of the input variables is their ease and speed of measurement, encompassing both fine grain and coarse grain sizes. In contrast, previous studies mostly focused on one of these grain sizes.

The table provided highlights a comparison of methods used in current study with those employed in previous research for estimating cohesion (c) and internal friction angle (ϕ). The focus is on various models and their performance, measured by Root Mean Square Error (RMSE) values. In this study, I utilized a feedforward artificial neural network, achieving RMSE values of 0.53 for cohesion and 0.39 for internal friction angle. These values indicate high accuracy and are among the best when compared to previous research methods. The methods used in earlier studies include ANN, Multiple Linear Regression (MLR), Random Forest (RF), Gradient Boosting Machine (GBM), and Adaptive Neuro-Fuzzy Inference System (ANFIS). This study stands out due to the high accuracy of the ANN model, as demonstrated by the significantly lower RMSE values for both cohesion and internal friction angle. This suggests that my model is a more accurate and reliable prediction tool for soil properties compared to those used in many previous studies.

Table 8. Comparison Between Several Studies for Obtaining Shear Strength Parameters.

Model	Year	Method	RMSE-c	RMSE- ϕ
This study	2024	ANN	0.41	0.25
Zhu et al.	2022	ANN	3.39	4.08
Bala et al	2022	MLR	1.19	3.33
Mohammadi et al	2022	ANN	8.59	2.29
Ghoreishi et al	2021	ANN	0.04	4.43
Ly et al	2021	RF	3.323	-
Rezaee et al	2021	GBM	2.30	1.52
Pham et al	2021	ANFIS	0.07	0.08
Falamaki et al.	2018	ANN	4.70	2.3
Iyeke et al	2016	ANN	8.33	4.77
Roy et al	2014	MLR	-	0.82

In this chapter, the analysis and results of the research were detailed. Hydraulic conductivity values were first obtained through experiments. These values were then estimated using indirect methods, followed by multiple linear regression analysis to further refine the estimates. Artificial Neural Network (ANN) models were developed for predicting hydraulic conductivity, cohesion, and internal friction angle parameters. Additionally, a simultaneous modeling approach for shear strength parameters was implemented using ANNs, demonstrating their effectiveness in accurately estimating multiple soil parameters concurrently.

5- Theses

In this chapter, the main achievements of this research have been highlighted and explained in a general context. To provide a clearer understanding, some of the visualized graphs and plots presented in previous chapters as well as some new ones are revisited here.

Thesis 1

It is proven that artificial neural network (ANN) models can estimate the most important hydraulic and geotechnical parameters, hydraulic conductivity, cohesion and angle of internal friction, with an accuracy worth for preliminary phases of planning based on weight percentages of soil particles (clay, silt, sand and gravel).

My ANN models which employed feedforward method predicts values with higher accuracy than the other known and published methods. Based on my analysis, the ANN model for hydraulic conductivity, which has two hidden layers with 1 and 3 neurons respectively, and the joint model for cohesion and internal friction angle, which has three hidden layers with 3, 2, and 4 neurons respectively as well as an output layer with 2 neurons, demonstrated the highest accuracy.

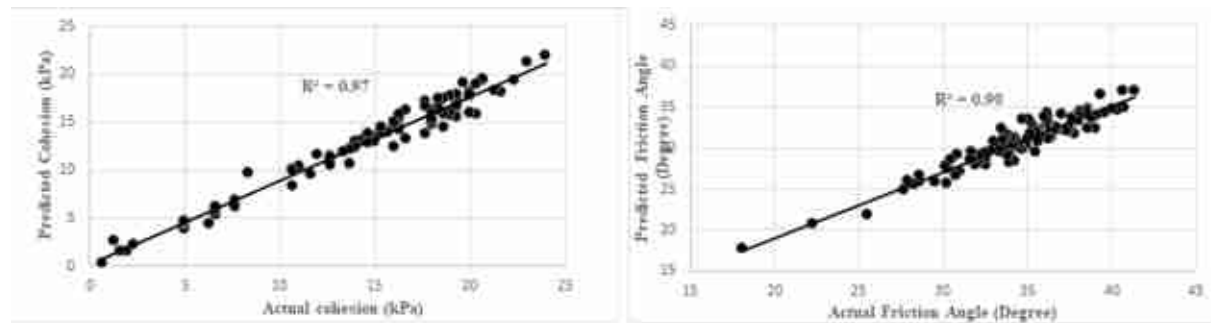
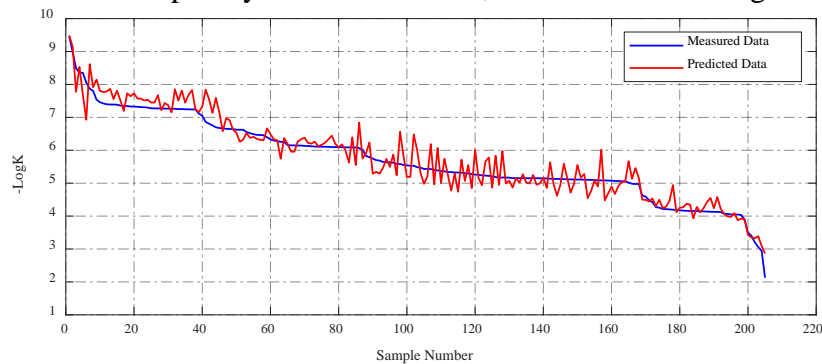


Figure 63 Comparison of results Obtained by ANN and Laboratory Tests (Source: Own Compilation).

My innovative programs utilize artificial neural networks (ANNs) to predict soil parameters, employing a specific function for neuron activation. The designed Programs for predicting both hydraulic conductivity and shear strength parameters follow the below fundamental mathematical concepts.

$$y_k = \tanh\left(\sum_{j=1}^m w_{kj}a_j + b_k\right)$$

$$a_j = \tanh\left(\sum_{i=1}^n w_{ji}x_i + b_j\right)$$

Where:

- y_k is the output of the k-th neuron in the output layer,
- a_j is the output of the j-th neuron in the hidden layer, computed using the tanh activation function,
- w_{ji} is the weight connecting the i-th input to the j-th neuron,
- x_i is the i-th input to the network,
- b_j is the bias of the j-th neuron,
- w_{kj} is the weight connecting the j-th neuron in the hidden layer to the k-th neuron in the output layer,
- and b_k is the bias of the k-th neuron in the output layer.

Thesis 2

The accuracy of the ANN model I developed and suggested for use is independent of grain size, unlike most other models and developed methods that are typically suitable only for fine-grain or coarse-grain soils.

My ANN model has been designed to effectively predict soil parameters regardless of whether the soil is predominantly fine-grained or coarse-grained. This capability ensures that my model provides reliable results across a wide range of soil types. Importantly, my approach simplifies the initial assessment of site soils for preliminary design by focusing solely on essential soil components. This streamlined methodology not only enhances efficiency but also ensures that my model can swiftly provide reliable insights into soil behavior across diverse environmental conditions. Thus, my ANN-based approach represents a practical and effective tool for engineers and researchers seeking rapid and insightful soil assessments during early project phases. Figure 64 shows simplified input and output box of the program.

Input Data	Output Data
Weight Percentage of Clay :	Hydraulic conductivity:
Weight Percentage of Silt:	Cohesion:
Weight Percentage of Sand :	Internal friction Angle:
Weight Percentage of Gravel :	

Figure 64. Simplified Input and Output Box of the Program (Source: Own Compilation).

The scatter plots for hydraulic conductivity, cohesion, and internal friction angle display residuals that are randomly distributed around the horizontal axis as shown in figure 65. This random distribution indicates that the residuals do not show any systematic pattern or trend. In other words, the errors in the predictions do not vary with the magnitude of the predicted values. This lack of systematic bias across the predicted values suggests that the model's performance is consistent and

unbiased across different samples. The absence of such patterns confirms that the ANN model is robust, and its predictions are not dependent on the grain size of the soil.

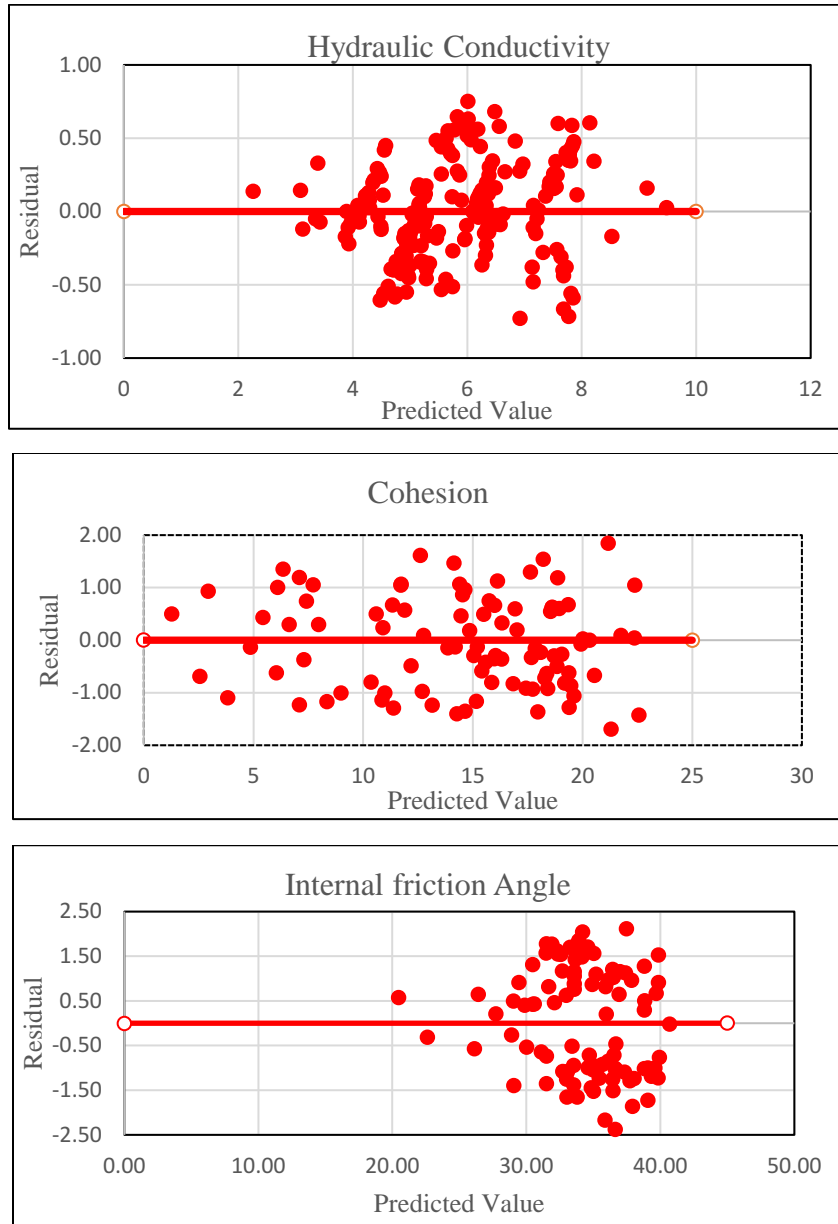


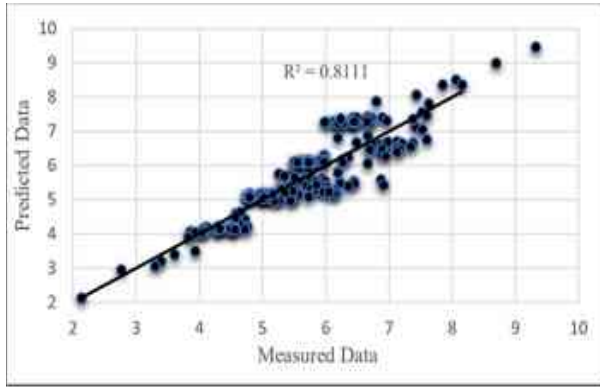
Figure 65. Plot of Residuals vs. Predicted Values (Source: Own Compilation).

Thesis 3

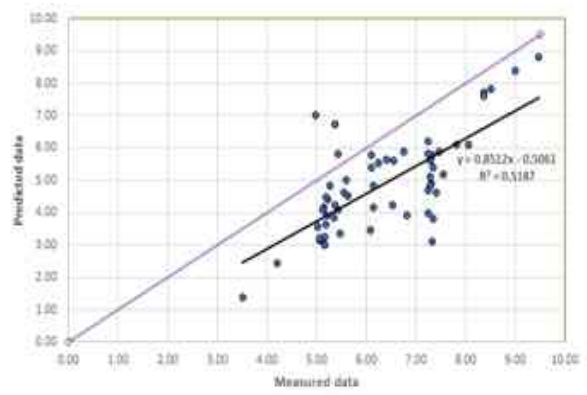
It was proven that artificial neural network method provides more reliable and accurate predicted hydraulic conductivity values independently form grainsize distribution than multiple linear regression or any known empirical correlation or pedotransfer function.

According to figure 66, the results showed that artificial neural network model is more reliable than the others as it presented higher accuracy. I utilized laboratory measurements to evaluate the

performance of various methods and models in predicting hydraulic conductivity. The indirect method, specifically the Carrier and Beckman approach, showed a moderate correlation with an R^2 value of 0.5187. However, this method consistently underestimated hydraulic conductivity values, highlighting the need for refinement or complementary methods. Multiple linear regression analysis demonstrated a better fit with an R^2 value of 0.81, indicating that 82% of the variability in the real data was explained by the model. My artificial neural network (ANN) models achieved superior accuracy with an R^2 value of 0.92 and an RMSE of 0.0768. This high level of accuracy underscores the effectiveness of ANNs in capturing the underlying patterns in the data.



Multiple linear Regression Method



Indirect Method

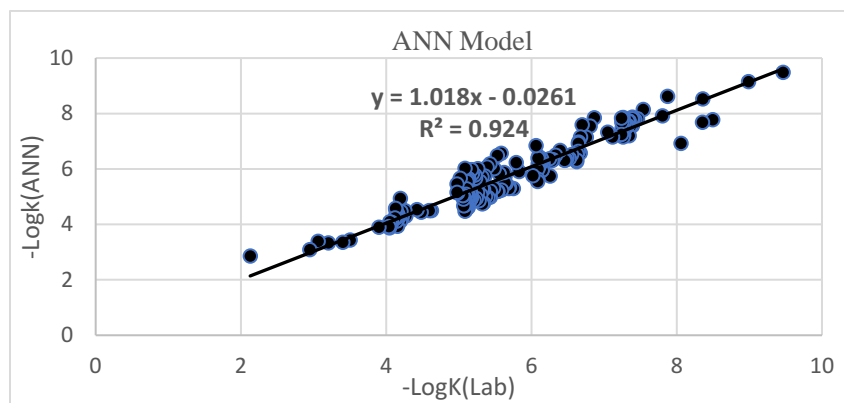


Figure 66. Estimation Accuracy of Different Models (Source: Own Compilation).

Thesis 4

It is statistically proven by using the artificial neural network model that the extremities in grainsize (gravel and clay content) indicate a higher effect on the hydraulic conductivity of the material than weight percentages of silt and sand.

By examining the effect of each parameter in the compositions and the amount of hydraulic conductivity changes, as shown in the figure 67, it becomes evident that clay and gravel exhibit a considerable slope as well as less deviation, indicating their substantial influence. In contrast, silt and sand do not display a noticeable slope, suggesting their lesser impact on the hydraulic gradient parameter. This highlights the critical importance of gravel and clay, where even minor variations significantly affect the hydraulic conductivity predictions. To plot the figures, the data was sorted in descending order to help understand the performance of the model; this means that as the number of tests increases, the values of $-\text{Log K}$ drop.

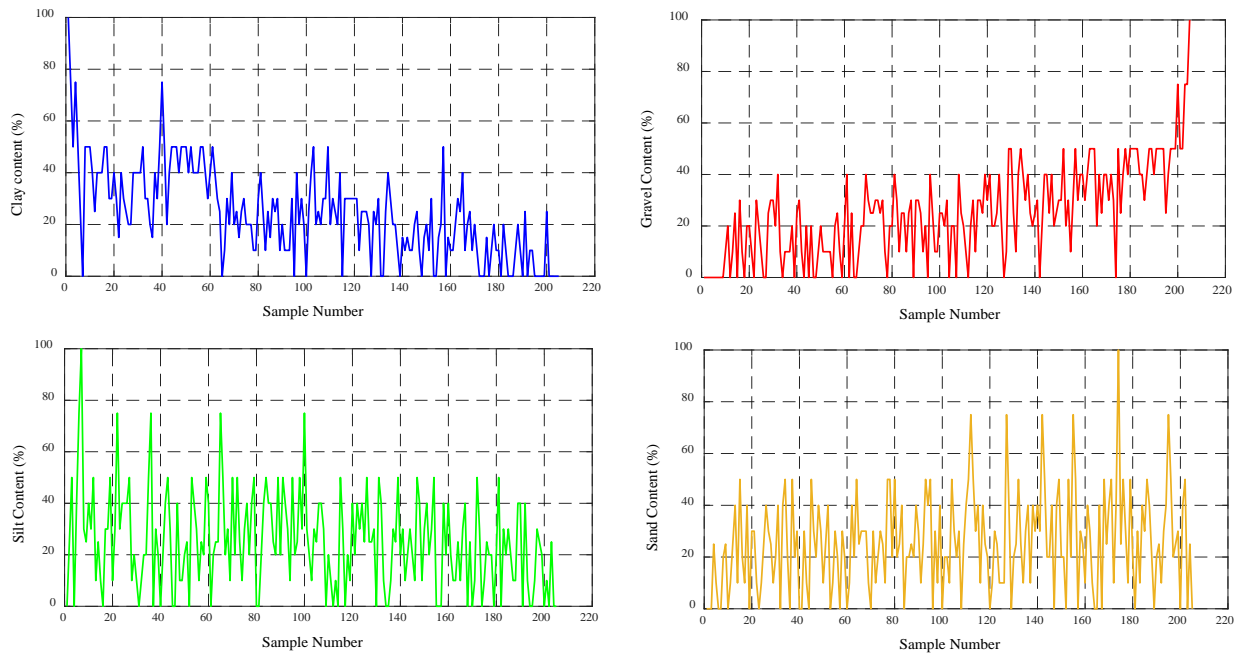


Figure 67. Effect of Each Component on Output (Source: Own Compilation).

Thesis 5

Simultaneous ANN model of cohesion and internal friction proved that contrary to separate ANN models, it achieved statistically significant improvements in predictive performance and accuracy.

The joint approach not only delivered higher accuracy but also demonstrated greater consistency in predictions. This model effectively captures the complex interactions between cohesion and internal friction. Unlike traditional models that handle each output independently, my joint model captures interdependencies and correlations between outputs, enhancing predictive accuracy and computational efficiency. The design of a joint model involves configuring the neural network architecture with multiple output neurons corresponding to each variable of interest and utilizing

a multi-output loss function during training to optimize predictions across all outputs simultaneously. The decision to opt for a joint model was primarily driven by the understanding that cohesion and friction angle exhibit correlated behavior in soil mechanics. By having multiple neurons in the output layer, the neural network learned to optimize weights and biases across both outputs simultaneously. The shared input features (Weight percentage of clay, silt, sand and gravel) are processed through hidden layers that extract relevant patterns and relationships. These hidden layers act as shared representations, contributing to the prediction of both shear parameters and leveraging correlations between them. The network architecture comprises hidden layers with sizes [3, 2, 4], utilizing hyperbolic tangent sigmoid ('tansig') functions for activation in the hidden layers and a linear ('purelin') function for the output layer.

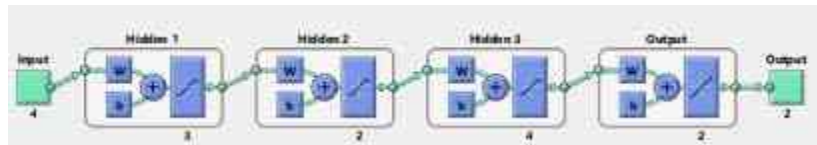


Figure 68. Neural Network Structure for Simultaneous Model of shear strength parameters (Source: Own Compilation).

Comparison of Simultaneous Model and Separate Models statistically demonstrated higher accuracy as mentioned in table 1.

Table 9. Comparison of Simultaneous Model and Separate Models

Model	Year	Method	RMSE-c	RMSE- ϕ	R ² -c	R ² - ϕ
Simultaneous Model	2024	ANN	0.41	0.25	0.97	0.90
Separate Model- ϕ	2024	ANN	-	0.39		0.82
Separate Model- c	2024	ANN	0.53	-	0.81	

6- Summary

In this study, I aimed to estimate geotechnical and hydrogeological parameters of soil based on grain size distribution using artificial neural networks (ANNs). Recognizing the limitations of traditional methods and the potential of ANNs to offer more accurate predictions, I collected a comprehensive dataset of laboratory-tested soil samples. This data was used to train and validate ANN models, with a focus on predicting hydraulic conductivity and shear strength parameters. My goal was to demonstrate the efficacy of ANN in soil parameter estimation and its potential to enhance geotechnical engineering practices.

Regarding the hydraulic conductivity estimation, this study involved extensive laboratory testing to observe the effects of different combinations of gravel, sand, silt, and clay on hydraulic conductivity. I employed several approaches for comparison, including indirect methods, Multiple Linear Regression (MLR), and ANN. Indirect methods relied on empirical calculations based on easily measurable soil parameters. The MLR approach created a statistical model linking several soil properties to hydraulic conductivity, while ANN utilized machine learning to model complex, nonlinear relationships among soil parameters. The indirect method, specifically the Carrier and Beckman approach, showed a moderate correlation with an R^2 value of 0.5187. However, the method consistently underestimated hydraulic conductivity values, highlighting the need for refinement or complementary methods. MLR analysis demonstrated a relatively good fit with an R^2 value of 0.81, indicating that 82% of the variability in the real data was explained by the model. Although MLR provided better accuracy, it still fell short of my desired predictive performance. My ANN models achieved superior accuracy with R^2 value of 0.92 and an RMSE of 0.0768. This high level of accuracy underscores the effectiveness of ANNs in capturing the underlying patterns in the data. Notably, I could not find any previous studies using the same input variables to predict hydraulic conductivity, making this study unique. However, the comparison with similar studies showed that my model outperformed others, highlighting the significant impact of clay and gravel on hydraulic conductivity.

In this study, I also modeled the shear strength parameters, cohesion and internal friction angle, using artificial neural networks in two distinct approaches: separately and simultaneously. Initially, I developed separate ANN models for each parameter, achieving good accuracy with R^2 values of 0.81 for cohesion and 0.82 for the internal friction angle. These values indicate a relatively good degree of precision. However, considering the interrelation between cohesion and internal friction angle, I also developed a simultaneous ANN model to predict both parameters concurrently. This joint modeling approach leveraged the inherent relationship between the parameters, leading to an improved predictive performance. The simultaneous ANN model not only presented higher accuracy than what achieved by the separate models but also demonstrated enhanced consistency and robustness in the predictions. The new R^2 values exhibited substantial increases, with R^2 value of 0.97 for cohesion and 0.90 for friction angle. Moreover, The RMSE values for the joint model were better than those obtained from the separate models, further validating the effectiveness of this approach and placing the separate models among the best compared to previous research methods. The simultaneous model thus provided a more comprehensive understanding of soil behavior, achieving a high degree of precision and placing my model among the best compared to previous research methods.

