

PREDICTION OF WATER RETENTION CURVES USING NEURAL
NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY



BY
MELEK TÜRKMEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

AUGUST 2020

Approval of the thesis:

PREDICTION OF WATER RETENTION CURVES USING NEURAL NETWORKS

submitted by **MELEK TRKMEN** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Ahmet Trer
Head of the Department, **Civil Engineering**

Assoc. Prof. Dr. N. Kartal Toker
Supervisor, **Civil Engineering, METU**

Assist. Prof. Dr. Onur Pekcan
Co-Supervisor, **Civil Engineering, METU**

Examining Committee Members:

Prof. Dr. Elçin Kentel Erdoğan
Civil Engineering, METU

Assoc. Prof. Dr. N. Kartal Toker
Civil Engineering, METU

Assoc. Prof. Dr. Nejan Huvaj Sarıhan
Civil Engineering, METU

Assist. Prof. Dr. Onur Pekcan
Civil Engineering, METU

Prof. Dr. Taha Taşkiran
Civil Engineering, Yildirim Beyazıt University

Date: 04.08.2020



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Melek Türkmen

Signature :

ABSTRACT

PREDICTION OF WATER RETENTION CURVES USING NEURAL NETWORKS

Türkmen, Melek
Master of Science, Civil Engineering
Supervisor: Assoc. Prof. Dr. N. Kartal Toker
Co-Supervisor: Assist. Prof. Dr. Onur Pekcan

August 2020, 115 pages

Soil water-retention curve (WRC) relates tension in soil water to soil suction. WRC information has pivotal importance for revealing the behavior of unsaturated soils. Methods for obtaining retention curves are either too expensive or time consuming. Instigated by the demand on fast predictions, this study expressed a composition of a 88 NN designs opened up with (i) hyperparameter tuning, (ii) reexamination of expressions of WRC and GSD. The data was extracted from UNSODA database which encompasses a broad type of soils and widely varied suction ranges, without excluding or subsampling any of the textural group or suction ranges of observations as most of the existing studies did. This inclusive approach rendered the originality of the study and yet spawned a series of problems in methodology and low accuracy in predictions. Among those models, Fredlund and Xing (1994) model held the highest accuracy measure, R^2 , which varied from 0.51 to 0.85.

Keywords: Unsaturated Soils, Water-Retention Curves, Neural Networks, Parameter Optimization.

ÖZ

SU TUTMA EĞRİLERİNİN YAPAY SİNİR AĞLARI TEKNİĞİ İLE TAHMINİ

Türkmen, Melek
Yüksek Lisans, İnşaat Mühendisliği
Tez Yöneticisi: Doç. Dr. N. Kartal Toker
Ortak Tez Yöneticisi: Dr. Öğr. Üy. Onur Pekcan

Ağustos 2020, 115 sayfa

Su tutma eğrisi (STE) su muhtevası ile zemin emmesini ilişkilendirir. STE bilgisi, suya doymun olmayan zeminlerin davranımını açıklamakta önemlidir. Bunun için geliştirilmiş çoğu metod ya zaman alıcı ya da pahalıdır. Hızlı tahminlere olan talebin teşvik ettiği bu çalışma, (i) hiperparametre optimizasyonu, (ii) su tutma ve dane büyüklüğü dağılımı temsillerinin yeniden değerlendirilmesi ile derinleştirilerek 88 yapay sinir ağı tasarımının bir kompozisyonunu sunar. Mevcut çalışmaların çoğunda gözlemlendiğinin aksine, tekstürel sınıfı veya emme aralığı göz önünde bulundurulmaksızın UNSODA veri tabanından elde edilen veriler kullanıldı. Bu kapsayıcı yaklaşım, çalışmayı özgün kılarken, metodolojide bir dizi problem ve tahminlerin doğruluğunun düşmesine sebebiyet verdi. Geliştirilen modeller arasında Fredlund ve Xing (1994) modeli, 0.51 ile 0.85 arasında değişen R^2 değerlerinin en yükseğini yakaladı.

Anahtar Kelimeler: Kelimeler: Suya Doymun Olmayan Zeminler, Su-Tutma Eğrileri, Yapay Sinir Ağları, Hiperparametre Optimizasyonu.



To my family

ACKNOWLEDGMENTS

I am thankful to anyone who has shared their support for me. First of all, I have been incredibly lucky to have my supervisor Assoc. Prof. Dr. Kartal Toker in my corner. To me, he has been an amazing teacher as he constantly shared his knowledge and inspiration. He has been a great supervisor for his constructive guidance, encouragement, his never-ending patience, clear feedbacks, and support in all respects that he has been providing me throughout the entire process.

I want to express my deepest gratitude to my co-advisor Assist. Prof. Dr. Onur Pekcan for his invaluable feedbacks, patience, and support through the process.

I gratefully acknowledge BAP (117M330) project for supporting this study.

I benefited the conversations with Metehan Yayla that helped me to clarify part of the problems with Chapter 4.

I am especially thankful to my dearest friends Eren Burak Tutuř, Püren řenyuva, Yunus Emre lmez, Berkan Sylemez, Mustafa Kangl for their helps and friendship. I am grateful to Glden Ersan and Atilla Eypoęlu for providing me comfort in their home in quarantine days during the COVID-19 pandemic. Emre Duman shared his room without hesitation in my time of need. I am especially, deeply grateful to Gamze Tanık who took care of me, and helped to me in the darkest times.

Finally, I am thankful to my family from the bottom of my heart for supporting me under any circumstances.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vi
ACKNOWLEDGMENTS	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	xii
LIST OF FIGURES	xv
LIST OF ABBREVIATIONS	xviii
CHAPTERS	
1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Scope of Work	2
1.3 Thesis Outline	3
2 LITERATURE REVIEW	5
2.1 Grain Size Distribution	5
2.2 Soil Suction	5
2.3 Water Retention Curve	6
2.4 Determination of Water Retention Curves	9
2.4.1 Laboratory Measurement	10
2.4.1 Estimation Methods	11
3 NEURAL NETWORKS	19
3.1 Preliminaries of Neural Networks	19
3.1.1 Neurons	19

3.1.2	Feedforward Neural Networks	21
3.1.3	Training of FFNNs	22
3.1.3.1	Backpropagation Algorithm.....	23
3.1.3.2	Principal Component Analysis.....	29
4	METHODOLOGY.....	33
4.1	Data Collection.....	33
4.2	Data Preparation	34
4.2.1	Criteria for Data Elimination	34
4.2.2	Input Variables	37
4.2.3	Target Variables	38
4.2.3.1	Effective Volumetric Water Content.....	40
4.2.3.2	Fitting Parameter Estimation	40
4.2.3.3	WRC Parameter Estimation.....	40
4.2.4	Data Scaling	46
4.3	Model Assessment	46
4.3.1	Network Creation	46
4.3.2	Network Initialization.....	51
4.3.3	Complexity Control	52
5	RESULTS.....	57
5.1	Model Comparison Criteria.....	57
5.2	Summary of Results.....	58
5.3	Comparisons for Individual Outputs	63
5.3.1	Residual Volumetric Water Content.....	63
5.3.2	Desaturation Rate	64

5.3.3	Inflection Point Location	66
5.3.4	Suctions at AEV, $\theta_e=0.2$, $\theta_e=0.5$, $\theta_e=0.8$	68
5.4	Group Outputs	71
5.5	Effect of Variable Transformation	75
5.6	Effect of Complexity	80
5.7	Effect of Dimensionality	85
5.8	Example Plots	88
6	CONCLUSION	97
6.1	Summary and Discussions	97
6.2	Recommendations for Future Work	99
REFERENCES		101
APPENDICES		
A.	NETWORK GENERATIVE MATLAB CODE	109

LIST OF TABLES

TABLES

Table 4-1 Range of suction – water content points according to textural groupings of UNSODA.	34
Table 4-2 Particle size ranges according to TS/EN-ISO 17892-4 (Craig, 2012, p.4)	37
Table 4-3 Descriptive statistics of predictors.	38
Table 4-4 Descriptive statistics for target variables.	39
Table 4-5 Descriptive statistics of FX fitting parameters (i) raw data and, after applying (ii) the statistical based method for skewed multivariate data, (iii) assemblage of univariate outliers.	44
Table 4-6 Number of nodes in single layered (top row) and two-layered (bottom row) networks.	54
Table 5-1 Model performances along with topology and number of observations of each set. Models used in this text are shown bold.	59
Table 5-2 Top 5 models according to NSE factor of test. PBIAS and NSE are for test sets.	62
Table 5-3 Top 5 models according to PBIAS of test set. PBIAS and NSE are for test sets.	62
Table 5-4 Metrics indicating the test errors of predictions of residual volumetric water content, θ_r	63
Table 5-5 MAEs and NSEs of the models targeting residual volumetric water content, θ_r , arranged in ascending order of mean absolute test error.	63
Table 5-6 Metrics indicating the test errors of slope of desaturation line (DSR) predictions.	65
Table 5-7 Number of hidden neurons and R^2 values for test sets of DSR models.	66
Table 5-8 Metrics indicating the test errors of inflection point (INF) predictions.	66

Table 5-9 Number of hidden neurons, sample size, and the test errors in terms of R^2 and NSE of inflection point (INF) predictions.	67
Table 5-10 Number of: nodes in each layer, training equations, weights and training data points for Model 49.	67
Table 5-11 Metrics indicating the test errors of air-entry value (AEV) predictions.	69
Table 5-12 Metrics indicating the test errors of predictions of suction value at effective volumetric water content of 0.2, $\psi(0.2)$	69
Table 5-13 Metrics indicating the test errors of predictions of suction value at effective volumetric water content of 0.5, $\psi(0.5)$	70
Table 5-14 Metrics indicating the test errors of predictions of suction value at effective volumetric water content of 0.8, $\psi(0.8)$	70
Table 5-15 Metrics indicating the test errors of predictions of suction values at effective volumetric water content of 0.2, 0.5, and 0.8 ($\psi(0.2;0.5;0.8)$).	72
Table 5-16 Metrics indicating the test errors of air-entry value, slope of desaturation line, and residual volumetric water content group (AEV-DSR- θ_r) predictions.	72
Table 5-17 Metrics indicating the test errors of air-entry value, slope of desaturation line, and volumetric water content at inflection point (AEV-DSR-INF) predictions.	73
Table 5-18 Metrics indicating the test errors of predictions of Fredlund and Xing (1994) equation (FX) parameters, a, m, n, θ_r	73
Table 5-19 Relation between the water contents estimated from test set of Fredlund and Xing (1994) fitting parameters and the experimental data, and MADs.	74
Table 5-20 The best models from 11 target variable ranked according to R^2 results.	74
Table 5-21 Applied transformations on fitting parameters of Fredlund and Xing (1994) model (either on a , or a, m, n , and θ_r all together), corresponding	

correlation values, R, between outputs and targets of train, validation, and test sets,
and coefficient of determination of test set. 76

Table 5-22 Differentiated topologies over Model 86 and corresponding R^2 values.
First row belongs to Model 86.....80

Table 5-23 Eigenvalues and explained variances by each principal component. ...85



LIST OF FIGURES

FIGURES

Figure 2.1. Characteristic parameters defining retention curves and zones of desaturation (Fredlund and Rahardjo, 2012, p. 6).....	7
Figure 2.2. Zones of desaturation (Sillers et al., 2001).....	9
Figure 2.3. Categorization of the proposed models based on suction ranges (Fredlund et al., 2011).....	13
Figure 3.1. A neuron is a function of its parameters and input features	20
Figure 3.2. (a) Sigmoid function, and, (b) hyperbolic tangent function (Orr and Müller, 1998).....	21
Figure 3.3. A feedforward network with a single hidden layer.....	22
Figure 3.4. Notation for BP algorithm	24
Figure 3.5. Bias unit with a fixed output of 1	25
Figure 3.6. Gradient descent for mean square error error function (Priddy and Keller, 2005).....	28
Figure 4.1. Textural class distribution of 575 soil samples in UNSODA, according to TS/EN-ISO 17892-4 classification system.	33
Figure 4.2. Data points highlighted in red box obeyed correction criteria.	35
Figure 4.3. Soil samples from (a) through (d) obeyed elimination criteria.	36
Figure 4.4. Water retention curve of sample 2250 before and after correction.	36
Figure 4.5. Constructed horizontal line (1) and the line passing through the first two points (2) on log-transformed suction – water content curve of sample 1043: sand soil from Live Oak, FL.	42
Figure 4.6. Red (dashed) line represents the fitted polynomial to maximum 5 suction points. Blue (dash-dot) line denotes the zero slope line of the red one, on log-transformed suction – water content curve of sample 2110: loamy sand soil from Lillington, NC.	43
Figure 4.7. Input-target matching process.	45
Figure 4.8. Each created layer is connected to its subsequent layer.....	47

Figure 4.9. Network is configured to 5 input and 4 output features.	47
Figure 4.10. Workflow for network generative code plugged in Appendices.	50
Figure 4.11. The first strategy followed in network generation process. Each run calls for a unique topology.	53
Figure 4.12. Execution process	55
Figure 5.1. The resulting regression plots without standardization.....	77
Figure 5.2. The resulting regression plots when targets were subjected only to standardization.....	77
Figure 5.3. The resulting regression plots when log-translation was applied on parameter a , only.....	78
Figure 5.4. The resulting regression plots when all the target features were log- transformed.....	78
Figure 5.5. The resulting regression plots when parameter a was Box-Cox transformed.....	79
Figure 5.6. The resulting regression plots when all the targets were Box-Cox transformed.....	79
Figure 5.7. Regression plot for Model 86.	81
Figure 5.8. Regression plot for Model 86.1.	81
Figure 5.9. Regression plot for Model 86.3.	82
Figure 5.10. Regression plot for Model 86.4.	82
Figure 5.11. Training state of Model 86.1.	83
Figure 5.12. Performance of Model 86.1.....	84
Figure 5.13. Performance of Model 86.....	84
Figure 5.14. Diagnostic plot for two principal components.....	86
Figure 5.15. Resulting regression plots after robust PCA analysis.	87
Figure 5.16. Comparison of estimated and measured WRCs of Sample 10709.	88
Figure 5.17. Comparison of estimated and measured WRCs of Sample 10884.	89
Figure 5.18. Comparison of estimated and measured WRCs of Sample 10888.	89
Figure 5.19. Comparison of estimated and measured WRCs of Sample 10890.	90
Figure 5.20. Comparison of estimated and measured WRCs of Sample 11175.	90

Figure 5.21. Comparison of estimated and measured WRCs of Sample 11274.	91
Figure 5.22. Comparison of estimated and measured WRCs of Sample 11550.	91
Figure 5.23. Comparison of estimated and measured WRCs of Sample 11553.	92
Figure 5.24. Comparison of estimated and measured WRCs of Sample 11554.	92
Figure 5.25. Comparison of estimated and measured WRCs of Sample 11580.	93
Figure 5.26. Comparison of estimated and measured WRCs of Sample 11587.	
Model 70 predicted the shape of the curve with a downward shift.	93
Figure 5.27. Comparison of estimated and measured WRCs of Sample 11605.	94
Figure 5.28. Comparison of estimated and measured WRCs of Sample 11606.	94
Figure 5.29. Comparison of estimated and measured WRCs of Sample 11607.	95

LIST OF ABBREVIATIONS

AEV: Air-entry value

NN: Neural Networks

PTF: Pede-transfer Functions

UNSODA: Unsaturated Soil Database



CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Retention curves describe the relation between tension in soil water and corresponding amount of water under equilibrium conditions. This relation becomes a fundamental notion for unsaturated soil mechanics as WRC parameters can be plugged into other equations for assessment of the pillars of physical processes in vadose zone, e.g., water flow and storage (Richards, 1931; Qian and Rahardjo, 2016), shear strength (Vanapalli et al., 1996) and volume change of unsaturated soils (Rahardjo et al., 2001), as well as for agricultural applications, e.g., estimates of WRC are required for crop water demand estimation in order to develop sustainable water saving strategy designs (Karandish and Šimůnek, 2016). Yet, in spite of its pivotal importance, determination of WRC still stays costly and/or time-consuming. Conventional methods to obtain WRC cannot capture the whole suction range, hence, needs to switch techniques and requires time. Estimation methods are either computationally intensive and requires experienced labor or have their own limitations (Fredlund and Rahardjo, 2012). These make it problematic to implement unsaturated soils theory in standard practice.

On the other hand, Neural Networks (NNs) as a chain of numerical computations which can be considered as an extensive regression tool, have the advantage over their counterparts for their rapid execution and the ability to adapt to almost any multivariate data of complex nonlinear systems. Due to their mapping potential, NNs have gained a general agreement in geotechnical practice and are commonly used in prediction of WRC, as well (Haghverdi et al., 2012; Pachepsky and Timlin, 1996; Schaap et al. 1998; Koekkoek and Booltink, 1999). This study contributes to existing

methodologies by embodying a large database without excluding or subdividing any data according to soil texture nor suction range.

1.2 Scope of Work

This study centers on fast and accurate determination of WRC, and delves on the backpropagation based neural networks: investigating, developing, and analyzing the repetitive design processes, and encoding and displaying the design parameter – prediction accuracy relationships. The intent is to explore potential factors influencing model performance and to provide further refinement in prediction accuracy by aligning the hyperparameters and reconsidering the descriptions of WRC.

In this regard, the NN design practice in this study is notably accumulated on differentiation of the generated models in terms of:

- Model topology: developing multiple alternative NN topologies to find the simplest model for which the complexity of the problem and the solution matches, so the NN model is able to learn without loss in generalization capability.
- Hyperparameters: varying the modes of a fixed complexity in terms of optimization algorithm, stopping criteria, and learning rate.
- Data processing: making sense of measured data for NN to interpret, by deriving descriptive features of WRC to form a number of input – target configurations for a better understanding of the descriptive qualities of different predictors. Shape of retention curve of a soil is controlled by its pore size distribution in relatively lower suctions and by textural type of the soil in relatively higher suctions (Jain et al., 2004), which are directly linked to porosity and grain size distribution (GSD) of the soil (Hwang and Choi, 2006). Thus, it is of value to extract the predictors from GSD data and porosity.

In line with the intent, a total of 575 soils with textural classifications ranging from sands to clays and laboratory measured drying retention data of Unsaturated Soil Database (UNSODA) is adopted. 88 models are generated and simultaneously trained with varying parameters which are stated above for the optimal tradeoff between performance and model capacity. Necessity of data processing and outlier elimination is revealed as a by-product through the process. Finally, model performances are reported and ranked according to the Mean Squared Error (MSE), coefficient of determination (R^2), Percent Bias (PBIAS), and Nash and Sutcliffe Efficiency (NSE) factor based evaluation.

In this way, pre-developed studies are desired to be extended by (i) approaching WRC in multiple scenarios both for point- and parametric-WRC representations and (ii) hyperparameter tuning in NN design stage.

1.3 Thesis Outline

The next chapter presents brief information on WRC, its determination techniques, and ends with a comprehensive review on existing studies. Chapter 3 provides a background for neural networks and the adopted techniques. Chapter 4 addresses the workflow to synthesize the two concepts, design tasks for NN model development in a step-by-step manner along with data processing. Chapter 5 summarizes the major findings of this study. A brief discussion of unmet performance ratings is presented in Chapter 6 which also concludes the thesis. Network generation algorithm is placed in Appendices.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a brief overview of water retention curves.

2.1 Grain Size Distribution

A detailed representation of GSD from which the pore size distribution can be correlated, provides a reliable basis for WRC prediction (Fredlund and Rahardjo, 2012, p. 40). Thus, representations of grain size distribution data form the primary argument of many WRC estimation techniques (Fredlund et al. 1997; Fredlund et al. 2002). Further details about the studies conducted on this relation can be found in Fredlund and Rahardjo (2012, p. 255).

2.2 Soil Suction

Soil suction (ψ) is defined as “energy per unit volume” (Klute, 1986). As such, it is a pressure, consisting two components: matric suction and osmotic suction, as follows:

$$\psi = (u_a - u_w) + \pi \quad (2.1)$$

where:

u_a : atmospheric pore-air pressure

u_w : pore-water pressure

π : osmotic suction.

Capillary effects along with the adsorptive forces (aroused from electrical and van der Waals fields) constitutes matric suction. Matric suction is the essential component of soil suction that directly affects the mechanical behavior and soil properties.

Osmotic suction is related to ion concentration. It affects the ability to retain water, but does not do much to its mechanical behavior.

As the amount of soil water changes, the fluid phase in the soil displaces (recede or advance) according to the pressure gradient that arose from suction change added to the gravitational force. This whole (drying and wetting) process is explained by retention curves.

2.3 Water Retention Curve

The water retention curve is the variation of the amount of soil water versus applied suction. As shown in Figure 2.1, a physical description of a typical monotonic drying retention curve can be made by two transitions : air-entry value (AEV), and residual conditions, i.e., residual water content (θ_r), and corresponding suction value (ψ_r). Desaturation rate (DSR), and saturated water content (θ_s) are the additional descriptors.

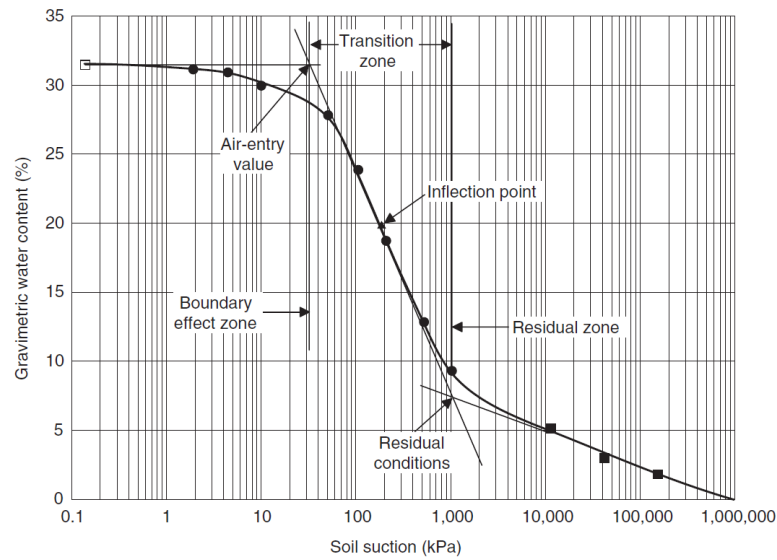


Figure 2.1. Characteristic parameters defining retention curves and zones of desaturation (Fredlund and Rahardjo, 2012, p. 6)

The characteristic parameters of the retention curve are directly affected by the physical properties of the soil. Air-entry value represents the average pore and grain sizes, while larger particles produce larger interparticle voids than the smaller ones (Arya and Paris, 1981). In addition, as the breadth of particle size distribution gets wider, pores get smaller due to effective packing; AEV gets higher. As moved from uniform sand to clay, the decrease in grain size and increase in gradation results in lateral shift of WRC. The uniformity of the soil designates flow path, so it provides a sense of the rate of water removal from the soil beyond AEV, that is, desaturation rate. As gradation gets poorer, the steepness of the retention curve in desaturation zone gets higher. Yet, no medium has perfectly packed particles. Even if every soil particle is the same, the pores will not be similar. Therefore, the drop in the retention curve cannot be completely steep. On the other hand, the amount and type of clay minerals predominate the amount of water adsorbed in the soil. Hence, θ_r increases as fines content increases. Additionally, the initial density (void ratio) defines the amount of saturated water content, θ_s . On the whole, grain and pore structure and distribution have a major importance on WRC.

The vertical axis of the retention curves may be written in terms of altered designations, i.e., gravimetric water content, volumetric water content, degree of saturation, normalized or dimensionless volumetric water content depending on the problem at hand. For example, the use of WRC in degree of saturation form would not be suitable for water storage problems as it results in discontinuities due to non-smoothness around AEV; however, it would provide convenience in computing unsaturated hydraulic conductivity. On the horizontal axis of the retention curve, soil suction is plotted on a logarithmic scale. As represented in Figure 2.2, soil suction ranges from a fraction of 1 kPa to 10^6 kPa, that is, the highest suction value that a soil can attain (Fredlund and Xing, 1994).

The amount of soil water determines the importance of each component of total suction relative to each other. At relatively low suction values, capillarity contributes more to soil suction; signifying that matric suction predominates in the lower suction values, whereas at relatively high suction values adsorptive forces become the dominant mechanism; implying that total suction is of interest in higher range of suction. In this context, Fredlund and Pham (2006) approached WRC as three zones:

- i. Low suction zone: (marked as capillary saturation zone in Figure 2.2) ranges from a fraction of 1 kPa to AEV. In low suction zone, soil remains saturated due to capillary forces; amount of water content is assumed constant up to a certain threshold. Once this threshold is reached, meniscus at largest pore throat cannot be curved further and breaks, i.e., air intrudes the pores, water starts to drain. This threshold is named as Air Entry Value, AEV, and is dependent on GSD.
- ii. Intermediate suction zone: (marked as desaturation zone in Figure 2.2) ranges from AEV to residual conditions. Beyond AEV, in intermediate zone most of the pore water (bulk water) drains; air occludes pores. When all the bulk water is finished, water content keeps decreasing due to shrinkage of

pendular rings. Removal of water is insignificant once residual conditions are reached.

- iii. High suction zone: (marked as residual saturation zone in Figure 2.2) ranges from residual suction to 10^6 kPa. Vapor flow becomes dominant, signifying that total suction is of interest in this zone. Water flow is insignificantly small, but, may be in the form of film flow or humidity exchange with air. Desaturation terminates at zero water content, that is, oven-dry conditions (10^6 kPa).

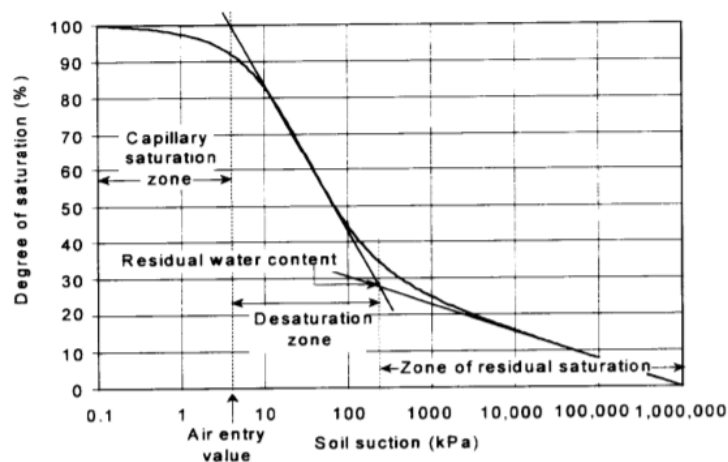


Figure 2.2. Zones of desaturation (Sillers et al., 2001)

The physical conditions of a soil (such as interfacial tension and capillary pressure, both of which are functions of water content) are directly linked to the retention curves and are associated with many geotechnical problems, such as water infiltration and redistribution, solute transport, aeration etc. In this respect, determination of retention curves is important for understanding unsaturated soil behavior and bringing solutions to many problems of geotechnical practice.

