Technical University of Crete
School of Environmental Engineering
Postgraduate Studies Program
Environmental Engineering

Predicting Spatial and Temporal changes in groundwater levels using Artificial Neural Networks and Geostatistical methods

Πρόβλεψη Χωροχρονικών μεταβολών στάθμης υπογείων υδάτων με χρήση Τεχνητών Νευρωνικών Δικτύων και Γεωστατιστικών μεθόδων.

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by

Evdokia Tapoglou
Environmental Engineer, M.Sc.

Examination Committee: Prof. George Karatzas (Supervisor)
Prof. Nikolaos Nikolaidis (Advisory committee)
Ass. Prof. Ioannis Nikolos (Advisory committee)
Prof. Constantinos Chrysikopoulos
Prof. Ioannis Tsanis
Prof. Dionissios Hristopoulos
Prof. Athanasios Loukas

Chania
29/05/2015

Examination Committee

GEORGE KARATZAS – Supervisor
Professor,
School of Environmental Engineering,
Technical University of Crete,

NICOLAOS NIKOLAIDIS – Advisory Committee
Professor,
School of Environmental Engineering,
Technical University of Crete,

IOANNIS NIKOLOS – Advisory Committee
Associate Professor,
School of Production Engineering and Management,
Technical University of Crete,

CONSTANTINOS CHRYSIKOPOULOS
Professor,
School of Environmental Engineering,
Technical University of Crete,

IOANNIS TSANIS
Professor,
School of Environmental Engineering,
Technical University of Crete,

DIONISSIOS HRISTOPOULOS
Professor,
School of Mineral Resources Engineering,
Technical University of Crete

ATHANASIOS LOUKAS
Professor,
Department of Civil Engineering,
University of Thessaly

Τίτλος Διδακτορικής Διατριβής:

Predicting Spatial and Temporal changes in groundwater levels using Artificial Neural Networks and Geostatistical methods - Πρόβλεψη Χωροχρονικών μεταβολών στάθμης υπογείων υδάτων με χρήση Τεχνητών Νευρωνικών Δικτύων και Γεωστατιστικών μεθόδων

«Η έγκριση της παρούσης Διδακτορικής Διατριβής από τη Σχολή Μηχανικών Περιβάλλοντος του Πολυτεχνείου Κρήτης δεν υποδηλώνει αποδοχή των γνωμών της συγγραφέως» (Ν. 5343/1932, άρθρο 202, παρ. 2)

To my family

Acknowledgments

This PhD thesis is the result of a challenging journey, upon which many people have contributed and given their support. The tip of the iceberg starts with friends, family, and supervisor. Go a little deeper though, and we find many more who deserve credit: Colleagues, postdocs, proof-readers, participants, researcher communities, academics, office-mates, etc.

First and foremost I would like to express my sincere gratitude to my supervisor Prof. George Karatzas. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and exciting. His guidance helped me throughout the research and writing of this thesis. Special thanks deserve also the member of my advisory committee; Prof Nikolaos Nikolaidis and Ass. Prof. Ioannis Nikolos. Their support and guidance throughout my studies was valuable. I would also like to thank the members of the examining committee, Prof. Constantinos Chrysikopoulos, Prof. Ioannis Tsanis, Prof Dionissios Hristopoulos and Prof. Athanasios Loukas for taking the time to evaluate this doctoral work, their helpful observations and contributions towards its improvement.

In addition, I would like to thank the members of the Geoenvironmental Engineering Laboratory Dr. Emmanouil Varouchakis and Dr. Ioannis Trichakis, as well as all other former and current members and especially Dr. Zoi Dokou Dr. Anthi-Eirini Vozinaki, PhD Candidate Konstantina Kavouri, Dr. Nektarios Kourgialas. Their support both scientific, and psychological was extremely valuable over the years, while our coexistence in the laboratory was always pleasant.

I would also like to thank all of my friends who supported me in writing, and incented me to strive towards my goal. Lastly, and most importantly, a special thanks to my family. Words cannot express how grateful I am to my parents Stelios and Dionysia and my brother Nikos, for all of the sacrifices that they've made on my behalf. They raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

Table of Contents

List of Tables

List of Figures

Περίληψη

Σκοπός της παρούσας διατριβής είναι η δημιουργία ενός προγράμματος χωρικής και χρονικής προσομοίωσης του υδραυλικού ύψους ενός υδροφορέα, με χρήση μεθόδων υπολογιστικής νοημοσύνης και γεωστατιστικών μεθόδων. Τα Τεχνητά Νευρωνικά Δίκτυα είναι η μέθοδος που επιλέχθηκε για τη χρονική προσομοίωση, καθώς έχει αποδειχθεί ήδη στην βιβλιογραφία ότι η χρήση της φέρει καλά αποτελέσματα χωρίς την απαίτηση πολύπλοκων και δυσεύρετων δεδομένων εισόδου. Η μεθοδολογία του Kriging ακολουθήθηκε για τη χωρική προσομοίωση και παρεμβολή των αποτελεσμάτων των τεχνητών νευρωνικών δικτύων στο χώρο. Για το συνδυασμό των δύο αυτών μεθοδολογιών χρησιμοποιήθηκε ένα σύστημα ασαφούς λογικής.

Το πρώτο βήμα που ακολουθείται στην προτεινόμενη προσέγγιση είναι η συλλογή όλων των διαθέσιμων δεδομένων. Οποιαδήποτε παράμετρος μπορεί να συνδεθεί με το υδατικό ισοζύγιο μπορεί να χρησιμοποιηθεί σαν παράμετρος εισόδου στα τεχνητά νευρωνικά δίκτυα. Στην παρούσα διατριβή συλλέχθηκαν δεδομένα και έγινε προσομοίωση του υδραυλικού ύψους για δύο περιοχές μελέτης μια στην Βαυαρία της Γερμανίας και μια στο Μαϊάμι της πολιτείας Φλόριντα των ΗΠΑ. Οι δύο αυτές περιοχές έχουν πολύ διαφορετικά χαρακτηριστικά μεταξύ τους και για αυτό το λόγο η επιτυχής προσομοίωση του υδραυλικού ύψους σε αυτές μπορεί να επιβεβαιώσει την αξιοπιστία του μοντέλου. Στη συνέχεια προσδιορίζονται οι βέλτιστες χρονικές υστερήσεις παραμέτρων όπως οι κατακρημνίσεις και το ύψος ή/και η παροχή των επιφανειακών υδάτων.

Ακολουθεί η προσομοίωση με χρήση τεχνητών νευρωνικών δικτύων. Ανάλογα με την περιοχή μελέτης και τα χαρακτηριστικά των διαθέσιμων δεδομένων, δοκιμάστηκαν διάφορες αρχιτεκτονικές, έτσι ώστε να επιλεχθεί το κατάλληλο δίκτυο για την προσομοίωση. Στην πρώτη περιοχή μελέτης δοκιμάστηκε η χρήση συστήματος ασαφούς λογικής για την επιλογή των κατάλληλων γειτόνων που στην συνέχεια χρησιμοποιούνται από τον αλγόριθμο του kriging. Η αποτελεσματικότητα της χρήσης της μεθόδου ασαφούς λογικής αποδείχθηκε συγκρίνοντας τα αποτελέσματα της μεθόδου με αυτά που προκύπτουν χωρίς τη χρήση της.

Τέλος, εφαρμόζεται η μεθοδολογία της παρεμβολής με kriging, με χρήση τριών διαφορετικών βαριογραμμάτων σε κάθε περίπτωση. Τα αποτελέσματα αξιολογούνται μέσω μιας σειράς δεικτών σφάλματος σε δεδομένα διασταυρωμένης επικύρωσης και προσδιορίζεται το πιο κατάλληλο θεωρητικό μοντέλο βαριογράμματος για κάθε περιοχή μελέτης ξεχωριστά. Για την πρώτη περιοχή μελέτης στη Βαυαρία, καταλληλότερο βαριόγραμμα ήταν το δυναμονομικό με τιμή για τη ρίζα του μέσου τετραγωνικού σφάλματος (RMSE) ίση με $7.6 \cdot 10^{-3}$ m. Για την δεύτερη περιοχή μελέτης καταλληλότερο βαριόγραμμα ήταν το εκθετικό με τιμή για το δείκτη σφάλματος RMSE ίση με 0.962 m. Βασική διαφορά που οδηγεί σε αυτή την απόκλιση στις τιμές του σφάλματος ήταν ότι στην πρώτη περίπτωση προσομοιώνεται η διαφορά του υδραυλικού ύψους, ενώ στη δεύτερη περιοχή μελέτης προσομοιώνεται το υδραυλικό ύψος αυτό καθ' αυτό. Η επιλογή διαφορετικών παραμέτρων εξόδου έγινε καθώς στη πρώτη περίπτωση οι τιμές που προσομοιώθηκαν είχαν μεγάλες αρχικές τιμές, με μεγάλη μεταβλητότητα μεταξύ των σημείων με δεδομένα, ενώ στη δεύτερη περίπτωση τα δεδομένα υδραυλικού ύψους είχαν ομοιόμορφες, σχεδόν κανονικοποιημένες, τιμές.

Για την επιβεβαίωση του μοντέλου και την πιστοποίηση της ακρίβειάς του, πραγματοποιήθηκε ανάλυση αβεβαιότητας. Αρχικά προσδιορίστηκαν τα διαστήματα εμπιστοσύνης των παραμέτρων εξόδου των τεχνητών νευρωνικών δικτύων εφαρμόζοντας την ποσοστιαία μέθοδο. Στη συνέχεια μέσω ανάλυσης Monte Carlo, προσδιορίστηκαν η αβεβαιότητα της παραμέτρου εξόδου στην εκπαίδευση των τεχνητών νευρωνικών δικτύων με τη μορφή διαστημάτων εκτίμησης, αλλά και η επίδραση τους στα αποτελέσματα του kriging. Σε τελευταίο στάδιο μελετήθηκε η αβεβαιότητα των παραμέτρων εξόδου στις παραμέτρους του kriging με χρήση συνθετικών δεδομένων. Τα αποτελέσματα των παραπάνω διεργασιών έδειξαν ότι η χρησιμοποιούμενη μεθοδολογία ήταν αποτελεσματική, ακριβής και σταθερή ως προς τα αποτελέσματα και τα σφάλματά της.