The results of this study hold significant practical implications for the field of geotechnical engineering. By leveraging artificial neural networks to estimate hydraulic conductivity and shear strength parameters from grain size distribution, I offer a method that is both time-efficient and cost-effective compared to traditional in situ and laboratory testing methods. This advancement is particularly valuable in scenarios where rapid decision-making is crucial, such as during the preliminary design phase of engineering projects. The ability to quickly and reliably predict soil parameters can streamline project planning and execution, reducing delays and associated costs. Moreover, the high accuracy of my ANN models, as demonstrated by superior RMSE values compared to conventional methods, underscores the potential for these models to provide more precise assessments of soil behavior, thereby enhancing the safety and reliability of geotechnical structures.

The key feature of my input variables is their ease and speed of measurement as well as encompassing both fine grain and coarse grain sizes. This comprehensive approach contrasts with previous studies that mostly focused on one grain size. This research makes a substantial contribution to the field of estimating soil properties by showing how artificial neural networks (ANNs) can improve the precision and dependability of predictions made in geotechnical engineering. According to the research, using machine learning models, such as artificial neural networks (ANNs), can result in more informed choices and effective engineering techniques. By leveraging easily measurable soil properties and advanced machine learning techniques, I have developed models that offer superior predictive performance. Future research could explore further refinement of ANN models and the inclusion of additional soil properties to enhance prediction accuracy even further. This study sets a precedent for integrating advanced computational methods in geotechnical engineering, paving the way for more accurate and reliable soil property estimation.

It is important to recognize the limits of the neural network model established in this study, even though it offers useful insights into the assessment of soil properties such as hydraulic conductivity, cohesion, and internal friction angle. Significant obstacles include the data's regional specificity, possible laboratory errors, the omission of pertinent variables, model sensitivity, and the dangers of extrapolation. Future studies could address these constraints by increasing the dataset size, adding more input variables, and enhancing the model's interpretability and resilience. These efforts would improve the model's applicability in many circumstances and lead to a more thorough understanding of soil qualities.

List of research related to the dissertation

Eteraf, H., Kovács, B., & Mikita, V. (2024). Advanced Approach Integrating Methods to Estimate Hydraulic Conductivity of Different Soil Types with Support from a Machine Learning Model, *Open Geoscience Journal*. De Gruyter, ISSN: 2391-5447. (submitted)

Eteraf, H., Kovács, B., & Mikita, V. (2024). Enhanced Methodology for Estimating Soil Shear Strength Parameters Across Varied Soil Types Using Machine Learning Integration, *Open Geoscience Journal*. De Gruyter, ISSN: 2391-5447. (Submitted)

Eteraf, H., Kovacs, B., Mikita, V., & Delshad, Z. (2023). Effect of particle size distribution on shear strength of soil. In *MATEC Web of Conferences* (Vol. 385, p. 01043). EDP Sciences. DOI: 10.1051/mateconf/202338501043 (Published)

Eteraf, H., Mosallaei, A., Kovács, B., & Mikita, V. (2020). Estimation of hydraulic conductivity of soil from grain size distribution: A Review. *Muszaki Foldtudományi Közlemények*, 89(2), 316-326. (Published)

Eteraf, H., Mosallaei, A., Kovács, B., & Mikita, V. (2022). Effect of Water Content on Consolidation Parameters. *Research Developments in Geotechnics, Geo-Informatics and Remote Sensing*, 161–163. https://doi.org/10.1007/978-3-030-72896-0_35 (Published)

Eteraf, H., Kovács, B., & Mikita, V. (2019). Effect of the compaction energy on pre-consolidation stress and consolidation parameters of clayey soil-experimental study. In Szigyártó, I.-L., & Szikszai, A. (Eds.), *XV. Kárpát-medencei Környezettudományi Konferencia* (pp. 199-210). Cluj-Napoca, Romania: Ábel Kiadó. (Kárpát-Medencei Környezettudományi Konferencia, ISSN 1842-9815). Cluj-Napoca: Sapientia Hungarian University of Transylvania. (Published)

Eteraf, H., Madarász, T., Mosallaei, A., Kovács, B., & Mikita, V. (2022). A Review of the Role of Geosynthetic Clay Liners in Contaminated Site Remediation. *Geosciences and Engineering*, 10(15), 80–95. <https://doi.org/10.33030/geosciences.2022.15.080> (Published)

Mikita, V., Kovács, B., Czinkota, I., **Eteraf, H.**, & Pinjung, Z. (2022). The Investigation of the Stress Field Evolution in Soil Samples During Vertical Loads with Computer Tomography. *Research Developments in Geotechnics, Geo-Informatics and Remote Sensing*, 123–126. https://doi.org/10.1007/978-3-030-72896-0_27 (Published)

Bibliography

- A. Prieto, B. P. (2016). Neural Networks: An Overview of Early Research, Current Frameworks and New Challenges. *Neurocomputing*, 242-268.
doi:<https://doi.org/10.1016/j.neucom.2016.06.014>
- A. T. T. I. L. A. Nemes, W. J. (2004). Soil Texture and Particle-Size Distribution as Input to Estimate Soil Hydraulic Properties. In *Developments in Soil Science* (Vol. 30, pp. 47-70).
- A. Viana da Fonseca, C. F. (2015). Improved Laboratory Techniques for Advanced Geotechnical Characterization towards Matching In Situ Properties. In *Deformation Characteristics of Geomaterials* (Vol. 6). IOS Press. doi:<http://dx.doi.org/10.3233/978-1-61499-601-9-231>
- Abbaszadeh Shahri, A. (2016). Assessment and prediction of liquefaction potential using different artificial neural network models: a case study. *Geotechnical and Geological Engineering*, 807-815. doi:<https://doi.org/10.1007/s10706-016-0004-z>
- Abdolrasol, M. G. (2021). Artificial neural networks based optimization techniques: A review. *Electronics*, 143. doi:<https://doi.org/10.3390/electronics10212689>
- Abioye, S. O. (2021). Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges. *Journal of Building Engineering*. doi:<https://doi.org/10.1016/j.jobe.2021.103299>
- Agarap, A. F. (2018). *Deep learning using rectified linear units (relu)*. arXiv preprint. doi:<https://doi.org/10.48550/arXiv.1803.08375>
- Aimrun, M. S. (2004). Effective Porosity of Paddy Soils as an Estimation of its Saturated Hydraulic Conductivity. *Geoderma*, 197-203.
doi:<https://doi.org/10.1016/j.geoderma.2003.11.010>
- Akbulut, S. (2005). Artificial neural networks for predicting the hydraulic conductivity of coarse-grained soils. *Eurasian Soil Science/Pochvovedenie*, 392-398.
- Al-Doury, M. M. (2010). A Discussion About Hydraulic Permeability and Permeability. *Petroleum Science and Technology*, 1740-1749.
doi:<https://doi.org/10.1080/10916460903261715>
- Alghafri, A. S. (2021). Hebb's Theory of Learning in Cognitive Neuroscience. *International Journal of Humanities and Educational Research*, 259-265.
doi:<http://dx.doi.org/10.47832/2757-5403.5-3.23>
- Ameratunga, N. S. (2016). *Correlations of Soil and Rock Properties in Geotechnical Engineering*. doi:<https://doi.org/10.1007/978-81-322-2629-1>
- Arshad, R. R. (2013). Predicting saturated hydraulic conductivity by artificial intelligence and regression models. *International Scholarly Research Notices*. doi:<https://doi.org/10.1155/2013/308159>

- B. D. Klein, D. R. (1999). Data Quality in Linear Regression Models: Effect of Errors in Test Data and Errors in Training Data on Predictive Accuracy. *Informing Science*, 33. doi:<http://dx.doi.org/10.28945/599>
- B. Kuang, H. S. (2012). Sensing Soil Properties in the Laboratory, In Situ, and On-line: A Review. In *Advances in Agronomy* (pp. 155-223). Academic Press. doi:<https://doi.org/10.1016/B978-0-12-394275-3.00003-1>
- B. M. Das, N. S. (2017). *Fundamentals of Geotechnical Engineering*. Cengage Learning.
- Capra, M. B. (2020). Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead. *IEEE Access*, 225134-225180. doi:<https://doi.org/10.1109/ACCESS.2020.3039858>
- Carrier, W. D. (2003). Goodbye, Hazen; hello, Kozeny-Carman. *Journal of Geotechnical and Geoenvironmental Engineering*, 1054-1056. doi:[https://doi.org/10.1061/\(asce\)1090-0241\(2003\)129:11\(1054\)](https://doi.org/10.1061/(asce)1090-0241(2003)129:11(1054))
- Chandel, A. &. (2022). Grain-size model for hydraulic conductivity estimation of porous media. *Water Practice & Technology*, 1836-1848. doi:<http://dx.doi.org/10.2166/wpt.2022.101>
- Chapuis, R. P. (2004). Predicting the Saturated Hydraulic Conductivity of Sand and Gravel Using Empirical Models. *Canadian Geotechnical Journal*, 787-796. doi:<http://dx.doi.org/10.1139/t04-022>
- Chen, K. (2015). Deep and modular neural networks. In *Springer Handbook of Computational Intelligence* (pp. 473-494). doi:https://doi.org/10.1007/978-3-662-43505-2_28
- Chen, R. P. (2019). Prediction of maximum surface settlement caused by earth pressure balance (EPB) shield tunneling with ANN methods. *Soils and Foundations*, 284-295. doi:<https://doi.org/10.1016/j.sandf.2018.11.005>
- Chin, W. W. (1998). The partial least squares approach to structural equation modeling. *Modern Methods for Business Research*, 295-336.
- Citakoglu, H. (2017). Comparison of Artificial Intelligence Techniques for Prediction of Soil Temperatures in Turkey. *Theoretical and Applied Climatology*, 545-556. doi:<https://link.springer.com/article/10.1007/s00704-016-1914-7>
- Craig, R. F. (2004). *Craig's Soil Mechanics*. CRC Press. doi:<https://doi.org/10.4324/9780203494103>
- Darcy, H. (1856). *Les fontaines publiques de la ville de Dijon: exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau*. Victor Dalmont. doi:<https://doi.org/10.1111/jawr.1965.1.2.4>
- Dong, Y. H. (2020). Research on how human intelligence, consciousness, and cognitive computing affect the development of artificial intelligence. *Complexity*.

- Eldan, R. &. (2016). The power of depth for feedforward neural networks. *Conference on Learning Theory* (pp. 907-940). PMLR.
- El-Gohary, K. M.-K. (2017). Engineering approach using ANN to improve and predict construction labor productivity under different influences. *Journal of Construction Engineering and Management*.
- Elhakim, A. F. (2016). Estimation of Soil Permeability. *Alexandria Engineering Journal*, 55(3), 2631-2638. doi:<https://doi.org/10.1016/j.aej.2016.07.034>
- European Committee for Standardization. (2007). European Committee for Standardization. (2007). EN 1997-2:2007, Eurocode 7 — Geotechnical design — Part 2: Ground investigation and testing. European Committee for Standardization. <https://www.cen.eu>.
- F. Meskini-Vishkaee, N. D. (2018). Evaluation of Different Predictor Models for Detailed Soil Particle-Size Distribution. *Pedosphere*, 28(1), 157-164.
- G. W. Gee, D. O. (2002). Particle-size analysis. In *Methods of Soil Analysis: Part 4 Physical Methods* (pp. 255-293). Soil Science Society of America.
- Glorot, X. &. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, (pp. 249-256).
- Goh, A. T. (1995). Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering*, 143-151.
- H. Eteraf, A. M. (2020). Estimation of Hydraulic Conductivity of Soils from Grain-Size Distribution: A Review. *Muszaki Foldtudományi Közlemények*, 316-326.
- H. Eteraf, B. K. (2023). Effect of Particle Size Distribution on Shear Strength of Soil. (p. 01043). EDP Sciences.
- H. Q. Dou, T. C. (2014). Probabilistic Slope Stability Analysis Considering the Variability of Hydraulic Conductivity under Rainfall Infiltration–Redistribution Conditions. *Engineering Geology*, 1-13.
- H. Vereecken, J. M. (1989). Estimating the soil moisture retention characteristic from texture, bulk density, and carbon content. *Water Resources Research*, 897-899.
- H. Vereecken, M. W. (2010). Using pedotransfer functions to estimate the van Genuchten–Mualem soil hydraulic properties: A review. *Vadose Zone Journal*, 795-820.
- Hanne, T. D. (2017). Computational intelligence. In *Springer International Publishing* (pp. 13-41). Springer International Publishing.
- Hazen, A. (1911). Discussion: Dams on Sand Foundations. *Transactions of the American Society of Civil Engineers*, 199-203.

- Huang, Y. T. (1997). Fuzzy representation and reasoning in geotechnical site characterization. *Computers and Geotechnics*, 65-86.
- ISO 14688-2. (2004). Geotechnical Investigation and Testing—Identification and Classification of Soil—Part 2: Principles for a Classification. Comité Européen de Normalisation.
- J. C. Santamarina, G. C. (2004). Soil Behaviour: The Role of Particle Shape. *Advances in Geotechnical Engineering: The Skempton Conference: Proceedings of a Three Day Conference on Advances in Geotechnical Engineering, organised by the Institution of Civil Engineers and held at the Royal Geographical Society, London, UK, on 29–31* (pp. 604-617). Thomas Telford Publishing.
- J. H. M. Wösten, M. T. (1988). Parameter estimation for unsaturated soil hydraulic characteristics. *Soil Science Society of America Journal*, 892-898.
- J. Říha, L. P. (2018). Assessment of Empirical Formulae for Determining the Hydraulic Conductivity of Glass Beads. *Journal of Hydrology and Hydromechanics*, 337-347.
- Jalal, F. E. (2021). Predictive modeling of swell-strength of expansive soils using artificial intelligence approaches: ANN, ANFIS and GEP. *Journal of Environmental Management*.
- Jiadong, Q. O. (2024). Predicting clay compressibility for foundation design with high reliability and safety: A geotechnical engineering perspective using artificial neural network and five metaheuristic algorithms. *Reliability Engineering & System Safety*.
- K. C. Onyelowe, F. F. (2023). Selected AI optimization techniques and applications in geotechnical engineering. *Cogent Engineering*.
- K. E. Saxton, W. J. (1986). Estimating generalized soil-water characteristics from texture. *Soil Science Society of America Journal*, 1031-1036.
- Kamp, G. v. (2001). Methods for Determining the In Situ Hydraulic Conductivity of Shallow Aquitards – An Overview. *Hydrogeology Journal*, 5-16.
- L. L. Zeng, H. W. (2020). Hydraulic Conductivity of Naturally Sedimented and Reconstituted Clays Interpreted from Consolidation Tests. *Engineering Geology*, 105638.
- L. M. Arya, J. F. (1981). A physicoempirical model to predict the soil moisture characteristic from particle-size distribution and bulk density data. *Soil Science Society of America Journal*, 1023-1030.
- L. R. Ahuja, J. W. (1984). Macroporosity to characterize spatial variability of hydraulic conductivity and effects of land management. *Soil Science Society of America Journal*, 699-703.
- LeCun, Y. B. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2278-2324.
- Lu, P. &. (2003). Artificial neural networks and grey systems for the prediction of slope stability. *Natural Hazards*, 383-398.

- M. A. Abdelfattah, S. A. (2007). A Comparative Characterization and Classification of Soils in Abu Dhabi Coastal Area in Relation to Arid and Semi-Arid Conditions Using USDA and FAO Soil Classification Systems. *Arid Land Research and Management*, 245-271.
- M. Belkhatir, T. S. (2013). Effect of Fines Content and Void Ratio on the Saturated Hydraulic Conductivity and Undrained Shear Strength of Sand–Silt Mixtures. *Environmental Earth Sciences*, 70, 2469-2479.
- M. Cozzolino, E. D. (2018). *Geophysical Methods for Cultural Heritage Management*. Cham, Switzerland: Springer.
- M. Khalili-Maleki, R. V. (2022). Prediction of Hydraulic Conductivity Based on the Soil Grain Size Using Supervised Committee Machine Artificial Intelligence. *Earth Science Informatics*, 15(4), 2571-2583.
- M. M. Jasim, R. M.-K.-R. (2019). Prediction of Bearing Capacity, Angle of Internal Friction, Cohesion, and Plasticity Index Using ANN (Case Study of Baghdad, Iraq). *International Journal of Civil Engineering and Technology*, 2670-2679.
- Meyer, V. (2019). Sampling: The Ghost in Front of the Laboratory Door.
- Minasny, B. &. (2002). The neuro-m method for fitting neural network parametric pedotransfer functions. *Soil Science Society of America Journal*, 352-361.
- Minsky, M. L. (1969). *Perceptrons: An introduction to computational geometry*. MIT Press.
- More, S. B. (2018). Estimation of saturated hydraulic conductivity using fuzzy neural network in a semi-arid basin scale for murum soils of India. *ISH Journal of Hydraulic Engineering*, 140-146.
- More, S. B. (2022). Machine learning-based modeling of saturated hydraulic conductivity in soils of tropical semi-arid zone of India. *Sādhanā*, 26.
- N. K. Twarakavi, J. Š. (2010). Can Texture-Based Classification Optimally Classify Soils with Respect to Soil Hydraulics? *Water Resources Research*.
- N. Prieto-Taboada, C. I.-A. (2014). The Problem of Sampling on Built Heritage: A Preliminary Study of a New Non-Invasive Method. *Environmental Science and Pollution Research*, 21(21), 12518-12529. doi:<https://doi.org/10.1007/s11356-014-3119-x>
- Nathans, L. L. (2012). Interpreting multiple linear regression: A guidebook of variable importance. *Practical Assessment, Research & Evaluation*.
- Opsal, Ø. L. (2018). Geological Parameters and Shear Strength of Dry Till from the Southern Half of Norway in Relation to Bedrock Geology.
- O'shea, K. &. (2015). *An introduction to convolutional neural networks*. arXiv:1511.08458.
- P. V. Sivapullaiah, A. S. (1985). A New Approach for the Determination of Hydraulic Conductivity of Fine-Grained Soils. *Geotechnical Testing Journal*, 189-191.

- P. Y. Hicher, A. M. (1996). Identifying soil parameters by means of laboratory and in situ testing. *Computers and Geotechnics*, 153-170. doi:[https://doi.org/10.1016/0266-352X\(96\)00040-7](https://doi.org/10.1016/0266-352X(96)00040-7)
- Park, H. (2011). Study for application of artificial neural networks in geotechnical problems. In *Artificial neural networks-application* (pp. 303-336).
- Park, H. (2011). Study for Application of Artificial Neural Networks in Geotechnical Problems. In *Artificial Neural Networks - Application* (pp. 303-336).
- Pham, B. T. (2020). Using Artificial Neural Network (ANN) for prediction of soil. *Vietnam Journal of Earth Sciences*, 311-319.
- Pitts, W. &. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 115-133.
- Pratiwi, H. W. (2020). Sigmoid activation function in selecting the best model of artificial neural networks. *Journal of Physics: Conference Series*.
- Q. Gao, M. H. (2024). Geophysical Estimation of 2D Hydraulic Conductivity for Groundwater Assessment in Hard Rock. *Acta Geophysica*, 1-12. doi:<https://doi.org/10.1007/s11600-024-01310-w>
- Qin, H. G. (2020). Binary neural networks: A survey. *Pattern Recognition*.
- R. B. J. Brinkgreve, E. E. (2010). Validation of Empirical Formulas to Derive Model Parameters for Sands. In *Numerical Methods in Geotechnical Engineering* (pp. 137-142).
- R. Saravanan, K. P. (2020). Effect of Particle Grain Size on Its Shear Strength Behaviour of Soils. *IOP Conference Series: Materials Science and Engineering* (p. 032079). IOP Publishing.
- Rasamoelina, A. D. (2020). A review of activation function for artificial neural network. *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII)* (pp. 281-286). IEEE.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 386-408.
- Rumelhart, D. E. (1986). Learning representations by back-propagating errors. *Nature*, 533-536.
- S. Bilardi, D. I. (2020). Predicting the saturated hydraulic conductivity of clayey soils and clayey or silty sands. *Geosciences*, 393.
- S. C. Gupta, W. E. (1979). Estimating soil water retention characteristics from particle size distribution, organic matter percent, and bulk density. *Water Resources Research*, 1633-1635.
- S. Dołęgowska, A. G. (2016). Heterogeneous Areas—Identification of Outliers and Calculation of Soil Sampling Uncertainty Using the Modified RANOVA Method. *Environmental*

- Monitoring and Assessment*, 188, 1-11. doi:<https://doi.org/10.1007%2Fs10661-016-5584-9>
- S. K. Gupta, M. K. (2024). A Comprehensive Review of the Impacts of Chemicals, Soil Salinity, Organic Matter, and Land Use. *IOP Conference Series: Earth and Environmental Science* (p. 012032). IOP Publishing.
- S. K. Gupta, M. K. (2024). Hydraulic Conductivity of Soils: A Comprehensive Review of the Impacts of Chemicals, Soil Salinity, Organic Matter, and Land Use. *1st International Conference on Sustainable Experimentation and Modelling in Civil Engineering 2023. 1327*, p. 012032. Phagwara, India: IOP Publishing. doi:<http://dx.doi.org/10.1088/1755-1315/1327/1/012032>
- S. Nam, M. G. (2011). Determination of the Shear Strength of Unsaturated Soils Using the Multistage Direct Shear Test. *Engineering Geology*, 122(3-4), 272-280. doi:<https://doi.org/10.1016/j.enggeo.2011.06.003>
- S. Sharma, S. A. (2021). A Survey on Applications of Artificial Intelligence for Pre-Parametric Project Cost and Soil Shear-Strength Estimation in Construction and Geotechnical Engineering. *Sensors*, 21(2), 463. doi:<https://doi.org/10.3390/s21020463>
- S. Yoon, S. R. (2015). Estimation of saturated hydraulic conductivity of Korean weathered granite soils using a regression analysis. *Geomechanics and Engineering*, 101-113.
- Schaap, M. G. (2000). Using neural networks to predict soil water retention and soil hydraulic conductivity. *Water Resources Research*, 2123-2130.
- Schmidt, R. M. (2019). *Recurrent neural networks (RNNs): A gentle introduction and overview*. arXiv:1912.05911.
- Shakiba, F. M. (2020). Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. *IEEE Transactions on Industrial Electronics*, 10856-10867.
- Shepard, F. P. (1954). Nomenclature Based on Sand-Silt-Clay Ratios. *Journal of Sedimentary Research*, 151-158.
- Sivakugan, N. E. (1998). Settlement predictions using neural networks. *Australian Civil Engineering Transactions*, 49-52.
- Software, G. (2015). *Voxler User Guide: Visualizing 3D Volumetric Data*. Golden Software.
- Subhashini, M. a. (2020). Employee attrition prediction in industry using machine learning techniques. *International Journal of Advanced Research in Engineering and Technology* , 3329-3341.
- Taleb Bahmed, I. H. (2019). Prediction of geotechnical properties of clayey soils stabilised with lime using artificial neural networks (ANNs). *International Journal of Geotechnical Engineering*, 191-203.
- Terzaghi, K. (1943). *Theoretical Soil Mechanics*. John Wiley & Sons.