2.4 Determination of Water Retention Curves

Retention curves provide key information for deriving unsaturated soil property functions, such as permeability function, shear strength function, water storage

function, and thermal property functions (Fredlund, 2012). Despite its importance in solving unsaturated soils processes, determination of retention curves is a complex process. Retention curves can be obtained by laboratory measurements or by using estimation methods.

2.4.1 Laboratory Measurement

Amongst many available methods, conventional methods, capillary column method, syringe method, and evaporation method are the most common techniques used in the laboratory to determine the retention curves.

Conventional methods deal with three main steps to obtain a single point on WRC of one soil: i) imposing a known value of suction to soil by different techniques (e.g. pressure plate extractors, osmotic control, relative humidity control etc.), ii) allowing the moisture to come to equilibrium with the applied suction, iii) measuring the water content (ASTM, 2016). This sequence gives reliable measurements, but has limitations. Firstly, there is no suction control method to capture the entire suction range. Distribution of WRC on suction axis will always be wider than the range of suction which the adopted technique is applicable. To solidify, the use of the hanging column technique in relatively low suction values; axis translation techniques in the intermediate range of suction, and relative humidity control techniques in relatively high ranges of suction is required. This leads use of multiple methods. Another problem with conventional methods is that waiting for moisture equilibrium leads to large test durations. It takes around 1 day per point on the curve.

Capillary column method handles the latter problem by providing lots of points on the curve with one specimen. Capillary column method provides advantage for obtaining continuous curves, but it is most reliable for sandy soils under low suctions (ASTM, 2016).

Syringe method employs axis translation system where High Air Entry (HAE) ceramic is connected into a syringe. Then, whatever water is moving in or out, can

be measured by the syringe (ASTM, 2016). Hence, syringe method provides continuous curve rather than data points on WRC, by continuously measuring how much water flows in or out as suction changes. This enables the whole behavior of WRC to be revealed (scanning curve, wetting, drying, wetting again etc.), and eliminates the need for oven drying. Also, the time at which equilibrium is reached can be seen at that instant; no need for waiting to be sure as in conventional methods. However, the problem with syringe method is that the dissolved air can diffuse to the water side of HAE ceramic, and become gas again, occupying extra volume. This in turn misleads the measurements.

Evaporation method measures suction usually with a tensiometer (Toker et al., 2004). When the suction exceeds the range of the tensiometer, it determines the water content. This method can be used to obtain drying curve only and can go up as much as the range of tensiometer allows. Evaporation system never reaches static equilibrium because evaporation occurs from the surface, where the suction is larger.

Each of the techniques requires experienced labor. In addition, apparatuses are not easy to assemble. Measuring matric suction is challenging due to cavitation. Total suction is difficult to deal with because it requires accurate temperature control. Pore-water pressures are highly negative, which is another challenge in measurements.

In overall, although laboratory measurements lead reliable results (Fredlund et al., 2007), they are demanding considering the time and cost. An alternative way of obtaining WRCs is to estimate it based on textural information which has generally been found adequate.

2.4.1 Estimation Methods

The soil being a medium containing randomly distributed, interconnected pores, renders the voids distribution, which is related to GSD, to form the basis of WRC

estimation. In this regard, due to shorter duration and lower cost, estimating the retention curves based on GSD data is desirable.

Estimation methods used to obtain the retention curves can be classified into three categories (Zapata et al. 2000).

Category 1 estimates WRC by converting GSD data into pore size distribution to reveal the relation between distribution of water and pore pressures.

As one of the most used techniques, Pedo-transfer Functions (PTFs) converts GSD and other information, such as soil organic matter content and mineralogy, to the equivalent capillary pore size distribution on the basis of capillary model. PTFs assume spherical particles and cylindrical pores, treating soil as capillary tubes. Although Arya and Paris (1981) employed empirical factor for this type of uncertainties, in practice, there are pore volumes interconnected via pore throats. Pore throats control at what suction the pore would be drained. Pore network models (e.g. Sattari and Toker, 2016) use pore volumes to estimate pore water, and uses pore throats to obtain the suction required to drain the adjacent pore volume. This category is mostly reliable for coarse soils under low suctions.

Category 2 uses curve fitting to estimate WRC by correlating soil properties (e.g. GSD and porosity) to the selected data points on the curve (Gupta and Larson 1979; Rawls and Brakensiek 1982; Zapata 1999).

Category 3 estimates the retention curves by correlating soil properties to the fitting parameters of the existing mathematical models of WRC. Numerous models are proposed for representation of retention curves as outlined in Figure 2.3. In general, equations contain fitting parameters to locate the fixed points on WRC (i.e. AEV, residual/saturated conditions), and to capture the shape between these points. Three or four parameter models lose their simplicity in equation, but are more flexible to fit on a wider suction range; represent WRC better. Equation parameters can be obtained by fitting the model to the measured data using least-squares regression analysis.

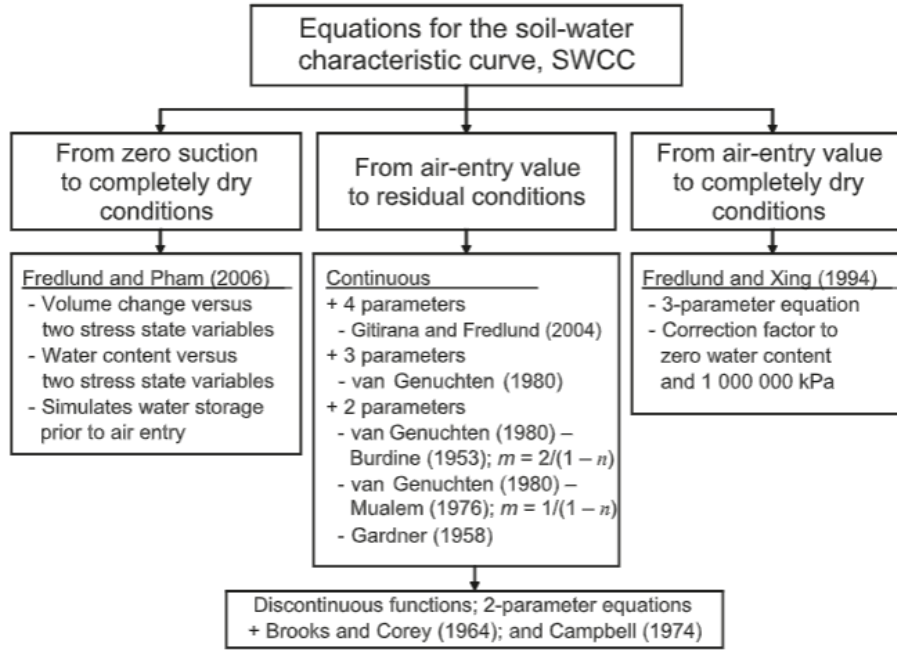


Figure 2.3. Categorization of the proposed models based on suction ranges (Fredlund et al., 2011)

Among the many proposed models, Brooks and Corey (1964), van Genuchten (1980), and Fredlund and Xing (1994) are the most adopted ones.

Brooks and Corey (1964) proposed a two-parameter, discontinuous function to describe WRC as:

$$\Theta_e = \left(\frac{\psi_{aev}}{\psi} \right)^\lambda \quad (2.2)$$

which may also be written in the form:

$$\theta = \begin{cases} \theta_s & \psi \leq \psi_{aev} \\ \theta_r + (\theta_s - \theta_r) \left(\frac{\psi_{aev}}{\psi} \right)^\lambda & \psi > \psi_{aev} \end{cases} \quad (2.3)$$

where:

Θ_e = volumetric water content normalized between saturation and residual conditions, $\Theta_e = (\theta - \theta_r) / (\theta_s - \theta_r)$. The degree of saturation, S , may be used in place of Θ_e if volume change is negligible.

ψ = soil suction,

ψ_{aev} = suction value at which soil starts to desaturate.

θ = volumetric water content,

θ_s = volumetric water content at saturation,

λ = pore size distribution index as an indicator of slope of desaturation line,

Relatively higher values of λ indicates a higher porosity and uniformity, and coarser grains if air-entry value remains constant (Lu and Likos, 2004).

It is the saturation of the portion beyond AEV up to residual suction that matters in Brooks & Corey function. The range in which it is applicable added to the absence of inflection point renders Brooks & Corey model be more appropriate for coarse grained soils, which drain in a relatively narrow range of suction. Brooks & Corey model is widely used because of its simplicity and parameters having physical meaning.

van Genuchten (1980) curve correlates its parameters, α , n , m to particle size distribution as:

$$\Theta_n = \frac{1}{\left[1 + (\alpha\psi)^n\right]^m} \quad (2.4)$$

where α (kPa) is the scale factor inversely proportional to mean pore diameter, hence in correlation with GSD, n and m are the shape factors related to pore size distribution and symmetry, respectively, $m = 1 - 1/n$, $0 < m < 1$, and Θ_n is the normalized volumetric water content. It is advantageous to present WRC (for its

applications) as a single continuous differentiable parametric expression. In this regard, van Genuchten equation is preferable as it is a sigmoidal type of function. However, it comes with limitations: first, it estimates WRC asymptotic to the horizontal line in the low suction range, whereas the actual behavior may show a sloped line. This also means a zero water storage around saturation, which is not realistic. Second, since the suction value may go up to 10^6 kPa, a correction factor is required to be applied in the high suction range. In this regard, Fredlund and Xing (1994) adopted a correction factor to obtain more realistic results, and proposed a model derived from pore size distribution function as:

$$\theta(\psi) = \frac{\theta_s (1 - \ln(1 + \psi/\psi_r)) / \ln(1 + 10^6/\psi_r)}{(\ln(\exp(1) + (\psi/a)^n))^m} \quad (2.5)$$

where the fitting parameters are:

a = the inflection point variable

n = steepness variable (slope at inflection point)

m = additional fitting parameter

ψ_r = residual suction

The units of a are in terms of stress. The inflection point will be a fraction of an order of magnitude larger than the point where WRC starts to bend downwards. The n fitting parameter designates the steepness of WRC (or the rate at which water is removed from the soil in response to increasing suction). A low n value signifies the gradual removal of water as soil suction is increased. While a high n value signifies the rapid removal of water once the soil starts to desaturate. The correction factor, $C(\psi)$, hidden in the above equation, forces the equation to reach zero water content at 10^6 kPa as:

$$C(\psi) = \left(1 - \frac{\ln(1 + \psi/\psi_r)}{\ln(1 + 10^6/\psi_r)} \right) \quad (2.6)$$

On the other hand, zones of WRC may be idealized by locating its parameters. AEV indicates the suction value at which the first intrusion of air takes place, that is, the start of desaturation. Consequently, WRC bends downwards at AEV, until the rate at which water removal is changed on a logarithm scale, i.e., residual water content (θ_r). These characterizations enable quantification of the parameters of WRC in terms of a graphical approach. If a tangent line is constructed from the inflection point of the retention curve, the intersection of the tangent line with the lines extended from the saturated and residual conditions can approximate the AEV and residual conditions of the soil. Therefore, WRC parameters can be obtained in graphical manner.

Apart from these, it is a common practice to use soft computing methods to relate suction - pore water relationship to WRC-related soil properties, e.g., Support Vector Regression (Lamorski et al., 2017), ANFIS (Karandish and Šimůnek, 2016), k-nearest neighbors algorithm (Botula et al., 2013), genetic algorithm (Johari et al., 2006), neural networks (Schaap et al., 1998) etc.

The first attempts were made by Pachepsky et al. (1996) by using grain size distribution data and bulk density as predictors of PTF. Schaap and Bouten (1996); Koekkoek and Booltink (1999) included organic matter content (OMC) to predict the water contents at selected matric suctions, and concluded that including OMC to inputs improved the prediction accuracy for the Dutch and Scottish soil databases they used.

Børgesen and Schaap (2005) concluded that detailed soil textural grouping (7 textural classes) results in no significant difference from a rough grouping (percentages of sand, silt, and clay) while including organic matter content (OMC) and bulk density in prediction of pointwise PTFs using Danish database. Johari and Javadi (2010) included five input variables: initial void ratio and gravimetric water content, clay and silt fractions, and logarithm of selected suction values to predict gravimetric water content values. Haghverdi et al. (2012) suggested a pseudo-continuous PTF model by including matric suction as input, and leaving the

corresponding water content as the only target. Saha et al. (2018) used separate NN models for plastic and non-plastic soils with one hidden layer of 20 nodes, and predicted Fredlund and Xing (1994) parameters one at a time.

Achieng (2019) indicated that radial basis function-based support vector regression outperformed other machine learning techniques amongst linear and polynomial kernels, one hidden layer neural network, and deep neural network for both drying and wetting retention data of loamy sand soil. Additionally, it is purported that RBF-based SVR models do not require soil physical properties but retention data which was also reported by Lamorski et al. (2017) for drying curves. It is also noted that DNN outputted superior simulations than ANN.

Pham et al. (2019) constructed 3-layered and 1-layered models with fixed number of neurons ([9 18 36 54] and [12 12 6], respectively) that output the water content corresponding to the suction value defined as input feature, and concluded that (i) three layered NN outperformed the standard one hidden layered NN (ii) pointwise PTFs resulted in superior performance than parametric PTFs. They benefited conventional outlier elimination methods for univariate data.

Contrarily, Bayat et al. (2013) reported that one hidden layered NN topologies perform better than those with multi-layers. They developed NN models of hidden nodes ranging from 3 to 30, to predict WRC at suction values 1, 5, 25, 50, and 1500 kPa. This implies that the generated models are and should be design specific.

In brief, the literature identifies that adopting basic index properties, e.g., GSD and density, usually results in satisfactory prediction of WRC. However, NN models are typically portrayed with either standard one hidden layer or slightly varied multi-layers of fixed number of neurons, which output usually a single point on the curve. The little discussed impact of hyperparameter tuning (in terms of optimization algorithm, stopping criteria, and learning rate) and multivariate target prediction (Fredlund and Xing (1994) fitting parameters, group WRC parameter prediction, group point-wise prediction) was the main incentive of this study. As cited by Pham et al. (2019), according to Géron (2017), although it may seem like an advantage,

flexibility of NN can be considered a pitfall as there are several hyperparameters needed to be properly adjusted. Hence, such an aim requires a design-based study through trial and error since most studies do not follow a clean path in this respect.



CHAPTER 3

NEURAL NETWORKS

This chapter presents a brief overview of Neural Networks (NNs), as they are not common tools of geotechnical engineering.

3.1 Preliminaries of Neural Networks

Neural Network (NN) is a nonlinear data processing technique widely preferred for its learning and generalization ability, and flexible nature. NNs map the relations between the predefined inputs and outputs. Once trained, for any independent data set that NN has not seen before, it gives quite fast predictions. The next section covers the basic concepts about neural networks: neuron, feedforward neural network, training, cost function and backpropagation. Then, a review of the applications in WRC prediction will be presented.

3.1.1 Neurons

A neural network is built from several interconnected unit processors, called neurons (or nodes). Each neuron receives impulses from other neurons via connections. The strength of an impulse on a connection is amplified or dampened by an adjustable scalar called weight. Weights are similar to slope in linear regression; they indicate how strongly a neuron influences another.

A neuron processes the input in two parts when fed by input $X = [\underline{x}_1, \underline{x}_2, \dots, \underline{x}_d]$ as demonstrated in Figure 3.1.

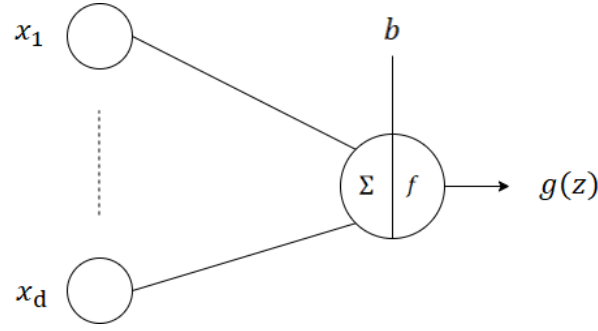


Figure 3.1. A neuron is a function of its parameters and input features

The first part multiplies each input by its corresponding connection weights, and adds an offset, called bias. The output of this linear combination of inputs is called the weighted sum, z , where:

$$z = \sum_{i=1}^n w_i x_i + b.$$

Usually bias is considered to be a member of weight matrix, then in matrix form $z = \underline{w}^T \underline{x}$. The second part of the neuron applies a nonlinear function called activation function, to the weighted sums. A smooth activation function should be selected for its derivative is to be used in training. The two activation functions: sigmoid and hyperbolic tangent provide nonlinearity in calculations. Sigmoid function has the exponential form:

$$g(z) = \frac{1}{1 + e^{-z}}.$$

Sigmoid function will squeeze its inputs in to the range $[0,1]$ as can be seen in Figure 3.2. Likewise, hyperbolic tangent (tanh) function lies in $[-1,1]$:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

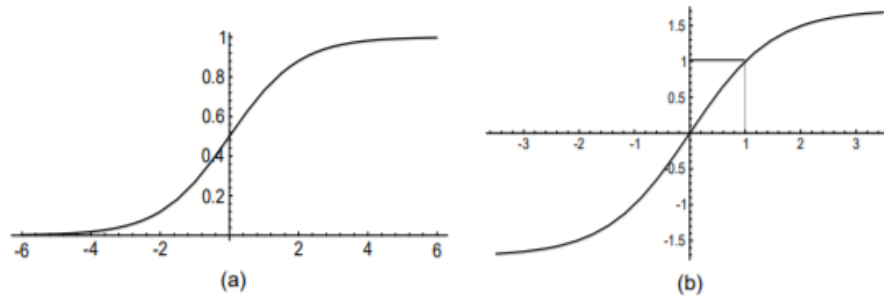


Figure 3.2. (a) Sigmoid function, and, (b) hyperbolic tangent function (Orr and Müller, 1998)

Afterwards, the output of a neuron becomes either an input to another neuron or the final output of the system.

3.1.2 Feedforward Neural Networks

A Feedforward Neural Network (FFNN) is a combination of a set of layers and neurons. Information flows only in the forward direction in FFNNs. The graph illustration of a three layered feedforward model is presented in Figure 3.3. The network topology is denoted as input-hidden-output for the example, 3-4-3, in Figure 3.3. The first layer, called input layer, does not process any data, it only serves to pass the features. Each neuron is fully connected to its following layer, feeding the outputs through the network. Between input and output layer, the hidden layers map the relationship between inputs and targets through a series of mathematical and matrix operations. Lastly, the neurons in the final layer give system outputs.

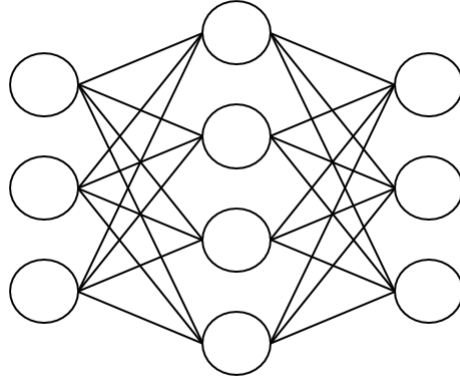


Figure 3.3. A feedforward network with a single hidden layer

3.1.3 Training of FFNNs

The aim of training is to search for a function in such a way that predictions are as close to targets as can be. Minimization of error underlies the notion of training, in which case the objective function takes the form:

$$\min_w E(w).$$

The distance between actual output $\{y\}_{i=1}^n$ and model output $\{\hat{y}\}_{i=1}^n$ is the residual, in the simplest form as:

$$r_i(w) = y_i - \hat{y}_i, \forall i$$

n being the sample size. Quadratic form for learning error is mostly used in fitting as:

$$E(w) = \sum_{i=1}^n r_i(w)^2 \quad (3.1)$$

Therefore, training is basically an optimization problem and can simply be solved by employing a nonlinear least square minimization method. Backpropagation tells how

to compute the gradients so that afterwards nonlinear minimization methods can be applied to correct the weights with some initial value.

3.1.3.1 Backpropagation Algorithm

Backpropagation algorithm systematically applies series of chain rule to compute error gradients. Backpropagation networks operate in two steps. In the first phase, information propagates in forward direction, taken by input layer, processed in hidden neurons and transferred to output layer. Then, error is calculated from generated output and target values. If the error is not small enough, it will back propagate from output layer to input layer, modifying the weights of each connection by an optimization algorithm. This iteration will keep going until the training error falls into an allowable range (He, Xu 2010).

Following notation will be used for the rest of the section shown in Figure 3.4. Superscripts denote the index of the layer and subscripts denote indices for neurons and connections.

The training set with n observations is denoted as $X = \{(x_i, y_i)\}_{i=1}^n$ where x denotes the inputs and y denotes targets.

w_{jk}^l : the weight for the connection to node j at layer l , from source k in layer $l-1$

b_j^l : bias for node j in layer l

a_j^l : net output for node j at layer l

z_j^l : weighted sum for node j at layer l

g : activation function for hidden neurons. g applies nonlinearity in element-wise manner to weighted sums, a .

g_o : activation function for output neurons

L : number of layers, excluding the input layer.

δ_j^l : error for node j at layer l

δ^L : error for the last layer, L .

α : learning rate

$f'(x)$: derivative of $f(x)$

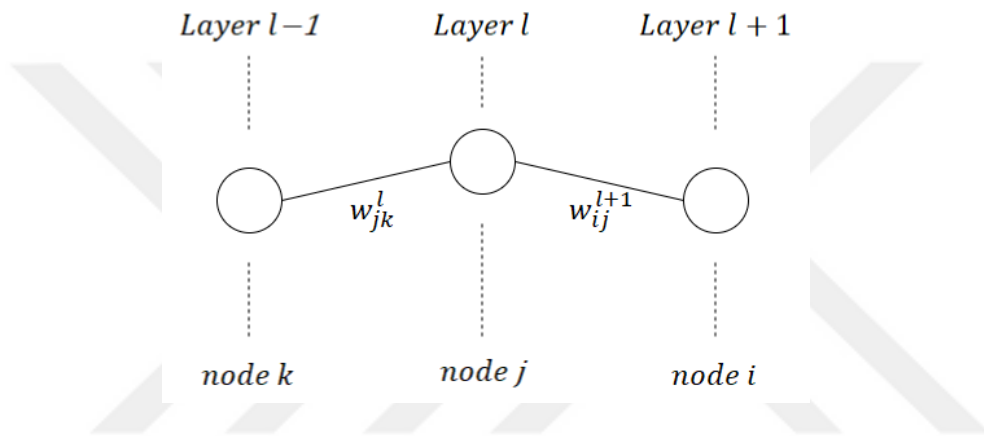


Figure 3.4. Notation for BP algorithm

When network with a single hidden layer is fed with input, outputs will be calculated all the way to last layer. The following series of equations are compiled from Krawczak, (2013); Priddy and Keller, (2005).

$$\begin{aligned}
 a^0 &= x \\
 z^1 &= w^1 a^0 + b_1 \\
 a^1 &= g(z^1) \\
 z^2 &= w^2 a^1 + b_2 \\
 a^2 &= g(z^2)
 \end{aligned} \tag{3.2}$$

In vectorized form, summary of feedforward propagation is:

$$\begin{aligned} z^l &= w^l \cdot a^{l-1} + b^l \\ a^l &= g(w^l \cdot a^{l-1} + b^l) \end{aligned} \quad (3.3)$$

where a^l is composed of a_j^l elements and z^l is composed of elements:

$$z_j^l = \sum_{j=1}^n w_{jk}^l a_k^{l-1} + b_j^l. \quad (3.4)$$

For the sake of simplicity, the bias vector is plugged in the weight matrix. For each element of vector b^l , notation becomes:

$$w_{j0}^l = b_j^l.$$

Then, the weighted sum takes the form:

$$z_j^l = \sum_{j=0}^{n-1} w_{jk}^l a_k^{l-1}. \quad (3.5)$$

and scheme of a single neuron would appear as illustrated in Figure 3.5.

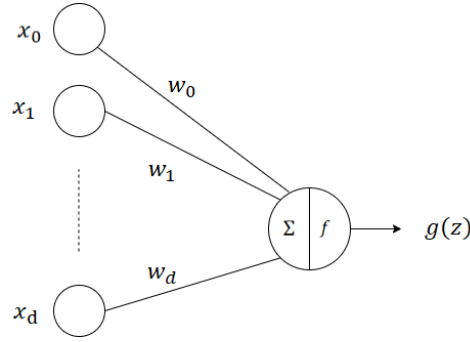


Figure 3.5. Bias unit with a fixed output of 1

The derivative of error function with respect to any weight in the system is:

$$\frac{\partial E(w)}{\partial w_{jk}^l} = \frac{\partial E}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l}. \quad (3.6)$$

The second term is known in Equation 3.6. Recalling Equation 3.5, the second term becomes:

$$\frac{\partial}{\partial w_{jk}^l} \left(\sum_k w_{jk}^l a_k^{l-1} \right) = a_k^{l-1}. \quad (3.7)$$

But the first term on the right-hand side of Equation 3.6 is unknown and is named as δ_j^l :

$$\delta_j^l = \frac{\partial E}{\partial z_j^l}. \quad (3.8)$$

This decomposition in Equation 3.6. leads to definition of error. It is defined over weighted sums, before the nonlinear conversion of activation function. Applying chain rule, Equation 3.8 takes the form:

$$\delta_j^l = \frac{\partial E}{\partial z_i^{l+1}} \frac{\partial z_i^{l+1}}{\partial z_j^l}. \quad (3.9)$$

Again the first term in the right hand side is unknown, call it δ_i^{l+1} . The second term is:

$$\frac{\partial z_i^{l+1}}{\partial z_j^l} = \frac{\partial z_i^{l+1}}{\partial a_j^l} \frac{\partial a_j^l}{\partial z_j^l},$$

where $\frac{\partial z_i^{l+1}}{\partial a_j^l} = \sum_j w_{jk}^{l+1}$, and $\frac{\partial a_j^l}{\partial z_j^l} = \frac{\partial g(z_j^l)}{\partial z_j^l} = g'(z_j^l)$. Then Equation 3.9 yields:

$$\delta_j^l = g'(z_j^l) \sum_i \delta_i^{l+1} w_{ij}^{l+1}. \quad (3.10)$$

Or, in matrix form as:

$$\delta^l = ((w^{l+1})^T \delta^{l+1})(g'(z^l)) \quad (3.10b)$$

Equation 3.10 is key to backpropagation. Now, the error in each layer can be computed in terms of its subsequent layer's error. The $(w^{l+1})^T$ term in the equation enables backpropagation of error. Once the error in last layer is computed, each gradient can be calculated. At the last layer L ,

$$\delta_j^L = \frac{\partial E}{\partial z_j^L} \quad (3.11)$$

where $a_j^L = z_j^L$ for output layer. Then, error for output layer is:

$$\delta_j^L = \frac{\partial E}{\partial a_j^L}. \quad (3.12)$$

Now that δ^L is known, using Equation 3.10 δ^l can be computed in previous layer and so on up to first layer. Knowing all δ^l and a^l , derivative of output with respect to each weight can be computed as $\frac{\partial E(w)}{\partial w_{kj}^l} = \delta_j^l a_k^{l-1}$.