Η καινοτομία της παρούσας διδακτορικής διατριβής είναι η ανάπτυξη ενός μοντέλου με βάση τα δεδομένα το οποίο μπορεί να χρησιμοποιηθεί για την προσομοίωση του υδραυλικού ύψους. Το μοντέλο αυτό μπορεί να χρησιμοποιηθεί για τη χωρική και χρονική πρόβλεψη του υδραυλικού ύψους, αλλά και άλλων παραμέτρων, χωρίς να είναι γνωστές οι γεωλογικές συνθήκες του πεδίου. Αντ΄ αυτού χρησιμοποιούνται δεδομένα που είναι διαθέσιμα για μεγάλο χρονικό διάστημα και είναι εύκολα στην καταγραφή τους, όπως για παράδειγμα μετεωρολογικά δεδομένα. Επιπροσθέτως η χρήση της ασαφούς λογικής για τον προσδιορισμό των γειτόνων του kriging αποτελεί καινοτομία της παρούσας εργασίας και μπορεί να βελτιώσει τα αποτελέσματα της μεθόδου. Η εφαρμογή του μοντέλου σε δύο διαφορετικού γεωλογικού υπόβαθρου περιοχές μελέτης, μια προσχωματική και μια καρστική, αποδεικνύει την αποτελεσματικότητα του κάτω από διάφορες συνθήκες.

Abstract

The purpose of this study is to create a spatial and temporal simulation program for the estimation of the hydraulic heads in an aquifer, using computational intelligence and geostatistical methods. Artificial neural networks were chosen for the temporal simulation, due to their proven ability, established in literature. This choice is further reinforced by the fact that their implementation does not require complex and hard to obtain input data. The methodology was completed by using the Kriging method for the spatial interpolation of the artificial neural network' simulation results. A fuzzy logic system was employed to combine these two methodologies.

The first step in the proposed approach is the collection of all available data. Any parameter related to the water balance can be used as input parameter for artificial neural networks. In this study, data were collected and the hydraulic head was simulated for two study areas; one in Bavaria, Germany and one in Miami, Florida, USA. These two regions have very different characteristics (geological, climatic and land use) and for this reason the successful simulation of hydraulic head in these regions can confirm the reliability of the model. Following data collection, the optimum time lags for parameters such as precipitation and surface water discharge and/or water depth on the aquifer were determined.

Next, the simulation using artificial neural networks was performed. Depending on the study area and the characteristics of the available data, several architectures were tested in order to select the appropriate network architecture for the simulation. For the first study site, for the selection of suitable neighbors cases with and without the use of a fuzzy logic system were tested, followed by the Kriging algorithm. In this way, the use the fuzzy logic system was proven to be the more effective.

Finally, the interpolation of the point hydraulic head estimation was performed by kriging using three different variogram models in each case. The results are evaluated through a series of cross-validation results error indicators, in order to determine the most appropriate theoretical variogram model for each study area. For the first study area in Bavaria, the power-law variogram was the most appropriate variogram yielding a root mean square error (RMSE) equal to $7.6 \cdot 10^{-3}$ m. For the second study area in Miami, the most appropriate variogram was the exponential one with error indicator of RMSE = 0.962 m. The key difference between the two case studies, leading to this deviation in RMSE values is that, the first case the hydraulic head change per time step was simulated, while in the second study area the hydraulic head itself was simulated. This difference in output parameters was due to the dataset involved in every case. In the first study area they were large variations in the hydraulic head, while at the second case study, the values of the dataset were more normalized.

Uncertainty analysis was performed in order to validate the model and verify its accuracy. Initially, the uncertainty of the hydraulic head simulation attributed to the artificial neural network simulations was determined by applying the percentile method. Next, using a Monte Carlo analysis, the output parameter uncertainty derived by the training of the artificial neural networks was calculated, followed by their respective effect in the output of the kriging. Finally, the uncertainty of the hydraulic head due to the kriging parameters was studied using synthetic data. The results of these processes showed that the methodology used was efficient, accurate and stable with regards to its results and errors.

The innovation of this thesis is the development of a data-based model which can be used to predict the hydraulic head in an aquifer. This model can then be used for the spatial and temporal simulation of the hydraulic head or any other parameter, without knowing the geological characteristics of the study area. Instead data available for a long time and is easy monitor, for example meteorological data can be used. Moreover, the use of fuzzy logic to determine the kriging neighbors is an innovation of this study and improves considerably the results. The implementation of the model in two study areas with different geological background, an alluvial and a karstified, proves its effectiveness under various conditions.

Chapter 1

Introduction

1.1 Scope and objective

Water is an important resource, necessary for the development of human life. Groundwater is the main source of drinking water in the planet, so the study of its quantity and quality is valuable, offering information important for its management and exploitation. However, the need for a wide range of data for the study area and for extensive knowledge of the aquifer makes the implementation of traditional modelling techniques challenging, time-consuming and with high uncertainty.

The objective of this study is the development of an alternative tool for groundwater hydraulic head simulation that does not make use of conventional modelling techniques and relies on the use of data which are easy to acquire. The use of machine learning techniques in hydrological studies has created new opportunities in the development of black-box models, while geostatistics are commonly used for the spatial interpolation of hydrological parameters.

In the present study, the use of Artificial Neural Networks in combination with the kriging interpolation method for the temporal and spatial hydraulic head variation in an aquifer is studied. The further use of machine learning techniques, such as fuzzy logic, which combines computational results with expert knowledge, for the improvement of initial results was also studied. Finally, the uncertainty analysis of the proposed methodology is also a main objective of the present study.

1.2 Innovation

The application of machine learning techniques and of geostatistical methods has been thoroughly tested in the past yielding, in most cases, better results than conventional modelling techniques. However, this is the first attempt to combine the strengths of these two methodologies, by applying Artificial Neural Networks and kriging for the temporal and spatial simulation of hydraulic head respectively. The main advantage of the developed methodology is that it does not require geological data and extensive knowledge of the study area. In this way, it is possible to simulate the hydraulic head variability, only by using meteorological and hydrological data as inputs.

The use of fuzzy logic to interconnect these methodologies is also a novelty, since its use was brought on to the methodology in order to tackle problems encountered while developing the algorithm. More specifically, it is the first time fuzzy logic is used for the kriging neighborhood determination.

The development of a Graphical User Interface to accommodate the use of the algorithm resulted in a fully functional program capable of executing the code with minimal user input and without the need of programming skills.

For the validation of the methodology, the algorithm was applied to two diverse case studies, the first one in an alluvial and the second one in a karstic aquifer. The derived simulation results, along with the uncertainty analysis results, endorse the applicability of the model irrespective of the geological background.

1.3 Thesis outline

The remainder of the present dissertation is organized as follows:

In **Chapter 2**, a comprehensive analysis of the methodologies involved is presented. The theoretical background and the principles for both Artificial Neural Networks (ANN) and Kriging interpolation are presented along with the overview of adjacent methodologies. Fuzzy logic and error measuring techniques are also analyzed in this chapter. Furthermore, the application of the involved methodology individually in hydrological studies is summarized.

In **Chapter 3**, the developed methodology is presented. The first part describes the ANN training procedure followed by the fuzzy logic methodology. The evaluation part of the algorithm is comprised of the Kriging and ANN testing methodologies. The Graphical User Interface developed to make the application of the algorithm user-friendly, and its functionality are also presented. Finally, the adopted uncertainty analysis methodology is described.

In **Chapter 4**, the two study areas where the developed methodology was tested are presented. Meteorological and hydrological data, as well as the geology and some general information about each area, are given. Parameters necessary for the development of the ANNs, the final input parameters, as well as information about the time lags between the parameters involved and their effect on the aquifer hydraulic head, are presented.

In **Chapter 5**, the results of the procedure are analyzed. That includes the ANN training and the evaluation results, the cross-validation and uncertainty analysis of the algorithm. Results are presented for a number of representative wells in each case, where ANN training is concerned. Simulation results for specific prediction points are demonstrated. The same procedure was followed for both case studies.

In **Chapter 6**, the conclusions of this research are summarized and evaluated, while in **Chapter 7** a series of recommendations along with topics for potential future research are presented.

1.4 Publications

In this chapter the publications in international peer-reviewed journals and international conferences are presented.

Journal Publications

Tapoglou E., Chatzakis A. and Karatzas G.P. (2015) Comparison of a black-box model to a traditional numerical model for hydraulic head prediction, Desalination and Water Treatment, Under Review

Tapoglou E., Karatzas G.P., Trichakis I.C. and Varouchakis E.A. (2014), A spatio-temporal hybrid neural network - kriging model for groundwater level simulation. Journal of Hydrology, 519, Part D(0), 3193-3203

Tapoglou E., Trichakis I.C., Dokou Z. and Karatzas G.P. (2013), Groundwater level forecasting under climate change scenarios using an artificial neural network trained with particle swarm optimization. Hydrological Sciences Journal Vol. 59(6), 1225-1239.

International Conferences

Aydin Sarikurt D., Yaksi K., Copty N., **Tapoglou E**., Dokou Z. and Karatzas G. (2015), Cosolvent Flushing for Remediation of Groundwater Contaminated with DNAPL, 7th International Conference on Porous Media.

Tapoglou E., Karatzas G.P., Trichakis I.C. and Varouchakis E.A. (2015), Uncertainty analysis of a combined Artificial Neural Network - Fuzzy logic - Kriging system for spatial and temporal simulation of Hydraulic Head. EGU2015-407, EGU General Assembly 2015.

Pantou Th. **Tapoglou E.** and Karatzas G.P. (2014), Hydraulic head simulation for Messara region, Crete using Artificial Neural Networks and different training algorithms. 10th International Hydrogeological Confess of Greece.

Chatzakis Al., **Tapoglou E.** and Karatzas G.P. (2014), Groundwater Simulation Using Artificial Neural Networks (ANNs) and Princeton Transport Code (PTC) - Performance Comparison. 12th International Conference "Protection and Restoration of the Environment – PRE12". Full paper on Proceedings pg. 2-8

Tapoglou E., Karatzas G.P., Trichakis I.C. and Varouchakis E.A. (2014), Temporal and Spatial prediction of groundwater levels using Artificial Neural Networks, Fuzzy logic and Kriging Interpolation. Geophysical Research Abstracts, Vol. 16, EGU2014-773, EGU General Assembly 2014.

Tapoglou E., Trichakis I.C., Dokou Z. and Karatzas G.P. (2012), Groundwater level forecasting using an artificial neural network trained with particle swarm optimization. Geophysical Research Abstracts, Vol. 14, EGU2012-2405, EGU General Assembly 2012.

Chapter **2**

Theoretical Background

2.1 Hydrological modelling

Models are used extensively to describe groundwater physical systems and can assist decision makers to act effectively. However, an exact model of the natural system is not feasible. A plethora of modelling techniques has been developed to simulate the natural system within some acceptable accuracy.

Mathematical modelling has a long history; its roots lie in the natural laws formulated throughout the years. Parameters involved in these rules must be defined through a calibration process based on the observed field data. The system description (equations) together with the calibrated parameters can serve as a model. Due to the formulation of a natural system with approximate reasoning, an imprecise solution is derived. In formulating the system's characteristics, restricting assumptions must be made, so that the process can be dealt with mathematically [*Abbott and Refsgaard*, 1996].