- U. Barman, R. D. (2020). Soil Texture Classification Using Multi Class Support Vector Machine. *Information Processing in Agriculture*, 318-332.
- U. Singh, P. K. (2023). Comparison of Saturated Hydraulic Conductivity Estimated by Surface NMR and Empirical Equations. *Journal of Hydrology*, 128929.
- USBR (United States Department of the Interior, B. o. (1990). *Design of Small Dams*. United States Department of the Interior, Bureau of Reclamation.
- Uyanık, G. K. (2013). A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences*, 234-240.
- V. Bolón-Canedo, D. P.-B.-B.-B.-M. (2011). Scalability Analysis of ANN Training Algorithms with Feature Selection. *Conference of the Spanish Association for Artificial Intelligence* (pp. 84-93). Berlin, Heidelberg: Springer Berlin Heidelberg.
- V. K. Singh, D. K. (2020). Modelling of soil permeability using different data driven algorithms based on physical properties of soil. *Journal of Hydrology*, 124223.
- Vanapalli, S. K. (2009). Shear Strength of Unsaturated Soils and Its Applications in Geotechnical Engineering Practice. New Castle, Australia.
- Vanapalli, S. K. (November 2009). Shear Strength of Unsaturated Soils and Its Applications in Geotechnical Engineering Practice. *4th Asia-Pacific Conference on Unsaturated Soils*, (pp. 579-598). New Castle, Australia.
- Wei, X. C. (2023). ANN deformation prediction model for deep foundation pit with considering the influence of rainfall. *Scientific Reports*.
- Weidong Hu, K. L. (2020). Active Earth Pressure against Rigid Retaining Walls for Finite Soils in Sloping Condition considering Shear Stress and Soil Arching. *Advances in Civil Engineering*. doi:<https://doi.org/10.1155/2020/6791301>
- Williams, C. G. (2021). Predictive modelling of soils' hydraulic conductivity using artificial neural network and multiple linear regression. *SN Applied Sciences*.
- Y. Lu, S. L. (2021). Hydraulic Conductivity of Gravelly Soils with Various Coarse Particle Contents Subjected to Freeze–Thaw Cycles. *Journal of Hydrology*, 126302.
- Y. Zhang, M. G. (2019). Estimation of saturated hydraulic conductivity with pedotransfer functions: A review. *Journal of Hydrology*, 1011-1030.
doi:<https://doi.org/10.1016/j.jhydrol.2019.05.058>
- Yilmaz, I. M. (2012). Neural computing models for prediction of permeability coefficient of coarse-grained soils. *Neural Computing and Applications*, 957-968.
doi:<http://dx.doi.org/10.1007/s00521-011-0535-4>
- Z. Han, J. L. (2020). Determining the Shear Strength and Permeability of Soils for Engineering of New Paddy Field Construction in a Hilly Mountainous Region of Southwestern China. *International Journal of Environmental Research and Public Health*, 17(5), 1555.

Zheng, Y. (2023). Optimization of computer programming based on mathematical models of artificial intelligence algorithms. *Computers and Electrical Engineering*. doi:<https://doi.org/10.1016/j.compeleceng.2023.108834>

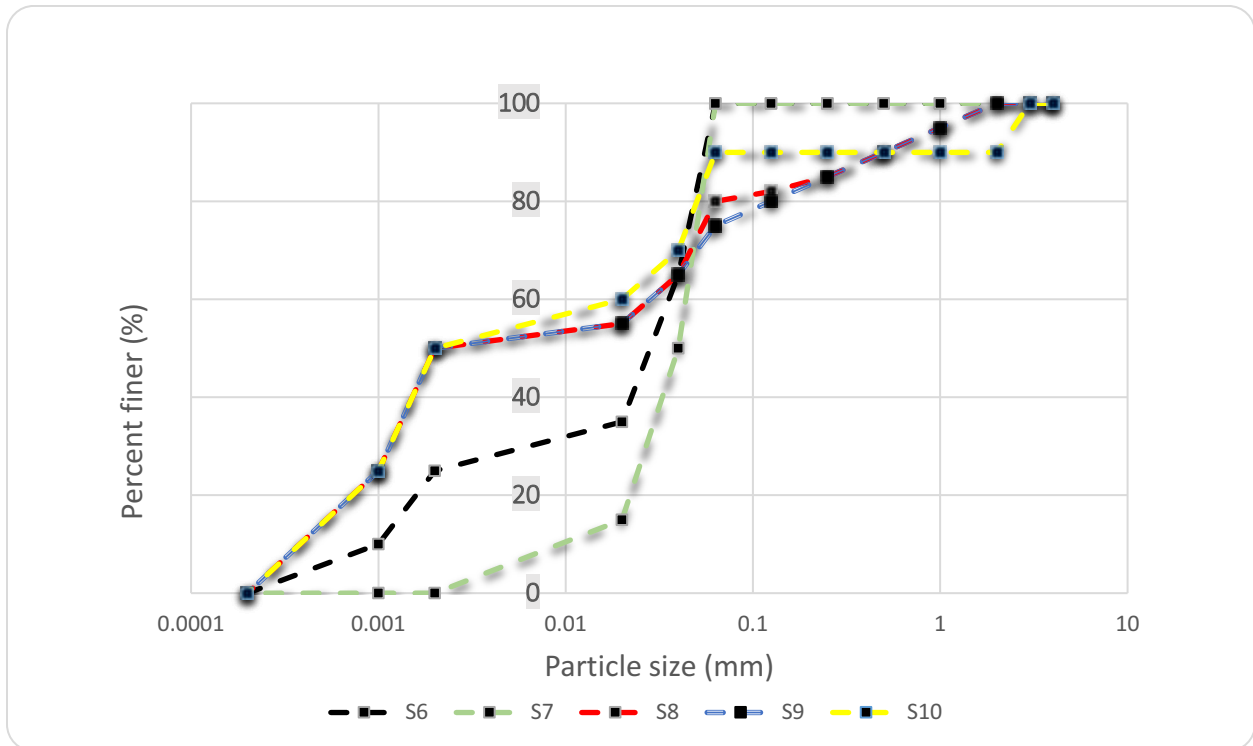
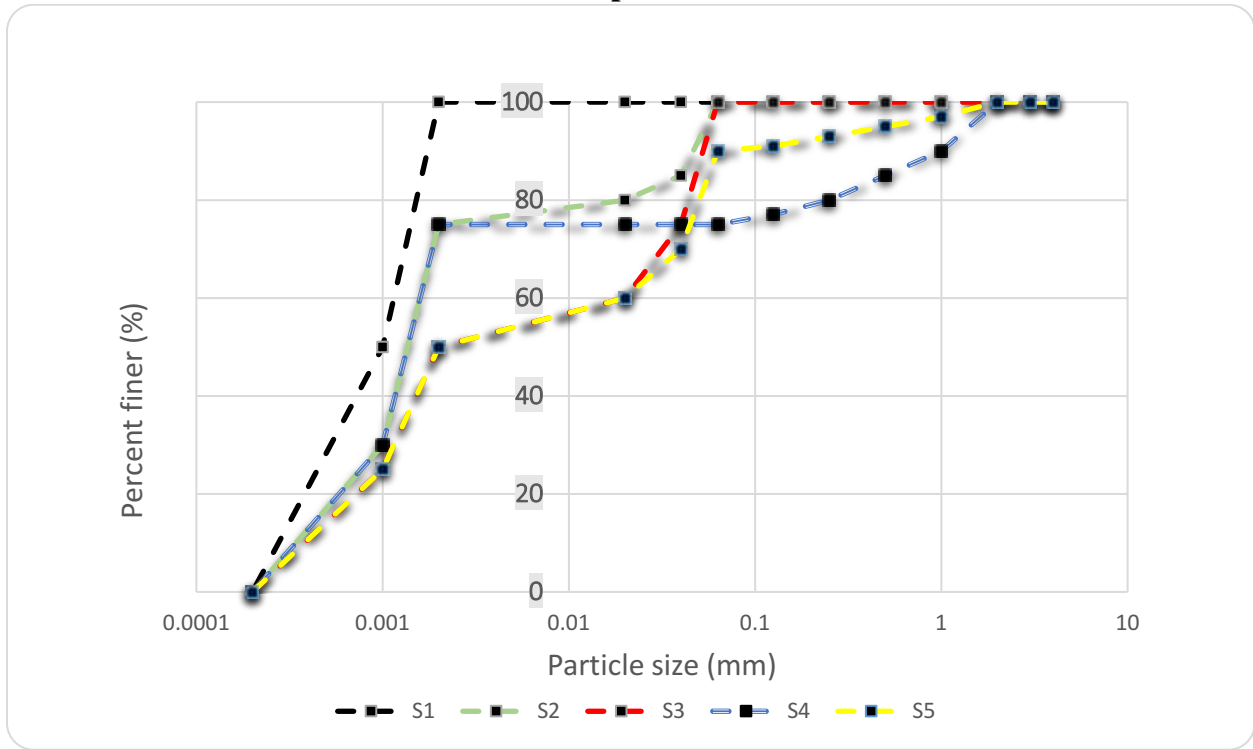
Zou, J. H. (2009). Overview of artificial neural networks. In *Artificial Neural Networks: Methods and Applications*. 14-22. doi:https://doi.org/10.1007/978-1-60327-101-1_2

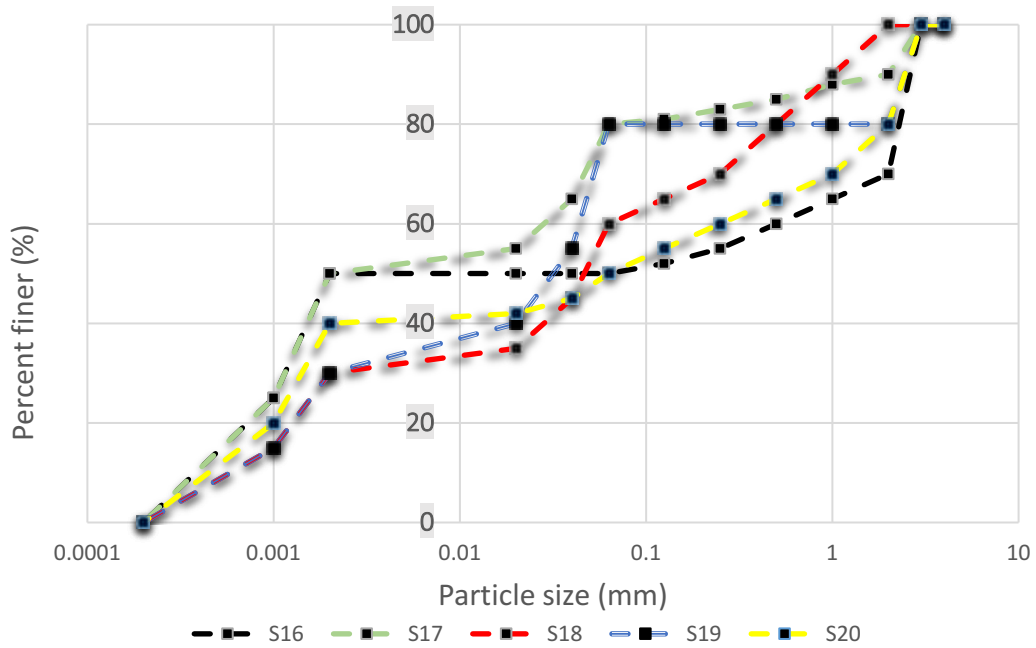
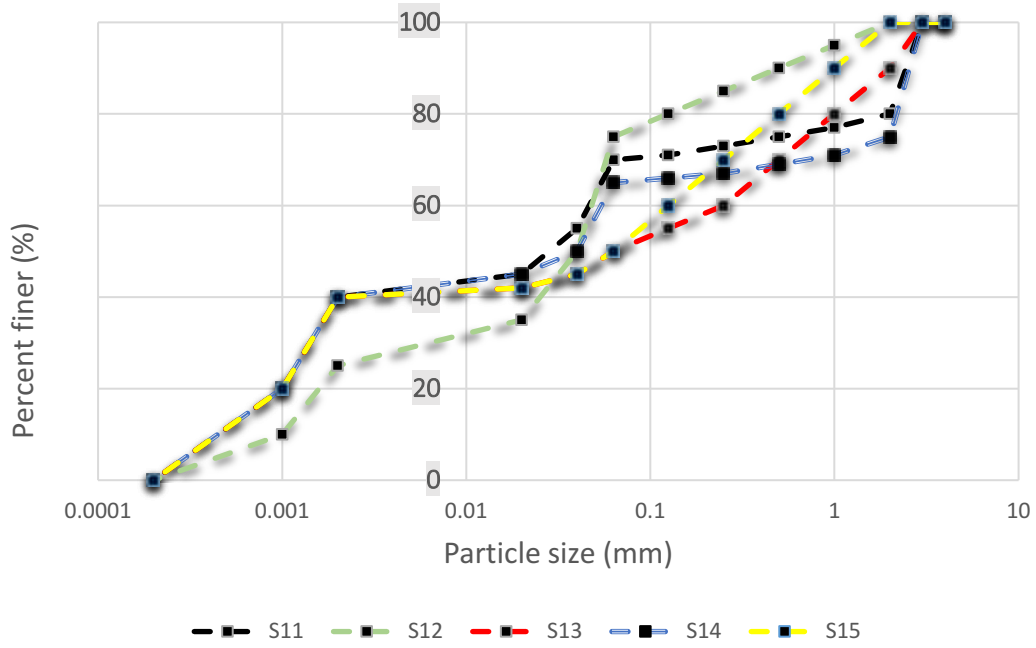
Annex 1. The percentage of soil components in each sample

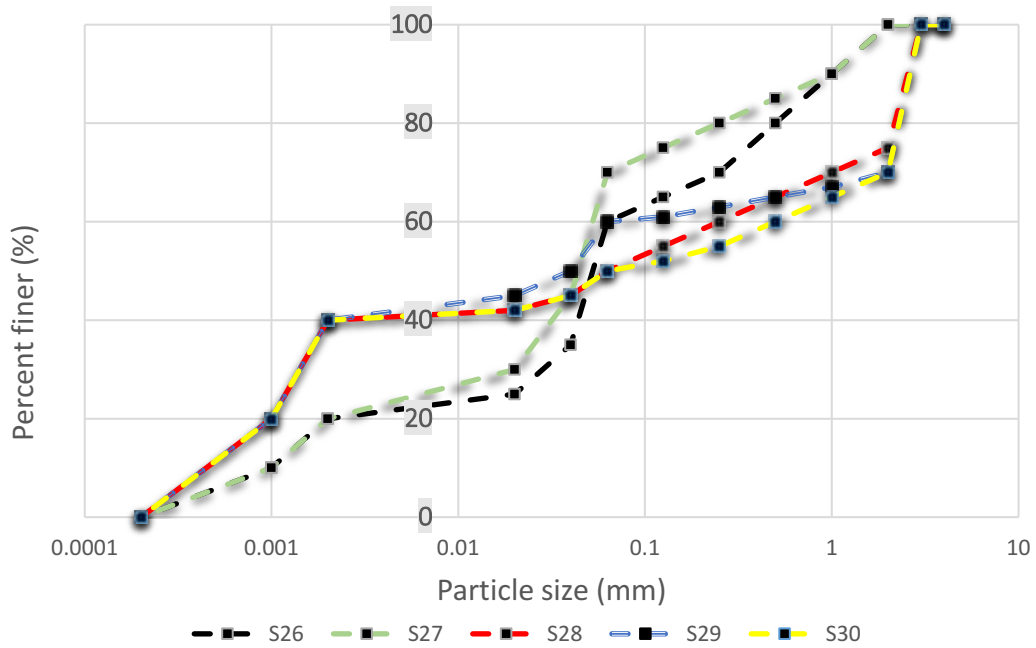
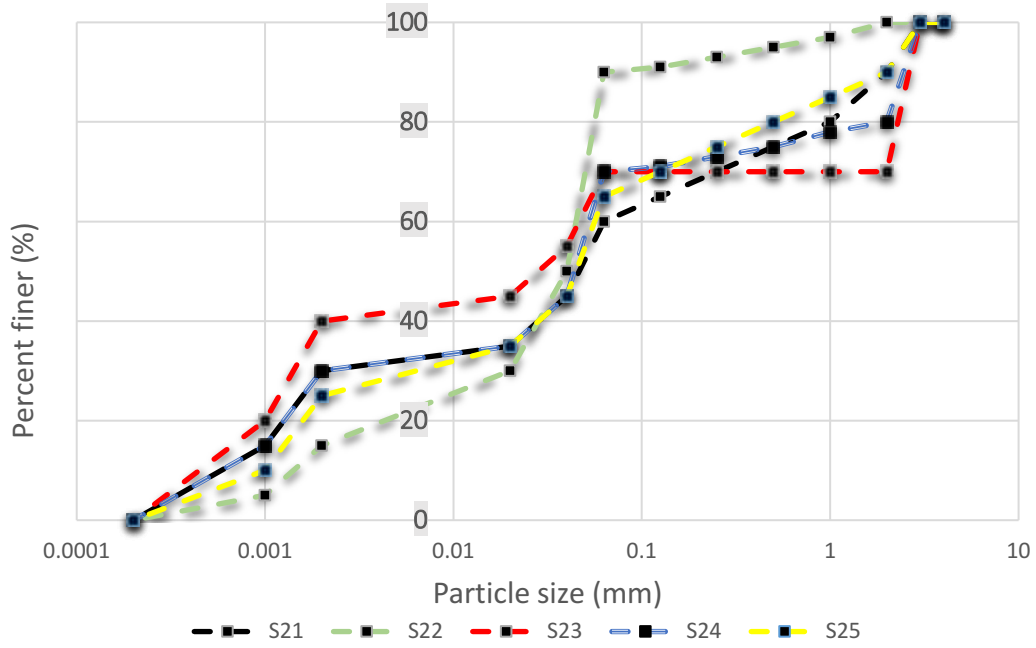
Number	Clay	Silt	Sand	Gravel	Number	Clay	Silt	Sand	Gravel
1	100	0	0	0	51	40	25	25	10
2	75	25	0	0	52	50	0	40	10
3	50	50	0	0	53	40	50	0	10
4	75	0	25	0	54	40	40	10	10
5	50	40	10	0	55	40	30	30	0
6	25	75	0	0	56	50	10	20	20
7	0	100	0	0	57	50	25	0	25
8	50	30	20	0	58	40	20	30	10
9	50	25	25	0	59	30	50	20	0
10	50	40	0	10	60	40	40	0	20
11	40	30	10	20	61	50	0	10	40
12	25	50	25	0	62	40	20	40	0
13	40	10	40	10	63	30	25	20	25
14	40	25	10	25	64	25	25	50	0
15	40	10	50	0	65	0	75	25	0
16	50	0	20	30	66	10	50	30	10
17	50	30	10	10	67	30	20	30	20
18	30	30	40	0	68	20	30	30	20
19	30	50	0	20	69	40	10	10	40
20	40	10	30	20	70	20	50	0	30
21	30	30	30	10	71	25	20	30	25
22	15	75	10	0	72	15	50	10	25
23	40	30	0	30	73	25	25	20	30
24	30	40	10	20	74	30	10	30	30
25	25	40	25	10	75	20	30	25	25
26	20	40	40	0	76	20	40	10	30
27	20	50	30	0	77	20	20	50	10
28	40	10	25	25	78	10	40	50	0
29	40	20	10	30	79	10	50	20	20
30	40	10	20	30	80	30	0	50	20
31	40	0	40	20	81	40	0	20	40
32	50	10	0	40	82	25	20	25	30
33	30	20	40	10	83	10	40	40	10
34	30	20	50	0	84	25	50	0	25
35	20	50	20	10	85	15	40	20	25
36	15	75	0	10	86	30	40	20	10
37	40	0	50	10	87	25	25	25	25
38	30	30	20	20	88	30	20	20	30
39	50	20	30	0	89	10	50	40	0
40	75	0	0	25	90	20	20	30	30
41	50	20	0	30	91	10	50	10	30
42	20	40	30	10	92	10	40	25	25
43	40	50	10	0	93	10	30	50	10
44	50	30	0	20	94	30	10	40	20
45	50	0	50	0	95	0	50	50	0
46	50	0	30	20	96	40	20	0	40
47	40	40	20	0	97	20	25	30	25
48	50	10	40	0	98	30	50	10	10
49	50	10	30	10	99	20	30	40	10
50	50	20	10	20	100	0	75	0	25