In summary, the intuition is as follows:

Step 1: Introduce training set $\{(x_i, y_i), \dots, (x_n, y_n)\}$ and initialize weights.

Step 2: Perform feedforward pass up to last layer and induce a^l for $l = 1, 2, \dots, L$.

Step 3: Using each target data, compute the error in output layer $\delta^L = (a^L - y)$.

Step 4: Perform backpropagation to compute the error in each layer excluding input and output layers (there is no error in input layer), $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot g'(z^l)$.

Step 5: Compute gradient of $E(w)$, with respect to each w_{jk}^l .

Step 6: Update weights for each iteration t as

$$w_{jk}^l(t+1) = w_{jk}^l(t) + \alpha \frac{\partial E(w)}{\partial w} (t) \quad (3.13)$$

in case of gradient descent. The network steps according to selected optimization method.

Step 7: Reiterate the computations in step 2-6 until error falls below a preset value, i.e., weights converge.

Gradient descent modifies weights in the opposite direction of error gradient, so that the error function decreases in the fastest way, until reaching a minimum. Learning rate (α) determines how small steps will be taken from one iteration to the next when updating the weights. As approaching to convergence the gradient term will get smaller and gradient descent will automatically take smaller steps.

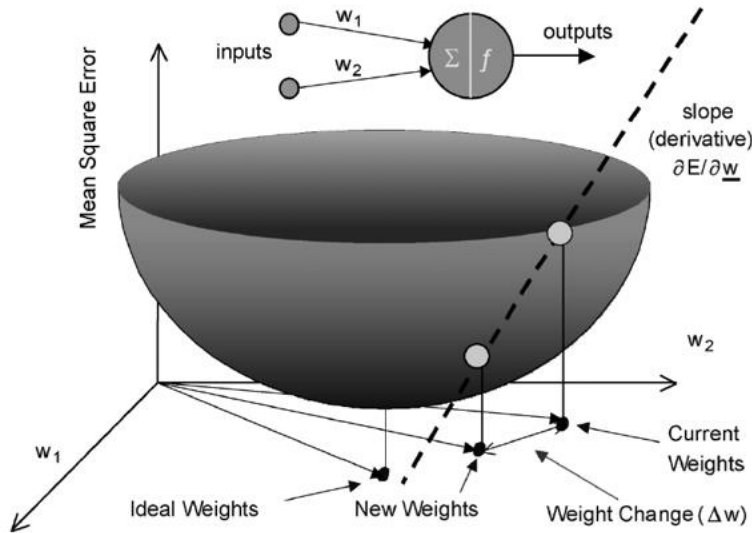


Figure 3.6. Gradient descent for mean square error error function (Priddy and Keller, 2005).

Figure 3.6 shows how gradient descent works on a simple convex objective function. However, error surface is usually non-convex and has lots of local minima in the case of high dimensional data. To remedy this shortcoming, training shall be

conducted with different initial weights which will be explained in detail in the following chapter. Details regarding to the algorithms used in this study: Gradient-Descent and Levenberg- Marquardt algorithm, can be found in (Ruder, 2016).

3.1.3.2 Principal Component Analysis

Principal component analysis (PCA) reduces dimensionality of its input $\underline{x} \in \mathbb{R}^d$ and projects it to a space spanned by principal directions: $\underline{y} \in \mathbb{R}^p$. If a data set $X = [\underline{x}_1, \dots, \underline{x}_n]_{d \times n}$ is projected on a vector $[\underline{u}_1]_{d \times 1}$, the linear transformation yields the projections $\underline{u}_1^T X$. PCA searches for directions such that variance of $\underline{u}_1^T X$ is maximized.

$$\max_{\underline{u}_1} \text{var}(\underline{u}_1^T X)$$

Let sample covariance of original matrix X is denoted by S , the objective is:

$$\max_{\underline{u}_1} \underline{u}_1^T S \underline{u}_1.$$

It is an optimization problem on a quadratic function, which has no upper bound.

To overcome the ill-definition, the problem is subjected to constraint $\underline{u}_1^T \underline{u}_1 = 1$:

$$\max_{\|\underline{u}_1\|=1} \underline{u}_1^T S \underline{u}_1 \quad (3.14)$$

since the direction of $\underline{u}_1$ is what matters, not its length. Now, Lagrangian is

$$L(\underline{u}_1, \lambda) = \underline{u}_1^T S \underline{u}_1 - \lambda(\underline{u}_1^T \underline{u}_1 - 1).$$

The derivative of L with respect to $\underline{u}_1$ is:

$$\frac{\partial L}{\partial \underline{u}_1} = 2S\underline{u}_1 - 2\lambda\underline{u}_1 \quad (3.15)$$

where $[\underline{u}_1^T]_{1 \times d} [S]_{d \times d} [\underline{u}_1]_{d \times 1}$ is a scalar and its derivative with respect to a vector yields another vector in same dimension. If Equation 3.15 is set to zero:

$$S\underline{u}_1 = \lambda \underline{u}_1.$$

This means eigenvalue and eigenvectors of covariance matrix S give saddle point of L . Plugging the above description into Equation 3.14 leads:

$$\underline{u}_1^T S \underline{u}_1 = \underline{u}_1^T \lambda \underline{u}_1$$

Since λ is a scalar:

$$\underline{u}_1^T S \underline{u}_1 = \lambda \underline{u}_1^T \underline{u}_1 = \lambda.$$

This means Equation 3.14 results in the corresponding eigenvalue for any eigenvector. PCA sorts eigenvalues in descending order. Then selects the eigenvector corresponding to the highest eigenvalue as the direction of the first principal component, PC1. PC2 explains the second highest variance in projected data in direction orthogonal to PC1, and so on. PCA employs singular value decomposition to succeed eigen decomposition of S .

$$X = U \Sigma V^T$$

where

U : eigenvectors of XX^T

Σ : diagonal matrix of eigenvalues of XX^T

V^T : eigenvectors of $X^T X$.

If X is centralized by subtracting the mean

$$\tilde{X} = (X_{d \times n} - M_{d \times n})$$

where $X = [\underline{x}_1, \dots, \underline{x}_n]_{d \times n}$, $M = [\underline{\bar{x}}, \dots, \underline{\bar{x}}]$, and $\underline{\bar{x}} = \frac{1}{n} \sum_{i=1}^n \underline{x}_i$. Then $\tilde{x}\tilde{x}^T$ becomes the covariance matrix of the original data X . If singular value decomposition is applied on $\tilde{x}$, columns of U gives eigenvectors (principal components) of S .

Now each original data point can be projected onto i th principal direction by $\underline{U}_i^T$ and get back to original space by $\underline{U}_i$. PCA analysis were conducted in order to reduce the dimensionality.



CHAPTER 4

METHODOLOGY

The following section describes the procedures and methods used in establishment of NN models in detail.

4.1 Data Collection

A variety of information of 575 soils was adopted from Unsaturated Soil Database (UNSODA) which had been developed by the U.S. Salinity Laboratory, U.S. Department of Agriculture, with suction - volumetric water content points on main drying retention curve of various type of soils (Figure 4.1) measured by different techniques (e.g., pressure plate, tensiometry, pressure outflow etc.) in addition to GSD data with an upper size of 2 mm, bulk density, and porosity (Nemes et al., 2001). Table 4-1 summarizes the range of retention data and corresponding textural group.

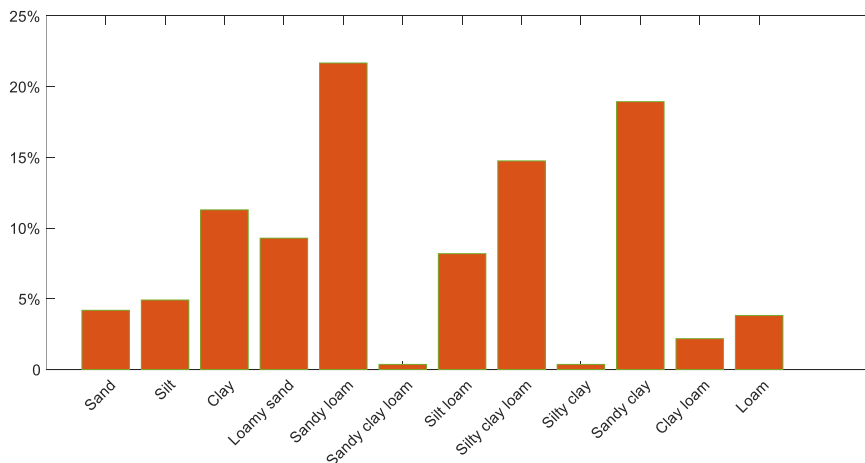


Figure 4.1. Textural class distribution of 575 soil samples in UNSODA, according to TS/EN-ISO 17892-4 classification system.

Table 4-1 Range of suction – water content points according to textural groupings of UNSODA.

Soil Texture	Suction, ψ (kPa)		Water Content, θ (cm ³ cm ⁻³)	
	maximum	minimum	maximum	minimum
Sand	83385	0	0.48	0.36
Silt	5886	0	0.61	0.40
Clay	1555	0	0.72	0.41
Loam	41202	0	0.84	0.36
Clay Loam	1471	0	0.76	0.37
Loamy Sand	1618	0	0.56	0.38
Sandy Clay Loam	4167	0	0.54	0.37
Sandy Clay	1500	0	0.47	0.37
Sandy Loam	1555	0	0.54	0.31
Silt Loam	276642	0	0.78	0.42
Silt Clay Loam	1622	0	0.80	0.47
Silty Clay	1550	0	0.70	0.32

4.2 Data Preparation

NN drives itself from the presented data, attributing great significance to reliability of the data as well as its amount.

4.2.1 Criteria for Data Elimination

The UNSODA retention data contained a considerable amount of scatter within relatively smaller sample size as stated in user's manual of UNSODA (version 1.0) by Leij et al. (1996), which was also observed by this study in version 2.0. Thereby a prudent treatment with minimum waste of data points and maximum compatibility to the rest of that sample's curve was conducted. In this sense, contaminated data

were not removed directly, but were corrected to a degree. The criteria for this correction were: (i) the water content of that data point, being larger (or smaller) than both of its neighboring observations' (Figure 4.2 (d)), (ii) if curve contained different water contents at the same suction (Figure 4.2 (a),(c)), or (iii) if water content of a data point was increased as response to increasing suction (Figure 4.2 (b)). These data points and their neighboring point, were paired, and replaced with the average of them, as exemplified in Figure 4.4.

The criteria for data elimination were: (i) data points scatter tremendously so composes a meaningless curve, and correction by interpolation is not possible (exemplified in Figure 4.3), and (ii) the samples containing no data regarding to either one of particle size or retention curve data were eliminated.

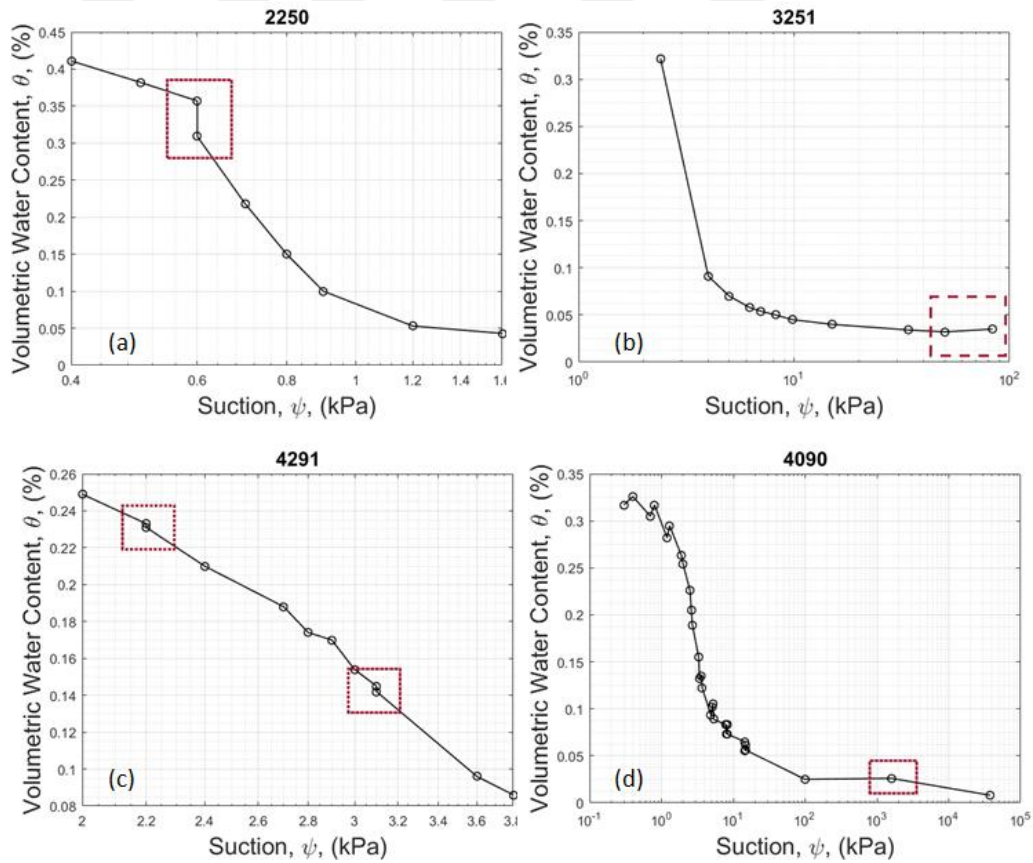


Figure 4.2. Data points highlighted in red box obeyed correction criteria.

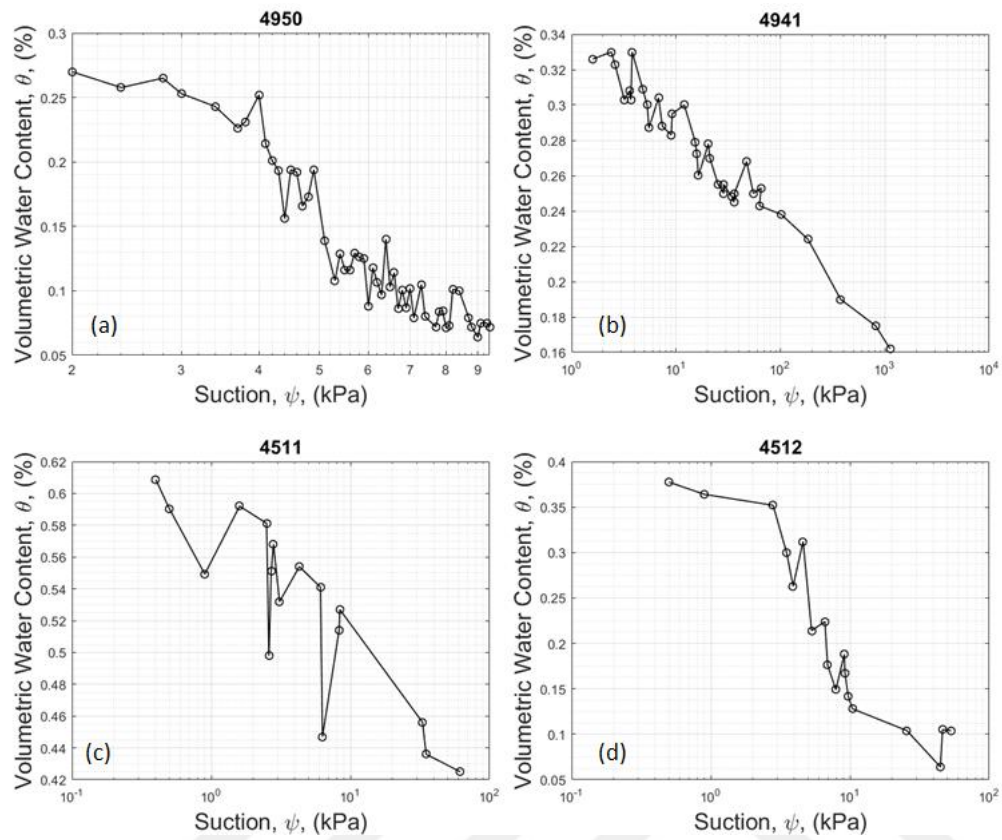


Figure 4.3. Soil samples from (a) through (d) obeyed elimination criteria.

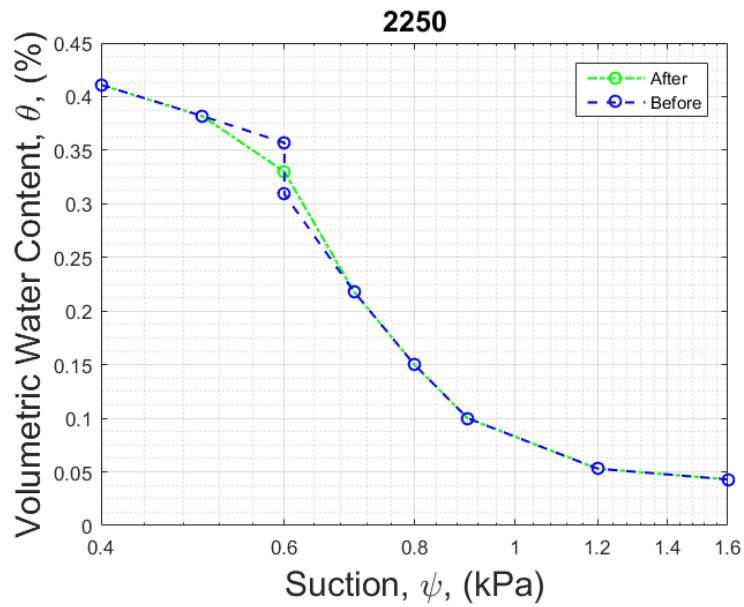


Figure 4.4. Water retention curve of sample 2250 before and after correction.

For those whose particle density and/or porosity information is missing, G_s is assumed as 2.65 g/cm^3 , and porosity was calculated as:

$$n = 1 - \frac{\rho_{bulk}}{G_s} \quad (4.1)$$

where n is porosity (%), ρ_{bulk} is bulk density (g/cm^3), and G_s is particle density (g/cm^3). Saturated water content was considered to be equal to maximum measured water content.

4.2.2 Input Variables

GSD data gathered from UNSODA was discretized as the physical indices: (i) percentages of each particle size range compatible with TS/EN-ISO 17892-4 classification system: percent clay; fine silt; medium silt; coarse silt; fine sand; medium sand; coarse sand (7-TXT), and (ii) particle diameter variables (D-Values): $(d_{10}, d_{20}, d_{30}, d_{40}, d_{60})$. Table 4-3 tabulates ranges of each input variable and corresponding sample size.

7 size fractions were extracted according to TS/EN-ISO 17892-4 classification boundaries shown in Table 4-2. Equivalent diameters were interpolated from UNSODA grain size data assuming a lognormal distribution.

Table 4-2 Particle size ranges according to TS/EN-ISO 17892-4 (Craig, 2012, p.4)

Clay	Silt			Sand			Gravel			Cobbles	Boulders
	Fine	Medium	Coarse	Fine	Medium	Coarse	Fine	Medium	Coarse		
	0.002	0.006	0.02	0.06	0.2	0.6	2	6	20	60	200
	0.01			0.1			1			10	
										100	
Particle size (mm)											

Shape of retention curves is secondarily governed by density of the soil, which was accounted by possessing porosity in input layer. Available OMC data was opted as an optional predictor as it influences the shape (Gupta and Larson, 1979; Rawls et

al. 1983) and may enhance the prediction accuracy (Schaap and Bouten (1996); Koekkoek and Booltink (1999). Other miscellaneous parameters affecting the shape of WRC, e.g., stress history, climatic changes, secondary soil structure etc. are not addressed in this study due to limited availability of data.

Table 4-3 Descriptive statistics of predictors.

Variable	Mean	St. Deviation	Minimum	Median	Maximum	Sample Size
D10 (mm)	0.05	0.05	0.00	0.05	0.28	278
D20 (mm)	0.06	0.07	0.00	0.03	0.50	372
D30 (mm)	0.09	0.09	0.00	0.07	0.56	480
D40 (mm)	0.12	0.11	0.00	0.10	0.63	537
D60 (mm)	0.21	0.16	0.01	0.18	0.81	570
OM (%)	2.85	9.47	0.01	0.94	88.40	388
Clay (%)	0.172	0.140	0.000	0.150	0.633	575
Fine silt (%)	0.082	0.071	0.000	0.066	0.303	575
Medium Silt (%)	0.108	0.086	0.000	0.090	0.348	575
Coarse Silt (%)	0.150	0.124	0.000	0.119	0.620	575
Fine Sand (%)	0.217	0.162	0.000	0.181	0.868	575
Medium Sand (%)	0.180	0.164	0.000	0.141	0.848	575
Coarse Sand (%)	0.088	0.107	0.000	0.040	0.638	575
Porosity (%)	0.448	0.085	0.256	0.438	0.820	575

4.2.3 Target Variables

Three target sets were derived from laboratory measured main drying branch of WRC and were comprised of (i) suction values at various water content: $\psi(\theta_e = 0.2; 0.5; 0.8)$, (ii) fitting parameters of Fredlund – Xing (1994) equation: a, n, m , and θ_r , (iii) retention curve variables which may also be considered

descriptors of unsaturated soil property functions (Zhai and Rahardjo, 2014): suction at which air intrusion starts, AEV; slope, s , and location, I , of point of maximum slope (the slope of desaturation line (DSR)); and/or residual water content, θ_r . Table 4.4 summarizes target variables.

Table 4-4 Descriptive statistics for target variables.

Target	Mean	St. Dev.	Minimum	Median	Maximum	Size
AEV	26.8	414.3	0.1	3.1	9925.6	575
DSR*	-0.28	0.26	-1.45	-0.19	2.23	569
θ_r	0.12074	0.09464	0.00092	0.09417	0.49021	327
I (wc)**	0.2879	0.1000	0.0400	0.2725	0.7505	575
$S(\theta_e = 0.2)$	236.1	1064.2	2.1	22.7	13175.2	287
$S(\theta_e = 0.5)$	35.21	87.18	1.42	8.23	1058.49	324
$S(\theta_e = 0.8)$	6.18	10.43	0.11	3.42	104.99	327
FX*	a	4.53	3.54	0.21	3.44	280
	m	0.58	0.38	0.01	0.55	
	n	3.61	2.88	0.46	2.64	
	θ_r	0.12019	0.09649	0.00092	0.09284	
AEV -	AEV	11.5	31.4	0.1	2.9	327
DSR -	DSR	-0.28	0.29	-1.45	-0.18	
θ_r	θ_r	0.1207	0.0946	0.0009	0.0942	
AEV -	AEV	11.5	31.4	0.1	2.9	569
DSR -	DSR	-0.28	0.29	-1.45	-0.18	
I	I	0.1207	0.0946	0.0009	0.0942	
$S(\theta_e)$	$\theta_e(0.2)$	236.1	1064.2	2.2	22.8	287
	$\theta_e(0.5)$	23.4	49.6	1.4	7.2	
	$\theta_e(0.8)$	5.6	10.1	0.1	3.3	

*Sample size is smaller due to removal of outlying observations **water content

4.2.3.1 Effective Volumetric Water Content

Effective water content, θ_e , was calculated by normalizing each sample's water content between its maximum and minimum as:

$$\theta_e = \frac{\theta - \theta_r}{\theta_s - \theta_r}$$

with θ_s being fixed as maximum measured water content value of the sample of interest and θ_r being the estimated residual water content, as will be explained below. In this way, retention data was scaled using volumetric water content.

4.2.3.2 Fitting Parameter Estimation

Available retention data was fitted to Fredlund – Xing (1994) equation adopting the correction factor in order to determine FX fitting parameters. The unknown parameters, a, m, n , were optimized at a 95% confidence interval using Excel Solver. Excel Solver treats MSE as the objective function and employs generalized reduced gradient algorithm to minimize it. Estimates of θ_r were temporarily discarded and later added as the fourth parameter. Saturation water content which can be derived from porosity and particle density, was excluded, in order to reduce the dimensionality.

4.2.3.3 WRC Parameter Estimation

It is possible to fix the position of parameters in WRC plane by constructing three lines that represents the inclinations of near-saturation – mid – end portions of WRC. This approach assumes that WRC parameters are in correlation with the input arguments: GSD and porosity (this can be exemplified by the correlations between: the inflection point on WRC and porosity, AEV and percent sand or porosity, residual conditions and percent fines etc., as mentioned in Chapter 2).

Two equations are required for this purpose: (i) AEV – inflection point and (ii) residual conditions.

Inflection point coincides the mode of pore size frequency distribution for a drying lognormal WRC (Fredlund and Xing, 1994). Hence, the two adjacent points that give the maximum slope were fitted to a line that considered as the tangent through the inflection point. The slope, s , between each consecutive suction-water content point was calculated as:

$$s = \frac{\theta_{i+1} - \theta_i}{\log(\psi_{i+1} / \psi_i)}. \quad (4.2)$$

Next, as illustrated in Figure 4.5, AEV was located at the intercept of the desaturation line and (i) the horizontal line that intersects the water content axis at maximum water content value (ii) the line passing through the first two points on WRC. As a third option, (iii) AEV was marked at the point of minimum curvature (maximum value with negative sign) on WRC. A quadratic function was fitted to each three consecutive points on the curve. Then, the middle point of the trio of points whose coefficient of the highest degree term in the fitted polynomials is minimum is taken as AEV, e.g., the red point on the curve in Figure 4.5. Eventually, the most consistent way to quantify AEV was noticed to be the first approach, considering the data at hand.

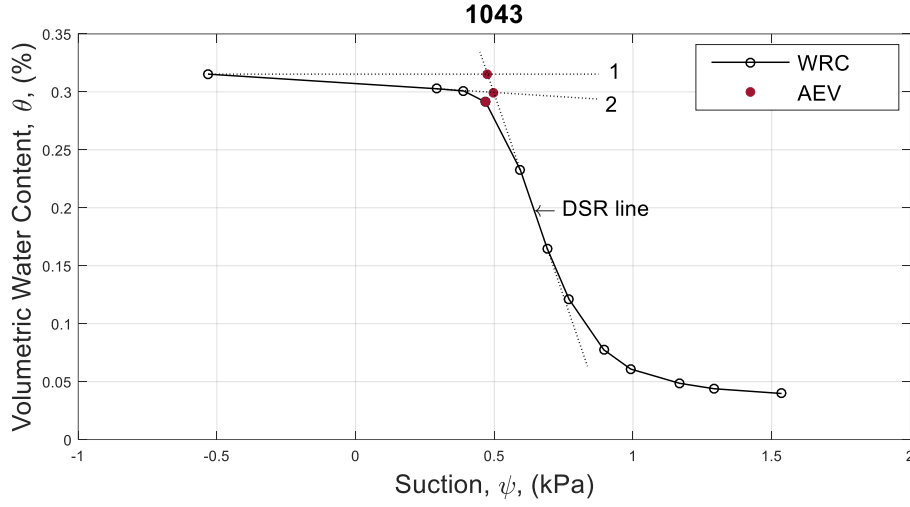


Figure 4.5. Constructed horizontal line (1) and the line passing through the first two points (2) on log-transformed suction – water content curve of sample 1043: sand soil from Live Oak, FL.