Models used for the simulation of an aquifer behavior can be divided into three categories: Physically based, Conceptual and Data-driven models [*ASCE*, 2000].

Physically based models use the general principles of the physical processes and describe the natural system in detail. A prerequisite in using these models is the extensive knowledge of the aquifer parameters and of the processes taking place within the aquifer. These models commonly use finite differences and/or finite elements representations of partial differential equations of mass and momentum. They do not require lengthy time series of hydrometeorological data, but they require the evaluation of a large number of parameters and initial conditions [*Abbott and Refsgaard*, 1996]. Conceptual models, often called grey-box models are composed of a series of interconnected storages representing physical elements in a catchment. Parameters typically represent mean values over the entire catchment. The equations used are semi-empirical, but they still lay their basis in physical processes. They are simple and easy to implement, but they require a large amount of field data for the calibration process.

Parallel to the advances in mathematical modelling, the technological revolution enabled the development of a new modelling technique, data-driven models. Taking advantage of the increased computer speed and of the cheaper data acquisition/storage/transmission, new branches of science have emerged. One of these, known as machine learning, is primarily concerned with learning patterns from observed data. Many of the natural systems are not known with sufficient precision to be modeled accurately. If observed data are available, a model representing the system can be built by discovering regularities in the data. This is the reason why they are commonly referred to as black-box models; because they do not take into consideration the processes happening within the system, but only their result. Depending on their origin they can be divided into three main groups, namely empirical hydrological methods, statistically based methods and hydroinformatics based methods [*Abbott and Refsgaard*, 1996]. These three categories can be distinguished by the following:

- Empirical hydrological methods lay their basis in empirical methods, but they use a selection of input-output parameters which are dictated by some physical understanding of the processes.
- Statistically based methods consist of regression and correlation models, which are used to determine a relationship between input and output.
- Hydroinformatics is a branch of hydrological sciences that has a strong interest in the use of machine learning techniques. A hydroinformatics system is defined as an electronic knowledge encapsulator that models the real process and can be used for the simulation and analysis of physical, chemical and biological processes in the water for the better management of the aquatic environment. Over the past ten years such methods have been used extensively in groundwater modelling.

The classification of the hydrological models is depicted in Figure 2-1.

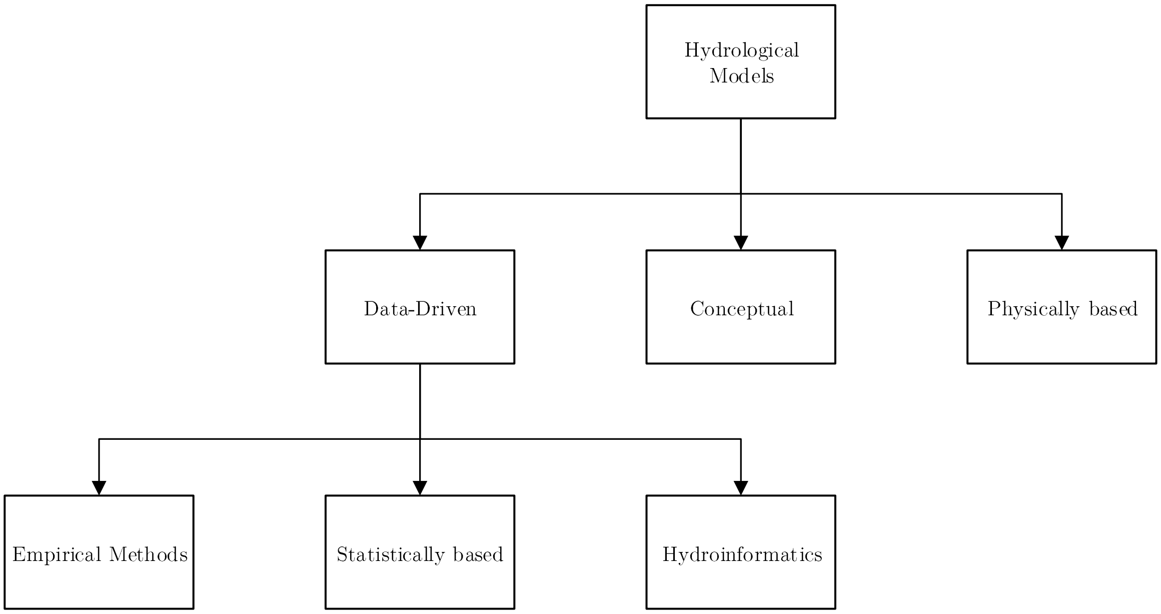


Figure 2-1: Classification of Hydrological models [*ASCE*, 2000]

2.2 Computational intelligence – Machine learning – Artificial intelligence

Artificial intelligence (AI) is the field of study that aims at the creation of machines or software with human-like reasoning and intelligence. The main goals of AI research includes knowledge, planning, learning, natural language processing, perception and the ability to move and manipulate objects. The first studies which included AI were developed in the 1950s, however the concept was introduced a long time before that [*Haykin*, 1999].

Later on there was a shift in AI research from its biological connection towards providing practical methods and tools in achieving solutions to engineering problems using the huge amount of data that is available in many situations [*Bhattacharya*, 2005]. This new line of research that has originated from AI has ties with several closely related fields: Computational Intelligence (CI), Machine Learning (ML) and Soft Computing (SC) [*Negnevitsky*, 2005].

These fields overlap in many cases and other classifications of the fields are possible.

Computational intelligence (CI) is a sub-branch of AI, studying adaptive mechanisms to enable intelligent behavior in complex environments. These mechanisms include AI paradigms that exhibit an ability to learn and adapt to new situations, to generalize and associate [*Engelbrecht*, 2007].

Machine learning (ML) is an interdisciplinary subject, which is enriched with concepts drawn from diverse fields, such as statistics, information technology, cognitive science, control theory and others, and is primarily concerned with learning from data. Many natural systems are not known with sufficient precision to be modelled accurately. If observed data are available then ML can be used to build a machine representing the system, by discovering regularities in the data [*Bhattacharya*, 2005]. Numerous research applications of ML have established its importance as a complement to the traditional physically based modelling. It is presently being integrated with almost all branches of science as an alternative complementary way of encapsulating knowledge.

Soft Computing (SC) includes methods which are more tolerant to imprecision and uncertain data. It became a formal area of study in Computer Science in the early 1990s [*Zadeh*, 1994]. SC techniques reflects human thinking more closely than traditional techniques, which are based on formal logical systems [*Negnevitsky*, 2005; *Reddy et al.*, 2014].

Artificial Neural Networks (ANNs), which are used extensively in this study, overlap these three categories and are mostly considered as an ML oriented method, while Fuzzy Logic (FL) is mostly categorized with SC methods. The relationship between Artificial intelligence algorithms is illustrated in Figure 2-2.

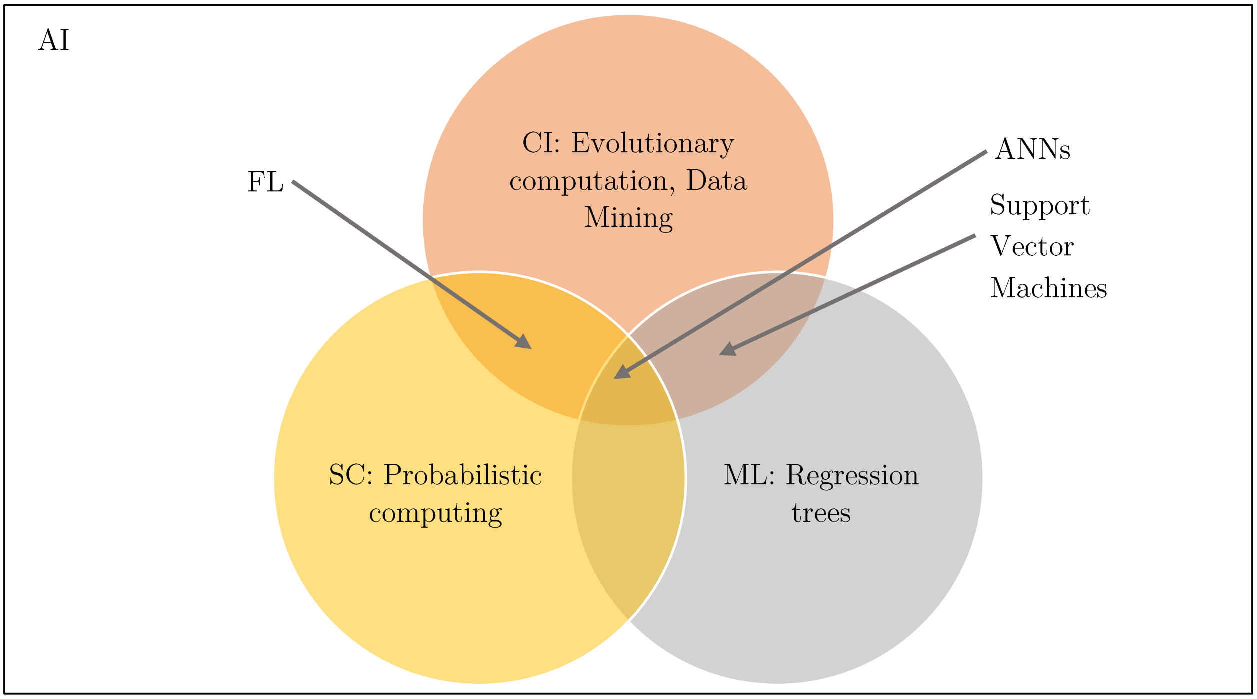

AI
CI: Evolutionary computation, Data Mining
FL
ANNs
Support Vector Machines
SC: Probabilistic computing
ML: Regression trees

Figure 2-2: Relationship between AI algorithms

2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) were constructed as an attempt to simulate, by mathematical means, an idealized form of the elementary processing units in the human brain and their interconnections, signal processing and self-organization capabilities [*Tettamanzi and Tomassini*, 2001]. According to *Schalkoff* [1997], an artificial neural network is *"a structure composed of a number of interconnected units. Each unit has an input/output characteristic and implements a local computation or function. The output of any unit is determined by its input/output characteristics, its interconnection to other units and external inputs. Although 'hand crafting' of the network is possible, the network usually develops an overall functionality through one or more forms of training".*

2.3.1 Historical overview

Artificial neural networks were first developed by *McCulloch and Pitts* [1943], who were inspired by the desire to understand the human brain and to emulate its functionality. By the 1960s the interest for ANNs was high, until 1969 when *Minsky and Papert* [1969] showed the limitations of the Perceptron. The research interest gradually declined until 1982-85 when *Rumelhart et al.* [1985] introduced a new theoretical framework for drafting the error propagation algorithm reverse (back propagation). Since then, artificial neural networks have found application in various fields such as computer science, electrical engineering, and robotics.