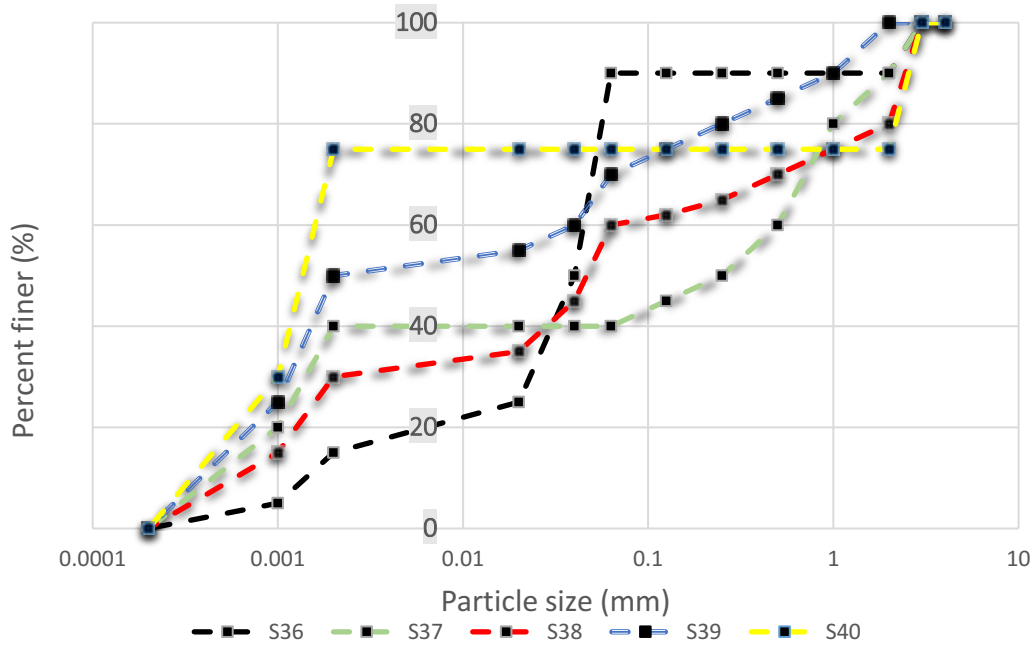
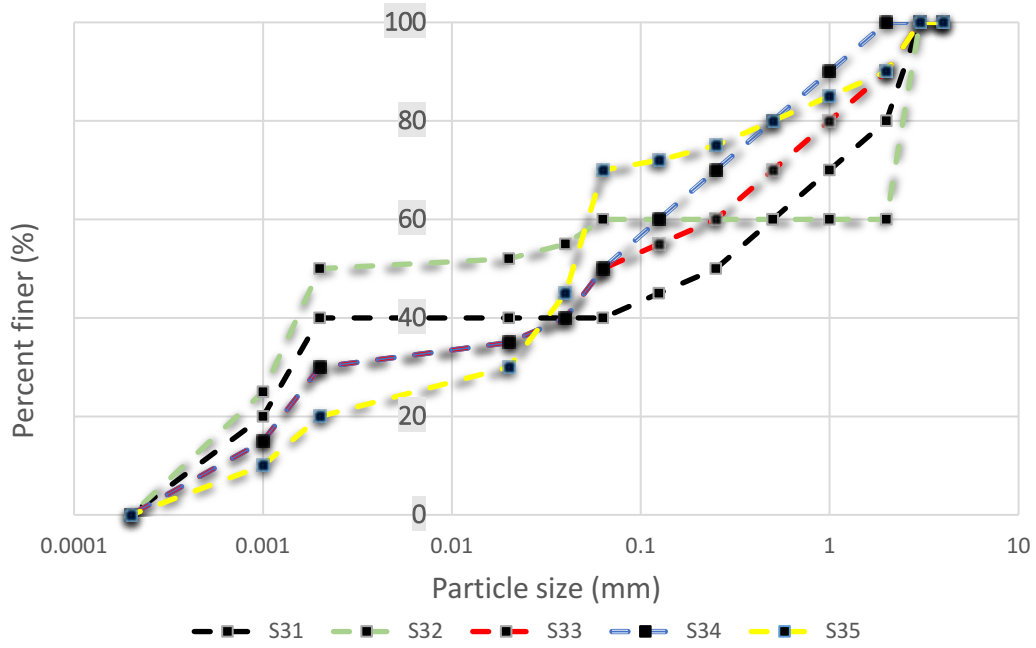
Number	Clay	Silt	Sand	Gravel		Number	Clay	Silt	Sand	Gravel
101	25	30	20	25		153	0	30	50	20
102	40	20	20	20		154	0	50	20	30
103	50	10	10	30		155	15	0	75	10
104	20	30	50	0		156	20	0	50	30
105	25	25	30	20		157	50	0	0	50
106	20	40	20	20		158	0	40	30	30
107	30	40	30	0		159	15	20	25	40
108	30	30	0	40		160	10	40	10	40
109	50	0	25	25		161	10	20	40	30
110	20	20	40	20		162	20	10	30	40
111	30	10	50	10		163	30	10	10	50
112	25	0	75	0		164	25	25	0	50
113	20	10	50	20		165	40	10	0	50
114	40	0	30	30		166	10	30	40	20
115	0	50	40	10		167	20	40	0	40
116	30	30	10	30		168	25	0	50	25
117	30	0	40	30		169	10	25	25	40
118	30	20	25	25		170	20	0	40	40
119	30	10	20	40		171	10	10	50	30
120	30	40	0	30		172	0	50	10	40
121	30	20	10	40		173	0	30	40	30
122	10	40	30	20		174	0	0	100	0
123	25	30	25	20		175	15	10	25	50
124	25	40	10	25		176	0	25	50	25
125	25	25	10	40		177	10	20	30	40
126	20	50	10	20		178	20	20	10	50
127	0	25	75	0		179	10	0	50	40
128	25	25	40	10		180	10	20	20	50
129	20	30	0	50		181	0	50	0	50
130	30	0	20	50		182	20	0	30	50
131	0	50	25	25		183	10	30	10	50
132	0	40	50	10		184	0	20	40	40
133	25	10	25	40		185	0	30	30	40
134	40	0	10	50		186	0	20	50	30
135	30	0	30	40		187	10	10	40	40
136	20	10	40	30		188	20	10	20	50
137	20	30	10	40		189	10	40	0	50
138	10	25	40	25		190	0	40	20	40
139	0	50	30	20		191	25	0	25	50
140	15	20	40	25		192	0	40	10	50
141	10	30	30	30		193	10	10	30	50
142	15	10	75	0		194	10	0	40	50
143	10	20	50	20		195	0	0	75	25
144	10	30	20	40		196	0	10	50	40
145	20	20	20	40		197	0	30	20	50
146	25	10	40	25		198	0	25	25	50
147	10	50	0	40		199	0	20	30	50
148	0	40	40	20		200	25	0	0	75
149	15	10	50	25		201	0	10	40	50
150	20	30	20	30		202	0	0	50	50
151	10	40	20	30		203	0	25	0	75
152	30	20	0	50		204	0	0	25	75
						205	0	0	0	100

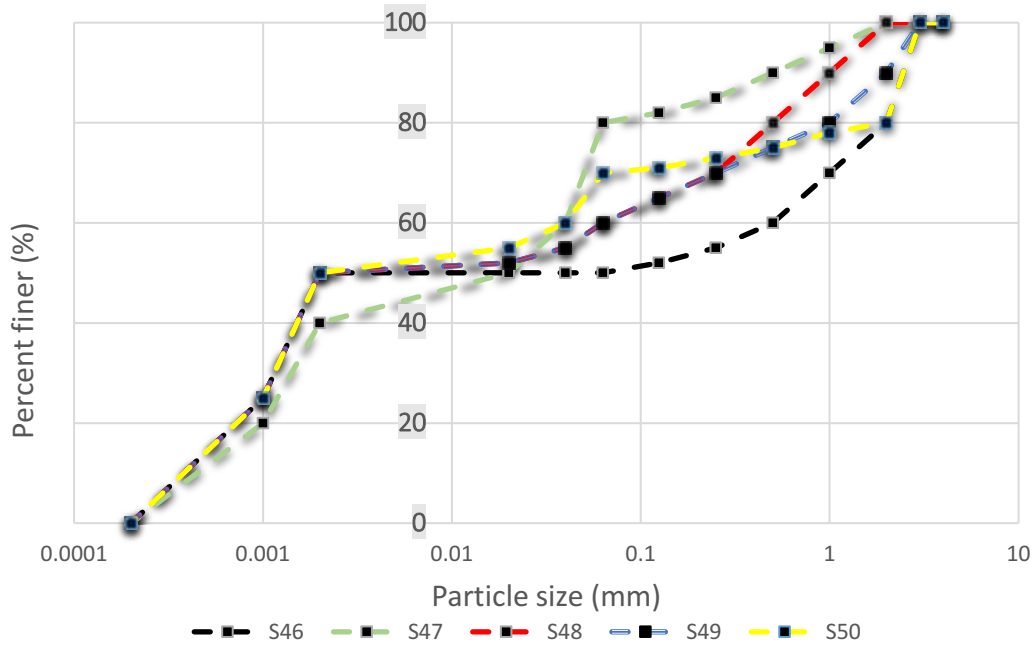
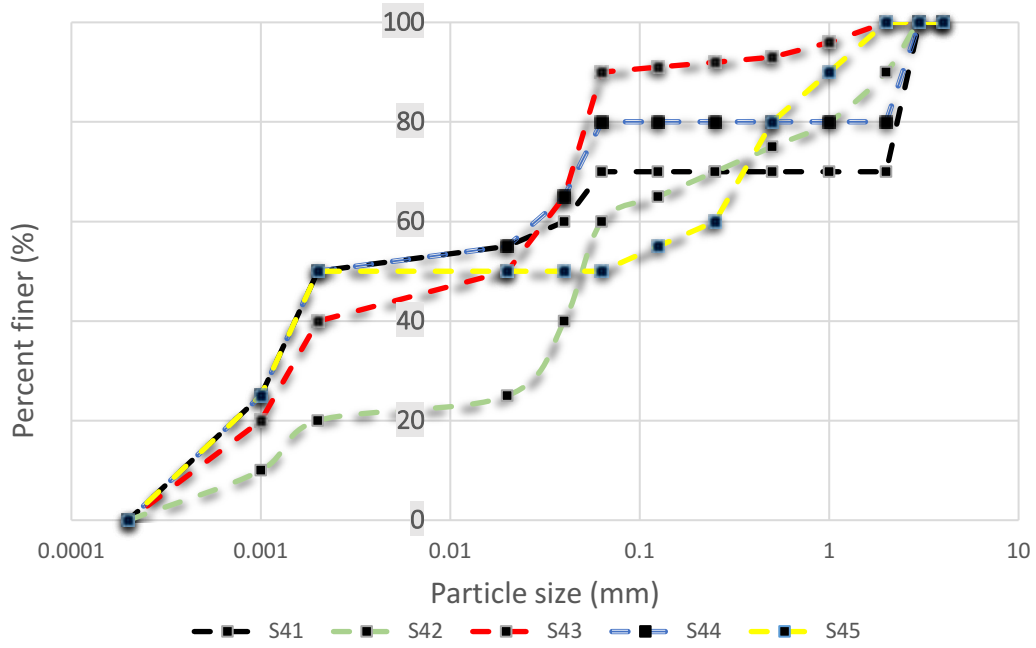
Annex 2. Grain size distributions for all samples

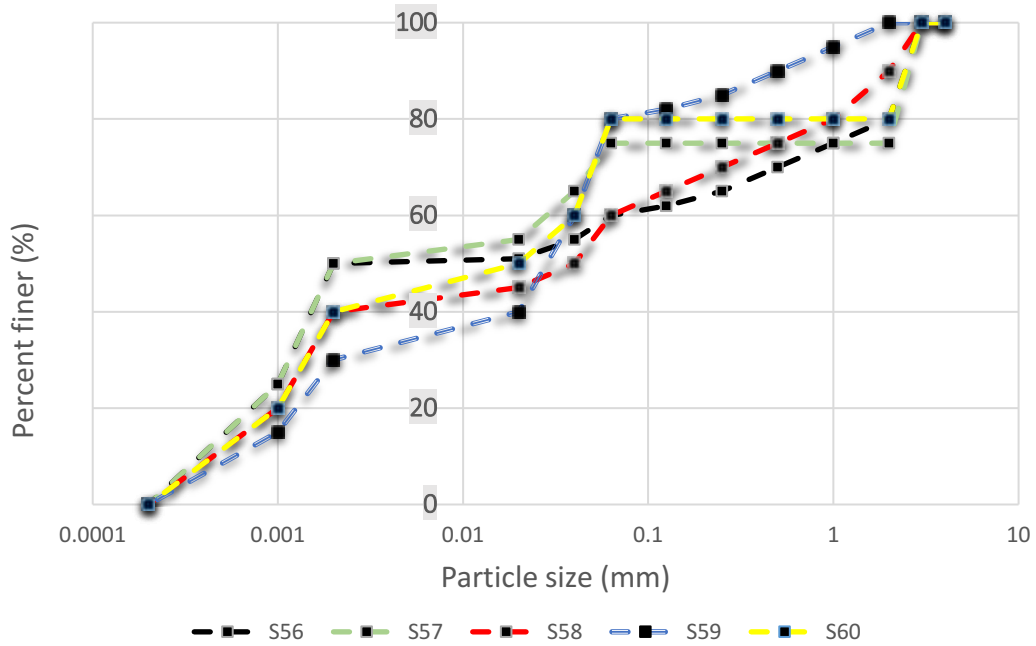
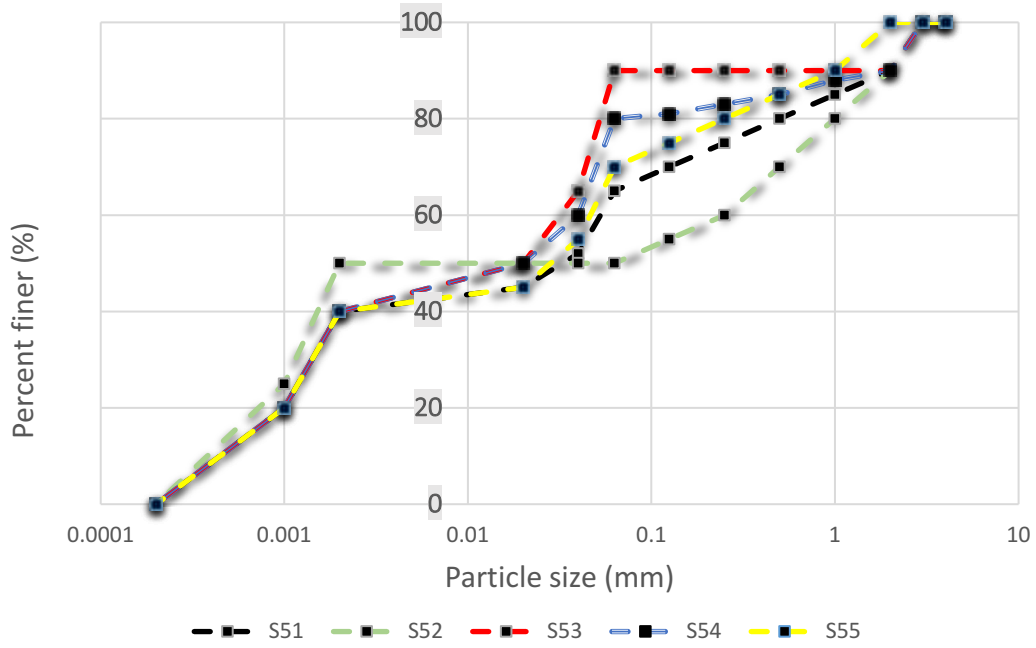


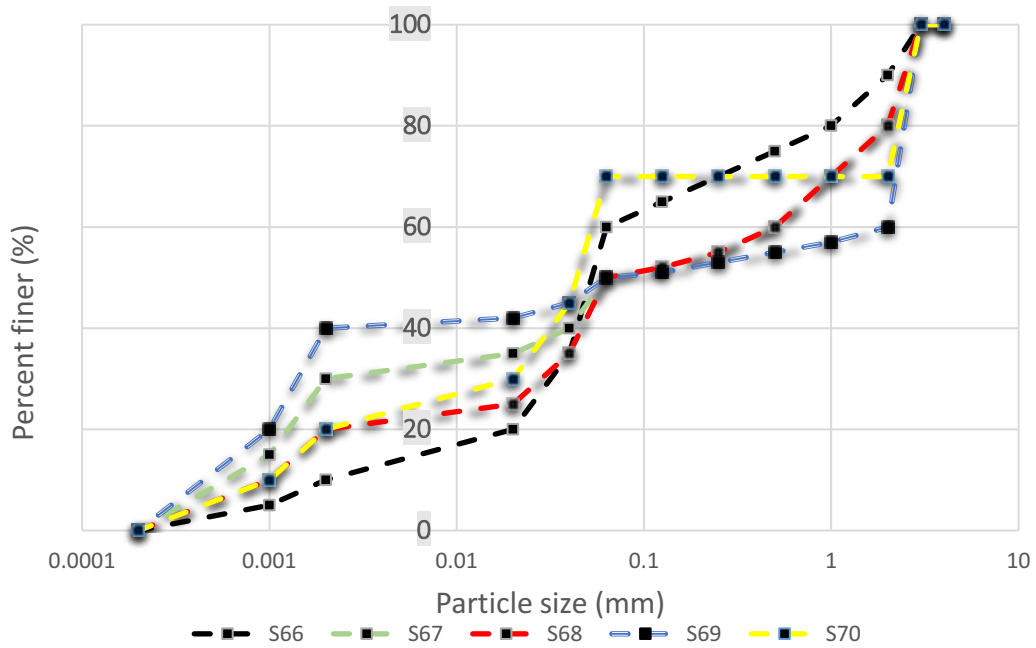
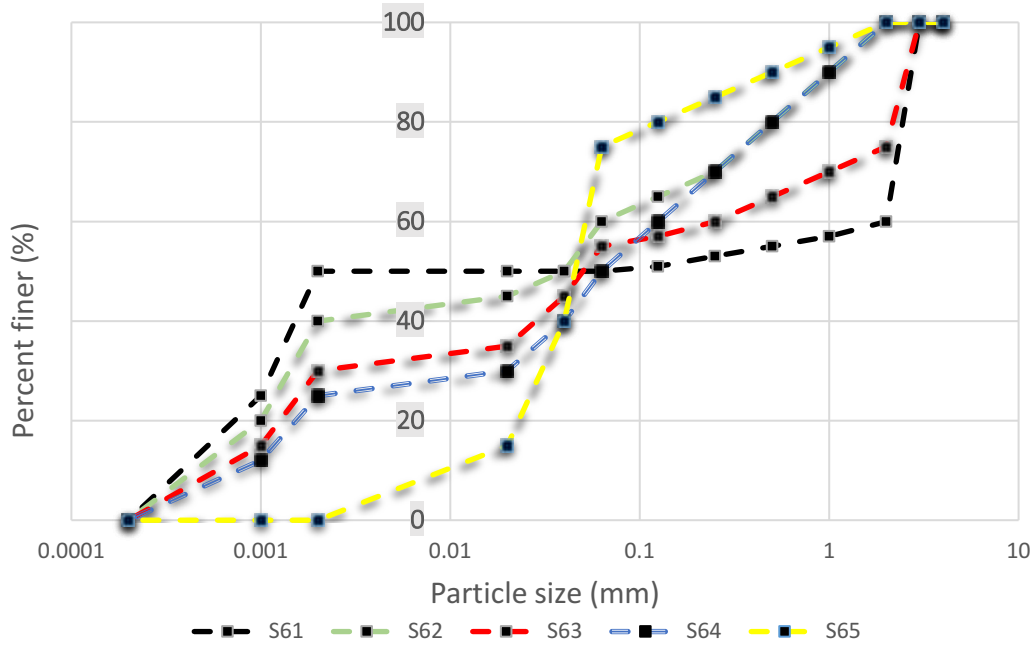


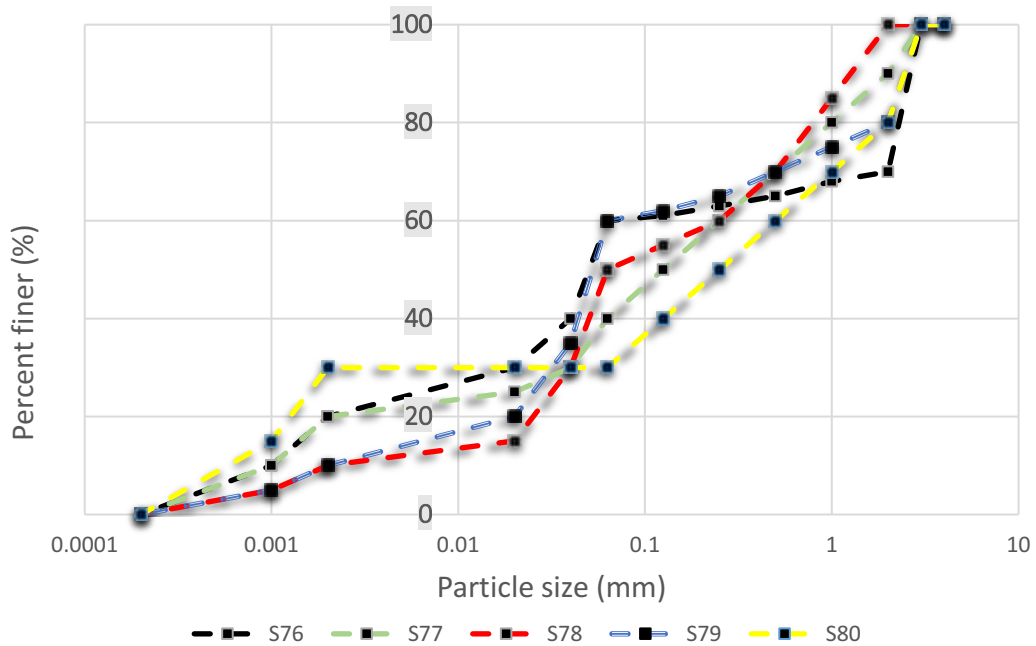
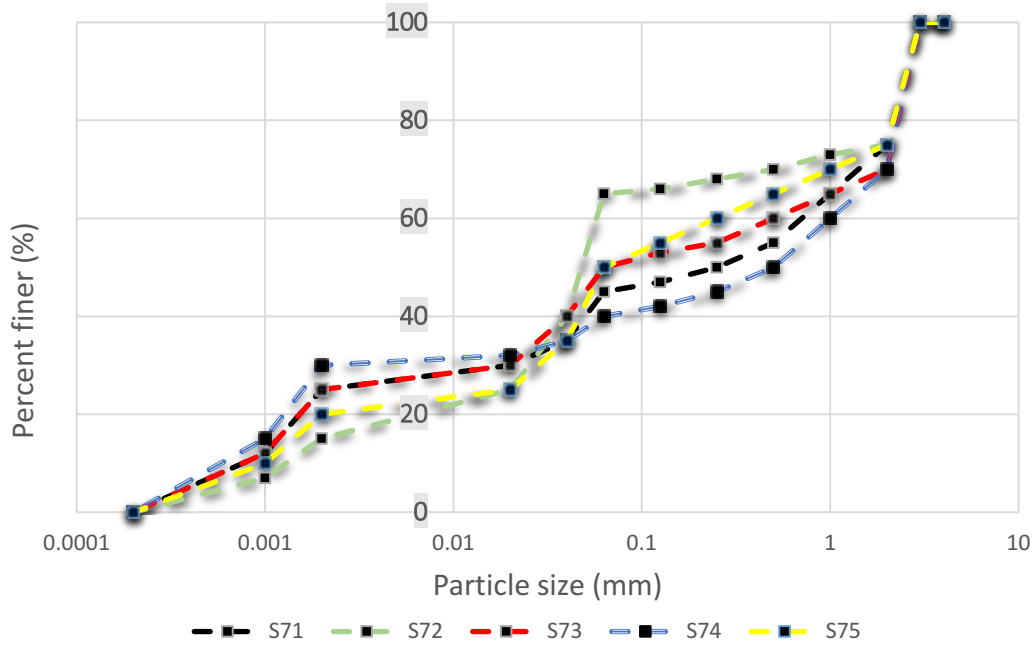


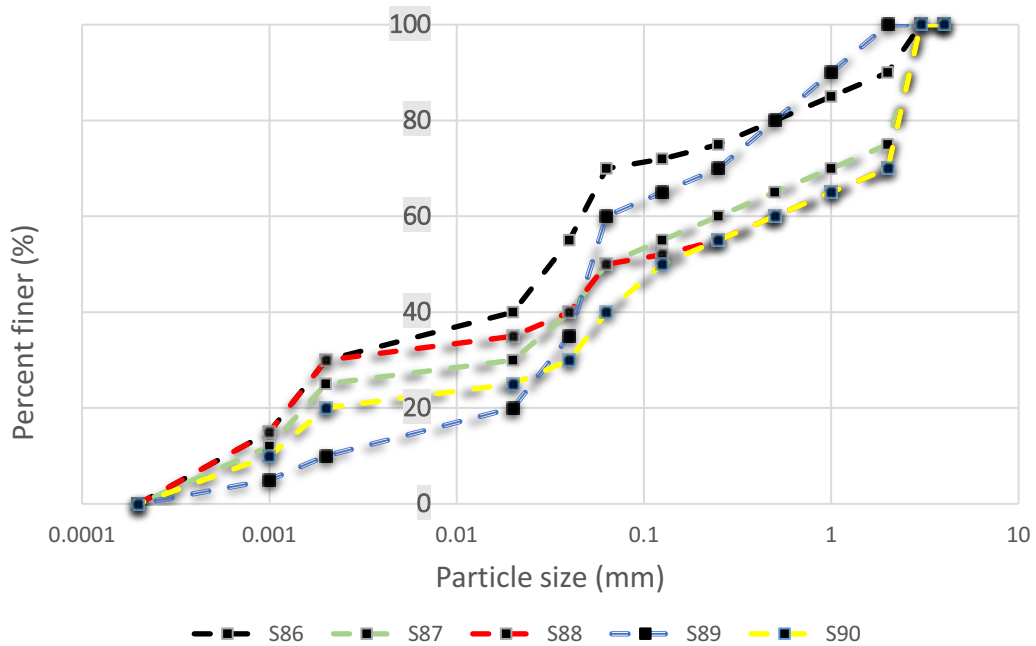
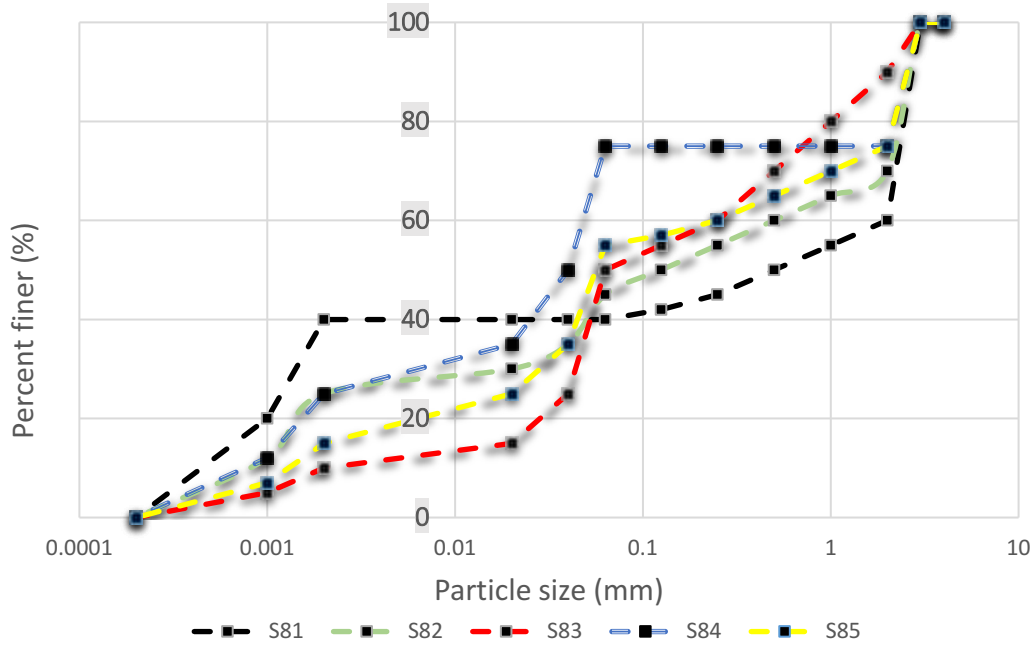