Assuming a lognormal pore size distribution, retention curve of a soil will yield an S-shape, symmetrical around the inflection point (Fredlund and Xing, 1994) in accordance with the capillary law. Then, the two parameters of pore size distribution: mean and standard deviation, will be related to AEV and inflection point, respectively. Consequently, residual conditions can be estimated as:

$$\begin{aligned}\psi_r &= 10^{2\log(\psi_i) - \log(\psi_a)} \\ \theta_r &= 2\theta_i - \theta_a.\end{aligned}\tag{4.3}$$

This relationship allows to locate residual conditions and predict complete WRC for those samples where the high suction segment is missing.

Another attempt to estimate θ_r was made by fitting a quadratic function to the 5 consecutive points at the highest suction. Subsequently, the intersection of water content axis and the line of zero slope of this polynomial was considered to be θ_r , as depicted in Figure 4.6. This procedure is only applicable for samples with convex shaped tails of retention curves.

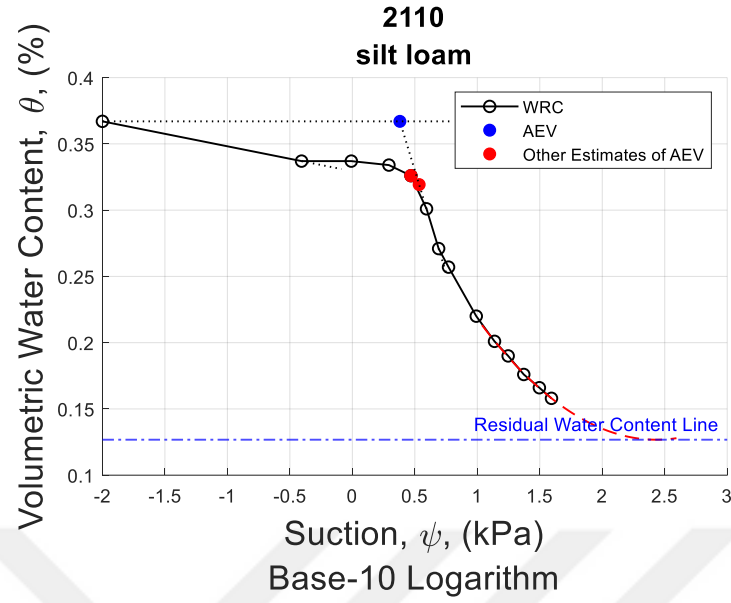


Figure 4.6. Red (dashed) line represents the fitted polynomial to maximum 5 suction points. Blue (dash-dot) line denotes the zero slope line of the red one, on log-transformed suction – water content curve of sample 2110: loamy sand soil from Lillington, NC.

Simplifications were made during derivation of parameters. The biggest complication was that all of the approaches were limited to premise that the database being consisted of ideal, complete, unimodal, monotonic retention curve data. Although alternative ways for locating AEV and θ_r were investigated, the considerable amount of scatter in results associated with errors arousing from manual drawing and erratic retention data, they are not addressed herein. Input-target matching is explained in Figure 4.7.

In case of univariate target, outlying data were diagnosed according to Tukey's rule. Data points that fall out of the range $[(Q1 - 1.5 \times IQR), (Q3 + 1.5 \times IQR)]$ with $Q1$ and $Q3$ being 25th and 75th percentiles, respectively, and $IQR = Q3 - Q1$ being the interquartile range of the parameter of interest.

It is worthy to note that fitting parameters of Fredlund and Xing (1994) Equation contained considerable amount of extreme values, notably in parameter a .

Mahalanobis distances of each data point to central location were compared with a critical value of the χ^2 distribution (Rousseeuw and Zomeren, 1990), to assign similarity, or dissimilarity, between observations (where Mahalanobis distance is basically equivalent to Euclidean distance transformed to another space that involves the sample covariance matrix). Since classical Mahalanobis distance approach is not robust against swamping and masking effects, the method suggested by Hadi (1992) was also adopted. In addition to the distance based methods, the statistical based method proposed by Hubert and Vereckeen (2008) which accounts for skewness in multivariate data.

Eventually, it seemed that assembling the univariate anomalies contained in each fitting parameter is much simpler and gave the best results. This is done by eliminating the top 3% portion of each parameter and then applying quartiles method mentioned above. Table 4.5 summarizes the three fitting parameters.

Table 4-5 Descriptive statistics of FX fitting parameters (i) raw data and, after applying (ii) the statistical based method for skewed multivariate data, (iii) assemblage of univariate outliers.

Variable	Mean	St.Dev.	Min	Max	Q1	Q2	Q3	Skewness	Kurtosis
Raw	a	6e5	e7	0.1	3e8	2.5	4.5	11.4	23.9
	m	5.8	78.4	0.0	1789.5	0.3	0.5	0.9	21.0
	n	4.6	12.4	0.2	200.0	0.9	1.5	4.1	9.3
Skew	a	8.9	12.7	0.1	108.1	2.4	4.0	9.5	2.6
	m	0.6	0.4	0.0	2.1	0.3	0.5	0.8	1.0
	n	3.0	3.5	0.2	29.5	0.9	1.6	4.0	2.2
IQR	a	4.5	3.5	0.2	16.8	2.2	3.4	5.3	1.7
	m	0.6	0.4	0.0	3.9	0.3	0.5	0.8	2.8
	n	3.6	2.9	0.5	15.0	1.4	2.6	5.2	1.3

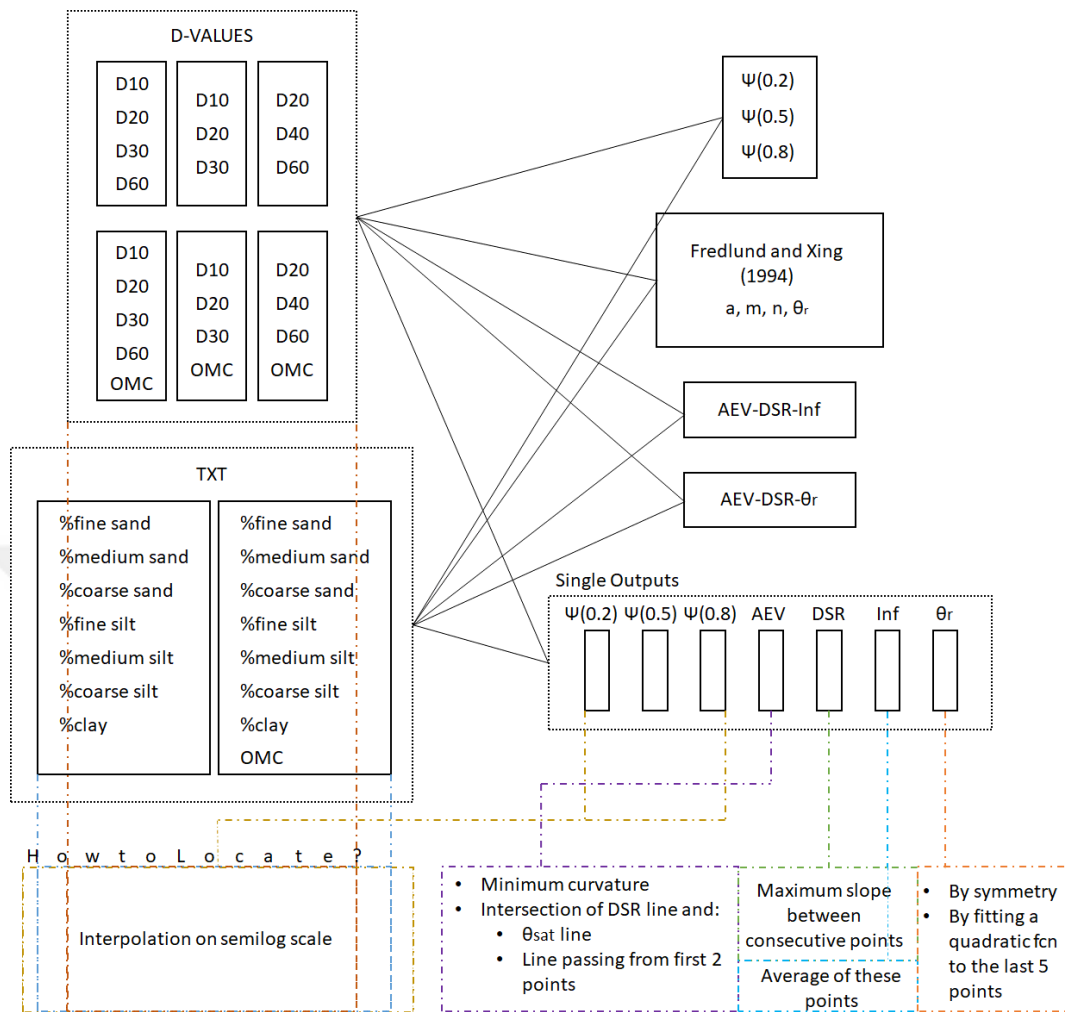


Figure 4.7. Input-target matching process.

4.2.4 Data Scaling

The purpose of data scaling is to put all features in a comparable range, so the different-scaled variables cannot reduce the quality of the models. Scaling of input data leads to faster learning and reduced possibility of getting stuck in local minima; whereas scaling of targets prevents exploding gradients.

From the three common normalization techniques: min – max, sigmoidal, and z-score normalization, z-score standardization was selected partly because of the existence of extreme valued observations that using min – max scaling would clump the small valued data, and partly to preserve the structure of data which sigmoidal transformation would violate. The data were scaled to have zero mean and unity variance as:

$$x_{std} = \frac{(x - \bar{x})}{\sigma} \quad (4.4)$$

where x_{std} is standardized data, σ is sample standard deviation, and $\bar{x}$ is the sample mean. Z-score normalization was used for each input and target data set without regard to underlying distribution.

4.3 Model Assessment

4.3.1 Network Creation

After assembling the data, a function was built to facilitate the generation of any possible modes and topology of NN models, and systematic variation of training parameters. This section explains a series of built-in functions called for model generation in the MATLAB environment. First, data were processed by `mapstd` function to have zero row mean and unity deviation by:

```
>> [xnorm, xsettings] = mapstd(x);
```

where data is denoted by x , scaled data is x_{norm} , and scale settings are recorded in x_{settings} so data can be unnormalized. The following call to `feedforwardnet` function creates an empty network object of assigned number of hidden neurons (Figure 4.8).

```
>> net = feedforwardnet ([hiddenSize]);
```

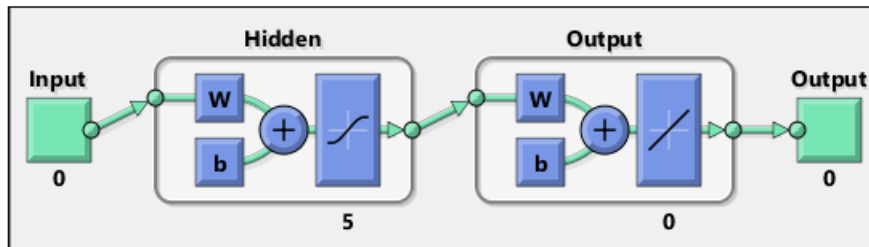


Figure 4.8. Each created layer is connected to its subsequent layer.

A network shall be configured to match to its input and targets before initializing its weights. The resulting object (Figure 4.9) can be configured to predefined input and target matrices, x and t here, by calling `configure` command.

```
>> net = configure (net,x,t);
```

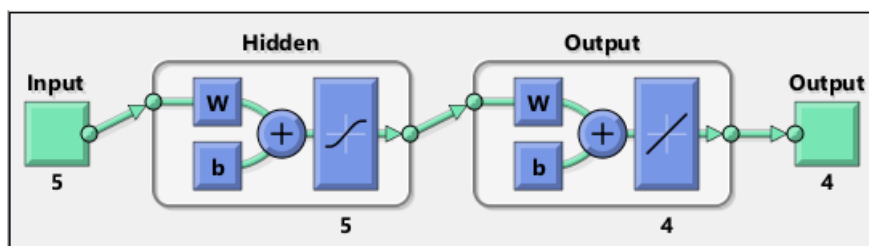


Figure 4.9. Network is configured to 5 input and 4 output features.

Properties of each layer is held at property `net.layers`. It should be noted that there are a number of network object and subobject properties. Detailed explanation can be found in MATLAB documentation. Here, only the key properties are addressed.

Nyugen-Widrow Algorithm was assigned to `initFcn` property. Transfer function was defined as hyperbolic tangent for each layer. Then, the layers of the current network object were initialized by `init()` function.

```
>> net.layers{i}.transferFcn = 'tansig';  
>> net.layers{i}.initFcn = 'initnw';  
>> net = init(net);
```

Here, index i refers to i th layer. Training options were specified in the following set of lines. For this specific design, Levenberg-Marquardt backpropagation was assigned to `trainFcn` property with random division of data into three subsets: training, validation, and test sets. The performance function is set with the `performFcn` property. For detailed options help

```
net.trainFcn = 'trainlm';  
net.performFcn = 'mse';  
net.divideFcn = 'dividerand';  
net.divideParam.trainRatio = trainRatio1;  
net.divideParam.valRatio = valRatio1;  
net.divideParam.testRatio = testRatio1;
```

Then, training properties are set to desired values. Object functions in the following series of lines are explicitly written so the user can override the default parameters. `max_fail` defines the number of consecutive validation failures before training to stop. Setting a maximum number of epochs ensures that training will stop. Performance goal is set to zero by default. `mu` is the adaptive value in Levenberg-Marquardt Algorithm, decreases by `mu_dec` and increases by `mu_inc`. Training will stop if any of these conditions are occurred or violated.

```
net.trainParam.max_fail = 6;  
net.trainParam.epochs = 1000;  
net.trainParam.goal = 0;  
net.trainParam.mu = 0.001;  
net.trainParam.mu_dec = 0.1;  
net.trainParam.mu_inc = 10;
```

Further training information will be displayed by setting `showCommandLine` to `true`. Then, `train` executes training according to the preset parameters, properties of `trainlm` in this case.

```
>> [net, tr] = train(net,x,t);
```

Once the training is stopped, the network's performance is stored in training record. This loop was closed after 20 number of trials to eliminate sensitivity to initial guesses. Then, the networks with best performance amongst them was simulated with `sim()` command. The simulations were then back scaled. Fi presents this workflow. This whole process is explained by Figure 4.10.





Figure 4.10. Workflow for network generative code plugged in Appendices.

4.3.2 Network Initialization

Before presenting the training set, each layer should be initialized with a reasonable configuration of its weights and biases. If the initial weights are too small, the net input to layers would yield to zero and the system would not move at all. Whereas, large initial values would end up with saturation of neurons where derivative of sigmoid function approaches to zero, network commences only roughly. The algorithm suggested by Nguyen and Widrow (1990) overcomes these problems by distributing each initial weight such that the active region of each neuron would span the input space uniformly. For example, if a system is trained to map a function of one variable with one hidden layer containing H hidden nodes and employing tanh activation, it will approximate the function over the region $[-1,1]$, which has length 2. This means each node spans a $2/H$ length. Nguyen-Widrow points that sigmoid is almost linear over the region $[-1,1]$

$$-1 < w_i x + w_{bi} < 1$$

yields

$$-1/w_i - w_{bi} < x < 1/w_i - w_{bi}$$

whose length is $2/w_i$. Thus,

$$\begin{aligned} 2/w_i &= 2/H \\ w_i &= H \end{aligned}$$

and w_{bi} is selected such that the bias intervals lie in between $[-1,1]$ and center of an interval is fixed at $-w_{bi}/w_i$. So w_{bi} is set to a random number between $-w_i$ and w_i .

In order to have slightly overlapping intervals, the algorithm sets $w_i = 0.7H$ and picks the interval of each neuron randomly. So each time the network is initialized,

different weights are assigned. Therefore, each generated network was initialized for a pre-defined number of times in this study.

4.3.3 Complexity Control

Complexity of the problem and the solution should match for NN to reliably extract the rules of the population stashed in the introduced finite sample data. The fitted function by NN can pass through all of the introduced points, or just few of them, depending on the number of generated unknown parameters.

$$Nw = (I + 1) \times H_1 + \dots + (H_i + 1) \times H_{i+1} + \dots + (H_n + 1) \times O$$

where plus one term counts for the bias and:

Nw : Number of unknown parameters, i.e., weights,

I : Number of neurons in input layer,

O : Number of neurons in output layer,

$\{H\}_{i=1}^{n-1}$: Number of nodes in the i th hidden layer.

As number of hidden neurons and layers increase, the number of unknown model parameters increase. If the complexity is unnecessarily large, i.e. overfitting, system passes through each point in training data, but performs poorly when met with unseen data. In other words, low training error and large difference between training and test errors are the symptoms of overfitting. Or, if underfitted, the system cannot approximate the true function accurately. A high training error may be interpreted as underfitting. It is essential, yet puzzling point in NN implementation that there are no strict rules for selecting the number of hidden neurons and layers.

Initially aimed to benefit the flexibility of NNs, many models were developed with hidden layers differentiated from 1 to 5, and nodes from 5 to 30 per layer, rather than

following an oppressed strategy by constituting the solution by some set of static parameters. Figure 4.11 demonstrates the flow chart of network generation process.

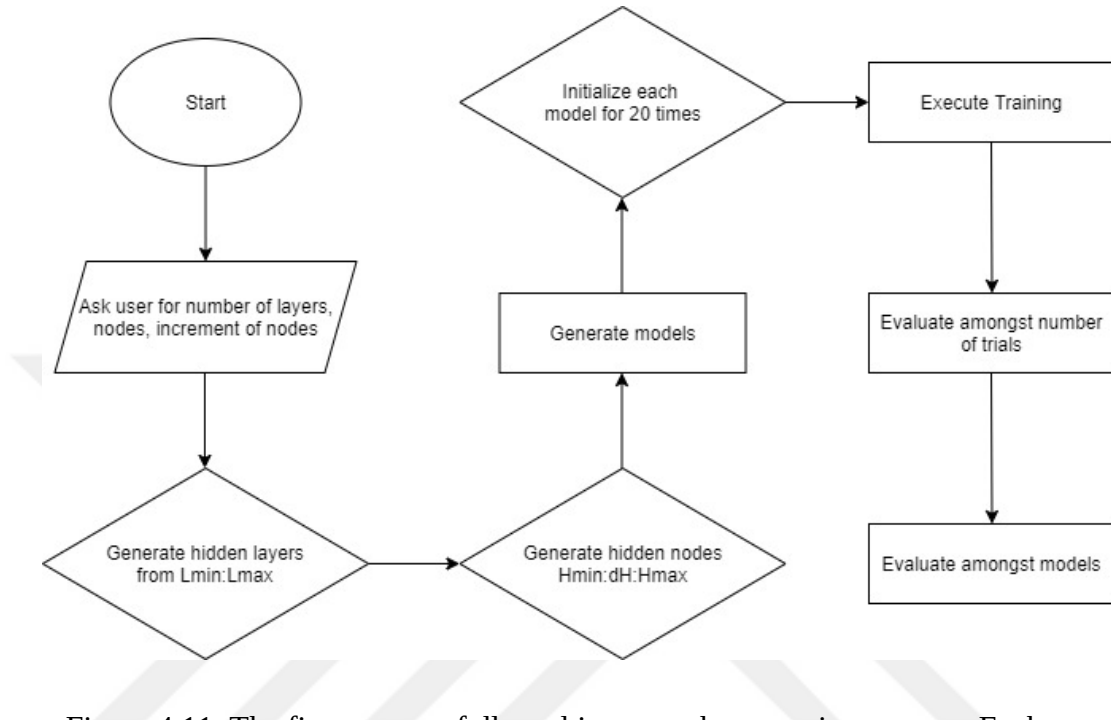


Figure 4.11. The first strategy followed in network generation process. Each run calls for a unique topology.

But, after initial runs, it has become clear that it suffices to use one or two hidden layered models for this problem and data. As can be inferred from Table 5.1, even relatively low complexities were satisfactory for training, where most models have two or at most three hidden layers (Demuth and Bale, 2000), and three hidden layered models can approximate almost any level of complexity (Rumelheart et al., 1986). Therefore, a set of models of single hidden layer and double hidden layer was constructed in MATLAB environment, such that number of nodes in hidden layers vary by $H_{\min} : dH : H_{\max}$, where $H_{\min}$ was assigned to 5 and $H_{\max}$ to 60, and dH is H_i as shown in Table 4-6.

Table 4-6 Number of nodes in single layered (top row) and two-layered (bottom row) networks.

H	5	10	20	40	60
H1-H2	5-5	10-10	20-20	40-40	60-60

Data were randomized and splitted into training (70%), validation (15%), and test (15%) sets, respectively. Since generalization performance of a model cannot be known straightforwardly as the unmet or independent data is unseen yet, a performance measurement on a distinct data from the training set, which is called validation data set, was used to estimate it. Validation set also provides an early stopping criteria to prevent excessive number of training iterations. Otherwise, if training set were to be used in performance measurements, overfitted networks would give the best results, which is not a desired solution. Therefore, models were evaluated and ranked based upon MSE of validation set.

Generated networks followed Levenberg-Marquardt algorithm (Børgesen and Schaap, 2005; Schaap et al., 1998) as the search strategy. The model with the lowest validation error was selected after each train. For the fixed complexity, test set was used to approximate generalization error of the fitted model.

Each input – target set is fed into a double for loop which trains each net for 20 trials, i.e., initializations. The flow chart illustrated in Figure 4.12 depicts the execution plan. Each input and target feature were combined as indicated in Figure 4-7, and trained 20 times for each complexity in Table 4.6. After the simulation, results were converted back to original scale.

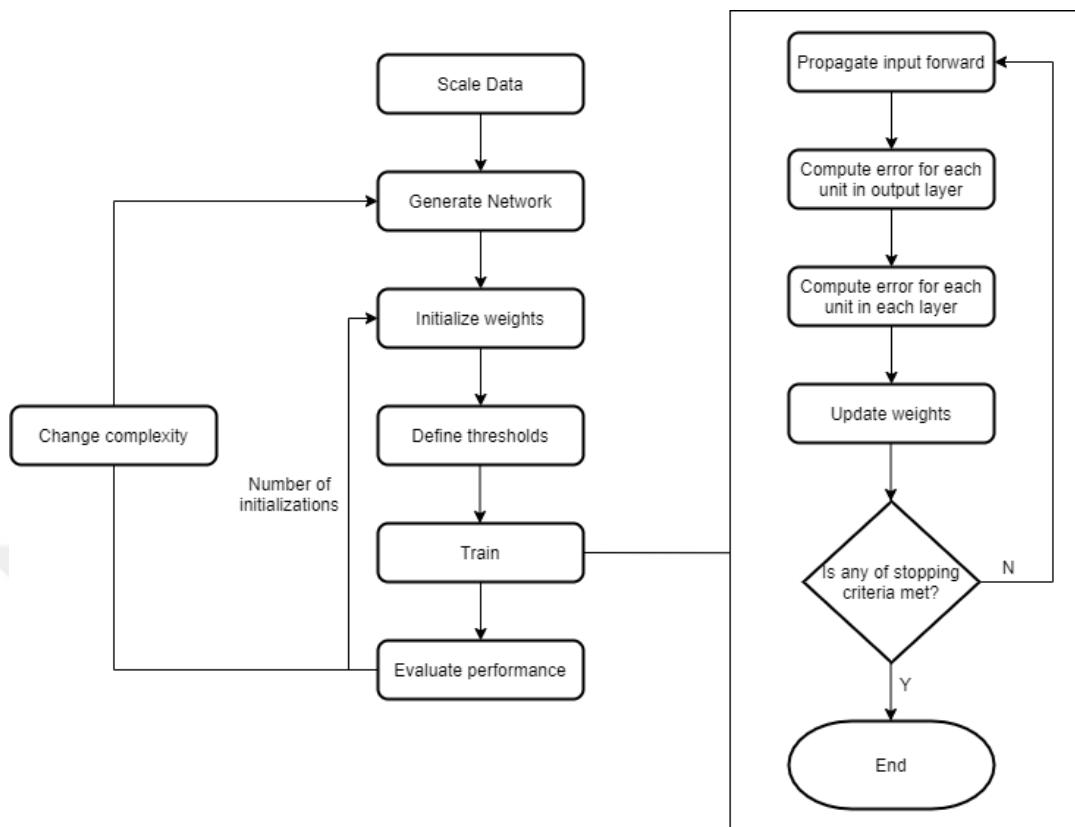


Figure 4.12. Execution process

CHAPTER 5

RESULTS

This section reviews graphical and statistical model evaluation aspects. Training results indicate the learning capability, while validation results are used in model ranking, and test results imply generalization capability of the model.

5.1 Model Comparison Criteria

Model performances were compared by adopting multiple performance statistics to prevent unduly emphasized aspects of performances: (i) Nash – Sutcliffe Efficiency (NSE) factor for the assessment of models and closeness of the predicted and actual values,

$$NSE = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

(ii) Percent Bias (PBIAS) to provide over- or under-estimation bias of a model by explaining deviations of predictions from the actual values,

$$PBIAS = \frac{\sum_{i=1}^n (y_i - \bar{y}) \times 100}{\sum_{i=1}^n y_i}$$

(iii) Mean Squared Error (MSE),

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 .$$

(iv) Mean Absolute Error (MAE), averaging the all absolute errors,

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}$$

where $\{\hat{y}\}_{i=1}^n$ are predictions, $\{y\}_{i=1}^n$ are observed values, and $\bar{y}$ is mean of observations for sample size, n , were used as comparison instruments for prediction accuracies across 88 models.

5.2 Summary of Results

Models that were trained for the same input – target configuration for different topologies and initializations were ranked based on MSE of validation set. Models that differ in input –target combination were ranked based on R^2 of test set in order to provide a measure of comparison with the literature. All of the model performances are compiled into and demonstrated in Table 5-1.

Table 5-1 Model performances along with topology and number of observations of each set. Models used in this text are shown bold.