In recent decades, more complex algorithms have been developed and, with the advent of more powerful computational tools, the application of ANNs has grown in different cases. The ability of neural networks to recognize relationships between input and output makes it possible to address complex problems.

2.3.2 Artificial neurons

Neurons are the key component of an ANN. They are elementary units which, when connected, they construct an ANN.

The first model of a neuron, named Threshold Logic Unit (TLU), was proposed by *McCulloch and Pitts* [1943]. A graphical representation of such a unit with n real-valued inputs x_i, each associated with a parameter w_i, is presented in Figure 2-3.

Parameter w_i is also known as synaptic weight, in analogy with biological synapses, the functional contacts between two nerve cells. A TLU performs a weighted sum operation (u) followed by a non-linear threshold operation (f). The weighted sum is also called the neuron activation and the thresholding operation is the activation function.

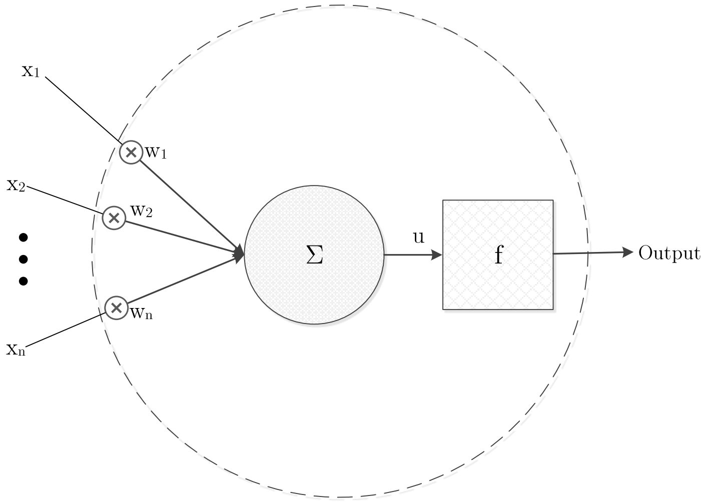


Figure 2-3: Artificial neuron model

Mathematically, the operations in one node can be described by Equations 2.1 – 2.2.

$$u = \sum_{j=1}^{n} w_j x_j \tag{2.1}$$

$$y = f(u) \tag{2.2}$$

2.3.3 Activation functions

Some of the most common activation functions are described below.
The *Threshold* activation function is described by Equation 2.3.

$$f(u) = \begin{cases} 1 & if\ u \geq 0 \\ 0 & if\ u < 0 \end{cases} \tag{2.3}$$

The graph for the above threshold activation function is presented in Figure 2-4.

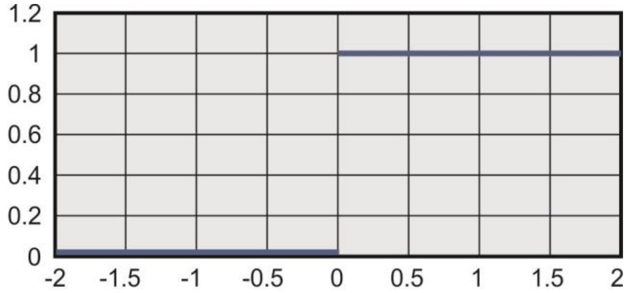


Figure 2-4: Threshold activation function

When using this model, the output of a node is equal to one when the input is positive and zero otherwise [*Kröse and Van Der Smagt*, 1993].

The *Partially linear* activation function is described by Equation 2.4 and a typical representation is given in Figure 2-5.

$$f(u) = \begin{cases} 1, & u \geq +\frac{1}{2} \\ u + \frac{1}{2}, & +\frac{1}{2} > u > -\frac{1}{2} \\ 0, & u \leq -\frac{1}{2} \end{cases} \tag{2.4}$$

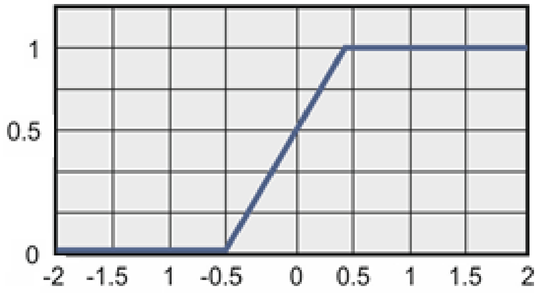


Figure 2-5: Partially linear activation function

The *Sigmoid* is one of the most common activation functions used in ANNs. A characteristic example of sigmoid activation function is the logistic function described by Equation 2.5.

$$f(u) = \frac{1}{1 + e^{-au}} \tag{2.5}$$

Parameter a in Equation 2-5 represents the slope of the sigmoid curve (Figure 2-6). The larger the value of a, the steeper the curve is [*Haykin*, 1999] .

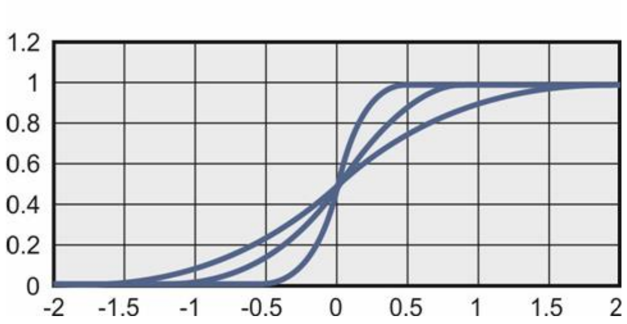


Figure 2-6: Sigmoid activation function for different values of parameter a

The sigmoid activation function is suitable in most cases because it is differentiable and it saturates, that is, it tends asymptotically to zero and one at the extremes.

2.3.4 Networks of artificial neurons

By connecting multiple Threshold Logic Units (TLUs), an interconnected network of neurons can be constructed. The computational capabilities of such a network are by far more useful than those of a single neuron. The way in which the neurons are interconnected defines the overall architecture of the network. Many different network architectures exist but the most important and well-known one is the feedforward multilayer network architecture, due to its simplicity and computational capabilities. In feedforward networks, an input pattern is transformed to an output pattern through a finite series of layers of nodes, some of which may have no connections to the input or output. In feedforward networks there are no feedback signals, i.e., the signal travels only forward (Figure 2-7).

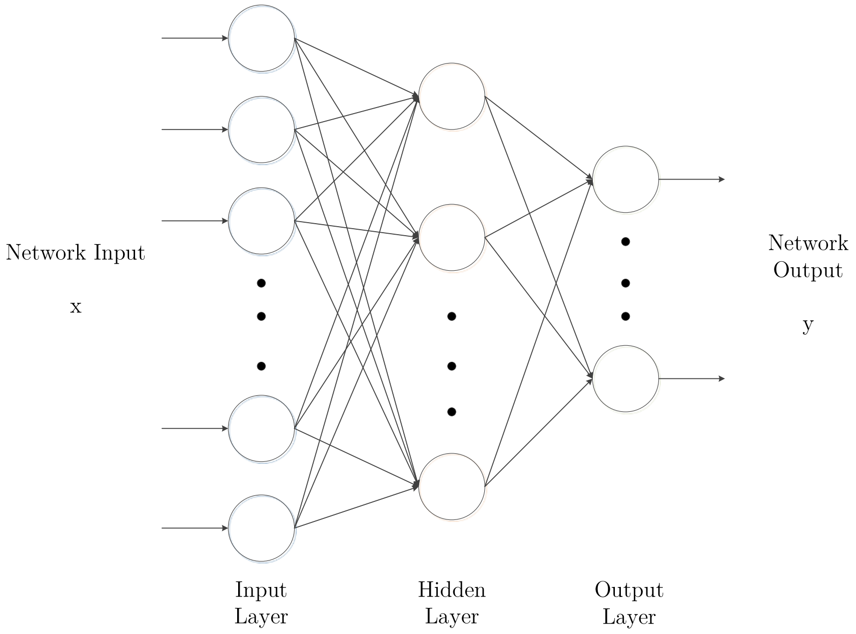


Figure 2-7: Typical Artificial Neural Network

The optimal architecture of an ANN has been extensively discussed in the past [*Gómez et al.*, 2006; *Mirchandani and Cao*, 1989] and there is not a universal rule that can be applied in all cases.

2.3.5 ANN training

In order for an ANN to acquire the appropriate synaptic weights w, the training process of the network must first take place. This procedure implies the incremental modification of the weights so as to improve a prespecified performance criterion. The adaptation of the weights may take place in a supervised or unsupervised way.

In supervised learning the network is presented with a set of known input/output pattern pairs. The learning procedure consists of updating the weights of the training algorithm at each iteration so that, for a given input, the error measure between the network's output and the known target value is reduced. This procedure is mainly used for regression problems. When using unsupervised learning, an input and the corresponding output are vital, in order to establish a relationship for the ANN to learn. In this case, there is no feedback provided on whether the given association is correct or not. It is mainly used for classification problems. In unsupervised learning, the categories in which data fall within are not known beforehand, and the learning process attempts to find appropriate categories for them, while in supervised learning these categories are known.

In the case of regression problem, the use of supervised learning is the most common. In order to train a network as such, the most common algorithms used are the generalized delta rule and the backpropagation training algorithms.

Generalized Delta Rule

Generalized Delta Rule (GDR) is a product-learning rule for a feedforward, multi-layer, structured neural network that uses learning by error correction. Network weights are adjusted to minimise an error based on a measure of the difference between target and calculated network output. The learning rule tries to reduce the Mean Squared Error (MSE) averaged over the training set [Kröse and Van Der Smagt, 1993]. The criterion function to be minimized is expressed in Equation 2.6.

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{m}(d_i - y_i)^2 \qquad 2.6$$

where d_i is the target output, y_i the calculated ANN output, m is the number of training sets and $\boldsymbol{w}$ is the vector of weights and $E(\boldsymbol{w})$ represents the error with a given set of weights. When training an ANN using the GDR, the idea is to make a change in weights proportional to the negative of the derivative of the error as measured on the current pattern with respect to each weight.

Taking the partial derivatives in Equation 2.6 with respect to the weights, the iterative form of the rule is formulated, as described by Equation 2.7 [Kröse and Van Der Smagt, 1993].

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \mu\,(d_k^t - y_k^t)\cdot \boldsymbol{x}_k^t \qquad 2.7$$

where μ is the learning rate; a parameter that controls the stability and the rate of convergence, $\boldsymbol{x}$ and $\boldsymbol{w}$ are the input and weight vectors and t and $t+1$ are the current and next iteration steps respectively.