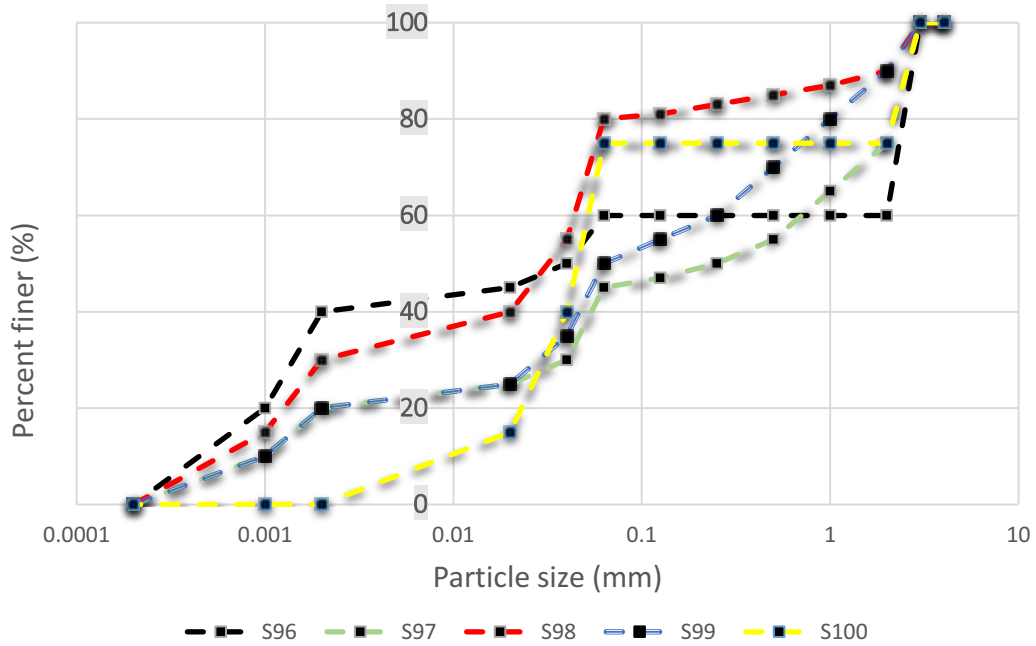
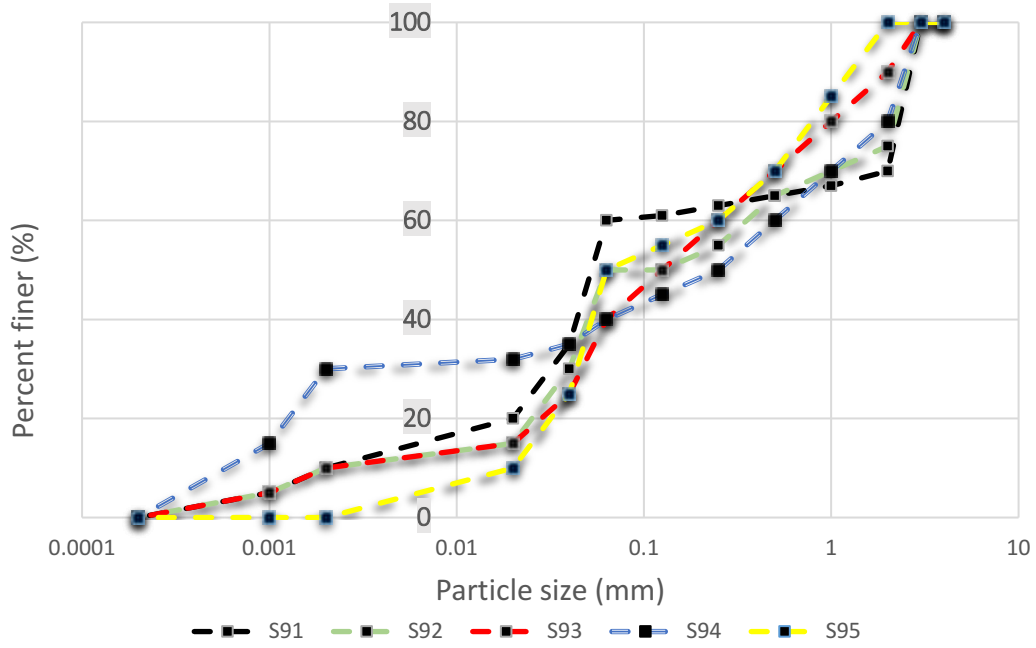


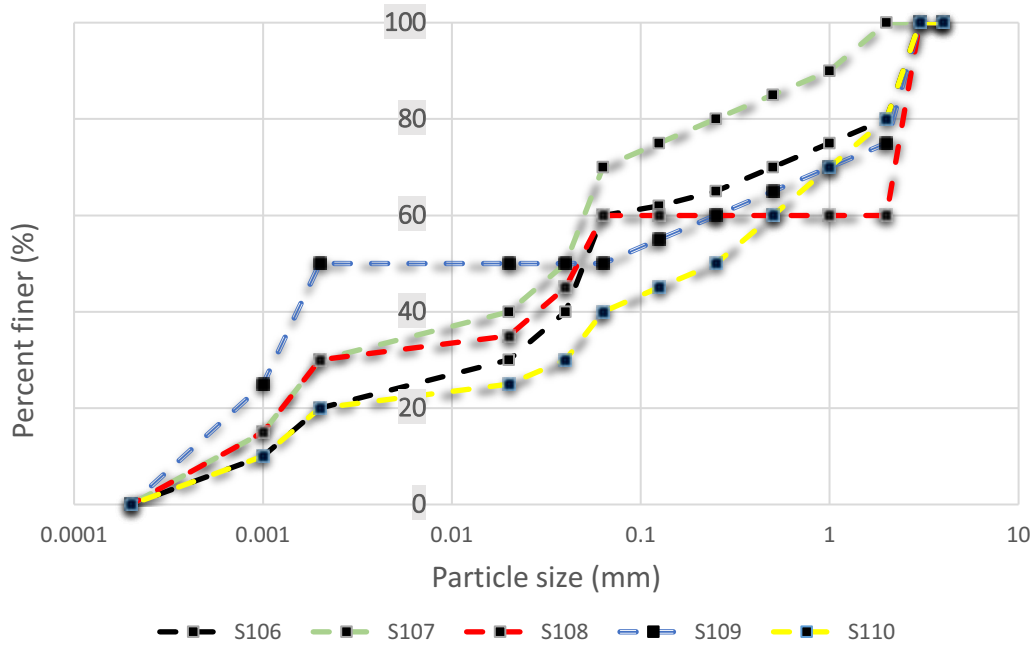
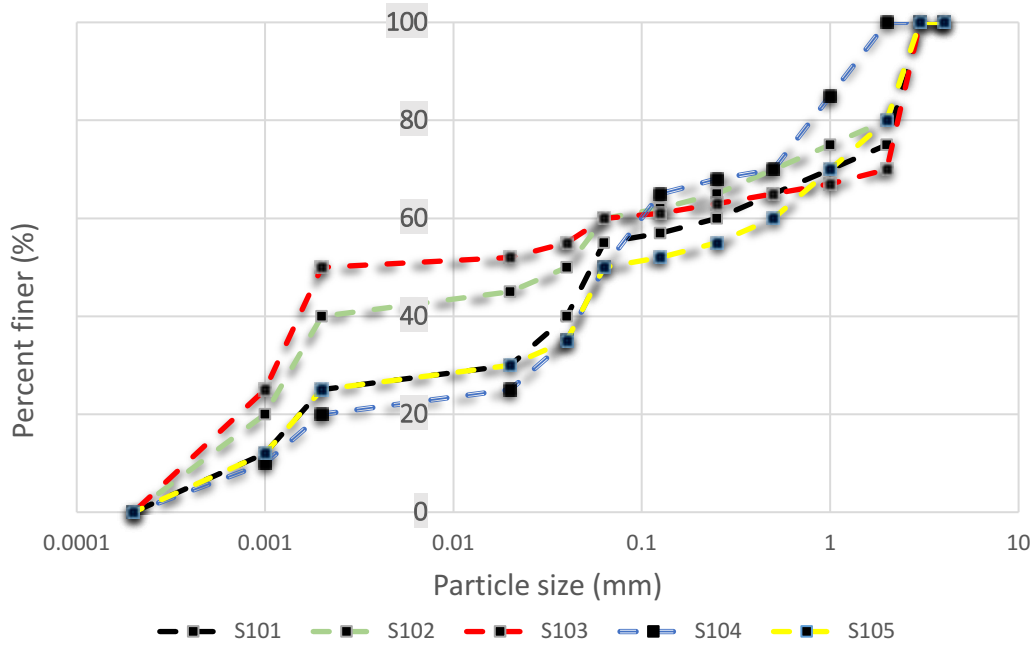


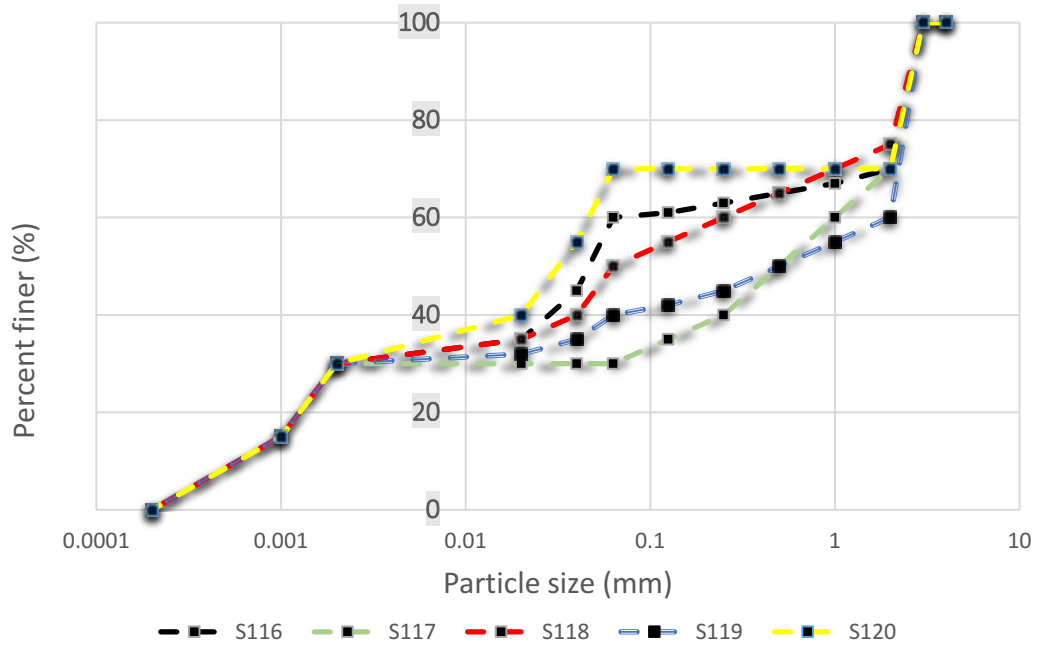
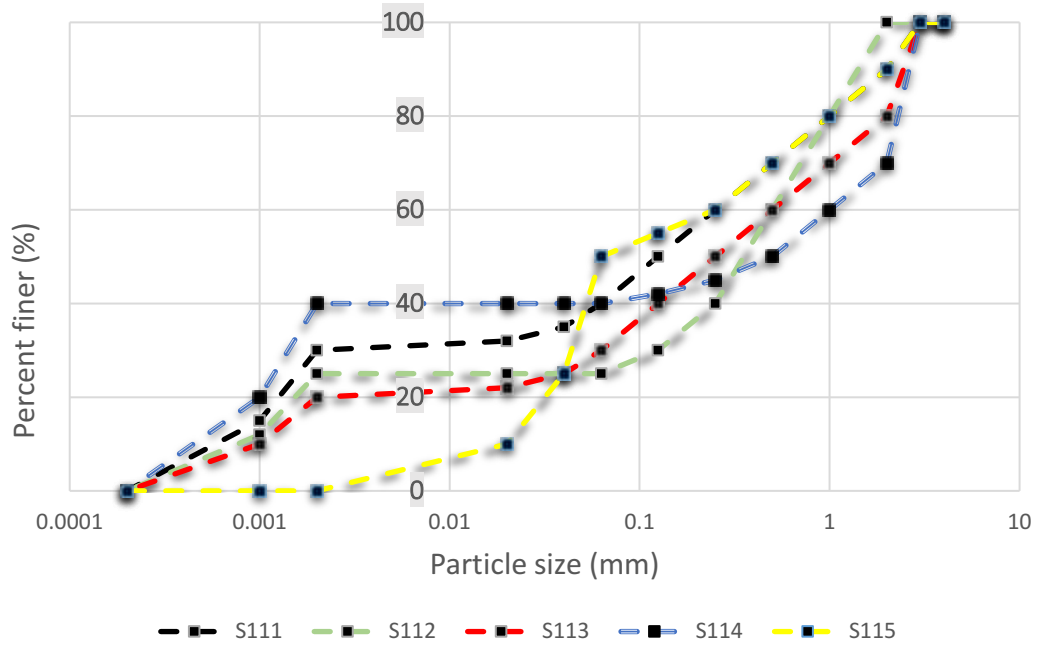


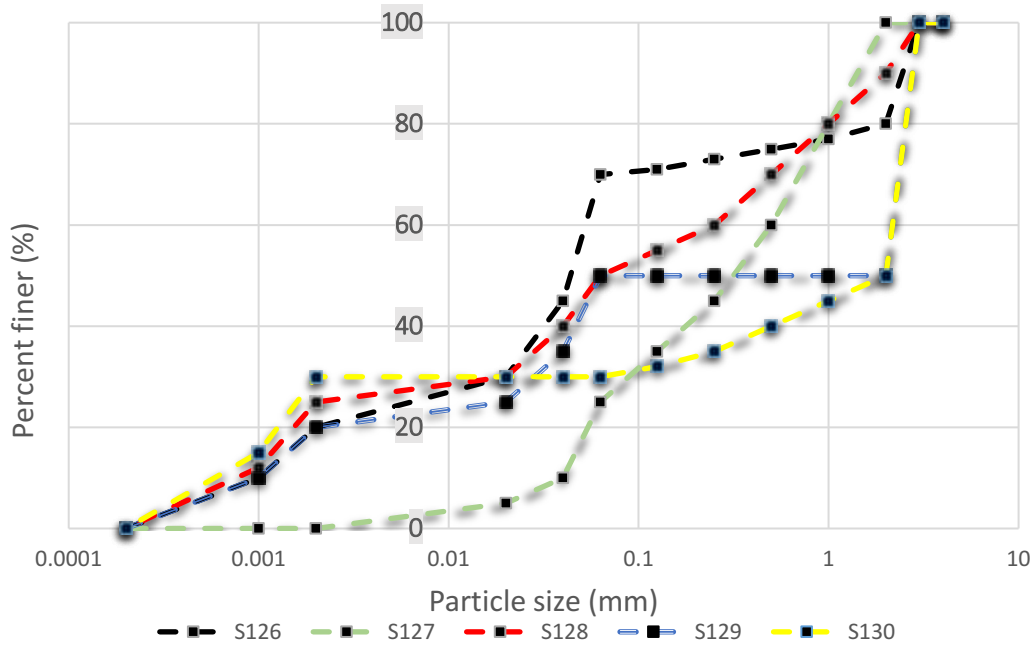
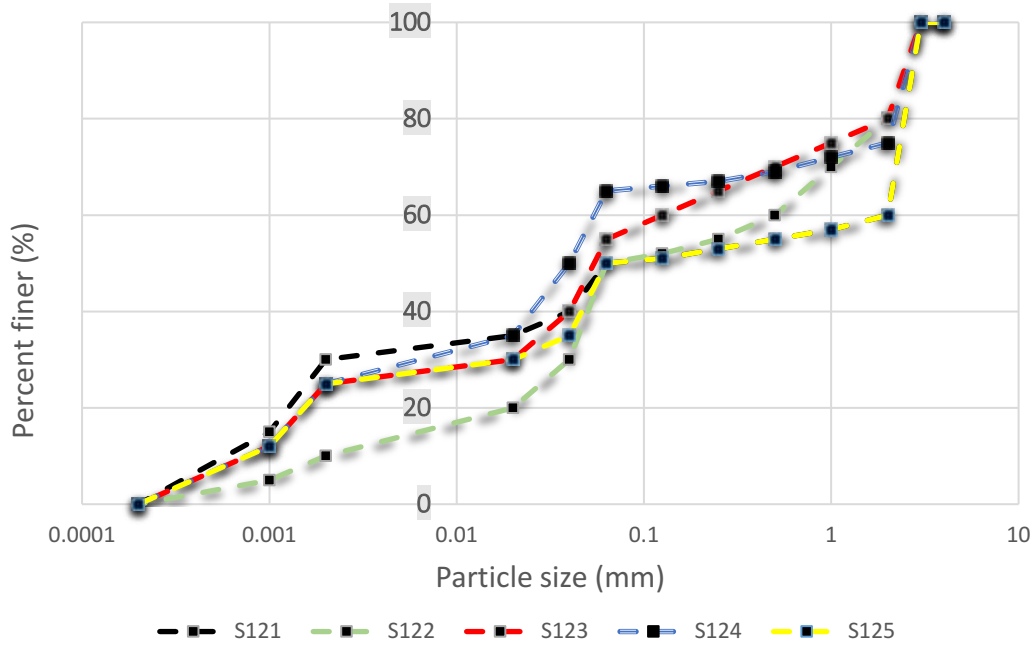


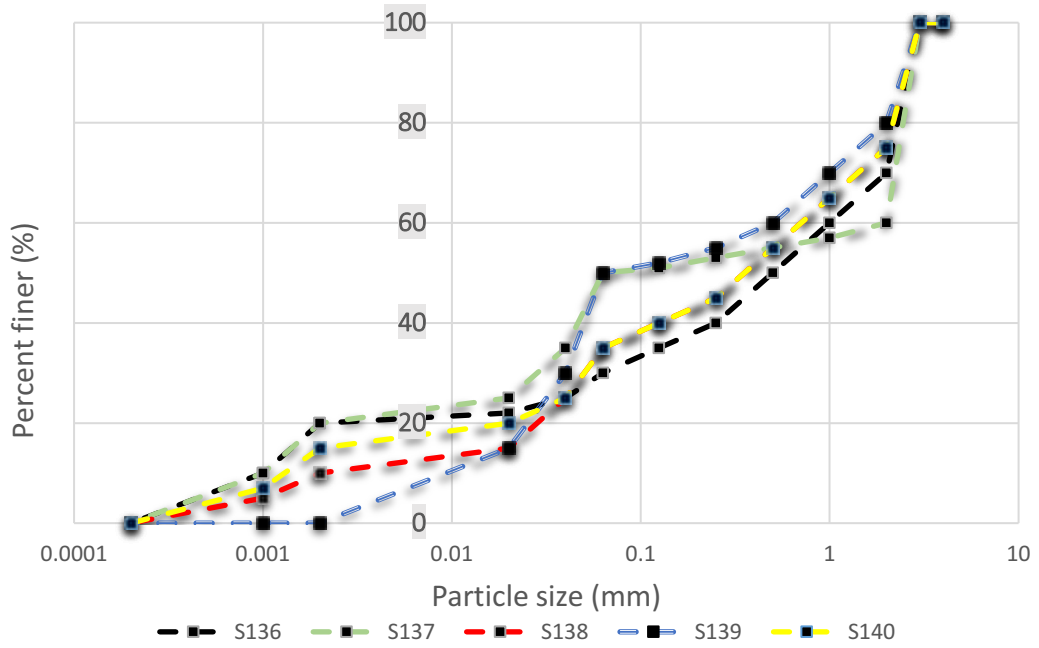
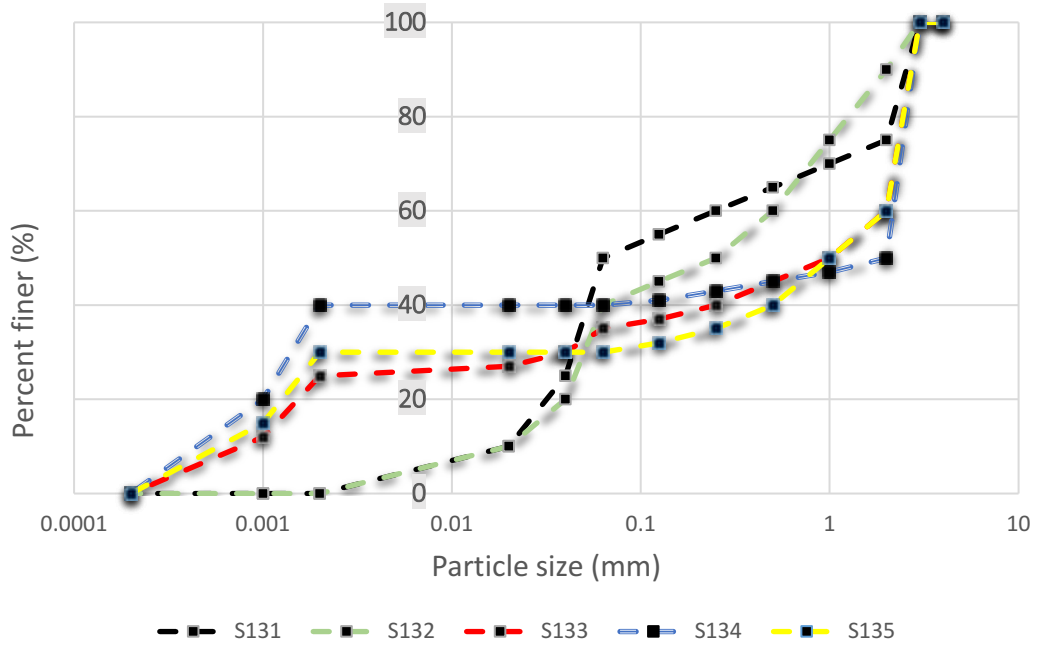