Model	Input Variables*	Hidden Nodes	Target Variables	MSE			Sample Size		
				Trn*	Val*	Test	Trn	Val	Test
1	D1236	10-10	θ_r	0.0006	0.0032	0.0029	110	24	24
2	D1236_OMC	5	θ_r	0.0005	0.0020	0.0041	58	12	12
3	D1236	5-5	DSR	0.0303	0.0395	0.1104	143	30	30
4	D1236_OMC	5	DSR	0.1625	0.1927	0.1890	76	16	16
5	D1236_OMC	20-20	INF	0.0114	0.1358	0.8754	76	16	16
6	D1236	20	INF	0.0203	0.2639	0.1613	147	31	31
7	D136	5	AEV	0.0582	0.1593	0.1521	147	31	31
8	D136_OMC	10	AEV	0.2101	0.4178	0.5706	76	16	16
9	D136	5	AEV-DSR-INF	0.0674	0.1214	0.1177	143	30	30
10	D136_OMC	5	AEV-DSR-INF	0.0410	0.3406	0.2279	76	16	16
11	D136_OMC	10	AEV-DSR- θ_r	0.0233	0.1294	0.3932	58	12	12
12	D136	5	AEV-DSR- θ_r	0.0539	0.3410	0.2032	106	23	23
13	D136	5	$\Psi(0.2)$	0.0781	0.2526	0.2530	107	23	23
14	D136_OMC	5-5	$\Psi(0.2)$	0.1131	1.1282	0.6789	56	12	12
15	D136	10-10	$\Psi(0.5)$	0.0261	0.0431	0.0796	110	24	24
16	D136_OMC	10	$\Psi(0.5)$	0.0131	0.0431	0.1023	58	12	12
17	D136	5	$\Psi(0.8)$	0.0294	0.1022	0.0828	110	24	24
18	D136_OMC	10	$\Psi(0.8)$	0.0145	0.0579	0.1527	58	12	12
19	D136	5	$\Psi(0.2;0.5;0.8)$	0.0366	0.1786	0.1815	107	23	23
20	D136_OMC	5	$\Psi(0.2;0.5;0.8)$	0.0017	0.1962	0.4324	56	12	12
21	TXT	10-10	θ_r	0.0058	0.0109	0.0081	229	49	49
22	TXT_OMC	20	θ_r	0.0018	0.0077	0.0141	124	27	27
23	TXT	20	DSR	0.0307	0.1497	0.0487	399	85	85
24	TXT_OMC	5	DSR	0.0090	0.0489	0.0253	236	50	50
25	TXT	20-20	INF	0.1880	0.3318	0.3000	403	86	86
26	TXT_OMC	5-5	INF	0.0971	0.3561	0.4704	236	50	50
27	TXT	40-40	AEV	0.0815	0.3548	0.3635	403	86	86
28	TXT_OMC	5	AEV	0.3803	0.4811	0.6870	236	50	50
29	TXT	10-10	AEV-DSR-INF	0.0461	0.0879	0.1714	399	85	85
30	TXT_OMC	10-10	AEV-DSR-INF	0.0290	0.1644	0.1762	236	50	50
31	TXT_OMC	10-10	AEV-DSR- θ_r	0.0923	0.1562	0.1378	124	27	27
32	TXT	5	AEV-DSR- θ_r	0.1476	0.3007	0.1838	229	49	49
33	TXT	20-20	$\Psi(0.2)$	0.0517	0.2615	0.6523	201	43	43
34	TXT_OMC	5-5	$\Psi(0.2)$	0.9034	0.6202	1.6241	109	24	24
35	TXT	40	$\Psi(0.5)$	0.0799	0.3081	0.2814	226	49	49
36	TXT_OMC	10	$\Psi(0.5)$	0.1928	0.3472	0.5788	124	27	27
37	TXT	5-5	$\Psi(0.8)$	0.0631	0.1300	0.1836	229	49	49
38	TXT_OMC	20	$\Psi(0.8)$	0.0001	0.4194	0.2670	124	27	27
39	TXT	5-5	$\Psi(0.2;0.5;0.8)$	0.0324	0.3044	0.2881	201	43	43

Table 5.1 Continued

Model	Input Variables*	Hidden Nodes	Target Variables	MSE			Sample Size		
				Trn	Val	Test	Trn	Val	Test
40	TXT_OMC	40	$\Psi(0.2;0.5;0.8)$	0.0251	0.3333	0.1978	109	24	24
41	TXT	10	FX	1.2870	4.2731	7.5533	196	42	42
42	TXT_OMC	5-5	FX	0.7828	6.5607	11.3036	101	21	21
43	D136	5-5	FX	0.6875	10.9747	8.1216	100	22	22
44	D136_OMC	10-10	FX	0.4236	2.8675	8.2293	51	11	11
45	D136	5	θ_r	0.0018	0.0029	0.0025	110	24	24
46	D136_OMC	10	θ_r	0.0002	0.0028	0.0038	58	12	12
47	D136	10	DSR	0.0242	0.1007	0.0819	143	30	30
48	D136_OMC	10	DSR	0.0188	0.1271	0.2549	76	16	16
49	D136_OMC	20-20	INF	0.0099	0.1129	0.1995	76	16	16
50	D136	10-10	INF	0.0314	0.0545	0.0772	147	31	31
51	D1236	10-10	AEV	0.0509	0.1458	0.0812	147	31	31
52	D1236_OMC	5-5	AEV	0.0183	0.2539	0.3690	76	16	16
53	D1236	10	AEV-DSR-INF	0.0222	0.1782	0.0709	143	30	30
54	D1236_OMC	5-5	AEV-DSR-INF	0.0434	0.1671	0.2210	76	16	16
55	D1236_OMC	10	AEV-DSR- θ_r	0.0095	0.4226	0.2925	58	12	12
56	D1236	5	AEV-DSR- θ_r	0.0639	0.1569	0.6406	106	23	23
57	D1236	5-5	$\Psi(0.2)$	0.3513	1.4074	0.7285	107	23	23
58	D1236_OMC	10-10	$\Psi(0.2)$	0.0001	0.1973	1.0878	56	12	12
59	D1236	20	$\Psi(0.5)$	0.0401	0.1067	0.0667	110	24	24
60	D1236_OMC	5-5	$\Psi(0.5)$	0.0053	0.1029	0.4624	58	12	12
61	D1236	10	$\Psi(0.8)$	0.0333	0.1537	0.0467	110	24	24
62	D1236_OMC	5	$\Psi(0.8)$	0.0001	0.1735	0.3116	58	12	12
63	D1236	5-5	$\Psi(0.2;0.5;0.8)$	0.0554	0.1107	0.1229	107	23	23
64	D1236_OMC	5-5	$\Psi(0.2;0.5;0.8)$	0.0000	0.3636	0.1329	56	12	12
65	D1236	5	FX	0.9717	6.1427	8.4365	100	22	22
66	D1236_OMC	5	FX	12.3371	15.6748	27.5132	51	11	11
67	D246	5	AEV-DSR-INF	0.0638	0.0951	0.0649	256	55	55
68	D246_OMC	10-10	AEV-DSR-INF	0.0203	0.0906	0.0912	157	33	33
69	D246	5-5	FX	1.5725	5.1727	25.2595	148	32	32
70	D246_OMC	10-10	FX	1.1179	13.0353	4.6392	77	17	17
71	D246	5	AEV	0.1543	0.1867	0.2065	260	56	56
72	D246_OMC	5-5	AEV	0.0771	0.4528	0.3079	157	33	33
73	D246_OMC	5	AEV-DSR- θ_r	0.0671	0.3790	0.5559	95	20	20
74	D246	5	AEV-DSR- θ_r	0.0654	0.2650	0.2166	166	35	35
75	D246	20	$\Psi(0.2)$	0.1312	0.1984	0.3830	156	33	33
76	D246_OMC	40	$\Psi(0.2)$	0.0504	0.2646	0.1681	87	19	19
77	D246	40	$\Psi(0.5)$	0.0701	0.1333	0.1049	169	36	36
78	D246_OMC	5-5	$\Psi(0.5)$	0.0123	0.1607	0.3022	95	20	20
79	D246	5	$\Psi(0.8)$	0.0477	0.0916	0.1577	170	36	36
80	D246_OMC	10-10	$\Psi(0.8)$	0.0262	0.1711	0.1261	95	20	20

Table 5.1 Continued

Model	Input Variables*	Hidden Nodes	Target Variables	MSE			Sample Size		
				Trn	Val	Test	Trn	Val	Test
81	D246	5-5	$\Psi(0.2;0.5;0.8)$	0.0553	0.4236	0.1157	156	33	33
82	D246_OMC	5-5	$\Psi(0.2;0.5;0.8)$	0.1124	0.4014	0.2947	87	19	19
83	D246	10-10	θ_r	0.0015	0.0023	0.0057	170	36	36
84	D246_OMC	5-5	θ_r	0.0007	0.0044	0.0024	95	20	20
85	D246	5-5	DSR	0.0230	0.0555	0.2262	256	55	55
86	D246_OMC	40-40	DSR	0.0147	0.0309	0.0839	157	33	33
87	D246_OMC	40	INF	0.1118	0.2634	0.2355	157	33	33
88	D246	5	INF	0.3391	0.2316	0.2233	260	56	56

*Other than porosity

NSE ranging in between -2.24 and 0.66 for test set with 32 negative values, indicated that models can barely explain the variance of presented data, while 4 models resulted negative NSE factor which ranged in between 0.93 and -0.34 for validation set. Although resulting NSE, 0.93, indicates good matching as NSE factor lies in between $(-\infty, 1]$, Model 76 produces overestimation bias, PBIAS of validation and test sets were -0.85 and -11.20. This indicated that NN efficiently learns but cannot generalize well. This is also in agreement with the 52% difference in test MSE values of top first and the fifth models ranked according to mean squared test errors.

On the other hand, a negative NSE value indicates that model produces higher variance in errors than what existed in observations. This means even fitting a straight line passing through the mean of observations would produce less error (NSE=0). In order to investigate the effect of parameters, results have been grouped according to MSE-, NSE-, MRE-, PBIAS-based test performances and notable ones were pointed out and discussed below.

Table 5-2 Top 5 models according to NSE factor of test. PBIAS and NSE are for test sets.

Model	PBIAS	NSE	MSE		
			Training	Val*	Test
78	-11.20	0.65	0.0123	0.1607	0.3022
58	2.01	0.62	0.0001	0.1973	1.0878
63	-1.78	0.59	0.0554	0.1107	0.1229
36	-4.30	0.56	0.1928	0.3472	0.5788
29	-13.20	0.53	0.0461	0.0879	0.1714

*Validation.

Table 5-3 Top 5 models according to PBIAS of test set. PBIAS and NSE are for test sets.

Model	PBIAS	NSE	MSE		
			Training	Val*	Test
1	-0.52	0.19	0.0006	0.0032	0.0029
7	-0.44	-0.56	0.0582	0.1593	0.1521
10	0.23	-0.17	0.0410	0.3406	0.2279
8	0.3	0.25	0.2101	0.4178	0.5706
83	0.53	-0.47	0.0015	0.0023	0.0057

*Validation.

Results were split according to its target group and presented from Table 5-4 to Table 5-14 for single output models, and from Table 5-15 to Table 5-20 for group output models. The columns are in descending order in terms of R^2 values.

5.3 Comparisons for Individual Outputs

5.3.1 Residual Volumetric Water Content

Table 5-4 Metrics indicating the test errors of predictions of residual volumetric water content, θ_r .

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
21	TXT	0.0081	0.71	0.51	2.91	0.35	0.66
84	D246_OMC	0.0024	0.63	0.40	19.31	0.31	0.68
45	D136	0.0025	0.46	0.21	-25.24	0.08	1.16
83	D246	0.0057	0.44	0.20	-0.93	0.15	1.88
2	D1236_OMC	0.0041	0.25	0.06	5.92	-0.12	0.76
46	D136_OMC	0.0038	0.15	0.02	-16.34	-2.24	1.49
22	TXT_OMC	0.0141	0.04	0.00	14.36	-0.70	1.07
1	D1236	0.0029	-0.13	0.00	0.44	-0.56	2.12

Table 5-5 MAEs and NSEs of the models targeting residual volumetric water content, θ_r , arranged in ascending order of mean absolute test error.

Model	84	22	1	21	45	46	83	2
MAE	0.668	0.682	0.686	0.725	0.797	0.808	0.946	1.189
NSE	0.31	-0.70	-0.56	0.35	0.08	-2.24	0.15	-0.12

Existence of small valued observations close to zero results in unduly high percentage errors, e.g., same absolute error may appear to be less problematic for Fredlund and Xing (1994) model than residual water content models. Therefore, mean absolute deviation was used rather than a percentage-based measure or a ratio scale in interpretation of the models embodying residual water content. High errors

in residual volumetric water content was a result of poorly defined residual conditions due to reasons stated in subsequent chapter. Yet, exact effect of model bias is not independent from the variance in targets, which is accounted by only the NSE factor amongst these metrics. Therefore, from Table 5-5, although Model 22 and Model 1 had smaller mean of absolute residuals, Model 21 and Model 84 shall be considered as the best two networks of this run.

On the other hand, the inconsistency between MAE and R^2 is due to the fact of R^2 being more beneficial in interpretation of multiple linear regression models. It explains the goodness of fit between outputs and targets which does not necessarily translated back to closeness of outputs and targets. This also manifested itself in Figure 5.16 through Figure 5.29 as Fredlund and Xing (1994) model (Model 70) captured the shape of the curve but missed the position whereas models having lower R^2 values (Model 63, point-wise suction predictions) captured both in more cases.

5.3.2 Desaturation Rate

All the metric except for PBIAS in agrees that a detailed textural grouping rather than a rough grouping gave closer DSR predictions to observations, as shown in Table 5-6. PBIAS of Model 47 indicates that predictions have lower tendency to deviate from observations, whereas Model 23 exhibited an underestimation bias.

Table 5-6 Metrics indicating the test errors of slope of desaturation line (DSR) predictions.

Model	Input	MSE	R	R^2	PBIAS	NSE	MAPE
23	TXT	0.05	0.73	0.54	4.50	0.53	0.61
47	D136	0.08	0.51	0.26	0.52	0.19	0.61
24	TXT_OMC	0.03	0.50	0.25	18.10	0.20	0.48
4	D1236_OMC	0.19	0.43	0.19	12.36	-0.05	0.44
48	D136_OMC	0.25	0.40	0.16	37.32	-0.18	0.38
86	D246_OMC	0.08	0.29	0.09	39.35	-0.68	0.58
85	D246	0.23	0.28	0.08	-2.72	0.08	0.45
3	D1236	0.11	0.26	0.07	2.79	0.02	0.68

Model 86 instigated a deeper investigation on hidden layer-neuron-performance relationship for its leapt complexity - lowered accuracy, as highlighted in Table 5-7. For further examination, number of hidden layers and neurons was differentiated over fixed input variables: D246, n, OMC, and target variable: slope of desaturation line. Table 5-7, unveils complexity, topology, and performance of models in terms of R^2 . Except for Model 86.2, each deep network in Table 5-7 worked better than Model 86, much of which was due to use of deep networks. Figure 5.7, shows that test and training data followed different distributions. Non-homogeneous division of data also contributes and explains the difference in R values of training and testing in Figure 5.7 (where use of k-fold cross validation would reduce this problem, as stated in future works). Detailed examination is explained in Section 5.6.

Table 5-7 Number of hidden neurons and R^2 values for test sets of DSR models.

Model	Inputs	Hidden Neurons	R^2_{test}
23	TXT	20	0.54
47	D136	10	0.26
24	TXT_OMC	5	0.25
4	D1236_OMC	5	0.19
48	D136_OMC	10	0.16
86	D246_OMC	40-40	0.09
85	D246	5-5	0.08
3	D1236	5-5	0.07

5.3.3 Inflection Point Location

Table 5-8 summarizes the resulting metrics for estimates of inflection point location.

Table 5-8 Metrics indicating the test errors of inflection point (INF) predictions.

Model	Input	MSE	R	R^2	PBIAS	NSE	MAPE
49	D136_OMC	0.20	0.87	0.75	6.37	0.39	0.13
88	D246	0.22	0.83	0.68	-1.23	0.48	0.19
6	D1236	0.16	0.79	0.63	0.86	0.36	0.15
87	D246_OMC	0.24	0.79	0.62	2.49	0.26	0.18
25	TXT	0.30	0.72	0.52	4.43	0.18	0.28
50	D136	0.08	0.64	0.41	1.36	-0.32	0.18
26	TXT_OMC	0.47	0.64	0.40	1.68	-0.58	0.21
5	D1236_OMC	0.88	0.60	0.36	11.27	-0.37	0.20

The only remarkable thing in estimates of location of inflection points was the need for relatively higher complexity levels. Table 5-9 and Table 5-10 summarizes the related information on inflection point models.

Table 5-9 Number of hidden neurons, sample size, and the test errors in terms of R^2 and NSE of inflection point (INF) predictions.

Model	Input	Hidden Neuron	R^2	NSE	Train	Val*	Test
49	D136_OMC	20-20	0.75	0.39	76	16	16
88	D246	5	0.68	0.48	260	56	56
6	D1236	20	0.63	0.36	147	31	31
87	D246_OMC	40	0.62	0.26	157	33	33
25	TXT	20-20	0.52	0.18	403	86	86
50	D136	10-10	0.41	-0.32	147	31	31
26	TXT_OMC	5-5	0.40	-0.58	236	50	50
5	D1236_OMC	20-20	0.36	-0.37	76	16	16

*Validation.

Table 5-10 Number of: nodes in each layer, training equations, weights and training data points for Model 49.

#	Input	Output	H1	H2	Eqn*	W**	Train***
Model 49	5	2	20	20	152	582	76

*Training equations. **Weights. ***Available training sample size

Above parameters in Table 5-10 are calculated according to

$$N_{trn} = N_{sample} - 2 \times (0.15 \times N_{sample})$$

$$N_{eqn} = N_{trn} \times O$$

$$N_w = (I + 1) \times H_1 + \dots + (H_i + 1) \times H_{i+1} + \dots + (H_n + 1) \times O$$

where $Neqn \gg Nw$ should be provided for a robust design, and:

N_{trn} : Number of training samples,

N_{sample} : Number of total samples,

N_{eqn} : Number of training equations,

Nw : Number of unknown parameters, i.e., weights,

I : Number of neurons in input layer,

O : Number of neurons in output layer,

$\{H\}_{i=1}^{n-1}$: Number of nodes in the i th hidden layer.

Then, the criterion yields that training sample size should be at least around 4 times larger than the available one shown in Table 5-10 (preferably several times more than this for a more stabilized design).

5.3.4 Suctions at AEV, $\theta_e=0.2$, $\theta_e=0.5$, $\theta_e=0.8$

Following tables from Table 5-11 to Table 5-14 tabulates the resulting metrics of the relevant target parameters.

Table 5-11 Metrics indicating the test errors of air-entry value (AEV) predictions.

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
51	D1236	0.08	0.74	0.54	19.94	0.44	0.70
52	D1236_OMC	0.37	0.54	0.29	-14.12	0.20	1.27
27	TXT	0.36	0.52	0.27	8.86	0.20	27.28
71	D246	0.21	0.45	0.21	-22.17	0.14	2.38
7	D136	0.15	0.38	0.15	-0.23	-0.17	0.82
72	D246_OMC	0.31	0.38	0.14	-16.54	0.02	0.72
28	TXT_OMC	0.69	0.24	0.06	-3.82	-0.08	1.13
8	D136_OMC	0.57	0.01	0.00	-0.53	-0.47	1.50

Table 5-12 Metrics indicating the test errors of predictions of suction value at effective volumetric water content of 0.2, $\psi(0.2)$.

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
58	D1236_OMC	1.09	0.87	0.76	-2.01	0.62	0.17
75	D246	0.38	0.78	0.60	11.42	0.13	0.66
14	D136_OMC	0.68	0.68	0.47	6.89	0.40	0.21
34	TXT_OMC	1.62	0.49	0.24	-10.03	-0.12	0.35
33	TXT	0.65	0.46	0.21	-5.12	-0.35	0.40
57	D1236	0.73	0.42	0.18	-3.00	0.16	0.42
13	D136	0.25	0.33	0.11	14.74	-0.54	0.49
76	D246_OMC	0.17	0.31	0.10	-8.00	-0.37	0.40

Table 5-13 Metrics indicating the test errors of predictions of suction value at effective volumetric water content of 0.5, $\psi(0.5)$.

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
78	D246_OMC	0.30	0.85	0.72	11.20	0.65	0.17
16	D136_OMC	0.10	0.77	0.60	18.41	-0.02	0.18
36	TXT_OMC	0.58	0.76	0.58	4.30	0.56	0.27
15	D136	0.08	0.75	0.56	23.00	0.42	0.33
77	D246	0.10	0.71	0.51	2.36	0.45	0.34
35	TXT	0.28	0.64	0.41	-2.24	0.35	0.31
60	D1236_OMC	0.46	0.56	0.32	-12.85	0.20	0.41
59	D1236	0.07	0.08	0.01	-8.91	-0.37	0.36

Accounting for soil organic matter content worked better for each predictor group particularly for D1236 by 0.31 increase in R^2 in Table 5-13.

Table 5-14 Metrics indicating the test errors of predictions of suction value at effective volumetric water content of 0.8, $\psi(0.8)$.

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
17	D136	0.08	0.66	0.43	1.20	0.43	0.52
79	D246	0.16	0.58	0.34	12.84	0.28	0.93
18	D136_OMC	0.15	0.56	0.31	-8.93	0.27	1.83
38	TXT_OMC	0.27	0.50	0.25	-6.94	-0.10	0.57
37	TXT	0.18	0.47	0.22	17.43	0.12	0.88
61	D1236	0.05	0.41	0.17	1.56	0.17	1.40
62	D1236_OMC	0.31	0.36	0.13	-32.80	-1.15	0.59
80	D246_OMC	0.13	0.20	0.04	-5.82	-0.30	1.02

Use of effective volumetric water content form appeared to be more reliable for the capillary saturation and desaturation zones (low to intermediate suction regions) than high suction region (Table 5-12 to Table 5-14). This conclusion was also supported by findings of Sillers (2001).

The effect of OMC depends on the proportional distribution of textural components as well as the amount and composition of OMC (Rawls et al., 2003). Thus, for such a diverse data set, any conclusion should be drawn in caution as its effect may not be divulged straightforwardly. However, OMC had particular help in estimation of $\psi(0.2)$ (Model 58) and $\psi(0.5)$ (Model 78), but not $\psi(0.8)$ (Model 17) (Table 5-12 to Table 5-14). The soil organic matter content having influence on both soil structure (a high PH value may make the PH dependent edge-charges tend to be negative and break down the flocculated structure to form a dispersed one such as kaolinite particles which also involved in part of UNSODA clays, being negatively charged at high PH values to provide proton equilibrium) and adsorption properties (particularly for sandy soils as potential hydrophobic organics, plant exudates produced by plant roots to enhance the nutrient availability and defend against desiccation stresses etc. surrender the particles and form a hydrophobic coating when water content falls below a critical threshold where such a coating would impact a larger proportion of particles of sandy soils due to lower surface area) which govern the high suction segment of WRC, may bring an explanation on those results.

5.4 Group Outputs

Following tables from Table 5-15 to Table 5-20 summarizes group output models' performances. Estimates obtained by Fredlund and Xing (1994) fitting parameters have the additional error stemmed from curve fitting. Table 5-17 compares the estimated water contents and measured data points for test set samples of each model.

Table 5-15 Metrics indicating the test errors of predictions of suction values at effective volumetric water content of 0.2, 0.5, and 0.8 ($\psi(0.2;0.5;0.8)$).

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
63	D1236	0.12	0.85	0.72	10.49	0.50	0.15
82	D246_OMC	0.29	0.84	0.70	0.05	0.43	0.31
81	D246	0.12	0.81	0.65	-0.79	0.42	0.16
20	D136_OMC	0.43	0.79	0.63	2.99	0.04	0.05
39	TXT	0.29	0.79	0.63	3.89	0.40	0.18
64	D1236_OMC	0.13	0.76	0.57	-2.38	-1.85	0.07
19	D136	0.18	0.73	0.54	21.22	-0.04	0.18
40	TXT_OMC	0.20	0.68	0.46	-7.59	0.00	0.18

Table 5-16 Metrics indicating the test errors of air-entry value, slope of desaturation line, and residual volumetric water content group (AEV-DSR- θ_r) predictions.

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
11	D136_OMC	0.39	0.84	0.71	20.55	0.22	0.20
12	D136	0.20	0.82	0.68	9.46	-0.05	0.39
56	D1236	0.64	0.76	0.59	10.40	0.04	1.68
55	D1236_OMC	0.29	0.74	0.54	-55.69	-0.85	1.63
31	TXT_OMC	0.14	0.73	0.54	3.95	-0.69	0.33
74	D246	0.22	0.70	0.49	9.68	0.02	2.38
32	TXT	0.18	0.60	0.36	21.45	-0.15	0.46
73	D246_OMC	0.56	0.57	0.32	-13.52	-0.43	0.49

Table 5-17 Metrics indicating the test errors of air-entry value, slope of desaturation line, and volumetric water content at inflection point (AEV-DSR-INF) predictions.

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
10	D136_OMC	0.23	0.89	0.80	4.55	0.39	0.15
54	D1236_OMC	0.22	0.87	0.76	-2.42	-0.11	0.65
9	D136	0.12	0.87	0.76	-0.63	0.26	0.20
68	D246_OMC	0.09	0.86	0.74	14.22	0.27	0.14
67	D246	0.06	0.86	0.73	-2.94	0.38	0.26
29	TXT	0.17	0.84	0.70	10.12	0.30	0.20
53	D1236	0.07	0.81	0.65	3.76	-0.49	0.21
30	TXT_OMC	0.18	0.73	0.53	-1.89	0.07	0.41

Table 5-18 Metrics indicating the test errors of predictions of Fredlund and Xing (1994) equation (FX) parameters, a, m, n, θ_r .

Model	Input	MSE	R	R ²	PBIAS	NSE	MAPE
70	D246_OMC	4.64	0.92	0.85	-8.58	0.27	0.19
65	D1236	8.44	0.75	0.56	-4.81	-0.45	0.18
66	D1236_OMC	27.51	0.74	0.55	5.14	-0.47	0.11
43	D136	8.12	0.73	0.53	-3.30	0.22	0.19
44	D136_OMC	8.23	0.71	0.51	4.96	-0.02	0.18
69	D246	25.26	0.71	0.50	5.23	0.06	0.16
41	TXT	7.55	0.68	0.46	0.32	0.08	0.21
42	TXT_OMC	11.30	0.47	0.22	-17.28	-1.73	0.28

Table 5-19 Relation between the water contents estimated from test set of Fredlund and Xing (1994) fitting parameters and the experimental data, and MADs.

Model	Input	R	R ²	MAD
44	D136_OMC	0.92	0.85	0.03481
43	D136	0.88	0.77	0.06294
65	D1236	0.87	0.75	0.04223
69	D246	0.86	0.73	0.06216
42	TXT_OMC	0.86	0.73	0.04699
70	D246_OMC	0.79	0.63	0.08940
41	TXT	0.78	0.60	0.06322
66	D1236_OMC	0.75	0.56	0.10362

Table 5-20 The best models from 11 target variable ranked according to R² results.