Backpropagation algorithm (BP)

The most commonly used ANN training method is backpropagation. The algorithm of error back propagation is a general method for calculating the weights of a network with more than one layer. The aim of the method is to train the weights based on the set of available input-output values, decreasing at each iteration of the algorithm the output error of the neural network. Initially, each

input vector propagates through the network and generates an output value. The network output value is compared with the desired value and the error is calculated. Then, this error propagates backwards, i.e., from the output level to the input level, and the weights of each synapsis are modified [*Kröse and Van Der Smagt*, 1993].
The steps of the algorithm are as follows:

1. Initialize randomly all unit weights in the network.
2. Apply an input set to the network.
3. Feed forward the inputs to determine all unit outputs.
4. Compare unit responses in the output layer with the desired or target response.
5. Compute and propagate an error sensitivity measure backwards through the network, using this as the basis for weight correction (Equation 2.7).
6. Minimize the overall error for each training set through unit weight adjustment.

2.3.6 Advantages and disadvantages of ANNs

Advantages

1. Non linearity: An ANN can be either linear or not depending on the activation function used. Non linearity can be a great advantage, especially in cases where a natural procedure is described (such as water level fluctuation).
2. Input-output mapping: After the training of the network, the user can have an overview of the correspondence between input and output values, by examining the neural weights. It is also possible, especially in small networks, to examine which parameters affect the output more significantly.
3. Adaptivity: ANNs have the ability to adapt their synaptic weights to environmental variations. This can be done by retraining the network in order to incorporate all immerging new trends.
4. Tolerance in errors: ANNs are tolerant to errors. Their performance will drop gradually in cases of datasets with extreme values, but they will still be able to provide a reasonable output.
5. Very-large-scale Integration (VLSI) implementability: The parallel architecture of an ANN makes it very fast. This feature makes it ideal for developing applications with very high technology systems integration [*Haykin*, 1999].
6. ANNs can be developed using various training methods and architectures depending on the application.

Disadvantages

1. ANNs behave as black boxes so they have limited ability to identify causal relationships.
2. Depending on the application, they may require great computational resources.
3. They are prone to overfitting.
4. Their development, in terms of architecture, can be empirical with various different rules in bibliography, and the best possible architecture cannot be easily identified [*Tu*, 1996].

2.3.7 ANNs in hydrology

Machine learning techniques have been used in the past in hydrological studies. An overview of these techniques and their implementations was presented by *See et al.* [2007]. Data-driven hydrological models used in the recent bibliography are also presented in a paper by *Solomatine and Ostfeld* [2008].

ANNs have been used extensively both in surface and groundwater hydrology. The purpose of these networks is to simulate the water budget, described by Equation 2.8,

$$\mathrm{\Delta S = I - O + P - EPT \pm Q} \qquad 2.8$$

where ΔS is the change in aquifer storage, I the inflow to the water basin, O the outflow from the water basin, P the precipitation, EPT the evapotranspiration and Q the pumping/recharge rate.

Rainfall-runoff processes have been successfully simulated using ANNs together with other methodologies (principle component analysis, timing error correction) in multiple studies in the past [*Abrahart et al.*, 2007; *Hu et al.*, 2007]. ANNs have also been used in surface hydrology in order to forecast floods [*Chen et al.*, 2013], to predict the frequency of floods in a region [*Aziz et al.*, 2014], and to calculate the flood resistance in existing cities [*Cruz et al.*, 2014].An overview of studies and methodologies where ANNs have been used with hydrological data was presented by *ASCE* [2000], *Maier and Dandy* [2000] and *Govindaraju* [2000].

As far as groundwater is concerned, ANNs have been used for diverse purposes. One of the most common purposes is for the definition of aquifer parameters. *Ajmera and Rastogi* [2009] used ANNs together with finite element models so as to solve the inverse problem and calculate the transmissivity of an aquifer. On the same note, *Samani et al.* [2007] used ANNs together with Principle Component Analysis to define the parameters of an aquifer (storativity, transmissivity and therefore drawdown). *Rizzo and Dougherty* [1994] characterized aquifer parameters spatially, by introducing a new concept, Neural Kriging. In this case the ANN is used as a spatial interpolation method and the structure of the ANN has many common points with the kriging interpolation method, hence, the name of the methodology. *Banerjee et al.* [2011] used ANNs in order to predict the salinity in an aquifer and to determine a safe pumping rate to maintain an acceptable groundwater salinity level. Their results were compared with the analytical results of a mathematical model, as well as with field measurements. They concluded that ANNs can be simpler in their implementation while being more accurate than mathematical models.

Finally, hydraulic head and hydraulic head change was also the focus of many studies using ANNs. Various input parameters have been used, including temperature, pumping rate, rainfall and conductivity [*Coppola et al.*, 2005; *Lallahem et al.*, 2005; *Nayak et al.*, 2006; *Tapoglou et al.*, 2013]. The architecture of ANNs compared to their performance and computational cost, was studied by *Mohanty et al.* [2010] and *Taormina et al.* [2012], using as output parameter the groundwater levels with hourly time step. *Trichakis et al.* [2011b] used hydraulic head change instead of hydraulic head, in order to increase the accuracy of the model. This model was developed in a karstic aquifer, which is considered highly complex and difficult to simulate with mathematical models. The ANN model yielded very good simulation results, while the confidence intervals were also calculated for the prediction by using a bootstrap methodology [*Trichakis et al.*, 2011a; *Trichakis et al.*, 2009].

In some cases the performance of ANNs for hydraulic head simulations was compared with that of other modelling techniques. *Mohanty et al.* [2013] compared the performance of an ANN with a commercial numerical model, MODFLOW, with the ANN model providing better predictions for the groundwater level over shorter time horizons. In another study [*Chatzakis et al.*, 2014], ANN performance was compared to the PTC numerical model.

2.4 Geostatistics

Geostatistics is a subcategory of statistics and refers to a collection of statistical techniques specialized to analyze and interpret spatially referenced data. Spatially referenced data are represented by triplets (x, y, z) where x and y are the spatial coordinates and z is the value of interest at location (x, y). Characteristic examples of this kind of data are the elevation of the piezometric surface over the areal extent of an aquifer [*Haan*, 2002].

The problems usually addressed by geostatistics are:

1. Point estimates at locations where measurements are not available
2. Estimates for the areal averages over the entire area or a part of it
3. Estimates of the uncertainty associated with point or areal average estimates
4. The distribution of values over areas of various sizes and locations

Spatial prediction models can be classified based on several aspects. Most importantly they can be classified according to the amount of statistical analysis included. According to *Hengl* [2007] two main categories can be identified:

- Mechanical/Empirical models: For these models arbitrary or empirical parameters are used. No estimate of the model error is available and usually no strict assumptions about the variability of a feature exist. Amongst the most known techniques that belong to this category are Thiessen polygons, and inverse distance interpolation.
- Statistical (Probability) models: In this case, the model parameters are estimated in an objective way, following the probability theory. The advantage of these models is that the predictions are accompanied with the estimate of the prediction error. There are at least three important categories of statistical models, namely, kriging, regression-based models and Bayesian-based models.

The Kriging interpolation method, which is used in this thesis, belongs to the second category. The principles of this method are presented in the next section.

2.5 Kriging interpolation

Kriging has been used for many decades as a synonym for geostatistical interpolation. It originated in the mining industry in the early 1950s as a means of improving ore reserve estimation. The technique was first published by *Krige* [1951], but it took almost a decade until the mathematician *Matheron* [1963] derived the formulas and basically established the whole field of linear geostatistics.

2.5.1 Estimation

A standard version of kriging is called Ordinary Kriging (OK) where the prediction of a parameter z at location s, $z(s)$, is based on the model given by Equation 2.9.

$$z(s) = \mu + \varepsilon'(s) \tag{2.9}$$

where μ is the constant stationary function (global mean) and $\varepsilon'(s)$ is the spatially correlated stochastic part of variation.

The predicted value at point s_o, $\hat{z}_{OK}(s_o)$ is a linear combination of n weighted function values at nearby sampling points, as given in Equation 2.10.

$$\hat{z}_{OK}(s_o) = \sum_{i=1}^{n} w_i(s_o) \cdot z(s_i) = \lambda^T \cdot z \tag{2.10}$$

where λ is the vector of kriging weights (w_i) and z is the vector of n observations at primary locations.

The kriging weights can be calculated using data at sampling points and constructing the appropriate variogram.

2.5.2 Variogram

The basic tool in geostatistics is the variogram, which is used to quantify spatial correlations between observations. Once a mathematical function has been fitted to the experimental variogram, this model can be used to estimate values at unsampled points.

The novelty that *Matheron* [1963] and *Gandin* [1963] introduced to the analysis of point data is the derivation and plotting of the semivariances. Semivariance is a measure of the dispersion of the observed data that fall below a target value in the data set [*Hengl*, 2007]. It represents the difference between neighboring values and can be calculated using Equation 2.11.

$$\gamma(h) = \frac{1}{2} E[(z(s_i) - z(s_i + h))^2] \tag{2.11}$$

where $z(s_i)$ is the value of target variable at some sampled location and $z(s_i + h)$ is the value of the neighbor at distance $s_i + h$. Suppose that there are n point observations, this yields $n \cdot (n-1)/2$ pairs for which a semivariance can be calculated. From the known values at various locations throughout the study area (Figure 2-8 a), it is possible to plot all semivariances versus their distances, which will produce a variogram cloud (Figure 2-8 b). Then, for the ease of interpretation, values are averaged for a standard distance called lag (Figure 2-8 c). This comprises the standard experimental variogram. What it is usually expected in this plot is for semivariances to be smaller at short distances, and to stabilize at some distance. Values of a specific variable, in this case the semivariance between data points, are similar at short distances, up to a certain distance where these values tend to reach a global value.