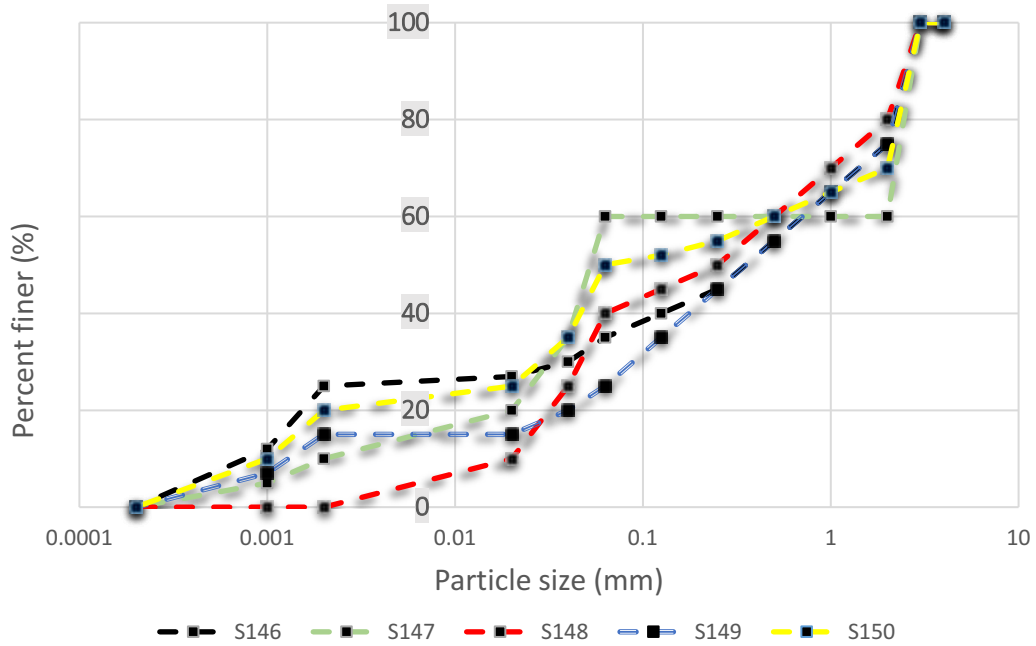
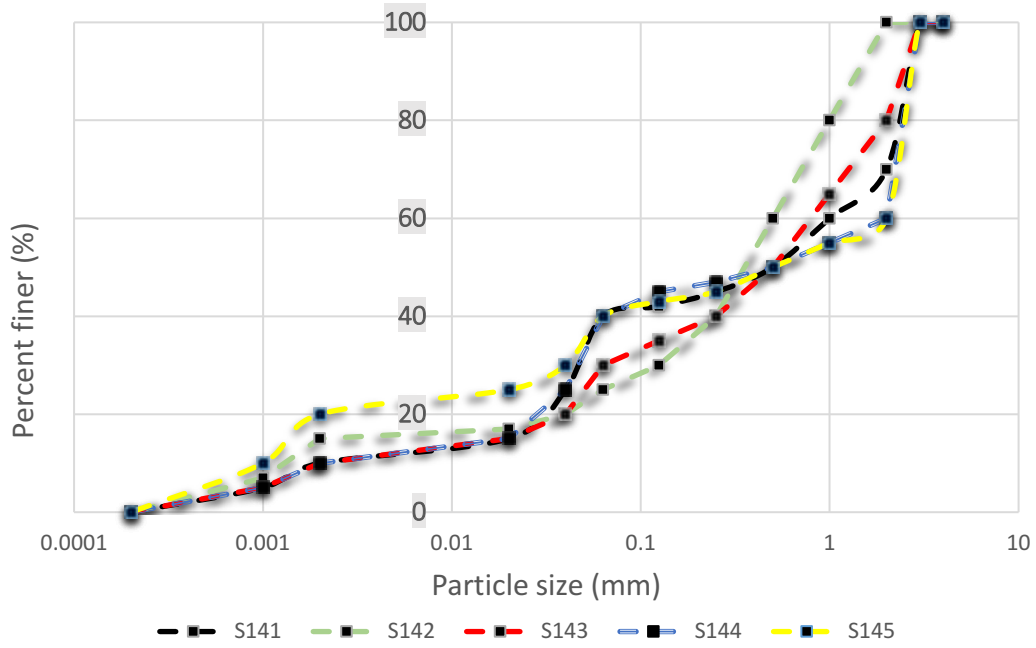


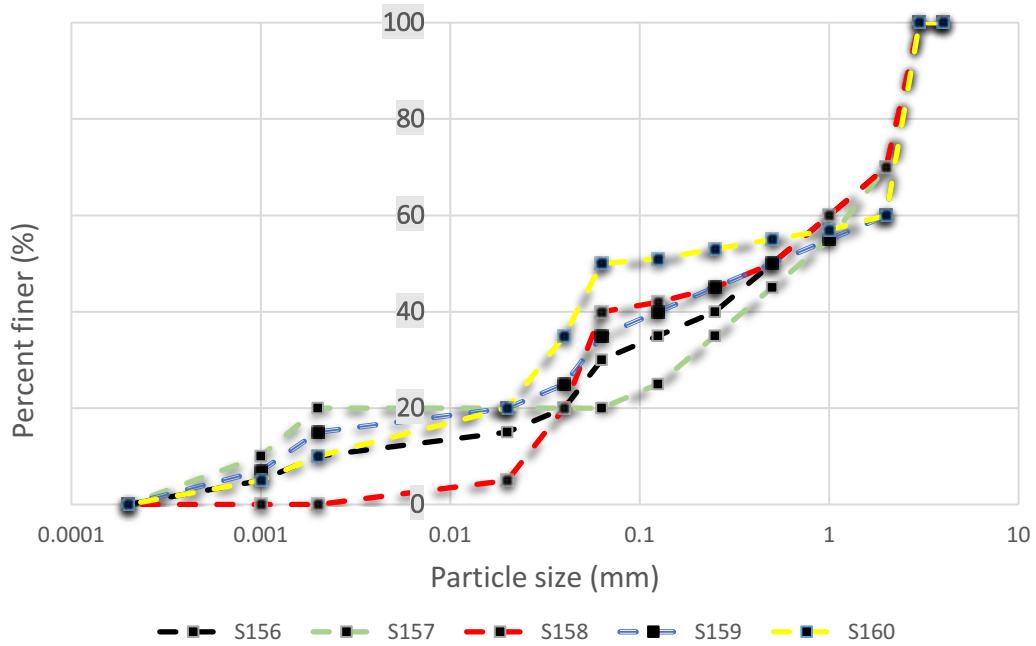
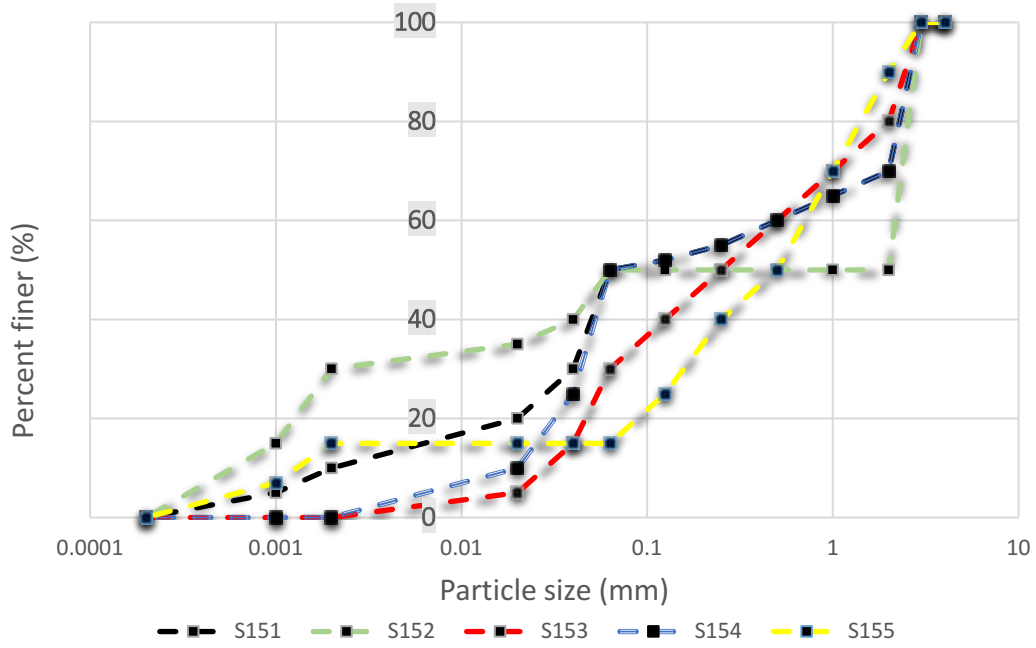


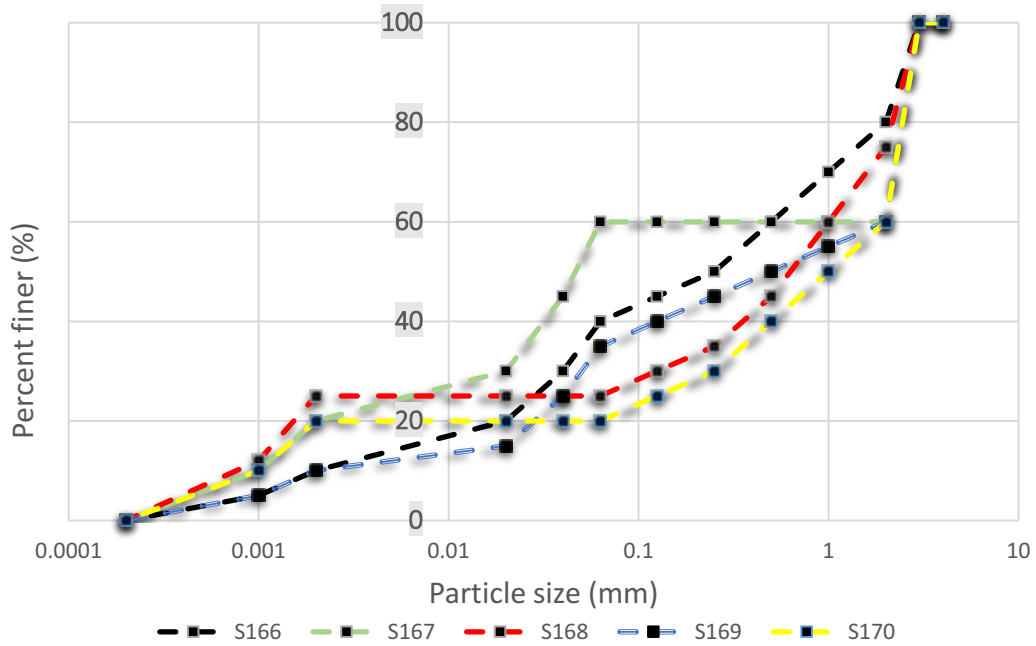
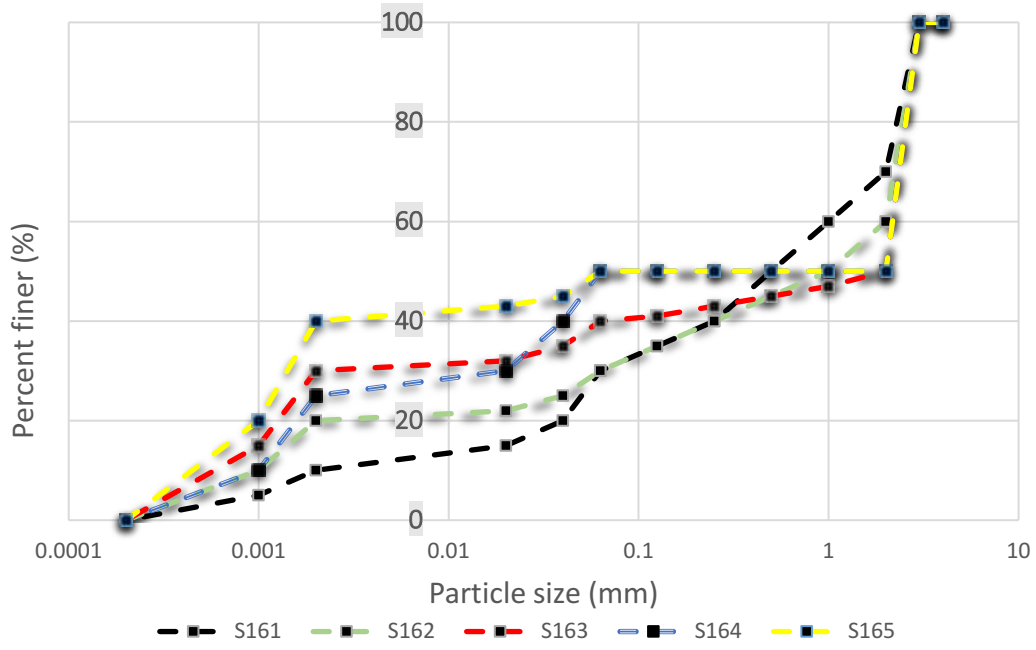


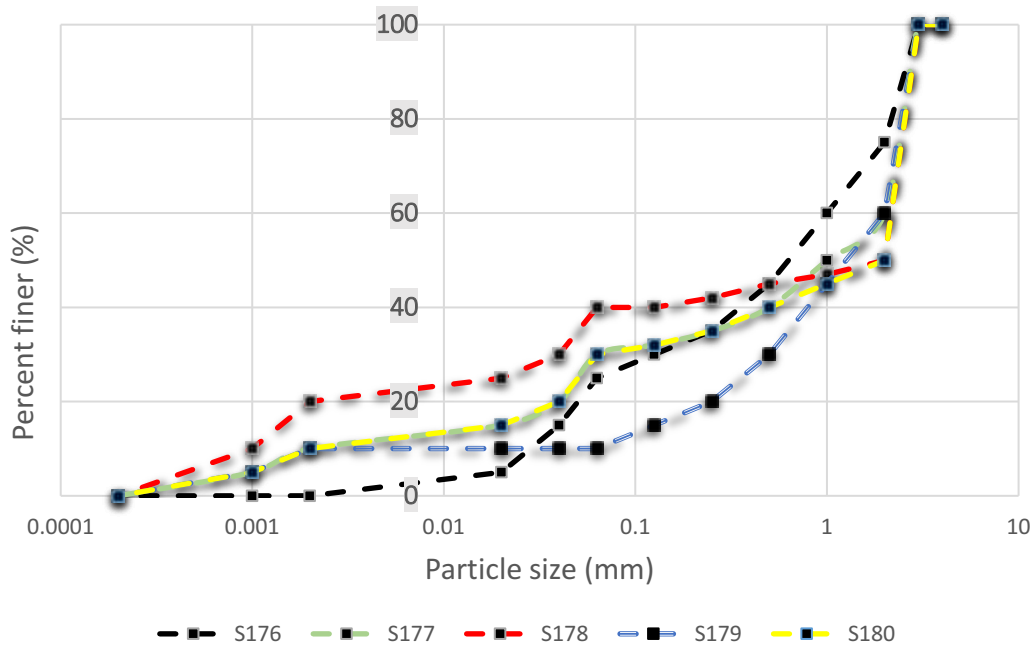
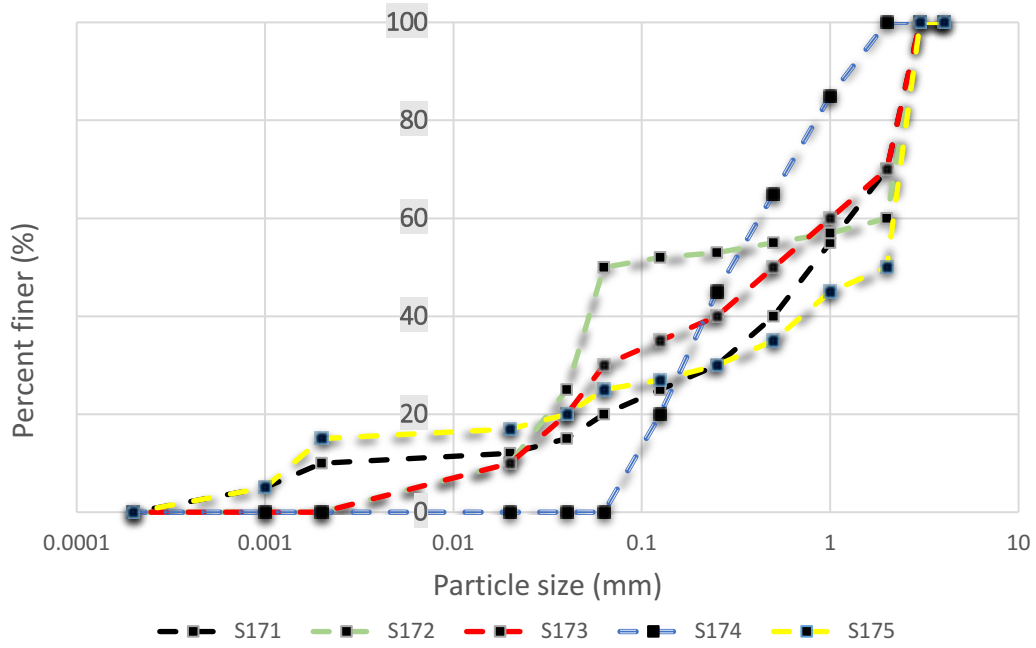


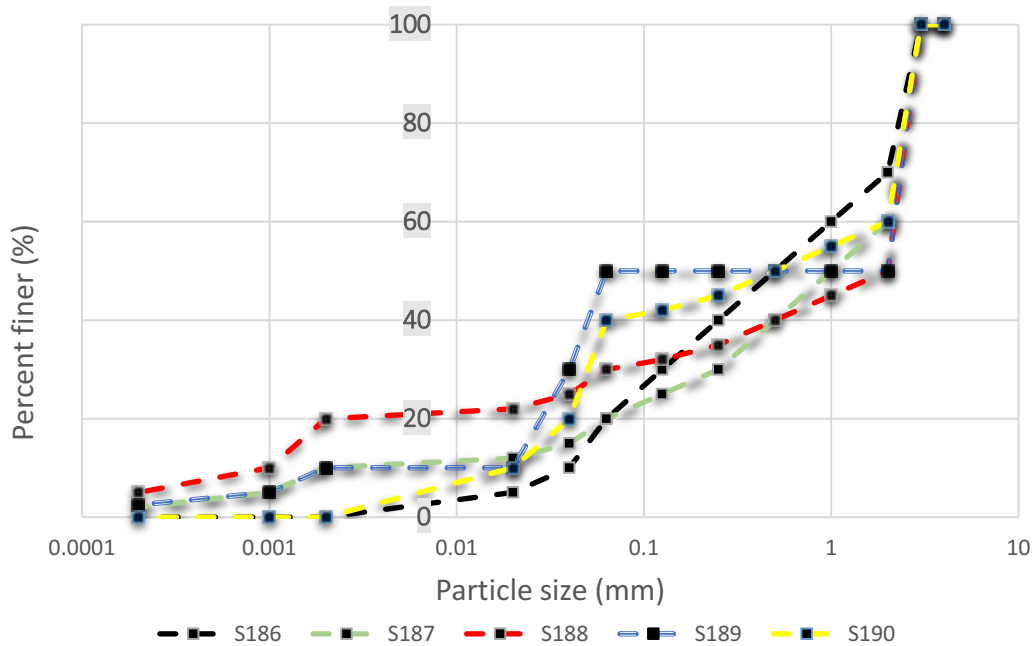
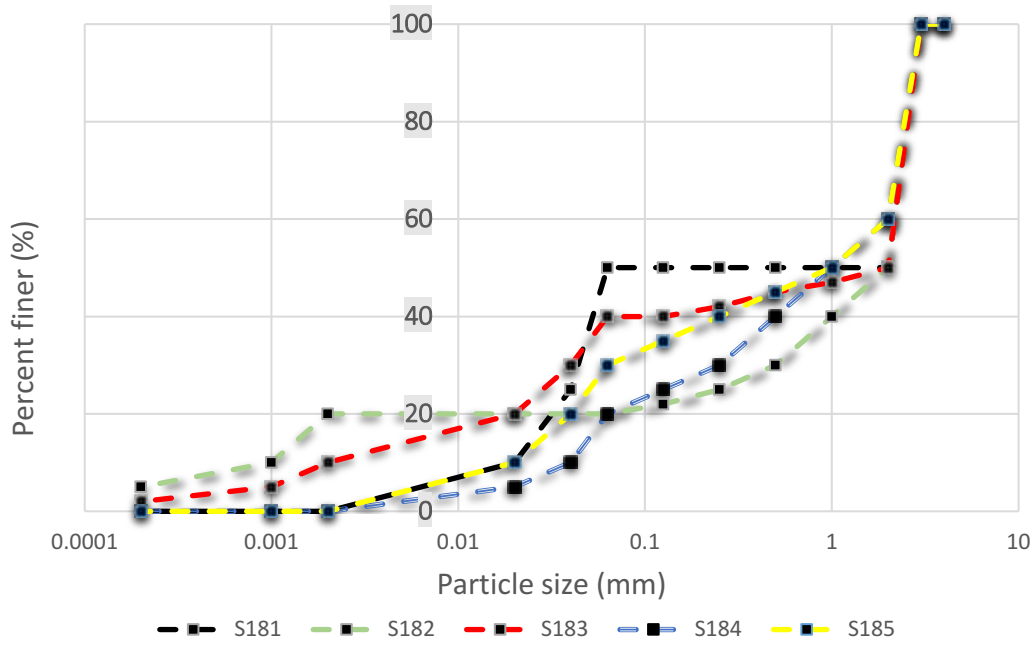