Model	Input	H*	Target	Trn - Val - Test**	R ²
70	D246_OMC	10-10	FX	77-17-17	0.85
10	D136_OMC	5	AEV-DSR-INF	76-16-16	0.80
58	D1236_OMC	10-10	$\Psi(0.2)$	56-12-12	0.76
49	D136_OMC	20-20	INF	76-16-16	0.75
63	D1236	5-5	$\Psi(0.2;0.5;0.8)$	107-23-23	0.72
78	D246_OMC	5-5	$\Psi(0.5)$	95-20-20	0.72
11	D136_OMC	10	AEV-DSR- θ_r	58-12-12	0.71
51	D1236	10-10	AEV	147-31-31	0.54
23	TXT	20	DSR	399-85-85	0.54
21	TXT	10-10	θ_r	229-49-49	0.51
17	D136	5	$\Psi(0.8)$	110-24-24	0.43

*Number of hidden neurons. **Train, validation, test set sample size.

Table 5-20 tabulates and ranks the best performed designs extracted from each target group. 5% ahead of its nearest competitor, FX achieved the highest R^2 value.

Single output models (Model 58, 49, 78; Model 51, 23, 21) worked parallel but should be considered along with their complementary output models as they are meaningless alone. Then, it is concluded that grouped outputs outperformed the single ones (Model 10 had greater R^2 (=0.80) by 20% from the average of single output models 58, 78, 17 (=0.64).

The worst performance models were Model 11 amongst group output models, and Model 17 amongst single output models. The common ground between these models may be the ill-defined parameters due to (i) effective water content form gives no information about volume change which produced erroneous targets at 0.8 axis, and (ii) curves which had no clear location of AEV nor residual conditions.

5.5 Effect of Variable Transformation

After presenting the results, part of the approaches applied for accuracy enhancement are exemplified below from Figure 5.1 to Figure 5.6 for Fredlund and Xing (1994) model, and results are tabulated in Table 5-21. All the target variables, a, m, n , and θ_r were standardized to zero mean and unity variance before presented to NN. However, R^2 values were bound to be lower due to lumping around the origin (Figure 5.1, Table 5-8). To overcome this, logarithm and Box-Cox transformations were applied on target features. Box-Cox transformation converts a distribution to normal distribution by maximizing the log-likelihood function (for each data point, calculating the probability to come from the normal distribution given some mean and variance values, then translating the product of densities into a sum) with respect to the transformation parameter, λ , to obtain the normal distribution parameters: mean and variance. If λ is not zero, then

$$x(\lambda) = \frac{x^\lambda - 1}{\lambda}$$

where x is the data transformed using Box-Cox. If λ is zero, data will be simply log-transformed on base e.

The two transformations (Box-Cox and logarithm on base-10) were applied on (i) parameter a only, and (ii) all four target variables of Fredlund and Xing (1994) model, a, m, n , and θ_r . The first argument did not do much for clumping or generalization for both of the transformations (Figure 5.2, Figure 5.4) which was reflected in results tabulated in Table 5.21, but the latter approach resulted in a 23% increase in R^2 under fixed topology. Regression plots from Figure 5.1 to Figure 5.6 showed that data were spread and fell on the regression line better when log-transformed, meaning increased association between targets and outputs as depicted in Table 5-21.

Table 5-21 Applied transformations on fitting parameters of Fredlund and Xing (1994) model (either on a , or a, m, n , and θ_r all together), corresponding correlation values, R , between outputs and targets of train, validation, and test sets, and coefficient of determination of test set.

Transformation		R-Train	R-Validation	R-Test	R^2_{test}
Base-10 Logarithm	a only	0.95	0.86	0.76	0.58
	a, m, n , and θ_r	0.97	0.97	0.92	0.85
Box-Cox	a only	0.94	0.87	0.76	0.58
	a, m, n , and θ_r	0.98	0.98	0.91	0.83
Standardization only		0.90	0.88	0.79	0.62
Without standardization		0.86	0.86	0.71	0.50

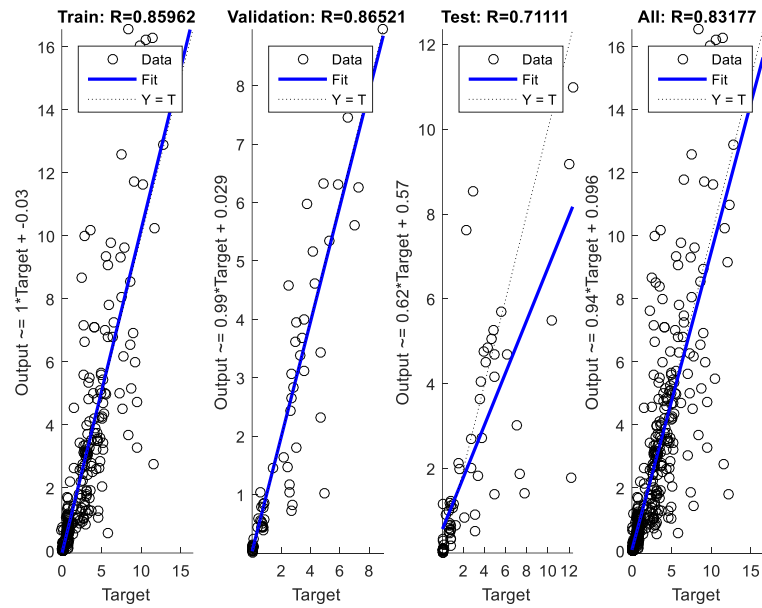


Figure 5.1. The resulting regression plots without standardization.

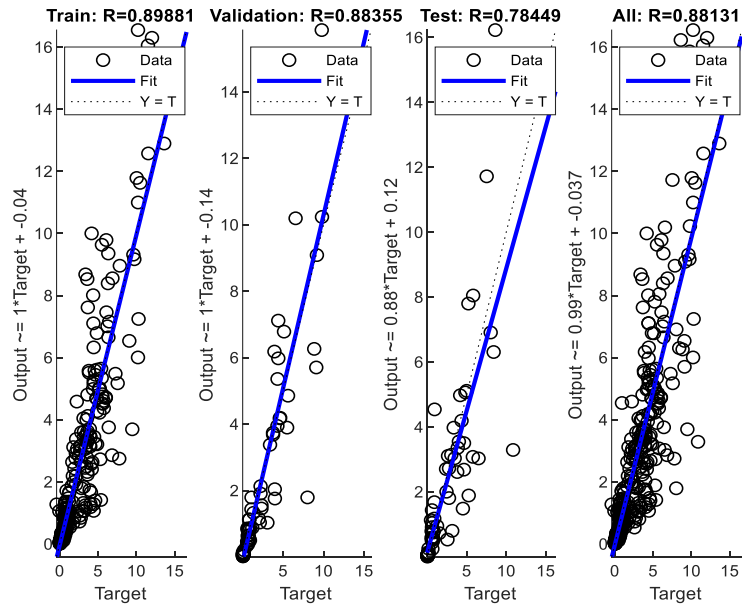


Figure 5.2. The resulting regression plots when targets were subjected only to standardization.

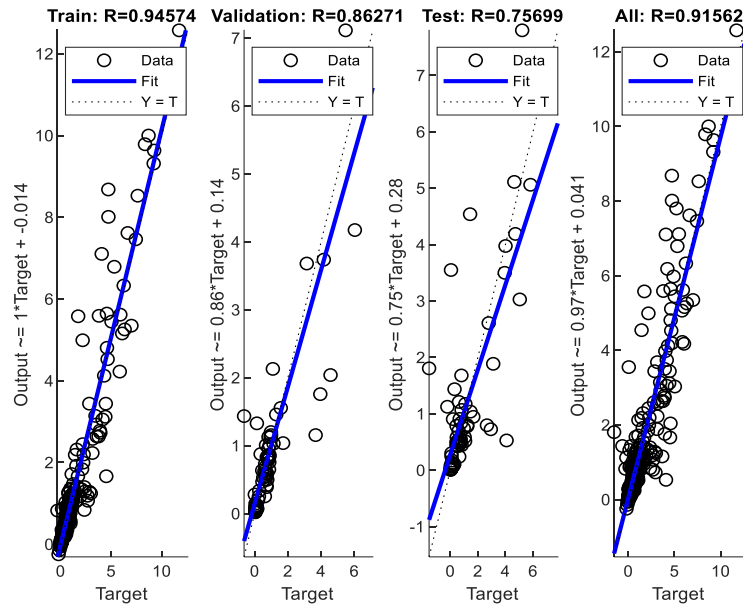


Figure 5.3. The resulting regression plots when log-translation was applied on parameter a , only.

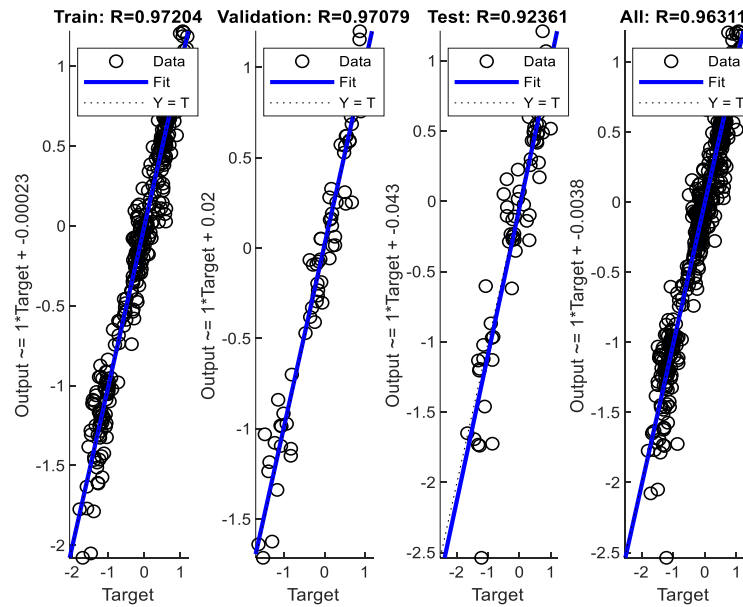


Figure 5.4. The resulting regression plots when all the target features were log-transformed.

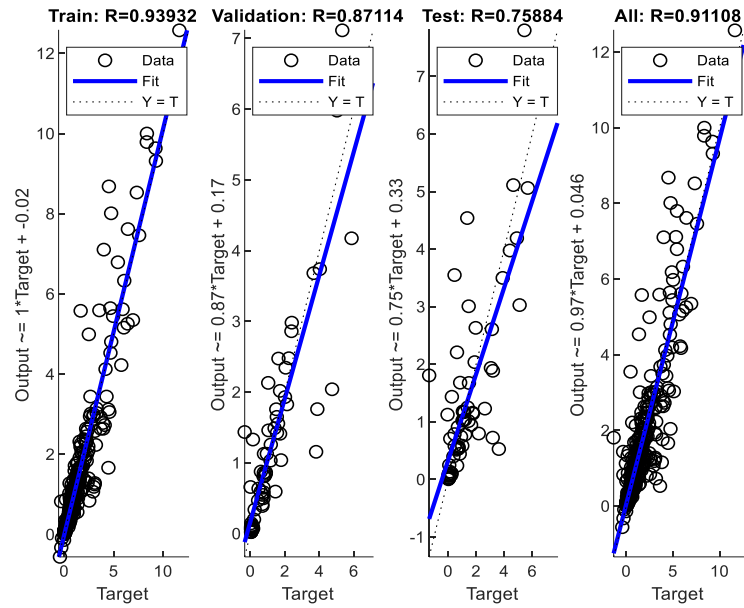


Figure 5.5. The resulting regression plots when parameter a was Box-Cox transformed.

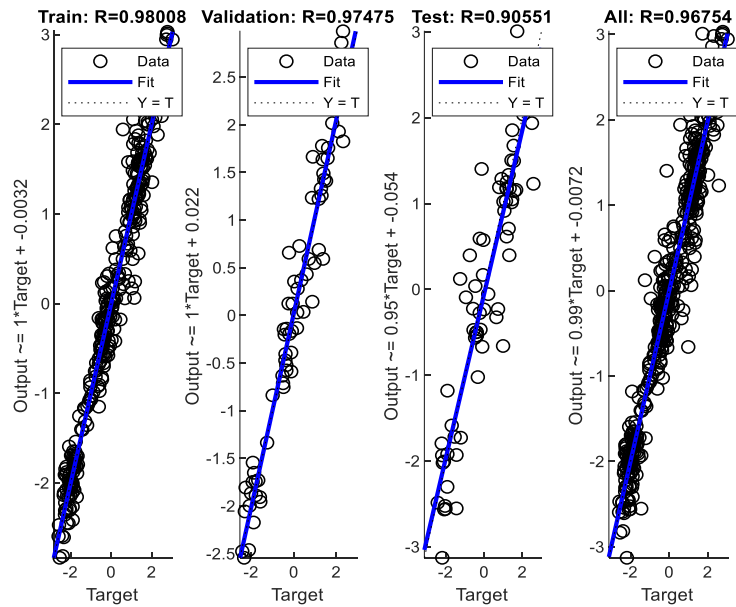


Figure 5.6. The resulting regression plots when all the targets were Box-Cox transformed.

5.6 Effect of Complexity

Increasing the number of hidden nodes (complexity) results in high variations of predictions from their expected due to overfitting, i.e., erratic, unstable networks that do not generalize well. The disparity between: (i) training and testing R values (shown in Figure 5.7), (ii) complexity and R^2 (shown in Table 5-22) clarified this as well. On the other hand, adding more hidden layers may (and did, except for Model 86.2) no harm to generalization capability and does not cause overfitting (since each layer is defined by its previous layer, each layer learns based on the previous one, preventing gradients to vanish before reaching the final layer rather than increasing the hidden nodes (complexity) of a shallow network.

Table 5-22 Differentiated topologies over Model 86 and corresponding R^2 values.

First row belongs to Model 86.

Model	Topology	Complexity	R^2_{test}
86*	5-40-40-1	1921	0.09
86.1	5-10-10-10-1	291	0.61
86.2	5-10-10-10-10-1	401	0.08
86.3	5-20-20-20-1	981	0.24
86.4	5-20-20-20-20-1	1401	0.56
86.5	5-10-10-10-10-10-1	511	0.12
86.6	5-20-20-20-20-20-1	1821	0.35

*Model 86 from Table 5.1

Model outputs and targets were transmitted onto the representative regression plots along with R values corresponding to train, validation, test, and all data, as depicted in Figure 5.7 through Figure 5.12. Regression plots indicate the deviations of network responses from the targets. Each circle represents output data. Blue line

represents the line fitted to outputs, or the deviation of outputs from the targets, dashed line denotes outputs are exactly equal to targets. The equations labeled in y-axis explains how much variation in outputs are explained by targets for multiple linear regression, the first term being the proportionality between them and the second is the bias that is to be added to scaled targets to make output and targets closer.

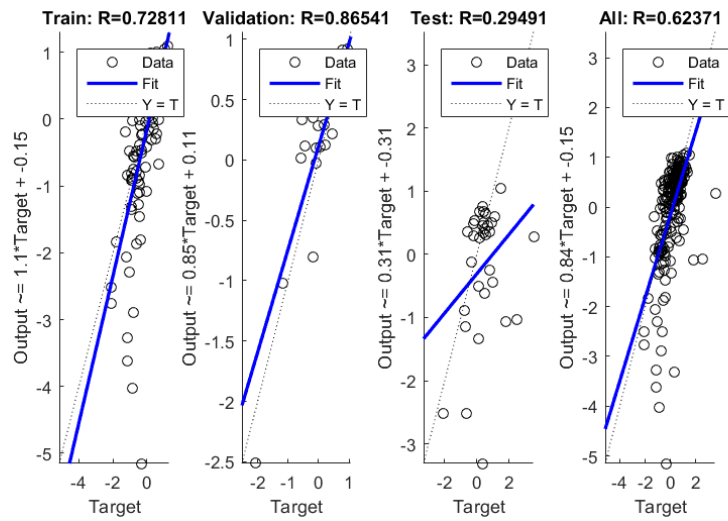


Figure 5.7. Regression plot for Model 86.

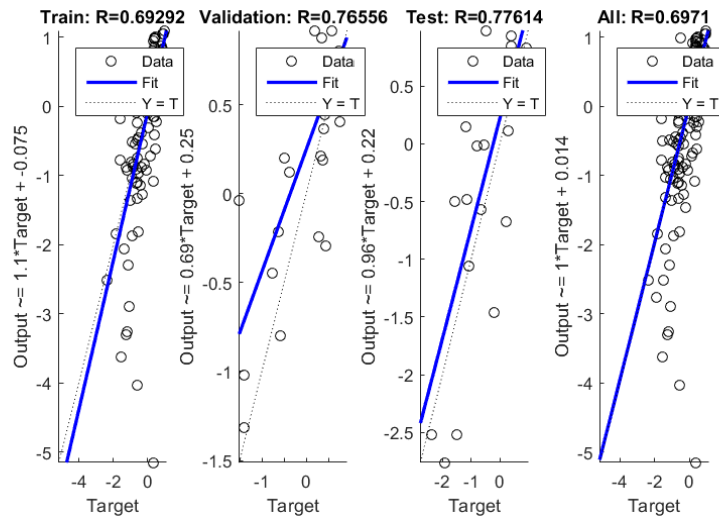


Figure 5.8. Regression plot for Model 86.1.

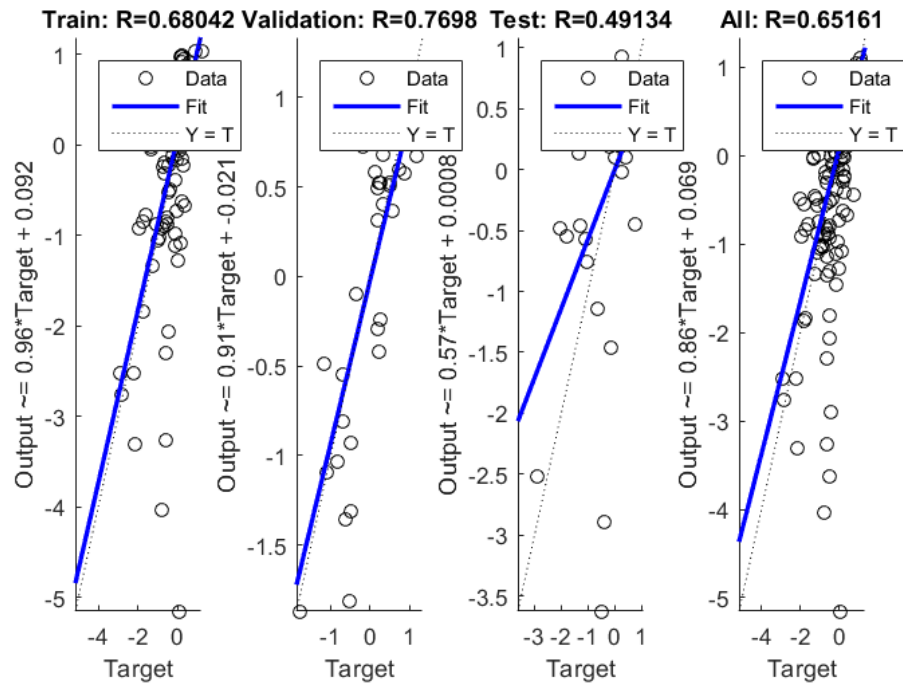


Figure 5.9. Regression plot for Model 86.3.

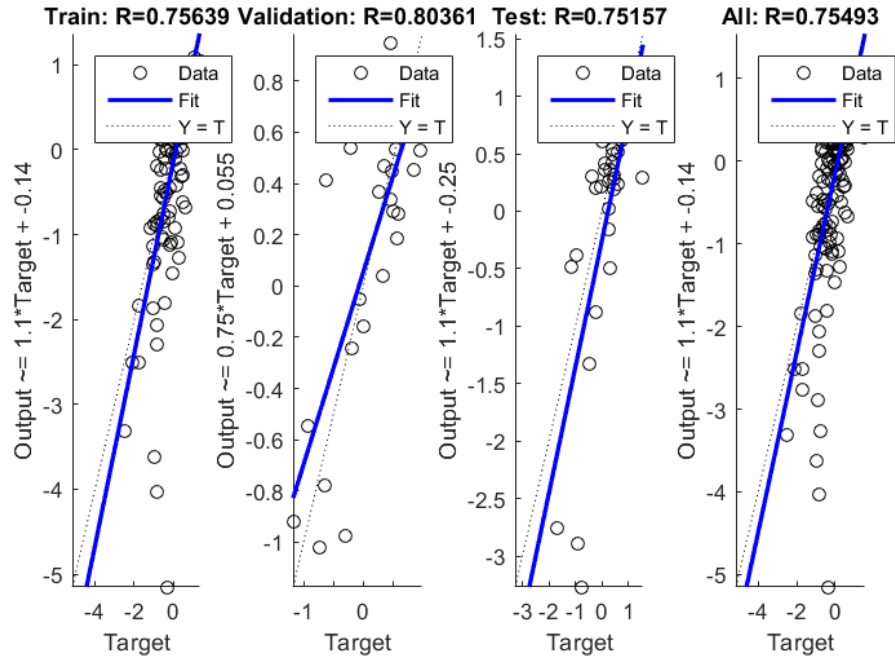


Figure 5.10. Regression plot for Model 86.4.

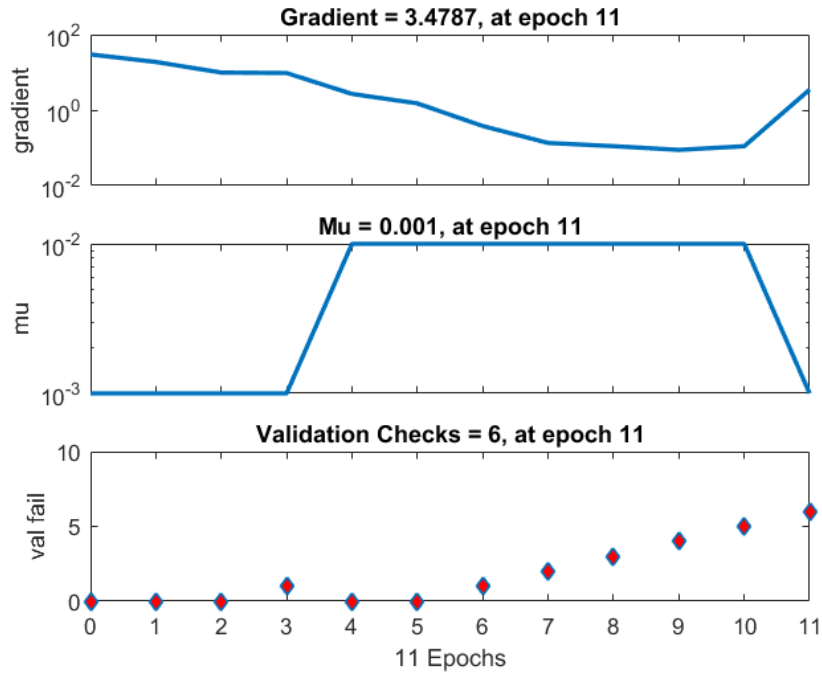


Figure 5.11. Training state of Model 86.1.

Figure 5.12 refers no significant problem, test and validation curves are similar. Training has stopped at fifth iteration (or epoch in this case, since the whole data were fed at each pass, i.e., batch size and sample size were equal) at which the validation error yielded its minimum. Figure 5.11. denotes the progress of training with six validation check errors which explains why training had stopped after six consecutive validation failures, at epoch 11. Eventually, for the sake of simplicity, emphasis was on the fixed topologies as other literature did.

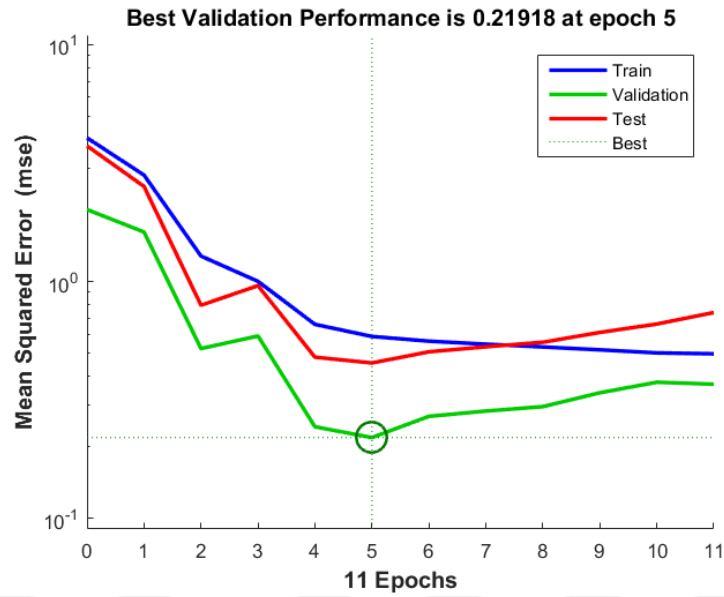


Figure 5.12. Performance of Model 86.1.

Figure 5.13 shows that test error had started to increase before validation error reached its minimum. If dissimilarity between test and validation curves were significant, that would be an indicator of overfitting.

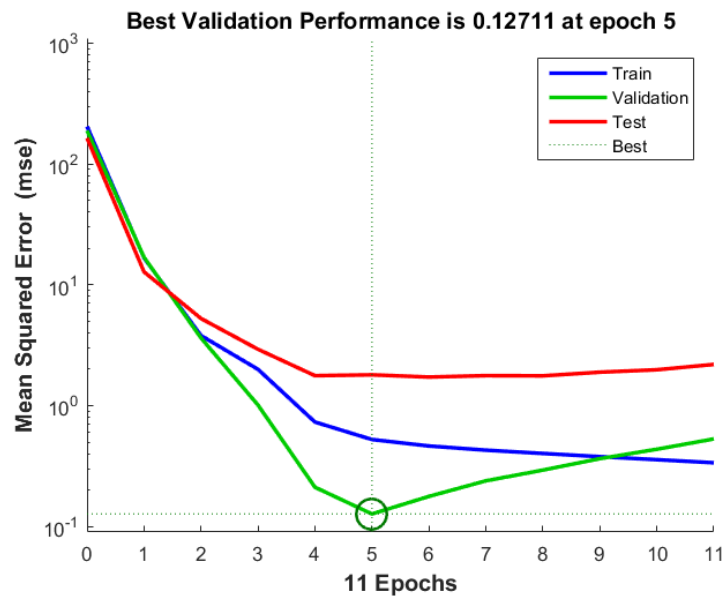


Figure 5.13. Performance of Model 86.