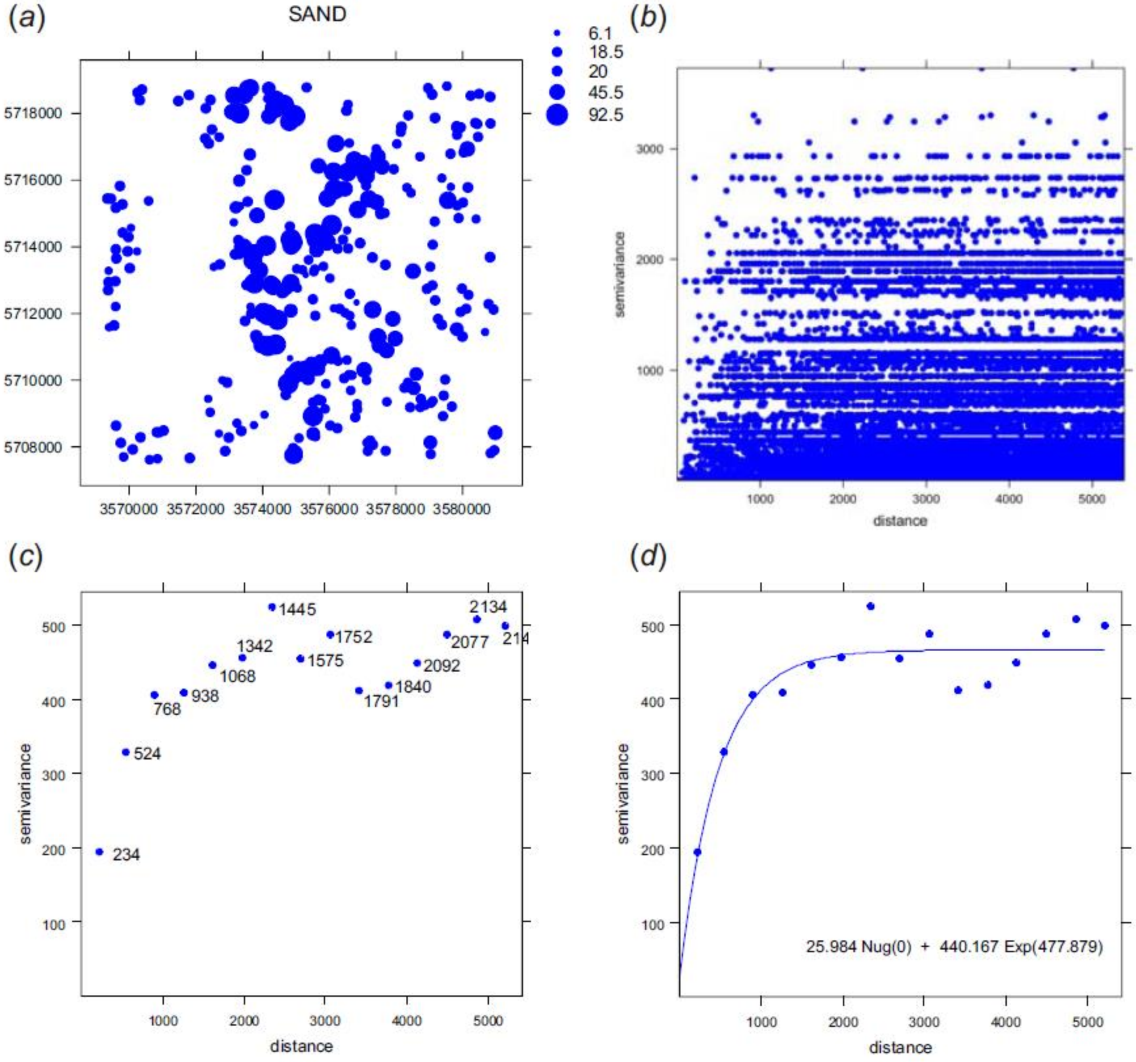


Figure 2-8: Steps in variogram model determination [*Hengl*, 2007]

Once the experimental variogram is calculated, it must be fitted to an authorized variogram model (Figure 2-8 d). Different variogram models, from the more traditional described in the following, to the more sophisticated like the Spartan and the Matern have been tested over the years. According to *Kitanidis* [1997], characteristic variogram models, suitable for hydrological data, are the linear, exponential, spherical and power-law (Table 2-1).

Table 2-1: Common authorized variogram models

Model	Equation
Linear	$\gamma(h) = C_o + b \cdot h$
Exponential	$\gamma(h) = \begin{cases} C_o + C_1[1 - e^{-\frac{h}{r}}], & h > 0 \\ C_o + C_1 \quad , & h = 0 \end{cases}$
Spherical	$\gamma(h) = \begin{cases} C_o + \left(\frac{3}{2}\frac{h}{r} + \frac{\frac{1}{2}h^3}{r^3}\right)C_1 \quad , & 0 <\ h < r \\ C_o + C_1 \quad , & h > r \end{cases}$
Power-Law	$\gamma(h) = C_o + th \cdot h^{2H}$

where $\mathrm{C_0} + \mathrm{C_1}$ is the sill, $C_o$is the nugget effect, r is the range of the variogram, th and b are coefficients and $2\mathrm{H}$ is the Hurst coefficient.

The parameters defined by fitting the theoretical variogram model to the experimental one are: range, nugget and sill (Figure 2-9). Range is the distance after which data are no longer correlated and is associated with the distance where the variogram levels off to the sill. The nugget effect is related to the amount of short range variability in the data and represents the distance at which the variogram intersects the y-axis. The value must be calculated using the first few empirical variogram points for the best fit. A nugget that is large relative to the sill is problematic and could indicate too much noise and not enough spatial correlation. The sill is the total variance where the empirical variogram appears to level off. Variogram points above the sill indicate negative spatial correlation, while points below the sill indicate positive correlation.

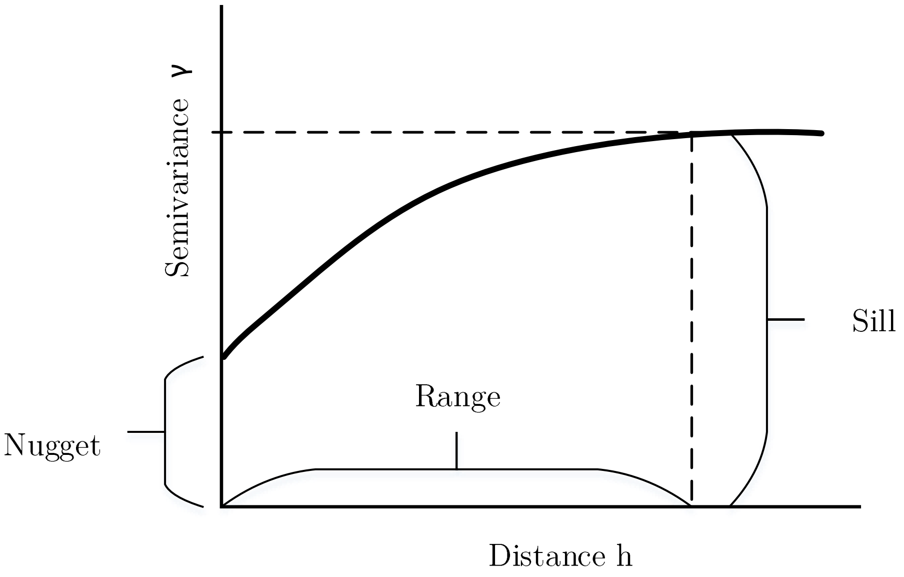


Figure 2-9: Characteristic variogram model parameters

The covariance function for different variogram models is given in Table 2-2 [*Kitanidis*, 1997; *Rasmussen and Williams*, 2006].

Table 2-2: Covariance function for different variogram models

Model	Equation
Linear	$c(h) = \sum_{d=1}^{D} b_d x_d x'_d$
Exponential	$c(h) = C_1 e^{-\frac{h}{r}}$
Spherical	$c(h) = \left(1 - \frac{3}{2}\frac{h}{r} + \frac{\frac{1}{2}h^3}{r^3}\right) C_1$
Power-law	$c(h) \propto h^{2H}, -1 < H < 0$

where d is the number of dimensions (in this case $d = 2$), x denotes the location is each dimension x' the location in each dimension of the closest data point and b a parameter different for every dimension.

The experimental variograms are commonly fitted to the theoretical, by iterative reweighted nonlinear least square estimation, using as stopping criteria either a sufficiently small step or a small function change, reaching a specified number of iterations or reaching some secondary boundary condition. The main difference between linear and nonlinear fitting is that in linear problems time complexity is fixed, whereas the solution of nonlinear problems is an iterative process, whose convergence speed is problem-dependent. Moreover, nonlinear methods generally have more tunable parameters, than linear ones and hence, lead to a better solution.

Using the estimated variogram, the weights can be calculated such that $\sum_{i=1}^{N} \lambda_i = 1$ and are given by Equation 2.12.

$$\boldsymbol{\lambda} = C^{-1} \cdot \boldsymbol{D} \qquad (2.12)$$

where $\boldsymbol{\lambda}$ is the vector of weights for $i = 1\ to\ N$ and μ for $i = N + 1$, μ is the Lagrange coefficient associated with the zero-bias constraint (Equation 2.13-a), Γ the matrix of covariances between data points (equation 2.13-b) and D is the vector of covariance between a prediction point and data points (equation 2.13-c).

$$\overset{(a)}{\lambda = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \mu \end{bmatrix}} \qquad \overset{(b)}{C = \begin{bmatrix} C_0 & \gamma(h_{12}) & & \gamma(h_{1n}) & 1 \\ \gamma(h_{21}) & C_0 & \cdots & \gamma(h_{2n}) & 1 \\ \gamma(h_{31}) & \gamma(h_{32}) & & \gamma(h_{3n}) & 1 \\ \vdots & & \ddots & & \vdots \\ \gamma(h_{n1}) & \gamma(h_{n2}) & \cdots & C_0 & 1 \\ 1 & 1 & & 1 & C_0 \end{bmatrix}} \qquad \overset{(c)}{D = \begin{bmatrix} \gamma(h_{p1}) \\ \gamma(h_{p2}) \\ \vdots \\ \gamma(h_{pn}) \\ 1 \end{bmatrix}} \quad (2.13)$$

The function $\gamma(\cdot)$ in equation 2.12-b and c is the variogram and it is used to relate the distance between observation and prediction points based on the data at the observation points.[*Varouchakis and Hristopulos*, 2013b]

The main advantage of statistical spatial prediction techniques, such as kriging, over other types of prediction and interpolation techniques, is that it can provide the prediction variance of the estimation. Kriging uses the described procedure in order to minimize the error variance. The error variance that remains at the end of kriging interpolation can be calculated by Equation 2.14 and it can be considered as a measure of associated uncertainty of the predicted values.

$$\hat{\sigma}^2 = \sum_{i=1}^{n} [\lambda_i \cdot \gamma(s_i - s_o)\,] + \mu \qquad (2.14)$$

2.5.3 Kriging in hydrological data

The interpolation technique of Kriging has been applied mostly for the spatial estimation of variables rather than the temporal, even though applications on spatio-temporal estimations are also common in recent literature [*Snepvangers et al.*, 2003; *Ta'any et al.*, 2009]. By taking into consideration the spatial distribution of data, kriging overcomes problems that other interpolation methods, such as the nearest neighbor method, the distance weighted method and polynomial interpolation may have. Basic concepts of the Kriging technique and its application to natural phenomena have been reviewed by the ASCE Task Committee on Geostatistical Techniques in Geohydrology [1990a; b], while *Kitanidis* [1997] analyzed the methodology involved when applying kriging to hydrological data.