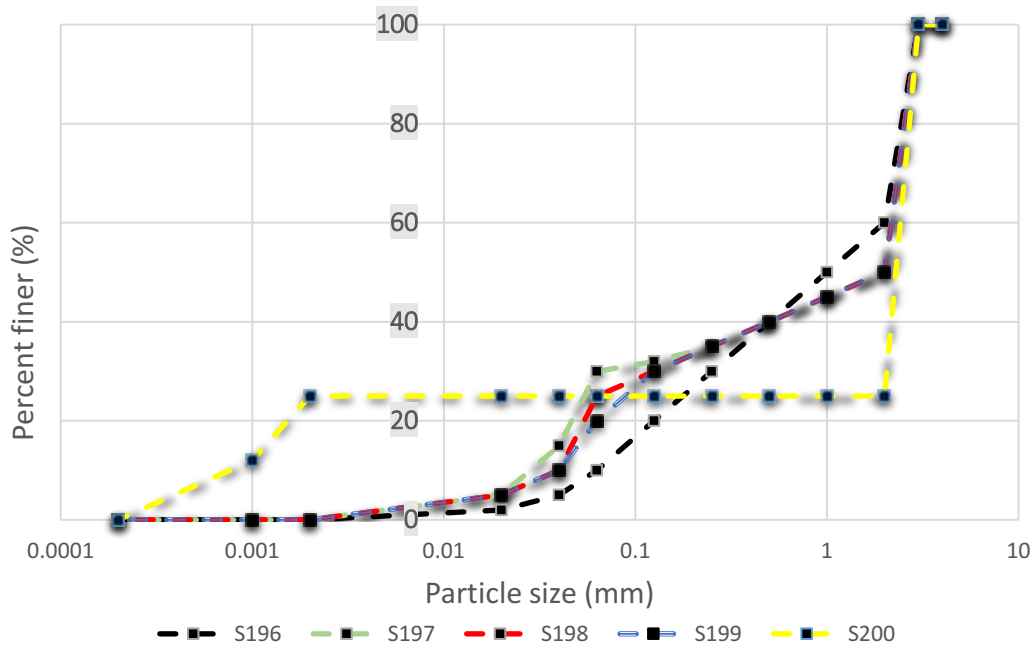
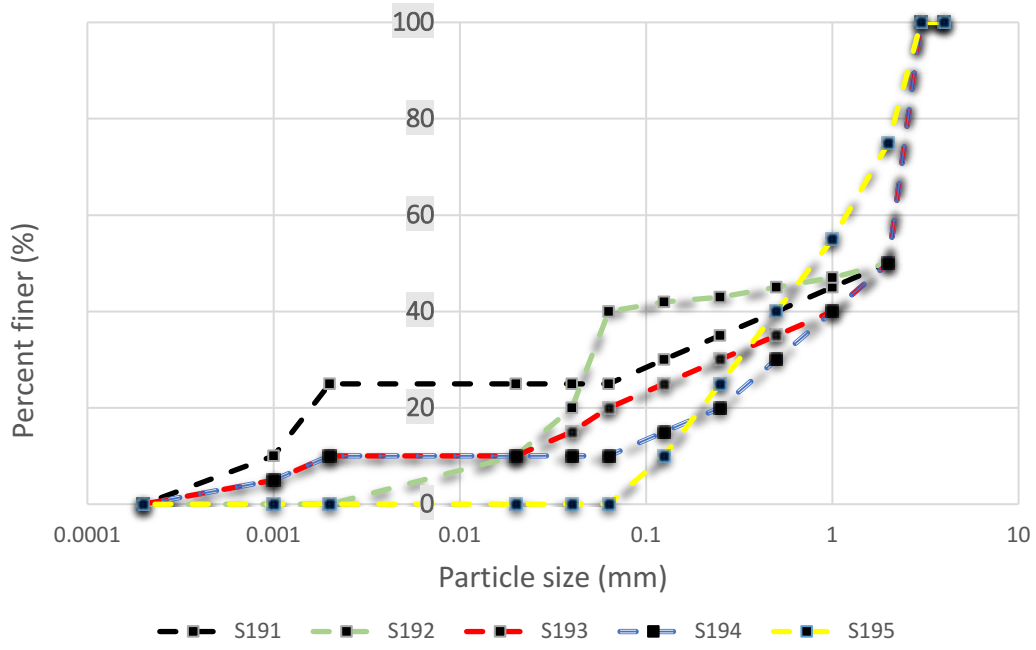


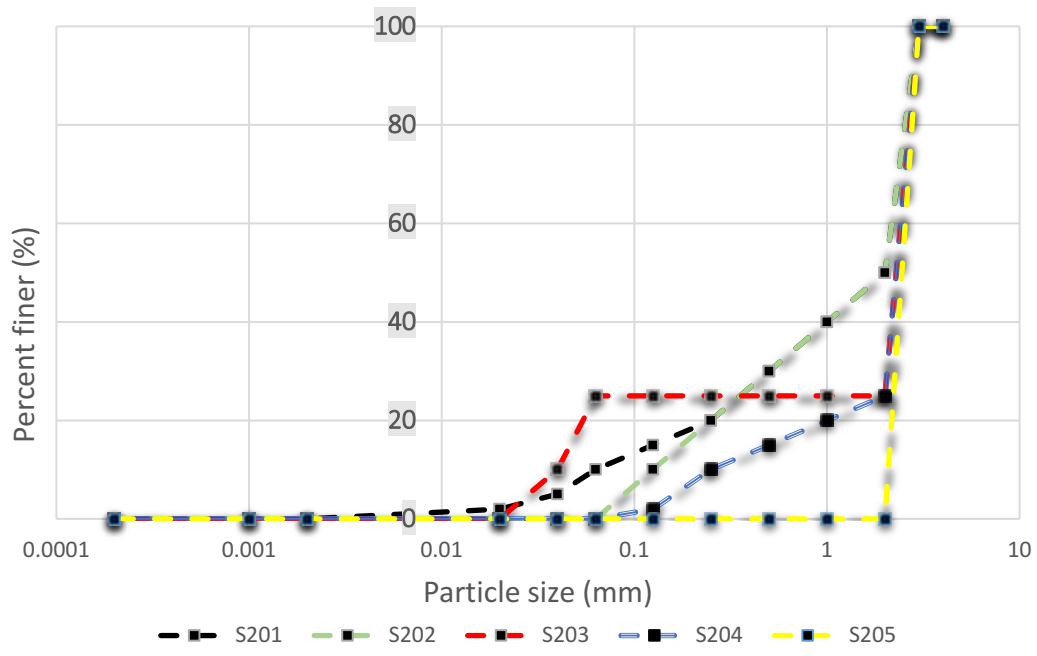




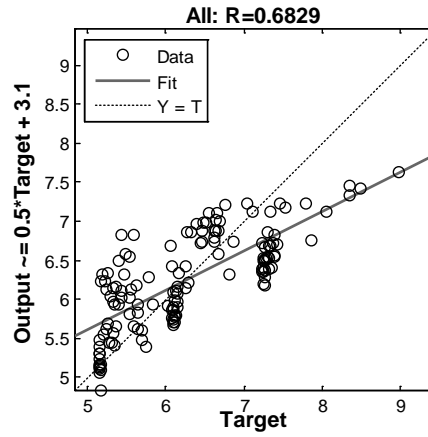
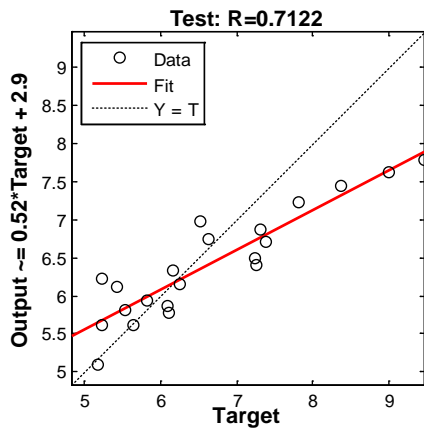
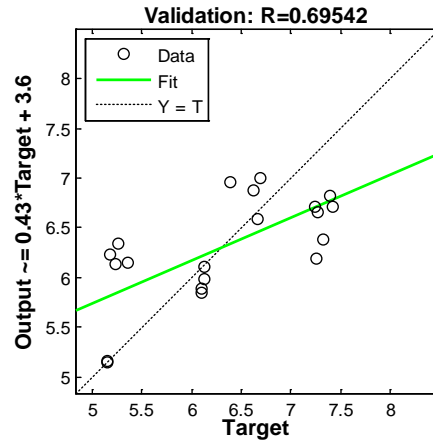
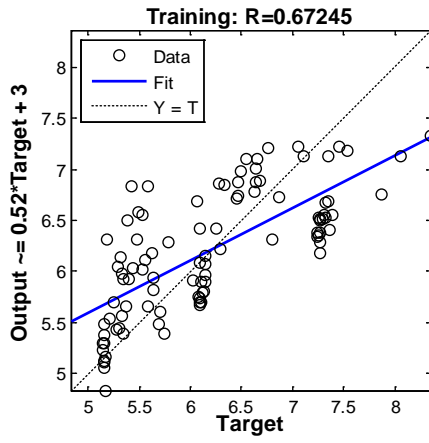




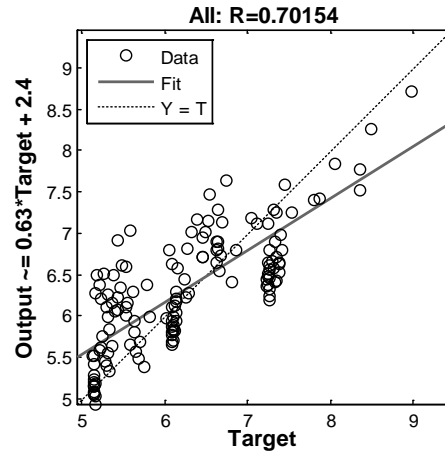
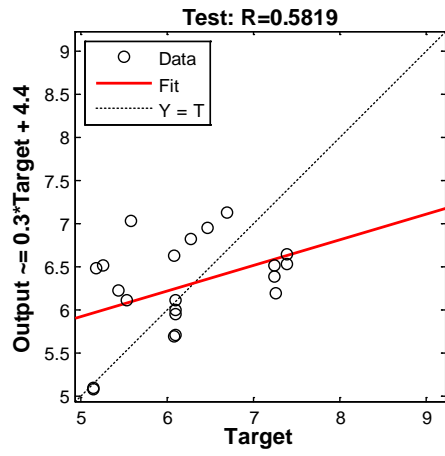
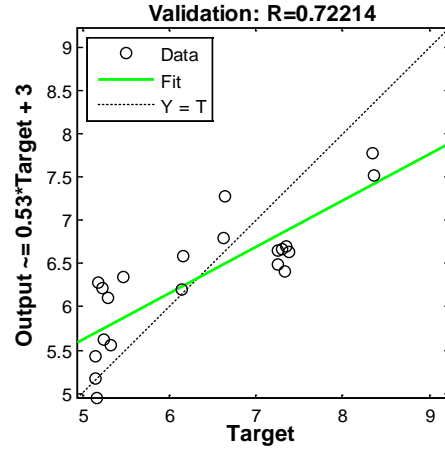
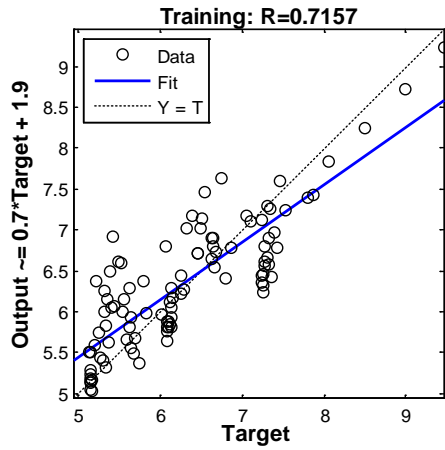




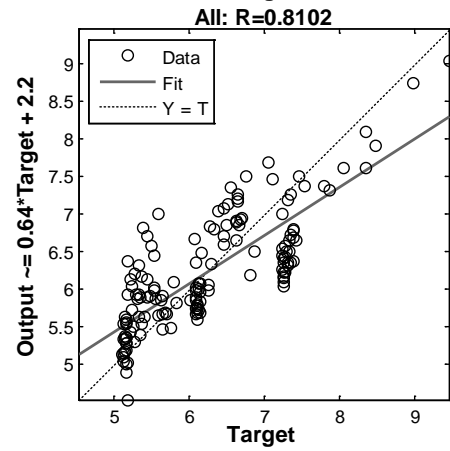
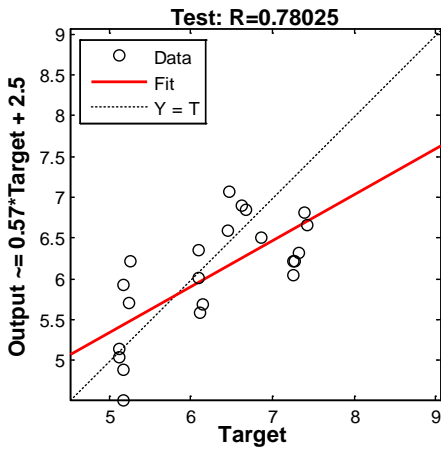
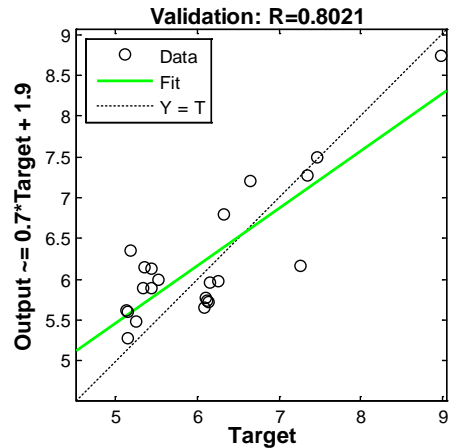
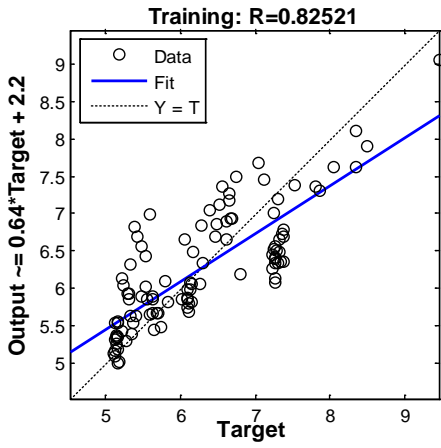
Annex 3. Regression analysis results for purpose of choosing best ANN model for K



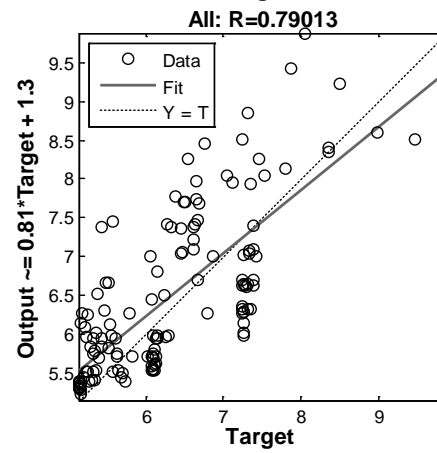
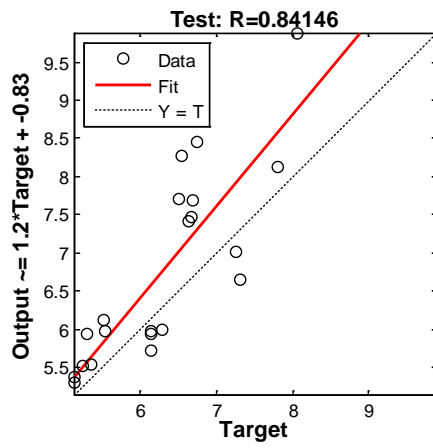
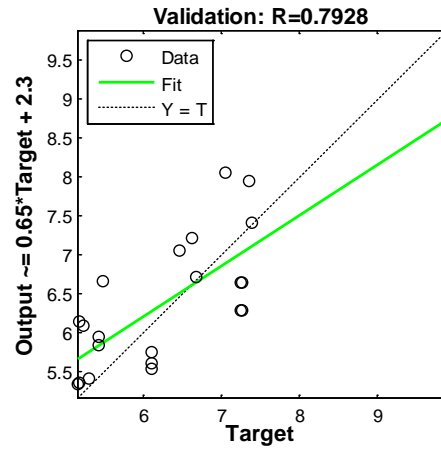
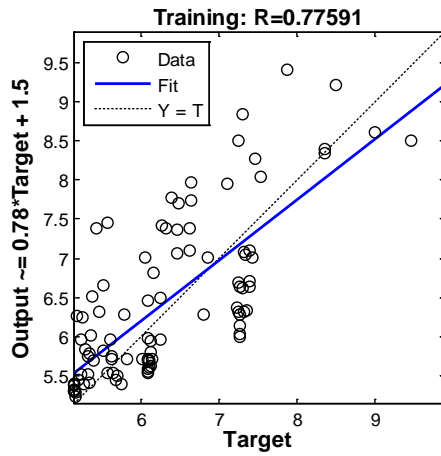
Model Number 1



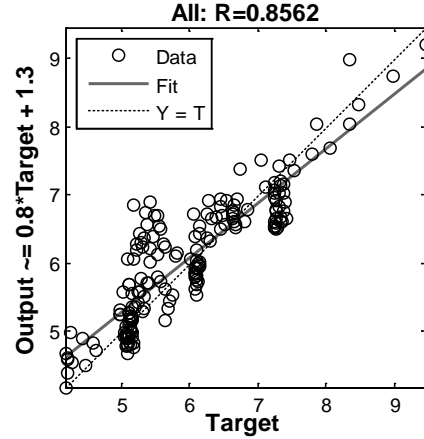
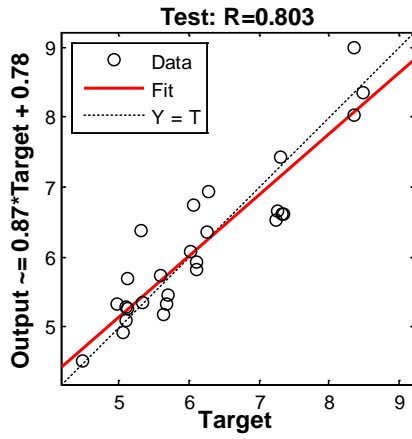
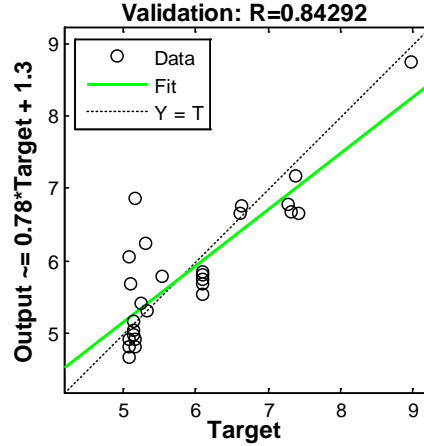
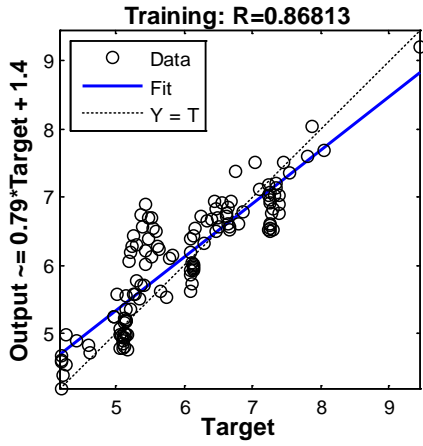
Model Number 2



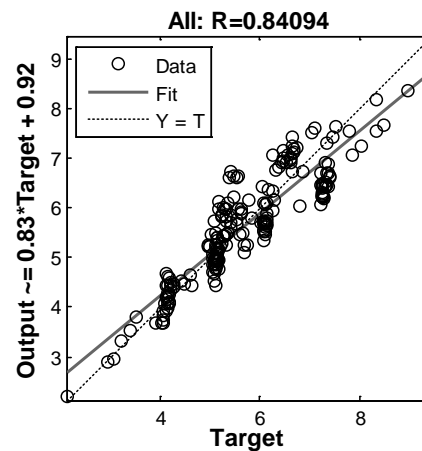
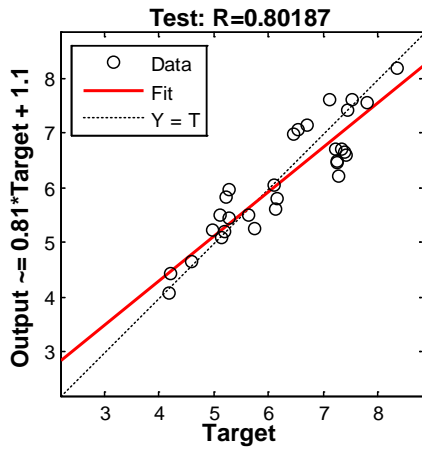
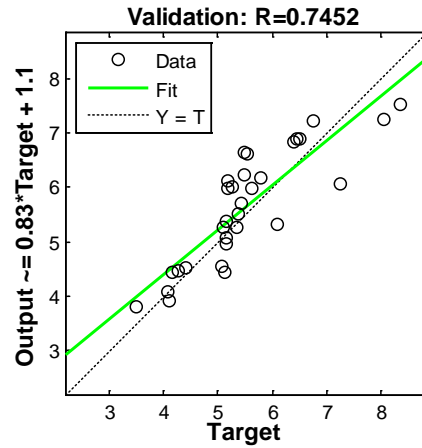
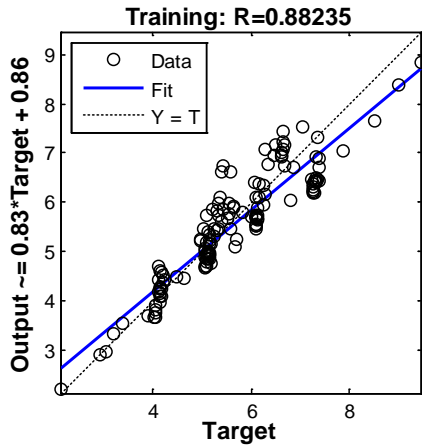
Model Number 3



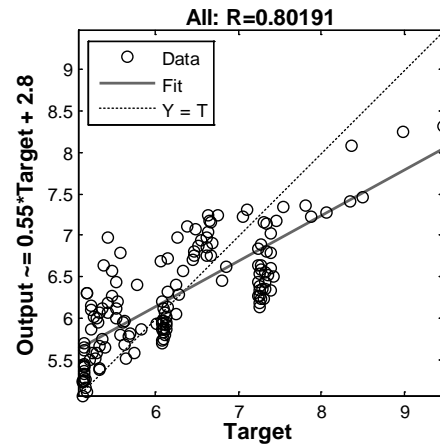
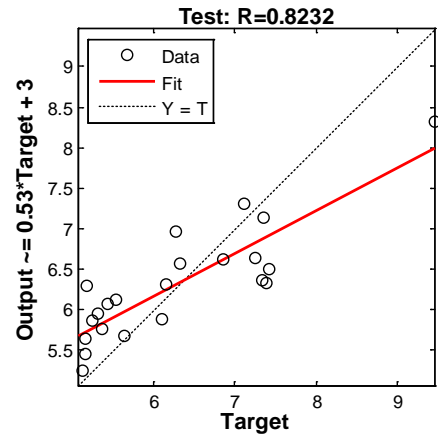
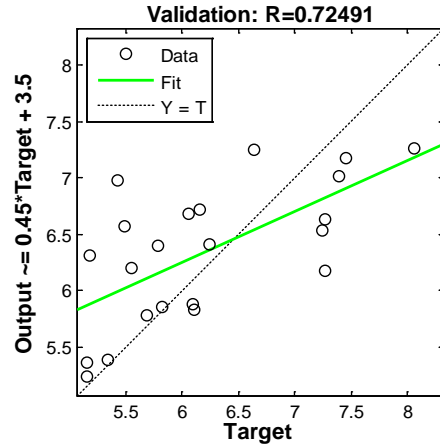
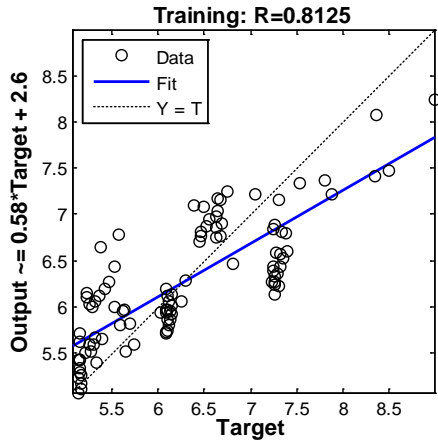
Model Number 5