5.7 Effect of Dimensionality

Apart from these, the effect of dimensionality was put into question by applying a series of principal component analysis on the models and exemplified on Fredlund and Xing (1994) model in following figures and tables. PCA decomposes the variation in data to orthogonal principal directions in a way that explained variation of projections is maximized. These directions, basically being the eigenvectors of sample covariance matrix (columns of U from the SVD of the data) were selected according to their corresponding eigenvalues, since eigenvalues denote the amount of variance accounted for by a principal component. Eigenvalues are put in descending order and explained variances by each principal component are tabulated in Table 5-23. Accordingly, selecting the top 2 principal components seemed enough for dimension reduction while preserving the data structure.

Table 5-23 Eigenvalues and explained variances by each principal component.

PC	Eigenvalue	Explained (%)	Σ Explained (%)
1	9.2488	67.76	67.76
2	4.3584	31.93	99.69
3	0.0417	0.31	99.99
4	0.0007	0.01	100

26 data points that fell out of region (4) were flagged as outlier according to orthogonal and score distances out of PCA analysis and eliminated. Visual inspection of these points can be done in Figure 5.14. Observations fell in: region (1) are marked as good PCA-leverage points, region (2) are bad PCA-leverage points, region (3) as orthogonal outliers, and region (4) as regular points. Outlyingness of an observation at 0.05 significance level was tested according to the cutoff lines drawn at $\sqrt{\chi_{p,0.975}^2}$ was exceeded or not. Cutoff values marked by red line are: 2.72 and 0.4856, for score distances within the principal subspace and for orthogonal distances to the principal subspace, respectively.

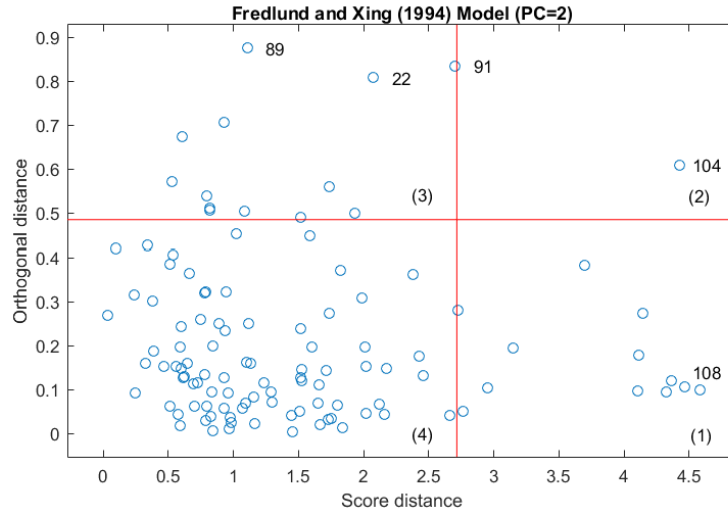


Figure 5.14. Diagnostic plot for two principal components.

Samples 104 and 91 in Figure 5.14 were identified as bad leverage points, i.e., they obey the data structure but are extreme valued. Observations 91, 22, 89 were assigned as orthogonal outliers and 108 was marked as good leverage point. Once the data were cleansed from contaminated samples, PCA analysis were conducted on MATLAB library LIBRA at which robust analysis were provided that accounts for the masking, swamping effects, data structure, anomalies regarding to dispersion etc. Further details are provided by Hubert et al. (2005).

The scores out of PCA analysis were used as new targets, which reduced the dimensionality from 4 to 2. However, applying a classical PCA did not improve the results nor did the robust PCA (Figure 5.15) with eliminated outliers based on (i) analysis robust to skewness in data, (ii) distance based analysis explained above.

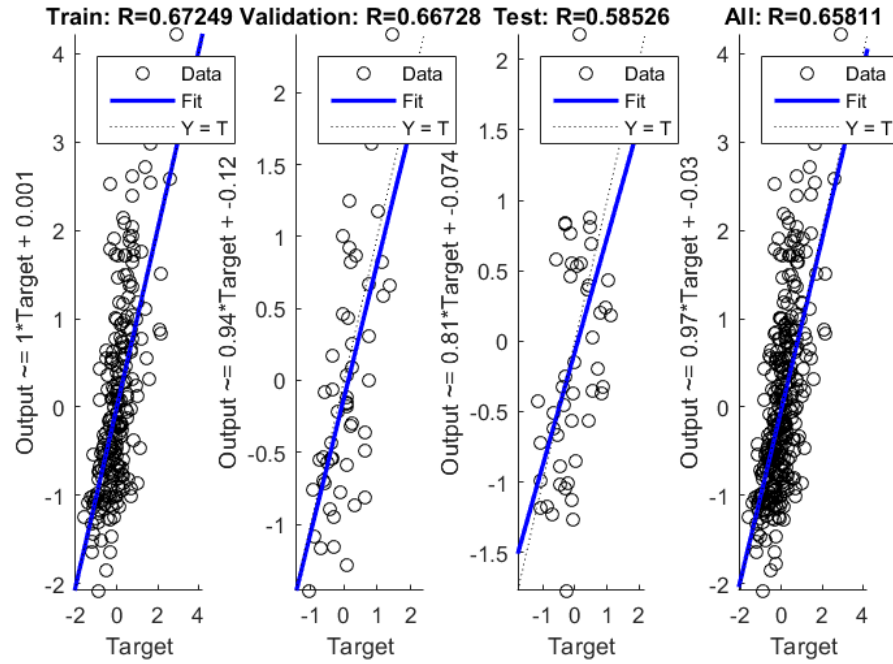


Figure 5.15. Resulting regression plots after robust PCA analysis.

This was expected because unsupervised PCA captures the directions of maximum variation, not dependence. However, preserving the information regarding to targets on those projections is much meaningful in this case. Therefore, supervised analysis should be conducted to have the subspace of maximum dependence. Various class of algorithms exist in literature to go into lower dimensions in a supervised manner: Fisher's discriminant analysis (linear dependence), Kernel Fisher discriminant analysis (linear dependence but nonlinear data is handled by going to higher dimensional space, obtaining a linear decision boundary, and apply the whatever linear techniques), metric learning methods, sufficient dimension reduction algorithms, Bair's supervised principal components, Hilbert-Schmidt component analysis. Further details can be found in review made by Ghogh et al. (2019).

Even then, the algorithm should maximize the nonlinear dependency between input and targets, not linear: projecting the data onto not a subspace anymore, but a submanifold. However, as nonlinear supervised PCA was not readily available, no dimension reduction was applied for any of the modeled networks.

5.8 Example Plots

The final part of this chapter demonstrates the WRC plots from Figure 5.16 to Figure 5.29. Water retention curves were plotted for independent data from SoilVision database against the simulated outputs from models which presented in . Networks' response to inputs provided by SoilVision database were calculated. Since OMC data was missing in SoilVision, for those models which contain predictor OMC, it was assumed as 1.52%, the average of OMC data in UNSODA. It should be noted that these samples have wider GSD range that modeled networks had never seen before.

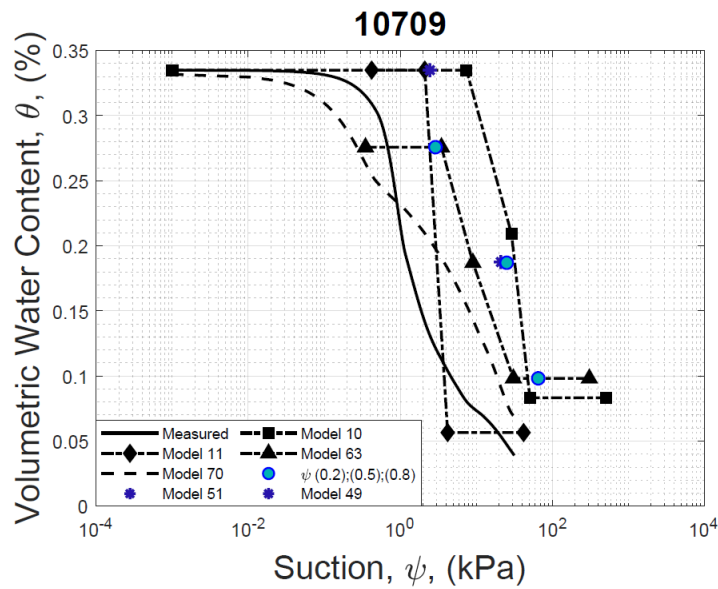


Figure 5.16. Comparison of estimated and measured WRCs of Sample 10709.

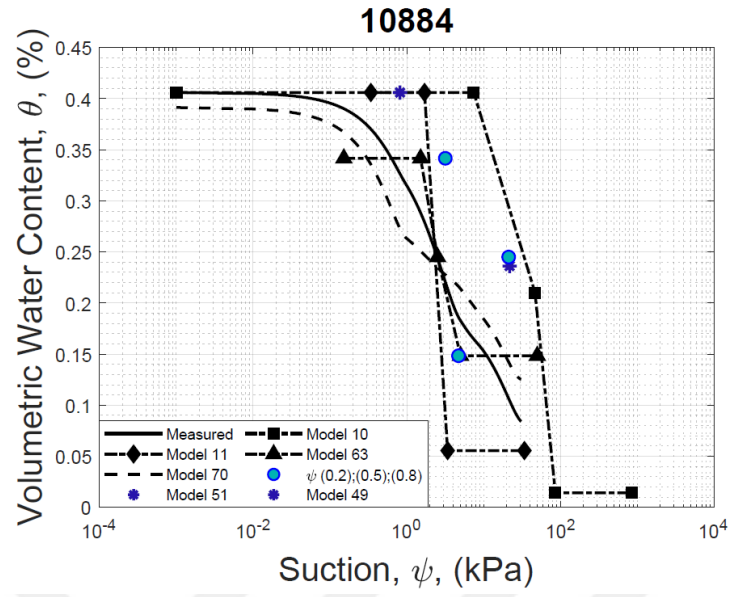


Figure 5.17. Comparison of estimated and measured WRCs of Sample 10884.

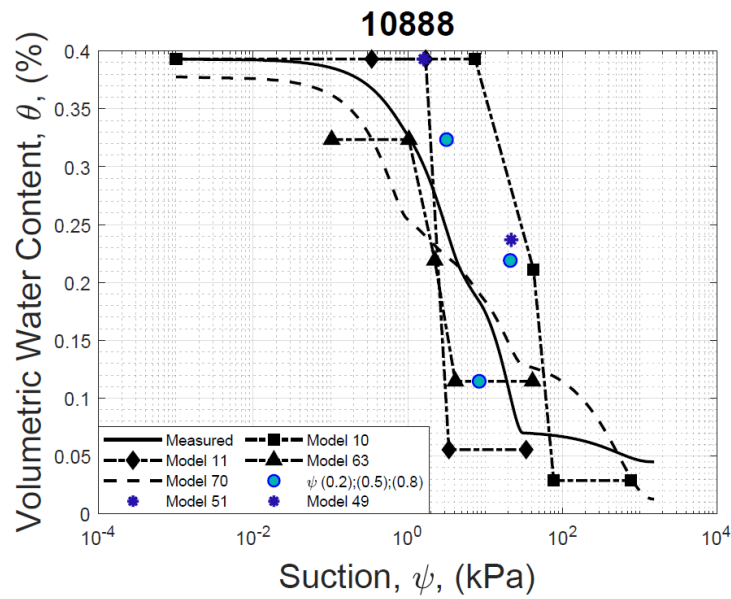


Figure 5.18. Comparison of estimated and measured WRCs of Sample 10888.

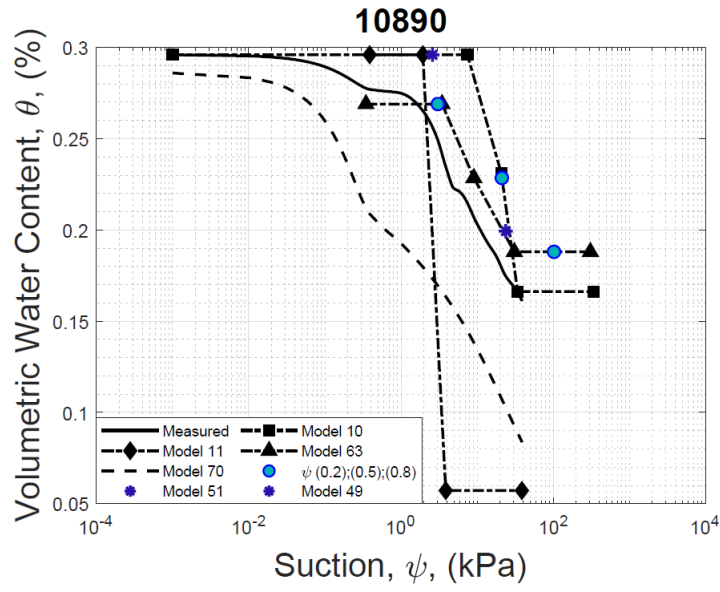


Figure 5.19. Comparison of estimated and measured WRCs of Sample 10890.

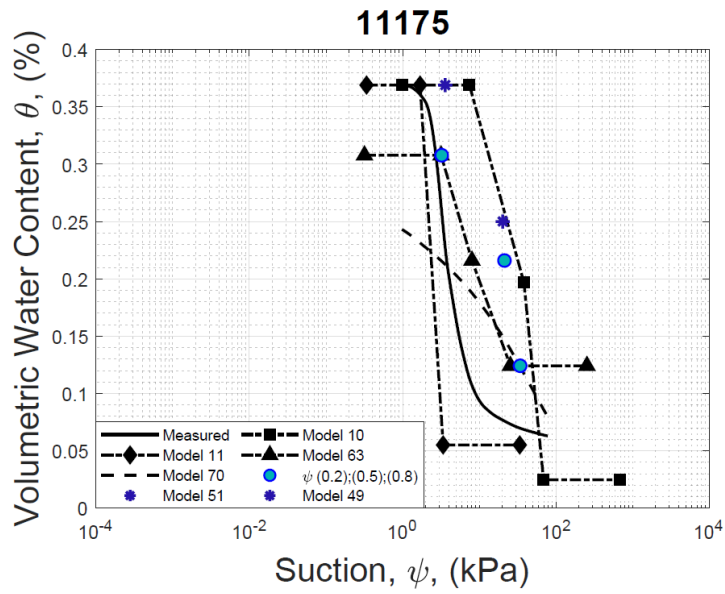


Figure 5.20. Comparison of estimated and measured WRCs of Sample 11175.

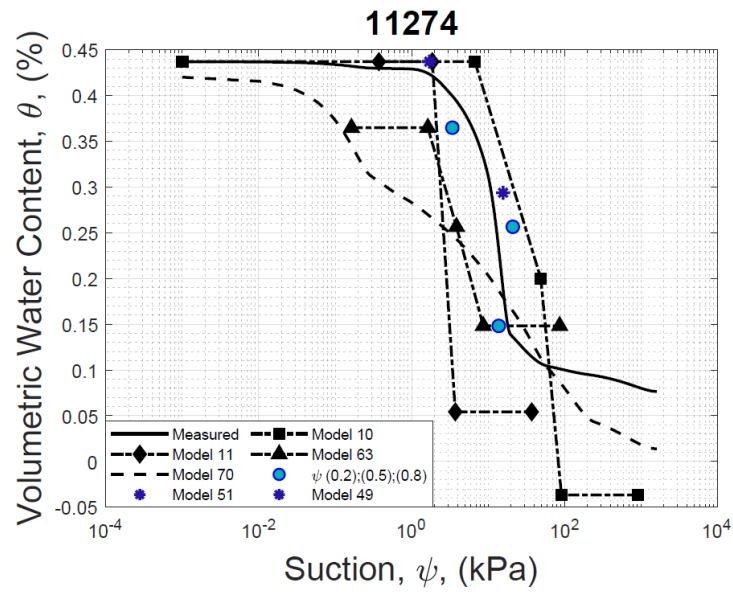


Figure 5.21. Comparison of estimated and measured WRCs of Sample 11274.

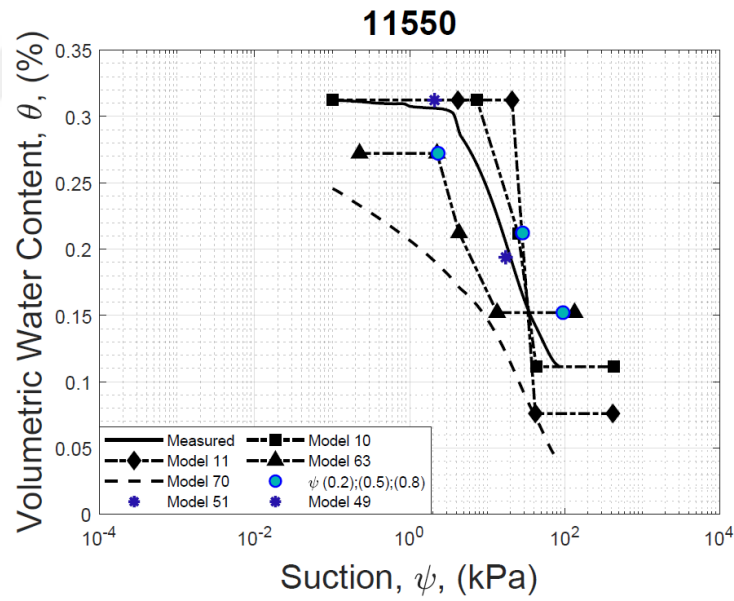


Figure 5.22. Comparison of estimated and measured WRCs of Sample 11550.

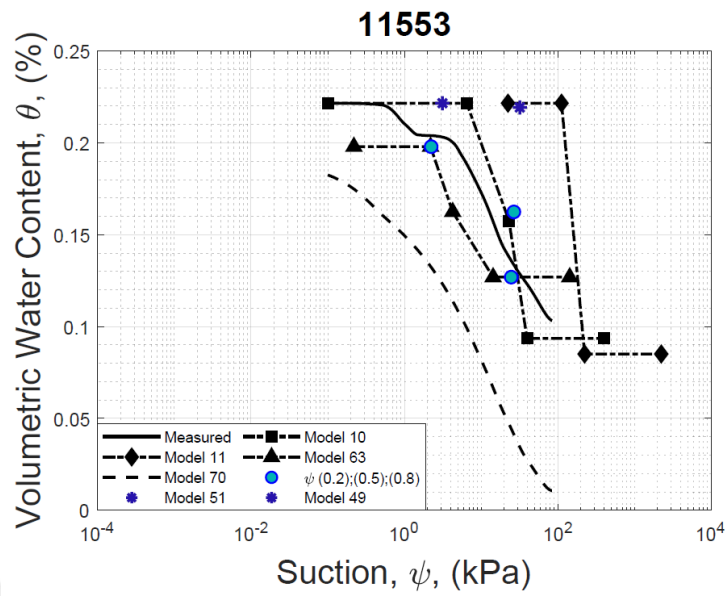


Figure 5.23. Comparison of estimated and measured WRCs of Sample 11553.

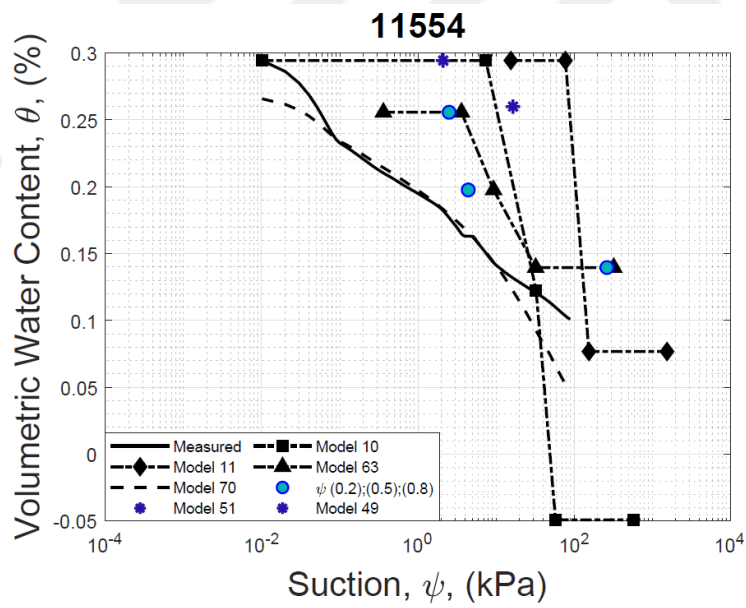


Figure 5.24. Comparison of estimated and measured WRCs of Sample 11554.

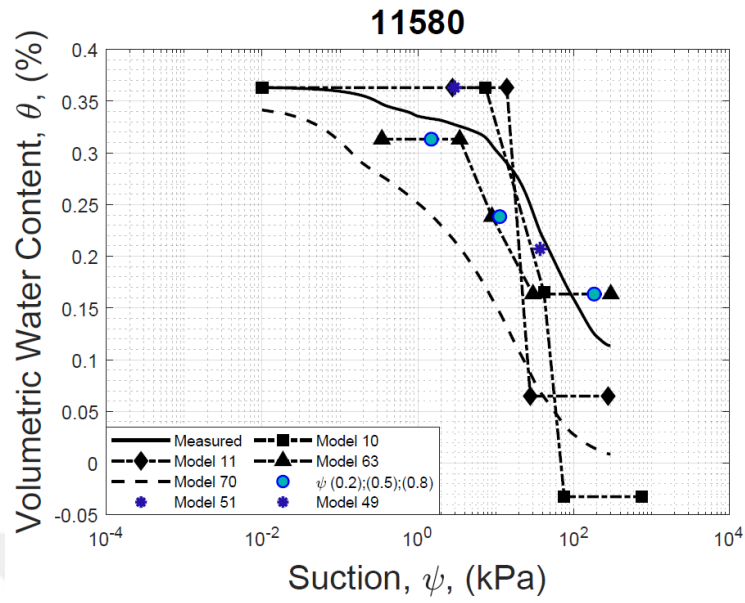


Figure 5.25. Comparison of estimated and measured WRCs of Sample 11580.

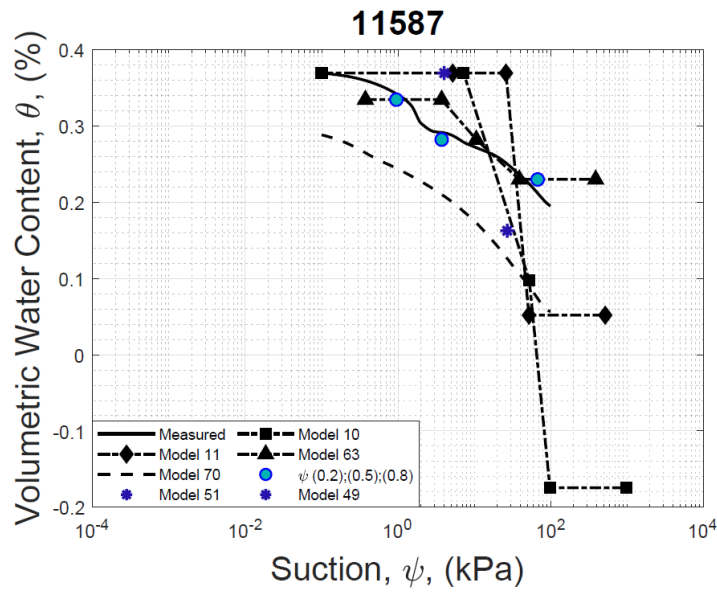


Figure 5.26. Comparison of estimated and measured WRCs of Sample 11587.

Model 70 predicted the shape of the curve with a downward shift.

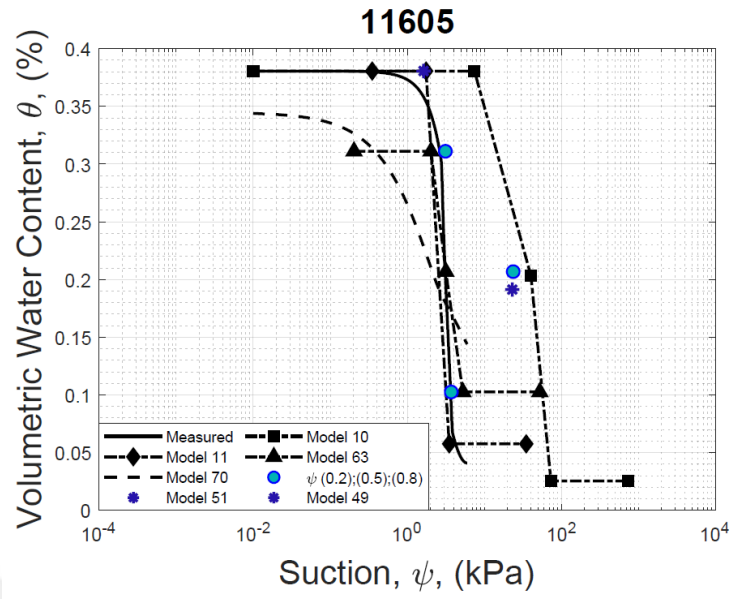


Figure 5.27. Comparison of estimated and measured WRCs of Sample 11605.

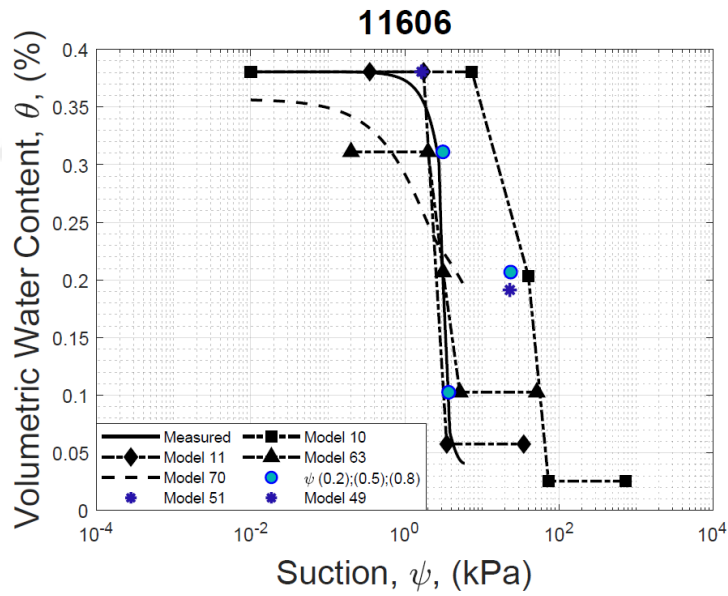


Figure 5.28. Comparison of estimated and measured WRCs of Sample 11606.