Kriging has been used extensively in applications other than the ones it was originally developed for [*Stytz and Parrott*, 1993]. As mentioned before, kriging has been used in hydrology and more specifically in groundwater applications. *Delhomme* [1978] first presented a series of case studies where kriging was used in automatic contouring, in measurement network design and other applications. *Aboufirassi and Mariño* [1983] and *Kumar* [2006], amongst others, used various kriging algorithms to estimate the groundwater level in different areas. The most commonly used kriging methods are the Ordinary [*Yang et al., 2008]* and the Universal [*Aboufirassi and Mariño, 1983*]. Other more accurate kriging methods such as Kriging with external drift and with incorporation of various parameters in the trend [*Varouchakis and Hristopulos*, 2013a] are also widely used. Auxiliary variables, such as digital elevation models or other geographical and physical parameters, can also be used in order to improve the accuracy of the estimations [*Desbarats et al., 2002*]. Different methods of kriging are compared in *Marinoni* [2003], while kriging has been compared to other interpolation methods by *Varouchakis and Hristopulos* [2013b]. *Loukas and Vasiliades* [2004] used kriging, amongst other interpolation techniques, together with machine learning techniques, for the spatiotemporal drought prediction.

Other than for simulating the hydraulic head in an aquifer [*Gundogdu and Guney*, 2007], Kriging has been used for the design and assessment of a groundwater monitoring network [*Baalousha*, 2010].

2.6 Fuzzy logic

2.6.1 Theoretical background

Fuzzy logic was first proposed by *Zadeh* [1965] as a means of representing and manipulating data that are not precise but rather fuzzy. A fuzzy set is a set without clear or sharp boundaries. Unlike a conventional (crisp) set (Figure 2-10-A) where an object either belongs or does not belong to the set, a fuzzy set allows for partial membership (Figure 2-10-B). A fuzzy set is represented by a membership function, which specifies the degree of membership within the set. A membership function for a fuzzy set *A* on the universe of discourse X is defined as $\mu_A: X \rightarrow [0,1]$, where each element of X is mapped to a value between 0 and 1.

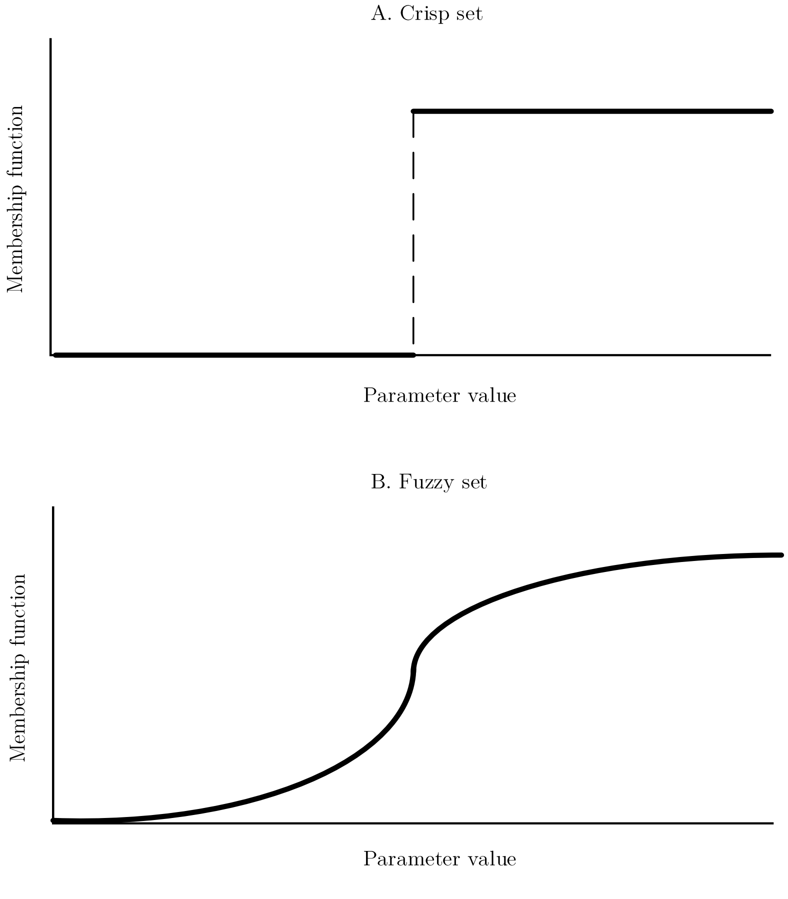


Figure 2-10: Typical Crisp and Fuzzy set

Membership functions allow to graphically represent a fuzzy set [*Negnevitsky*, 2005]. Because of the intermediate degrees of membership, a smooth transition from being a member to not being a member of a property can be achieved. The membership functions most commonly used are triangular, trapezoidal, Gaussian, generalized bell and sigmoidal Z- and S-functions [*Nguyen et al.*, 2010].

2.6.2 Building a fuzzy system

The typical process when developing a fuzzy logic system involves four steps: the fuzzification, rule evaluation, aggregation of rules and defuzzification (Figure 2-11).

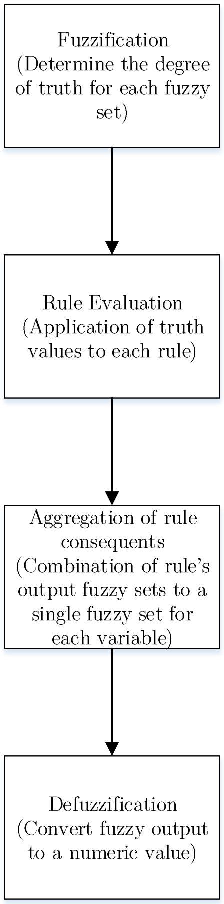
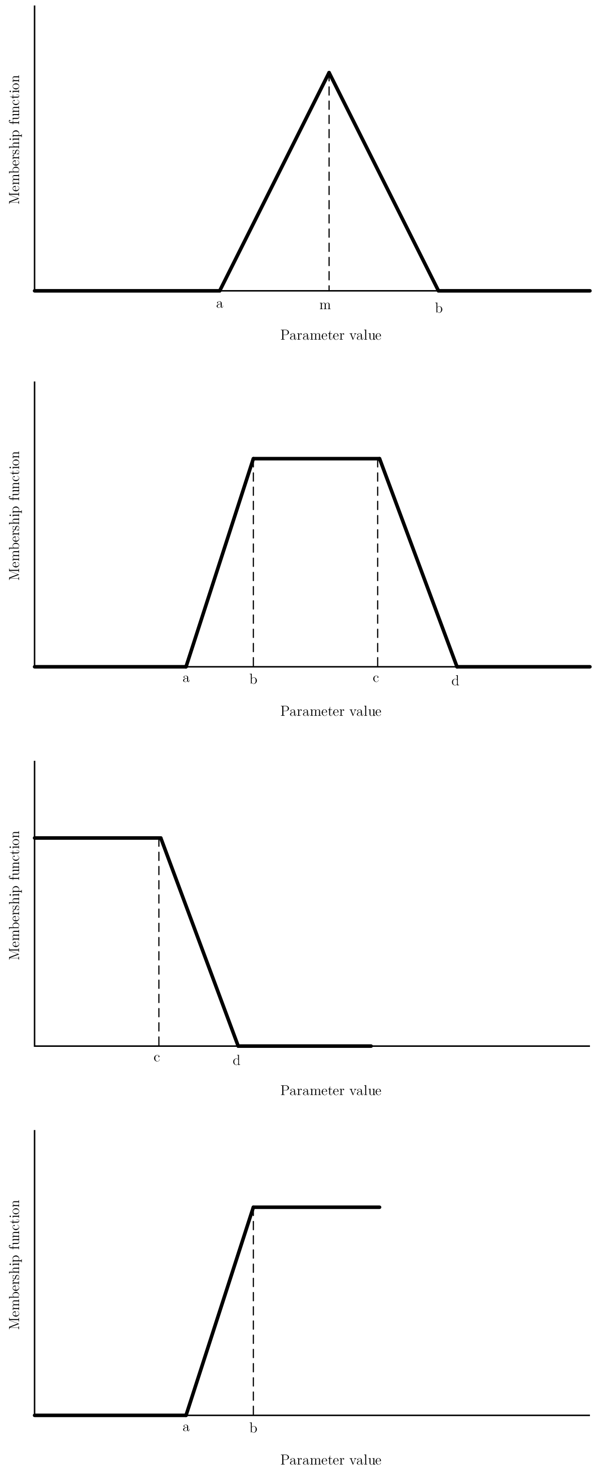

Fuzzification (Determine the degree of truth for each fuzzy set)
Rule Evaluation (Application of truth values to each rule)
Aggregation of rule consequents (Combination of rule's output fuzzy sets to a single fuzzy set for each variable)
Defuzzification (Convert fuzzy output to a numeric value)

Figure 2-11: Steps of fuzzy logic process

The first step in developing a fuzzy logic system is fuzzification where membership functions must be appointed to every linguistic function and the degree of truth for each of the fuzzy sets must be determined. Fuzzy set membership functions may have a variety of shapes (Figure 2-12). However, a triangular or trapezoidal shape can often provide an adequate representation of the expert knowledge. One of the key points is to maintain sufficient overlap in adjacent fuzzy sets for the fuzzy system to respond smoothly.



$$\mu_A(\chi) = \begin{cases} 0, & x < \alpha \\ \dfrac{x-a}{m-a}, & a < x \leq m \\ \dfrac{b-x}{b-m}, & m < x \leq b \\ 0, & x > b \end{cases}$$

$$\mu_A(\chi) = \begin{cases} 0, & x < \alpha \mid x > d \\ \dfrac{x-a}{b-a}, & a < x \leq b \\ 1, & b < x \leq c \\ \dfrac{d-x}{d-c}, & c < x \leq d \end{cases}$$

$$\mu_A(\chi) = \begin{cases} 0, & x > d \\ \dfrac{d-x}{d-c}, & c < x \leq d \\ 1, & x \leq c \end{cases}$$

$$\mu_A(\chi) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{b-a}, & a < x \leq b \\ 1, & x > b \end{cases}$$

Figure 2-12: Membership functions: triangular, trapezoid, R and L functions (from top to bottom)

Fuzzy sets, fuzzy rules and procedures must be encoded in order to incorporate fuzzy inference into expert system. During the fuzzification process takes, every value of a crisp input is mapped to a set of values expressing the degree to which the input belongs to each of the appropriate fuzzy sets. The numerical value of each input parameter must be converted into a probability of belonging in a Linguistic Subset (LS) of the fuzzy set. For example, in Figure 2-13 value v_1 has probability Mf1 of belonging to LS1, probability Mf2 of belonging to LS2 and zero probability of belonging to LS3. The same procedure is applied to all parameters involved.