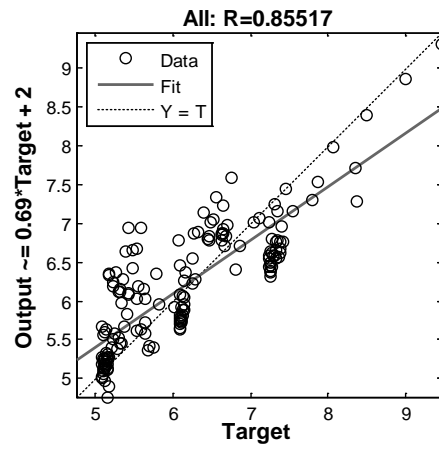
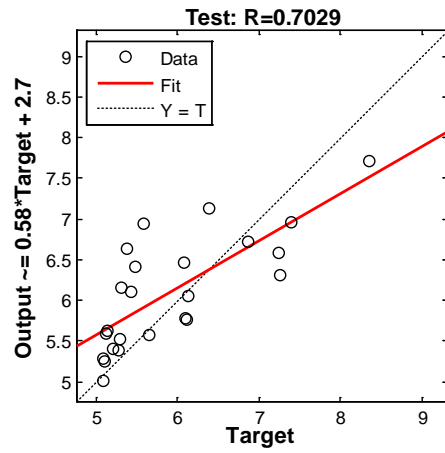
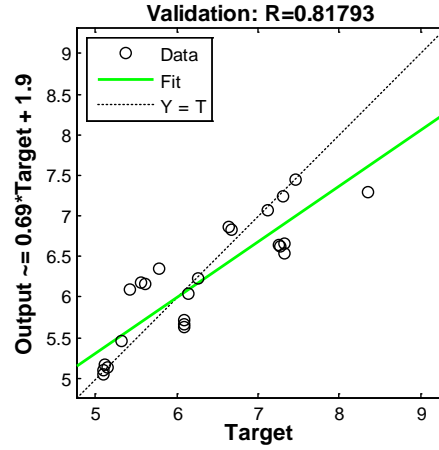
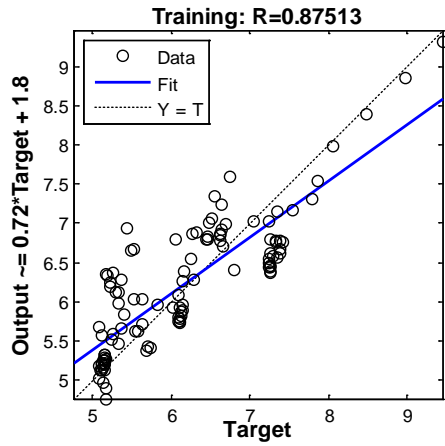
Model Number 6



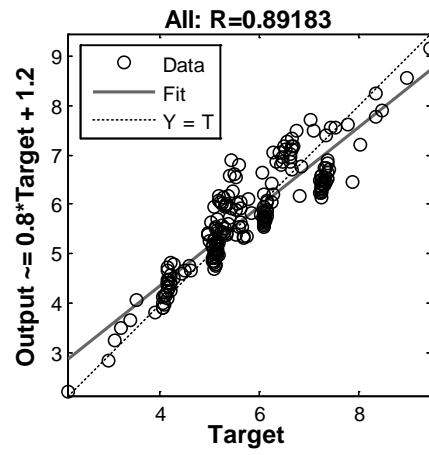
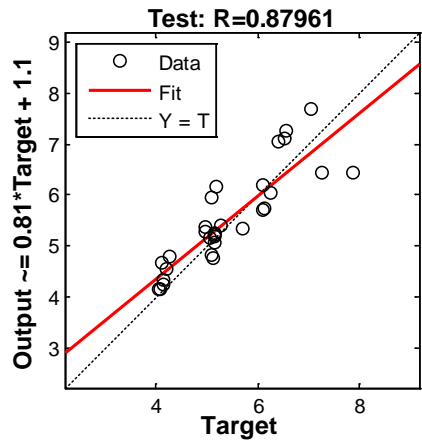
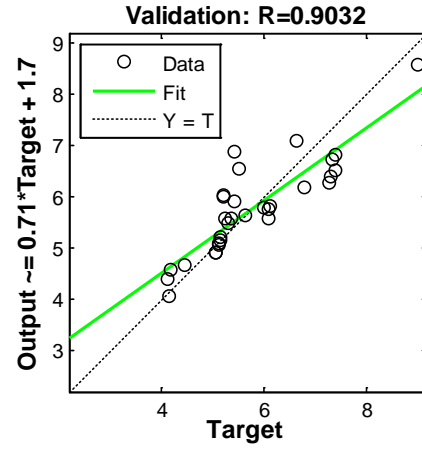
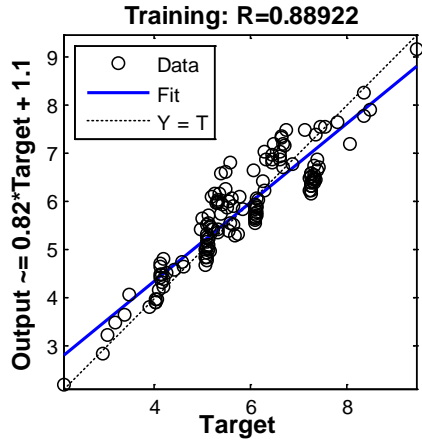
Model Number 7



Model Number 8



Model Number 9



Model Number 10

Annex 4: Source Codes for Artificial Neural Network Modeling

a) Code for hydraulic conductivity estimation

```
1 - close all
2 - clear all
3 - clc
4 - format shortE
5
6 - filename = 'data1.xlsx';
7 - sheet = 1;
8 - range = 'A2:F206';
9 - data = xlsread(filename, sheet, range);
10
11 - inputs = data(:, 1:4);
12 - targets = data(:, 5);
13
14 % Define the network structure
15 - net = feedforwardnet([1 3]);
16 - net.trainParam.showWindow = true;
17 - net.trainParam.epochs = 1000;
18 - net.trainParam.goal = 0.001;
19
20 % Train the network
21 - net = train(net, inputs', targets');
22
23 % Predict the outputs
24 - predicted_logK = net(inputs)';
25
26 % Calculate RMSE
27 - rmse = sqrt(mean((targets - predicted_logK).^2));
28 - disp(['RMSE: ', num2str(rmse)]);
29
30 % Calculate MAE
31 - mae = mean(abs(targets - predicted_logK));
32 - disp(['MAE: ', num2str(mae)]);
33
34 % Calculate MASE
35 - n = length(targets);
36 - mase = mae / mean(abs(diff(targets)));
37 - disp(['MASE: ', num2str(mase)]);
38
39 % Calculate R2
40 - ss_res = sum((targets - predicted_logK).^2);
41 - ss_tot = sum((targets - mean(targets)).^2);
42 - r2 = 1 - (ss_res / ss_tot);
43 - disp(['R2: ', num2str(r2)]);
44
```

```

45     %% Extracting Weights, Biases, and Activation Functions  %%
46     % (Unchanged from your original code)
47
48     % Display weights and biases
49     disp('Weights from input to first hidden layer:');
50     disp(net.IW{1,1});
51
52     disp('Biases of first hidden layer:');
53     disp(net.b{1});
54
55     disp('Weights from first hidden layer to second hidden layer:');
56     disp(net.LW{2,1});
57
58     disp('Biases of second hidden layer:');
59     disp(net.b{2});
60
61     disp('Weights from second hidden layer to output layer:');
62     disp(net.LW{3,2});
63
64     disp('Biases of output layer:');
65     disp(net.b{3});
66
67     % Display activation functions
68     disp('Activation function of first hidden layer:');
69     disp(net.layers{1}.transferFcn);
70
71     disp('Activation function of second hidden layer:');
72     disp(net.layers{2}.transferFcn);
73
74     disp('Activation function of output layer:');
75     disp(net.layers{3}.transferFcn);
76
77     %% Visualize the effect of each input parameter on output
78     clay_percent = inputs(:, 1);
79     silt_percent = inputs(:, 2);
80     sand_percent = inputs(:, 3);
81     gravel_percent = inputs(:, 4);
82
83     % Plot for clay percentage vs predicted -Log K
84     figure;
85     scatter(clay_percent, predicted_logK, 'b', 'filled');
86     xlabel('Clay Percentage (%)');
87     ylabel('Predicted -Log K');
88     title('Effect of Clay Percentage on Predicted -Log K');
89     grid on;

```

```

90
91     % Plot for silt percentage vs predicted -Log K
92     figure;
93     scatter(silt_percent, predicted_logK, 'r', 'filled');
94     xlabel('Silt Percentage (%)');
95     ylabel('Predicted -Log K');
96     title('Effect of Silt Percentage on Predicted -Log K');
97     grid on;
98
99     % Plot for sand percentage vs predicted -Log K
100    figure;
101    scatter(sand_percent, predicted_logK, 'g', 'filled');
102    xlabel('Sand Percentage (%)');
103    ylabel('Predicted -Log K');
104    title('Effect of Sand Percentage on Predicted -Log K');
105    grid on;
106
107    % Plot for gravel percentage vs predicted -Log K
108    figure;
109    scatter(gravel_percent, predicted_logK, 'm', 'filled');
110    xlabel('Gravel Percentage (%)');
111    ylabel('Predicted -Log K');
112    title('Effect of Gravel Percentage on Predicted -Log K');
113    grid on;
114
115    %% User input for prediction
116    data1 = input('Weight percentage of Clay: ');
117    data2 = input('Weight percentage of Silt: ');
118    data3 = input('Weight percentage of Sand: ');
119    data4 = input('Weight percentage of Gravel: ');
120
121    if (data1 + data2 + data3 + data4) == 100
122        disp('Percentage composition:');
123        Percentage_composition = [data1 data2 data3 data4];
124        outputs = abs(net(Percentage_composition));
125        disp('Estimation:');
126        disp(outputs);
127    else
128        disp('Error: The sum of percentages should be equal to 100')
129    end
130

```

b) Code for cohesion estimation (separate model)

```
1 - close all
2 - clear all
3 - clc
4 - format shortE
5
6 - filename = 'data-C and Fi.xlsx';
7 - sheet = 1;
8 - range = 'A2:H96';
9 - data = xlsread(filename, sheet, range);
10
11 - inputs = data(1:95, 1:4);
12 - targets = data(1:95, 5);
13
14 % Define the network structure
15 - net = feedforwardnet([3 1 4]);
16 - net.trainParam.showWindow = true;
17 - net.trainParam.epochs = 1000;
18 - net.trainParam.goal = 0.001;
19
20 % Train the network
21 - net = train(net, inputs', targets');
22
23 % Predict the outputs
24 - predicted_outputs = net(inputs)';
25
26 % Calculate RMSE (Root Mean Squared Error)
27 - rmse = sqrt(mean((targets - predicted_outputs).^2));
28 - disp(['RMSE: ', num2str(rmse)]);
29
30 % Calculate MSE (Mean Squared Error)
31 - mse = mean((targets - predicted_outputs).^2);
32 - disp(['MSE: ', num2str(mse)]);
33
34 % Calculate MAE (Mean Absolute Error)
35 - mae = mean(abs(targets - predicted_outputs));
36 - disp(['MAE: ', num2str(mae)]);
37
```



```

38  %% Extracting Weights, Biases, and Activation Functions %%
39
40  % Weights and Biases from input to first hidden layer
41  weights_input_to_hidden1 = net.IW(1,1);
42  biases_hidden1 = net.b(1);
43
44  % Weights and Biases from first hidden layer to second hidden layer
45  weights_hidden1_to_hidden2 = net.LW(2,1);
46  biases_hidden2 = net.b(2);
47
48  % Weights and Biases from second hidden layer to third hidden layer
49  weights_hidden2_to_hidden3 = net.LW(3,2);
50  biases_hidden3 = net.b(3);
51
52  % Weights and Biases from third hidden layer to output layer
53  weights_hidden3_to_output = net.LW(4,3);
54  biases_output = net.b(4);
55
56  % Display weights and biases
57  disp('Weights from input to first hidden layer:');
58  disp(weights_input_to_hidden1);
59
60  disp('Biases of first hidden layer:');
61  disp(biases_hidden1);
62
63  disp('Weights from first hidden layer to second hidden layer:');
64  disp(weights_hidden1_to_hidden2);
65
66  disp('Biases of second hidden layer:');
67  disp(biases_hidden2);
68
69  disp('Weights from second hidden layer to third hidden layer:');
70  disp(weights_hidden2_to_hidden3);
71
72  disp('Biases of third hidden layer:');
73  disp(biases_hidden3);
74
75  disp('Weights from third hidden layer to output layer:');
76  disp(weights_hidden3_to_output);
77
78  disp('Biases of output layer:');
79  disp(biases_output);
80
81  % Display activation functions
82  disp('Activation function of first hidden layer:');
83  disp(net.layers(1).transferFcn);
84
85  disp('Activation function of second hidden layer:');
86  disp(net.layers(2).transferFcn);
87
88  disp('Activation function of third hidden layer:');
89  disp(net.layers(3).transferFcn);
90
91  disp('Activation function of output layer:');
92  disp(net.layers(4).transferFcn);
93
94  % User input for percentage compositions
95  data1 = input('Weight percentage of Clay: ');
96  data2 = input('Weight percentage of Silt: ');
97  data3 = input('Weight percentage of Sand: ');
98  data4 = input('Weight percentage of Gravel: ');
99
100  if (data1 + data2 + data3 + data4) == 100
101      disp('Percentage composition:');
102      Percentage_composition = [data1 data2 data3 data4];
103      outputs = abs(net(Percentage_composition));
104      disp('Estimation:');
105      disp(outputs);
106  else
107      disp('Error: The sum of percentages should be equal to 100');
108  end
109

```


C) Code for internal friction angle estimation (separate model)

```
1 - close all
2 - clear all
3 - clc
4 - format shortE
5
6 - filename = 'data-C and Fi.xlsx';
7 - sheet = 1;
8 - range = 'A2:H96';
9 - data = xlsread(filename, sheet, range);
10
11 - inputs = data(1:95, 1:4);
12 - targets = data(1:95, 7);
13
14 % Define the network structure
15 - net = feedforwardnet([3 2]);
16 - net.trainParam.showWindow = true;
17 - net.trainParam.epochs = 1000;
18 - net.trainParam.goal = 0.001;
19
20 % Train the network
21 - net = train(net, inputs', targets');
22
23 % Predict the outputs
24 - predicted_outputs = net(inputs)';
25
26 % Calculate RMSE (Root Mean Squared Error)
27 - rmse = sqrt(mean((targets - predicted_outputs).^2));
28 - disp(['RMSE: ', num2str(rmse)]);
29
30 % Calculate MSE (Mean Squared Error)
31 - mse = mean((targets - predicted_outputs).^2);
32 - disp(['MSE: ', num2str(mse)]);
33
34 % Calculate MAE (Mean Absolute Error)
35 - mae = mean(abs(targets - predicted_outputs));
36 - disp(['MAE: ', num2str(mae)]);
37
```

```

38     %% Extracting Weights, Biases, and Activation Functions %%%
39
40     % Weights and biases from input to first hidden layer
41     weights_input_to_hidden1 = net.IW(1,1);
42     biases_hidden1 = net.b(1);
43
44     % Weights and biases from first hidden layer to second hidden layer
45     weights_hidden1_to_hidden2 = net.LW(2,1);
46     biases_hidden2 = net.b(2);
47
48     % Weights and biases from second hidden layer to output layer
49     weights_hidden2_to_output = net.LW(3,2);
50     biases_output = net.b(3);
51
52     % Display weights and biases
53     disp('Weights from input to first hidden layer:');
54     disp(weights_input_to_hidden1);
55
56     disp('Biases of first hidden layer:');
57     disp(biases_hidden1);
58
59     disp('Weights from first hidden layer to second hidden layer:');
60     disp(weights_hidden1_to_hidden2);
61
62     disp('Biases of second hidden layer:');
63     disp(biases_hidden2);
64
65     disp('Weights from second hidden layer to output layer:');
66     disp(weights_hidden2_to_output);
67
68     disp('Biases of output layer:');
69     disp(biases_output);
70
71     % Display activation functions
72     disp('Activation function of first hidden layer:');
73     disp(net.layers{1}.transferFcn);
74
75     disp('Activation function of second hidden layer:');
76     disp(net.layers{2}.transferFcn);
77
78     disp('Activation function of output layer:');
79     disp(net.layers{3}.transferFcn);
80
81     % User input for percentage compositions
82     data1 = input('Weight percentage of Clay: ');
83     data2 = input('Weight percentage of Silt: ');
84     data3 = input('Weight percentage of Sand: ');
85     data4 = input('Weight percentage of Gravel: ');
86
87     if (data1 + data2 + data3 + data4) == 100
88         disp('Percentage composition:');
89         Percentage_composition = [data1 data2 data3 data4];
90         outputs = abs(net(Percentage_composition));
91         disp('Estimation:');
92         disp(outputs);
93     else
94         disp('Error: The sum of percentages should be equal to 100');
95     end
96

```

D) Code for simultaneous Model to estimate cohesion and friction angle

```
1 % Load data from Excel file
2 data = readtable('data-C and F1.xlsx');
3
4 % Extract features and targets
5 X = table2array(data(:, 1:4)); % Clay, Silt, Sand, Gravel
6 y = table2array(data(:, 6:7)); % Cohesion, Friction Angle
7
8 % Define the architecture of the neural network
9 inputLayerSize = size(X, 2);
10 hiddenLayerSize = [5, 2, 4]; % Number of neurons in each hidden layer
11 outputLayerSize = size(y, 2);
12
13 net = feedforwardnet(hiddenLayerSize);
14
15 % Customize the network
16 for i = 1:length(hiddenLayerSize)
17     net.layers{i}.transferFcn = 'tansig'; % Hyperbolic tangent sigmoid function
18 end
19 net.layers(end).transferFcn = 'purelin'; % Linear for output layer
20
21 % Set up division of data for training, validation, testing
22 net.divideParam.trainRatio = 0.8;
23 net.divideParam.valRatio = 0.1;
24 net.divideParam.testRatio = 0.1;
25
26 % Train the network
27 [net, tr] = train(net, X', y');
28
29 % Extract weights and biases
30 weights_input_to_hidden1 = net.IW(1,1); % Weights from input to first hidden layer
31 biases_hidden1 = net.b(1); % Biases of first hidden layer
32
33 weights_hidden1_to_hidden2 = net.LW(2,1); % Weights from first hidden layer to second hidden layer
34 biases_hidden2 = net.b(2); % Biases of second hidden layer
35
36 weights_hidden2_to_hidden3 = net.LW(3,2); % Weights from second hidden layer to third hidden layer
37 biases_hidden3 = net.b(3); % Biases of third hidden layer
38
39 weights_hidden3_to_output = net.LW(4,3); % Weights from third hidden layer to output layer
40 biases_output = net.b(4); % Biases of output layer
41
42 % Display weights and biases
43 disp('Weights from input to first hidden layer:');
44 disp(weights_input_to_hidden1);
45
46 disp('Biases of first hidden layer:');
47 disp(biases_hidden1);
48
49 disp('Weights from first hidden layer to second hidden layer:');
50 disp(weights_hidden1_to_hidden2);
51
52 disp('Biases of second hidden layer:');
53 disp(biases_hidden2);
54
55 disp('Weights from second hidden layer to third hidden layer:');
56 disp(weights_hidden2_to_hidden3);
57
58 disp('Biases of third hidden layer:');
59 disp(biases_hidden3);
60
61 disp('Weights from third hidden layer to output layer:');
62 disp(weights_hidden3_to_output);
```

```

63
64 disp('Biases of output layer:');
65 disp(biases_output);
66
67 % Display activation functions
68 disp('Activation function of first hidden layer:');
69 disp(net.layers{1}.transferFcn);
70
71 disp('Activation function of second hidden layer:');
72 disp(net.layers{2}.transferFcn);
73
74 disp('Activation function of third hidden layer:');
75 disp(net.layers{3}.transferFcn);
76
77 disp('Activation function of output layer:');
78 disp(net.layers{4}.transferFcn);
79
80 % Evaluate the network on the entire dataset
81 y_pred_all = net(X');
82
83 % Transpose predictions to match the shape of y
84 y_pred_all = y_pred_all';
85
86 % Calculate (RMSE)
87 rmse_c = sqrt(mean((y(:, 1) - y_pred_all(:, 1)).^2));
88 rmse_phi = sqrt(mean((y(:, 2) - y_pred_all(:, 2)).^2));
89
90 % Calculate R squared
91 SS_tot_c = sum((y(:, 1) - mean(y(:, 1))).^2);
92 SS_res_c = sum((y(:, 1) - y_pred_all(:, 1)).^2);
93 R2_c = 1 - (SS_res_c / SS_tot_c);
94
95 SS_tot_phi = sum((y(:, 2) - mean(y(:, 2))).^2);
96 SS_res_phi = sum((y(:, 2) - y_pred_all(:, 2)).^2);
97 R2_phi = 1 - (SS_res_phi / SS_tot_phi);
98
99 % Display performance metrics
100 disp(['RMSE for Cohesion: ', num2str(rmse_c, '%.2f')]);
101 disp(['RMSE for Friction Angle: ', num2str(rmse_phi, '%.2f')]);
102 disp(['R^2 for Cohesion: ', num2str(R2_c, '%.2f')]);
103 disp(['R^2 for Friction Angle: ', num2str(R2_phi, '%.2f')]);
104
105 % Plotting the regression curves for all data
106 figure;
107 subplot(2, 1, 1);
108 scatter(y(:, 1), y_pred_all(:, 1));
109 hold on;
110

```



```

111 % Calculate and plot trendline for Cohesion
112 p_cohesion = polyfit(y(:, 1), y_pred_all(:, 1), 1); % Linear fit coefficients
113 y_fit_cohesion = polyval(p_cohesion, y(:, 1)); % Values of the trendline
114 plot(y(:, 1), y_fit_cohesion, 'r', 'LineWidth', 1.5); % Plot trendline
115 plot(y(:, 1), y(:, 1), 'k--'); % Perfect fit line
116 title('Actual vs Predicted for Cohesion');
117 xlabel('Actual Cohesion');
118 ylabel('Predicted Cohesion');
119 legend('Predictions', 'Trendline', 'Perfect Fit');
120
121 subplot(2, 1, 2);
122 scatter(y(:, 2), y_pred_all(:, 2));
123 hold on;
124
125 % Calculate and plot trendline for Friction Angle
126 p_friction = polyfit(y(:, 2), y_pred_all(:, 2), 1); % Linear fit coefficients
127 y_fit_friction = polyval(p_friction, y(:, 2)); % Values of the trendline
128 plot(y(:, 2), y_fit_friction, 'r', 'LineWidth', 1.5); % Plot trendline
129 plot(y(:, 2), y(:, 2), 'k--'); % Perfect fit line
130 title('Actual vs Predicted for Friction Angle');
131 xlabel('Actual Friction Angle');
132 ylabel('Predicted Friction Angle');
133 legend('Predictions', 'Trendline', 'Perfect Fit');
134
135 % Adjust figure properties
136 subplot('Actual vs Predicted Values with Trendlines');
137
138 % Prompt user for input
139 clay = input('Enter Clay percentage: ');
140 silt = input('Enter Silt percentage: ');
141 sand = input('Enter Sand percentage: ');
142 gravel = input('Enter Gravel percentage: ');
143
144 % Predict cohesion and friction angle
145 user_input = [clay, silt, sand, gravel];
146 user_output = net(user_input');
147
148 % Display the predictions
149 disp(['Estimated Cohesion: ', num2str(user_output(1), '%.2f')]);
150 disp(['Estimated Friction Angle: ', num2str(user_output(2), '%.2f')]);

```