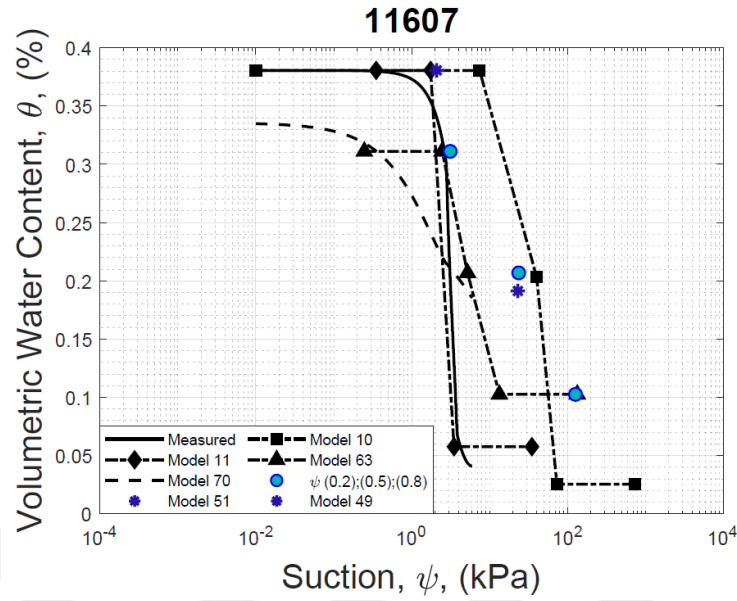


Figure 5.29. Comparison of estimated and measured WRCs of Sample 11607.

Eventually, for most of the plots, Model 10 overestimated AEV, producing rightward shifted curves. In most of the plots, Model 70 caught the shape of WRC but missed its position. All the models produced erroneous predictions, particularly Model 10 demonstrated the poorest estimates. The fact that GSD range of samples from SoilVision being never presented to the modeled networks during learning and missing OMC data, most probably had worsened the results. Out of the 13 soil samples that plotted above, 6 of them had grain sizes up to 50 mm and 2 samples had hydrometer test data, whereas UNSODA has GSD data ranging from 0.002 to 2 mm only. Soil textural information and GSD data can be obtained from SoilVision via sample IDs reported in plot titles.

On the other hand, the models containing effective water content form (Model 63 and singular suction outputs shown as blue dots) gave relatively closer outputs to the measured, particularly for those a well-defined point corresponding to AEV (11554,11553,11587) and/or residual conditions (all the samples except for: 11274,11175,10888) on the measured curve is absent. Thus, it can be concluded that for those samples that have no definable AEV or residual conditions, effective water content form held closer/similar predictions than other models.

CHAPTER 6

CONCLUSION

6.1 Summary and Discussions

The relative functionality of various input – output variables derived from nonparametric (point-wise prediction at AEV, residual conditions, inflection point location, and suctions at various effective water contents) and parametric (fitting parameters of Fredlund and Xing (1994) Equation) WRC approaches was concerned with NN hyperparameters in order to trigger better generalization performance. As opposed to the existing literature which follow a divisive approach on soil types, this study was performed on the complete data obtained from UNSODA, rendering specificity of the study which at the same time problematized the process and in turn, the results. Seven different soil textures defined by the EN system, particle diameter variables (D-Values), porosity, and/or soil organic matter content was assigned as predictors of WRC characteristic parameters which were obtained by parametric and non-parametric methods. The modeled networks were stabilized by minimizing number of hidden neurons and layers subjected to various constraints in terms of hyperparameters. After examining a number of facets of generated 88 models, the following observations were noted from these methods and results:

- The most accurate NN model, the one for predicting Fredlund and Xing (1994) model parameters for which the program outputs held the highest association with actual data. However, it should be noted that curve fitting error comes up as an additive source of error when predicting fitting parameters.
- Use of parametric approach in WRC estimation appeared to be also advantageous for providing a compact procedure when handling the measured data which is meaningful for automating the procedures for a large database.

- Through log translation of features that contain suction data, the modeled networks were able to better interpret the data. Logarithm translation was particularly worthwhile for Fredlund and Xing (1994) fitting parameters.
- Being the second best design, amongst nonparametric methods, manually drawn set of lines to estimate air-entry value, slope of desaturation, and inflection point produced more accurate predictions than models embodying suction values at predefined effective water contents.
- Prediction of WRC parameters as grouped outputs appeared to be a superior solution than assembling the single outputs.
- Taking into account different aspects of performance during model ranking is required to prevent a bogus sense of hierarchy in performances of models. To solidify, models that had lower R^2 values (e.g., Model 63) produced closer outputs to the actual data obtained from SoilVision than Fredlund and Xing (1994) model did, whereas Fredlund and Xing (1994) model caught the shape but missed the position of WRC, particularly when AEV and residual conditions were not clearly defineable.
- Taking into account predictor OMC provided a useful improvement particularly when modeling mid and lower suction segments of WRC.
- The existence of errors in the measured data made it cumbersome to apply nonparametric methods, in turn produced erroneous results.
- Prediction of suction values at predefined effective water contents agreed well with independent data from SoilVision.
- There were no strict relationship divulged between: OMC and required complexity; R^2 and number of epochs; R^2 , number of unknowns and number of available training equations for this available data.
- Increasing the number of trials for initial guesses is a sure way for enhancement. Likewise, despite the reduction in sample size, data pretreatment is required regardless of the input-target variables.

The performance of models depended critically on the data. Insufficient data, in terms of quantity, reflected itself as lower training accuracy, and in terms of quality, led poorly defined AEV and residual conditions, thus, governed the entire process in disguise of outlying, scattered target data. Missing substantial portions of WRC do not allow a strict definition of AEV or residual conditions. Therefore, predictions of water content at inflection point were more accurate than that of residual conditions. Yet, as absent data was not the sole source of errors, other components of error which stemmed from: (i) retention data obtained by different methods in different laboratories, (ii) incomplete curves, and (iii) experimental errors were neglected for their immanence, forcing to attribute the whole variance in errors to predictions.

6.2 Recommendations for Future Work

The points that this study put under question can be elucidated better in a different frame with (i) better quality data, (ii) other tools with different optimization algorithm considering the convergence problem of NN (Achieng, 2019; Schaap et al., 1998). Haghverdi et al (2012); Karandish and Simunek (2016) pointed out that ANFIS and SVM performed better than NN when predicting the WRC. In addition, NN does not reveal the relationships and significances of its features. In this sense, genetic algorithm based examinations may reveal more information regarding to working mechanisms of the predictions. On the other hand, WRC can be represented in a number of different manners. Jain et al. (2004) pointed out that PTF approach has the advantage to estimate complete WRC.

Moreover, use of k-fold validation is a sure way to enhance the performance of models for such small sample size (rather than simple cross validation which was used in this study). Addressing more of the influencers of WRC such as mineralogy, plasticity index, specific surface area, and enlarging the: (i) sample size, and (ii) range of suction covered in measurements would increase the accuracy which is heavily dependent on the database as any other regression method does.

Finally, for a more integrated representation of WRC,

- Estimations regarding the wetting branch of WRC to address the hysteresis,
- Field based data to address the disturbance effect,
- Estimation of shrinkage curve for those soils that undergo volume change as suction is increased, may be considered in future work.



REFERENCES

- Achieng, K. O. (2019). Modelling of soil moisture retention curve using machine learning techniques: Artificial and deep neural networks vs support vector regression models. *Computers & Geosciences*, 133, 104320. doi:10.1016/j.cageo.2019.104320.
- Arya, L. M., Paris, 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*, 45(6), 1023–1030. doi: 10.2136/sssaj1981.03615995004500060004x.
- ASTM International. (2016). D6836-16 Standard Test Methods for Determination of the Soil Water Characteristic Curve for Desorption Using Hanging Column, Pressure Extractor, Chilled Mirror Hygrometer, or Centrifuge. Retrieved from <https://doi.org/10.1520/D6836-16>.
- Bayat, H., Neyshaburi, M. R., Mohammadi, K., Nariman-Zadeh, N., Irannejad, M., & Gregory, A. S. (2013). Combination of artificial neural networks and fractal theory to predict soil water retention curve. *Computers and electronics in agriculture*, 92, 92-103.
- Børgesen, C. and Schaap, M., 2005. Point and parameter pedotransfer functions for water retention predictions for Danish soils. *Geoderma*, 127(1-2), pp.154-167.
- Botula, Y., Nemes, A., Mafuka, P., Van Ranst, E. and Cornelis, W.M. (2013), Prediction of Water Retention of Soils from the Humid Tropics by the Nonparametric *k*-Nearest Neighbor Approach. *Vadose Zone Journal*, 12: 1-17 vzj2012.0123. doi:[10.2136/vzj2012.0123](https://doi.org/10.2136/vzj2012.0123).
- Brooks, R. H., & Corey, A. T. (1964). Hydraulic properties of porous media and their relation to drainage design. *Transactions of the ASAE*, 7(1), 26-0028.

- Craig, R. F. & Knappett, J. A. (2012). *Craig's Soil Mechanics* (8th edition). New York, NY: Spon Press.
- Demuth, H., & Beale, M. (2000). *Neural Network Toolbox*. The MathWorks. *Inc.*, Natick, MA.
- Dreyfus, G. (2005). *Neural networks: Methodology and applications*. New York, NY: Springer Berlin Heidelberg.
- Fogel, D., Liu, D. & Keller, M. (2016). *Fundamentals of computational intelligence: Neural networks, fuzzy systems, and evolutionary computation*. New Jersey, NJ: John Wiley & Sons.
- Fredlund D.G., Pham QH. (2006). A volume-mass constitutive model for unsaturated soils in terms of two independent stress state variables. In *Proceedings of the 3rd International Conference on Unsaturated Soil*, Carefree, Ariz., ASCE, Vol. 1, pp. 105–136.
- Fredlund D.G., Xing A. (1994). Equations for the soil-water characteristic curve. *Canadian Geotechnical Journal*, 31(3), 521-532. doi: 10.1139/t94-061.
- Fredlund, D., Sheng, D., & Zhao, J. (2011). Estimation of soil suction from the soil-water characteristic curve. *Canadian Geotechnical Journal*, 48(2), 186-198. doi:10.1139/T10-060.
- Fredlund, D.G., Rahardjo, H., & Fredlund, M.D. (2012). *Unsaturated soil mechanics in engineering practice*. New York, NY: John Wiley & Sons.
- Fredlund, M. D., Wilson, G. W., and Fredlund, D. G. (1997). Prediction of the soil-water characteristic curve from the grain-size distribution curve. *Proc., 3rd Symp. on Unsaturated Soil*, Rio de Janeiro, Brazil, 13–23.

- Fredlund, M. D., Wilson, G. W., and Fredlund, D. G. (2002). Use of the grain-size distribution for estimation of the soil-water characteristic curve. *Canadian Geotechnical Journal*, 39(5), pp.1103-1117.
- van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil science society of America journal*, 44(5), 892-898.
- Géron, A. (2017). *Hands-on machine learning with scikit-learn and TensorFlow: concepts, tools, and techniques to build intelligent systems* O'Reilly media. *Sebastopol, CA*, 54-56.
- Ghojogh, B., Samad, M. N., Mashhadi, S. A., Kapoor, T., Ali, W., Karray, F., & Crowley, M. (2019). Feature selection and feature extraction in pattern analysis: A literature review. *arXiv preprint arXiv:1905.02845*.
- Gupta, S., & Larson, W. E. (1979). Estimating soil water retention characteristics from particle size distribution, organic matter percent, and bulk density. *Water resources research*, 15(6), 1633-1635.
- Haghverdi, A., Cornelis, W., & Ghahraman, B. (2012). A pseudo-continuous neural network approach for developing water retention pedotransfer functions with limited data. *Journal of Hydrology*, 442-443, 46-54. doi: 10.1016/j.jhydrol.2012.03.036.
- He, X. & Xu, S. (2010). *Process neural networks: Theory and applications*. New York, NY: Springer-Verlag Berlin Heidelberg.
- Hecht-Nielsen, R. (1989). Theory of the back propagation neural network in *Proceedings 1989 IEEE IJCNN*, pp. 1593–1605, IEEE Press, New York.
- Hubert, M., & Van der Veeken, S. (2008). Outlier detection for skewed data. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 22(3-4), 235-246.

- Hubert, M., Rousseeuw, P. J., & Branden, K. V. (2005). *Technometrics*. February, 1(200547), 1.
- Hwang, S. and Choi, S., 2006. Use of a lognormal distribution model for estimating soil water retention curves from particle-size distribution data. *Journal of Hydrology*, 323(1-4), pp.325-334.
- Jain, S., Singh, V., & van Genuchten, M. (2004). Analysis of Soil Water Retention Data Using Artificial Neural Networks. *Journal of Hydrologic Engineering*, 9(5), 415-420. doi: 10.1061/(asce)1084-0699(2004)9:5(415).
- Johari, A., Habibagahi, G., & Ghahramani, A. (2006). Prediction of soil–water characteristic curve using genetic programming. *Journal of Geotechnical and Geoenvironmental Engineering*, 132(5), 661-665.
- Johari, A., and Javadi, A. A. (2010). Prediction of soil-water characteristic curve using neural network. *Proceedings of the Fifth International Conference on Unsaturated Soils*. Barcelona, 461-466.
- Karandish, F., & Šimůnek, J. (2016). A comparison of numerical and machine-learning modeling of soil water content with limited input data. *Journal of Hydrology*, 543, 892-909. doi: 10.1016/j.jhydrol.2016.11.007.
- Kastanek, F. J., & Nielsen, D. R. (2001). Description of soil water characteristics using cubic spline interpolation. *Soil Science Society of America Journal*, 65(2), 279-283.
- Krawczak, M. (2013). *Multilayer neural networks: a generalized net perspective* (Vol. 478). Springer.
- Koekkoek, E., & Booltink, H. (1999). Neural network models to predict soil water retention. *European Journal of Soil Science*, 50(3), 489-495. doi: 10.1046/j.1365-2389.1999.00247.x.

Klute, A. (1986). Water retention: Laboratory methods, in methods of soil analysis, part 1- Physical and mineralogical methods (2nd ed.). Madison, WI: American Society of Agronomy.

Lamorski, K., Šimůnek, J., Sławiński, C., & Lamorska, J. (2017). An estimation of the main wetting branch of the soil water retention curve based on its main drying branch using the machine learning method. *Water Resources Research*, 53(2), 1539-1552. doi: 10.1002/2016wr019533.

Leij, F. J. (1996). The UNSODA unsaturated soil hydraulic database: user's manual (Vol. 96, No. 95). National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency.

Lu, N., Likos, W.J. (2004). *Unsaturated soil mechanics*. Hoboken, NJ: Wiley.

Nemes, A., Schaap, M., Leij, F., & Wösten, J. (2001). Description of the unsaturated soil hydraulic database UNSODA version 2.0. *Journal Of Hydrology*, 251(3-4), 151-162. doi: 10.1016/s0022-1694(01)00465-6.

Nguyen, D., & Widrow, B. (1990, June). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *1990 IJCNN International Joint Conference on Neural Networks* (pp. 21-26). IEEE.

Nielsen, R. H. (1989). Theory of the back propagation neural network in *Proceedings 1989 IEEE IJCNN*, pp. 1593–1605, IEEE Press, New York, 1989.

Orr, G. & Müller, K. (1998). *Neural networks: tricks of the trade*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Pachepsky, Y.A., Timlin, D., Varallyay, G., (1996). Artificial neural networks to estimate soil water retention from easily measurable data. *Soil Sci. Soc. Am. J.* 60, 727.

- Pham, H., & Fredlund, D. (2008). Equations for the entire soil-water characteristic curve of a volume change soil. *Canadian Geotechnical Journal*, 45(4), 443-453. doi: 10.1139/t07-117.
- Pham, B. T., Nguyen, M. D., Bui, K. T. T., Prakash, I., Chapi, K., & Bui, D. T. (2019). A novel artificial intelligence approach based on Multi-layer Perceptron Neural Network and Biogeography-based Optimization for predicting coefficient of consolidation of soil. *Catena*, 173, 302-311.
- Priddy, K.L. & Keller P.E. (2005). *Artificial neural networks: An introduction*. Bellingham, WA: SPIE Press.
- Qian, Z., & Rahardjo, H. (2016). Application of fitting parameters in best fit equation. *E3S Web of Conferences*, 9, 10008. doi: 10.1051/e3sconf/20160910008.
- Rahardjo, H., Kim, Y., & Satyanaga, A. (2019). Role of unsaturated soil mechanics in geotechnical engineering. *International Journal of Geo-Engineering*, 10(8). doi: 10.1186/s40703-019-0104-8.
- Rahardjo, H., Li, X., Toll, D.G., & Leong, E.C. (2001). The effect of antecedent rainfall on slope stability. *Geotechnical and Geological Engineering*, 19(3), 371-399. doi: 10.1023/A:1013129725263.
- Ren, X., Kang, J., Ren, J., Chen, X., & Zhang, M. (2020). A method for estimating soil water characteristic curve with limited experimental data. *Geoderma*, 360, 114013. doi: 10.1016/j.geoderma.2019.114013.
- Richards, L. (1931). Capillary conduction of liquids through porous mediums. *Physics*, 1(5), 318-333. doi: 10.1063/1.1745010.
- Rousseeuw, P. J., & Van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical association*, 85(411), 633-639.

- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature*, 323, 533–536.
- Saha, S., Gu, F., Luo, X., & Lytton, R. L. (2017). Prediction of Soil-Water Characteristic Curve Using Artificial Neural Network Approach. In *PanAm Unsaturated Soils 2017* (pp. 124-134).
- Sattari, A.S., & Toker, N.K. (2016). Obtaining soil–water characteristic curves by numerical modeling of drainage in particulate media. *Computers and Geotechnics*, 74, 196-210.
- Schaap, M., & Bouten, W. (1996). Modeling water retention curves of sandy soils using neural networks. *Water Resources Research*, 32(10), 3033-3040. doi: 10.1029/96wr02278.
- Schaap, M., (1998). Using neural networks to predict soil water retention and soil hydraulic conductivity. *Soil Tillage Res.* 47, 37–42.
- Schaap, M., Leij, F., & van Genuchten, M. (1998). Neural Network Analysis for Hierarchical Prediction of Soil Hydraulic Properties. *Soil Science Society of America Journal*, 62(4), 847-855. doi: 10.2136/sssaj1998.03615995006200040001x.
- Sillers, W. S., & Fredlund, D. G. (2001). Statistical assessment of soil-water characteristic curve models for geotechnical engineering. *Canadian Geotechnical Journal*, 38(6), 1297-1313.
- Sillers, W.S., Fredlund, D.G. & Zakerzaheh, N. (2001). Mathematical attributes of some soil–water characteristic curve models. *Geotechnical and Geological Engineering* 19(3), 243–283. doi: 10.23/A:1013109728218.

- Toker, N., Germaine, J., Sjoblom, K., & Culligan, P. (2004). A new technique for rapid measurement of continuous soil moisture characteristic curves. *Géotechnique*, 54(3), 179-186. doi: 10.1680/geot.2004.54.3.179.
- Vanapalli, S.K., Fredlund, D.G., Pufahl, D.E., and Clifton, A.W. (1996). Model for the prediction of shear strength with respect to soil suction. *Canadian Geotechnical Journal*, 33: 379–392.
- Wijaya, M., & Leong, E. (2016). Equation for unimodal and bimodal soil–water characteristic curves. *Soils And Foundations*, 56(2), 291-300. doi: 10.1016/j.sandf.2016.02.011.
- Zhai, Q., & Rahardjo, H. (2014). Variability in soil–water characteristic curve associated with fitting parameters. In *Conference Paper, UNSAT*.
- Zapata, C. E., Houston, W. N., Houston, S. L., and Walsh, K. D. (2000). “Soil-Water Characteristic Curve Variability.” In: C. D. Shackelford, S. L. Houston, and N-Y Chang (eds). *Advances in Unsaturated Geotechnics*. ASCE – GEO Institute Geotechnical Special Publication, No. 99. Also Proceedings of
- Zhan, T.L.T., Ng, C.W.W., Fredlund, D.G. (2007). Field study of rainfall infiltration into a grassed unsaturated expansive soil slope. *Canadian Geotechnical Journal*, 44(4), 392-408. doi: 10.1139/T07-001.

APPENDICES

A. NETWORK GENERATIVE MATLAB CODE

```
cd = 'C:\Users\labuser.PCLABS\Desktop\CreateNet' ;
folder = cd ;
b = 1 ;
A = [] ;
for k = 1:88
    a = 0 ;
    matFilename = sprintf('m%d.mat', k) ;
    matData = load(fullfile(cd, matFilename)) ;

    input = {matData.input} ;
    input = cell2mat(input) ;
    target = {matData.target} ;
    target = cell2mat(target) ;

    rng('default')
    j = 1 ;
    % Preprocessing
    if size(input,1) > size(input,2)
        input = input' ;
    end
    if size(target,1) > size(target,2)
        target = target' ;
    end
    % Standardization
    [xnorm, xsettings] = mapstd(input) ; %[I N]
    [tnorm, tsettings] = mapstd(target) ; %[O N]
    x = xnorm ;
```

```

t = tnorm ;
n = size(t,2);
% Data Divison
trainRatio1 = .7 ;
valRatio1 = .15 ;
testRatio1 = .15 ;
%Number of weight initializations
Ntrials = 20 ;
%Call pre-defined network topologies
modelDesign01 ;
L = length(model) ;
hr = L*4*Ntrials/60;
fprintf('Estimated completion in %.1f minutes\n',hr)
% Network Generation and Train
for m = 1:L
    %FFNN uses back-propagation of error
    net = feedforwardnet ([model(m).nodes]);
    %Train the generated model for Ntrials weight initializations
    for j=1:Ntrials
        %Configure so that its input, output, weight,
        %and bias dimensions match the input and target data
        net = configure(net,x,t) ;
        % emptying the init.param
        net.initFcn = 'initlay' ;
        nolayer = length(model(m).nodes) ;
        %Choose the Activation Function and Initialize each layer
        % according to Nguyen-Widrow Algorithm
        for l=1:nolayer
            net.layers{l}.transferFcn = 'tansig' ;
            net.layers{l}.initFcn = 'initnw' ;
        end
    end
end

```

```

net = init(net);
net.trainParam.showWindow = 0 ;
net.trainFcn = 'trainlm';
net.performFcn = 'mse' ;
tr.performFcn = 'mse' ;

% Random distribution of data indices
net.divideFcn = 'dividerand';
net.divideParam.trainRatio = trainRatio1 ;
net.divideParam.valRatio = valRatio1 ;
net.divideParam.testRatio = testRatio1 ;

net.trainParam.max_fail= 6 ;
net.trainParam.epochs = 1000 ;
net.trainParam.goal = 0 ;
net.trainParam.mu = 0.001 ;
net.trainParam.mu_dec = 0.1 ;
net.trainParam.mu_inc =10 ;

%Training
[net, tr] = train(net,x,t) ;
model(m).net{j} = net ;
model(m).tr(j) = tr ;

%Evaluation based on validation - MSE
yHat = sim(net,x(:,tr.valInd)) ;
yHatReverse = mapstd('reverse',yHat,tsettings) ;
model(m).yHatReverseScaled{j} = yHatReverse ;
model(m).yHat{j} = yHat ;
eVal = mse(yHat-t(:,tr.valInd)) ;

```

```

trOut = sim(net,x(:,tr.trainInd)) ;
model(m).trOut{j} = trOut ;
eTr = mse(trOut-t(:,tr.trainInd)) ;
trOutReversed = mapstd('reverse',trOut,tsettings) ;
model(m).trOutReversed {j} = trOutReversed ;
% Test simulation
yTest = sim(net,x(:,tr.testInd)) ;
yTestReal = mapstd('reverse',yTest,tsettings) ;
model(m).yTestReal{j} = yTestReal ;
model(m).yTest{j} = yTest ;
eTest = mse(yTest-t(:,tr.testInd));

model(m).wb{j} = getwb(net);
model(m).msetrn(j) = tr.best_perf ;
model(m).mseVal(j) = tr.best_vperf ;
model(m).mseTest(j) = tr.best_tperf ;
model(m).bestEpoch(j) = tr.best_epoch ;
model(m).numbEpoch{j} = tr.epoch ;

%the Nash-Suthcliff Efficiency
%test
val = target(:,tr.valInd); val = val'; yHatReverse =
yHatReverse';
tes = target(:,tr.testInd); tes = tes'; yTestReal =
yTestReal';
Etest = yTestReal - tes;
SSEtest = sum(Etest.^2) ;
utest = mean(tes) ;
SSUtest = sum((tes-utest).^2) ;
NSEtest = 1 - SSEtest./SSUtest ;
model(m).NSEtest(j,:) = NSEtest(:) ;
%val NSE

```

```

E = yHatReverse - val ;
SSE = sum(E.^2) ;
u = mean(val) ;
SSU = sum((val-u).^2) ;
NSE = 1 - SSE./SSU ;
model(m).NSE(j,:) = NSE(:) ;

%Percent BiasTest
SEtest = sum(Etest) ;
Stest = sum(tes) ;
PBIAStest = SEtest./Stest*100 ;
model(m).PBIAStest(j,:) = PBIAStest(:) ;
%Percent Bias
SE = sum(E) ;
Sval = sum(val) ;
PBIAS = SE./Sval*100 ;
model(m).PBIAS(j,:) = PBIAS(:) ;

%Mean Relative ErrorTest
n = size(val,1);
Stest = abs((tes-yTestReal)./(tes))*100 ;
MREtest = (1/n) * (sum(Stest)) ;
model(m).MREtest(j,:) = MREtest(:,) ;

%Mean Relative Error
S = abs((val-yHatReverse)./(val))*100 ;
MRE = (1/n) * (sum(S)) ;
model(m).MRE(j,:) = MRE(:,) ;
end

if m==round(L*0.2)

```

```

        display('20% completion')
    end
    if m==round(L*0.4)
        display('40% completion')
    end
    if m==round(L*0.6)
        display('60% completion')
    end
    if m==round(L*0.8)
        display('80% completion')
    end
end
% Record the lowest evaluation error within Ntrials nets
yHatReverse = yHatReverse'; yTestReal = yTestReal';
for i=1:L
    minimum = min(min(model(i).mseVal(:)));
    idx = find(model(i).mseVal == minimum) ;
    model(i).eValOpt = minimum ;
    model(i).eTrOpt = min(model(i).msetrn(idx)) ;
    model(i).eTestOpt = min(model(i).mseTest(idx)) ;
    model(i).BestTrial = model(i).net{idx};
    model(i).RawOutputVal(:, :) = model(i).yHatReverseScaled{idx};
    model(i).RawOutputTest(:, :) = model(i).yTestReal{idx};
    model(i).RawOutputTrain(:, :) = model(i).trOutReversed{idx};
    model(i).BestTrialOutput{:} = model(i).y{idx};
    model(i).Tr = model(i).tr(idx);
    model(i).OptPBIAS = mean(model(i).PBIAS(idx)) ;
    model(i).OptMRE = mean(model(i).MRE(idx)) ;
    model(i).OptNSE = mean(model(i).NSE(idx)) ;
    model(i).OptPBIASest = mean(model(i).PBIASest(idx)) ;
    model(i).OptMREtest = mean(model(i).MREtest(idx)) ;

```

```

        model(i).OptNSEtest = mean(model(i).NSEtest(idx)) ;
    end
    % Sort models according to valMSE
    i = [model(:).eValOpt] ;
    [StructSort, StructSortIdx] = sort(i, 'ascend') ;
    StructSortIdx = StructSortIdx' ;
    StructSorted = model(StructSortIdx) ;
end

```