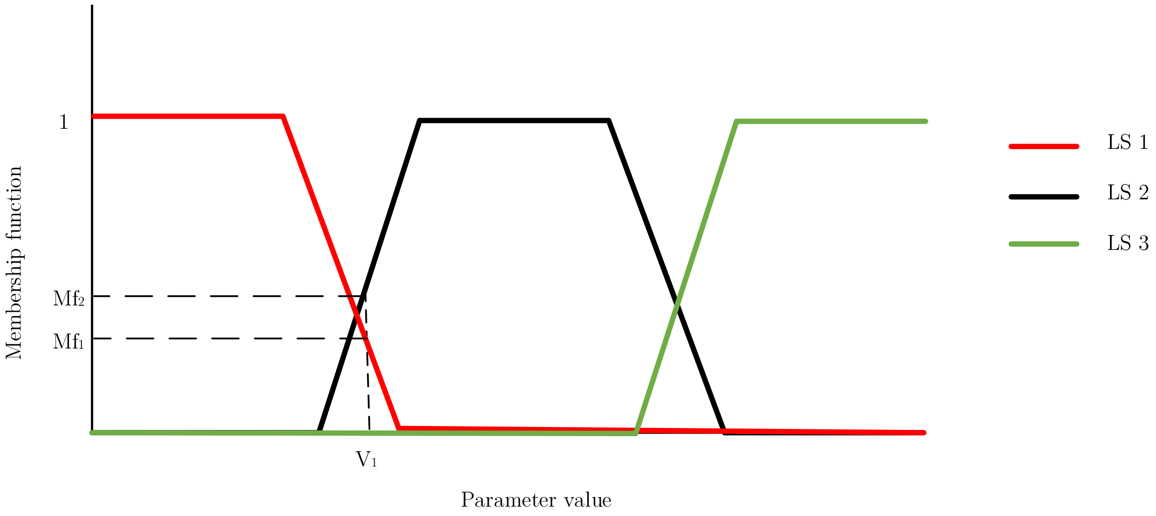


Figure 2-13: Fuzzification process

The next step is the rule evaluation where he truth value for the antecedents of each rule is computed and applied to the consequents of each rule. If a given rule has multiple antecedents, a fuzzy operator (e.g., AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number is then applied to the membership function. To evaluate the disjunction of the rule antecedents, we use the OR fuzzy operation. The most common operators used in linguistic fuzzy rules and their mathematical interpretation are presented in Table 2-3, where x and y are the probabilities calculated by the fuzzification process.

Table 2-3: Common fuzzy operators

Operator	Mathematical meaning
x AND y	$\min(x, y)$
x OR y	$\max(x, y)$
NOT x	$1 - x$

For example, consider two parameters, par1 and par2, whose values can fall into 3 categories each (L1, L2 and L3 for par1 and M1, M2 and M3 for par2) during the fuzzification process. If a rule states that the outcome is C1 when the par1 value belongs to L1 AND par2 value belongs to M3 then the probability that the outcome is C1 can be calculated using the operations in Table 2-3 and considering as x the probability $P(par_1 = L1)$ and y the probability $P(par_2 = M3)$. These probabilities are numerically defined during the fuzzification process. The final value of the probability of the outcome being C1 is expressed as: $P_{C1} = \min[P(par_1 = L1), P(par_2 = M3)]$.

The next step is the aggregation of the rule output, where the results of all the outputs are combined into a single fuzzy set for each output variable. Figure 2-14 shows how the output of a three-rule (C1, C2 and C3) system is aggregated into a single fuzzy set for the overall fuzzy output.

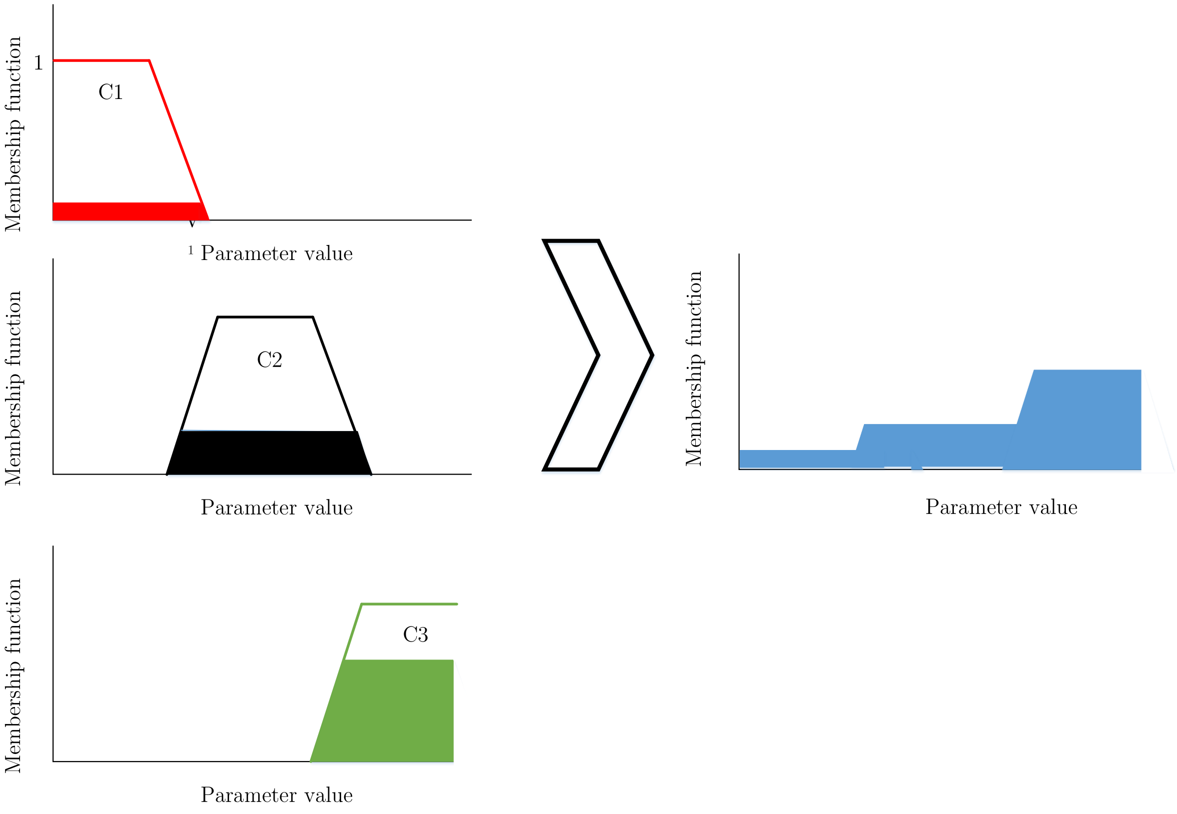


Figure 2-14: Aggregation of fuzzy rules

The last step is the defuzzification of the aggregate fuzzy set, where the fuzzy output set is converted to a numerical value. Numerous defuzzification methods are presented in the bibliography, the most popular being the centroid technique, which can be expressed mathematically by the Center of

Gravity (COG)[*Negnevitsky*, 2005]. The crisp output (COG) of the fuzzy system can be calculated using Equation 2.15:

$$COG = \frac{\int p_i(x)\, x\, dx}{\int p_i(x)\, dx} \tag{2.15}$$

where $p_i(x)$ is the aggregated membership function and x is the output variable.
A graphical representation of the defuzzification process is presented in Figure 2-15.

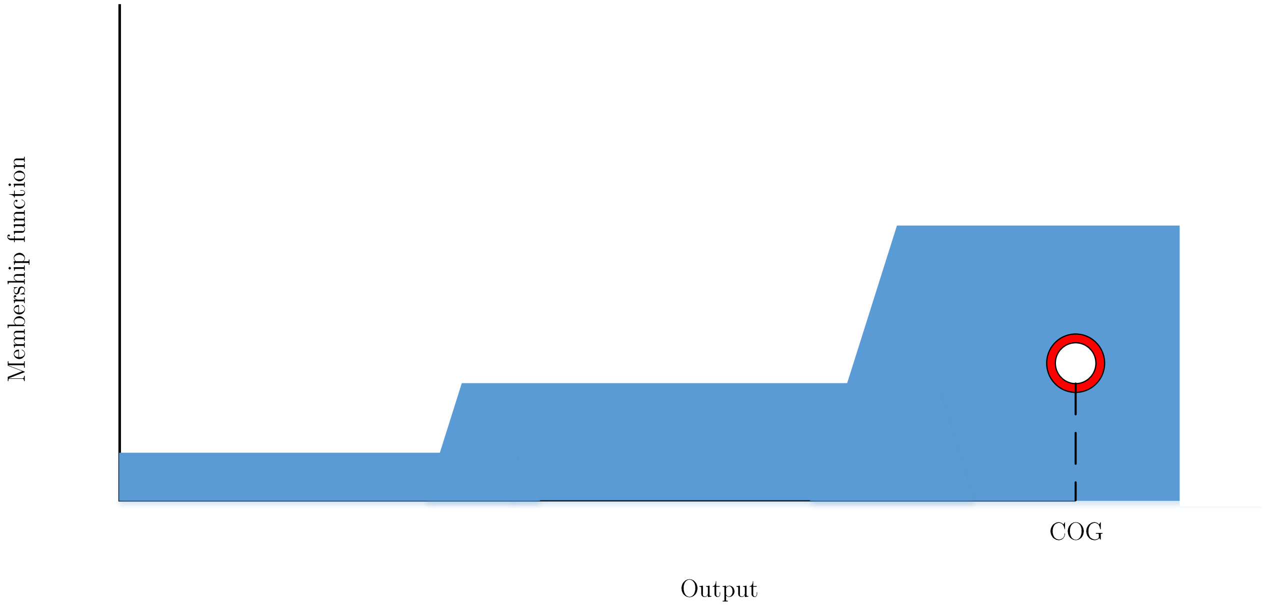


Figure 2-15: Defuzzification

The initial system should then be evaluated in terms of its results and fine-tuned in order to produce the optimal results.

2.6.3 Fuzzy logic in relevant applications

Fuzzy logic has been used in the past in multiple hydrological applications. Regression is one of the main applications, where a loose relation between parameters is known to exist, but only few data points are available. *Bogardi et al.* [1982] related the governing flow parameters to geological parameters so as to be able to describe groundwater flow using limited geological data. Fuzzy rule systems can replace traditional hydrological modelling tools. *Hundecha et al.* [2001] and *Nayak et al.* [2004] have both used fuzzy logic systems to replace rainfall-runoff models.

Fuzzy logic has been used with ANNs, creating Neuro-Fuzzy networks, which incorporate in ANNs the human-like reasoning style of fuzzy systems by using fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules [*Nayak et al.*, 2004; *Zounemat-Kermani and Teshnehlab*, 2008]. Fuzzy logic has been also used with the Kriging algorithm for the experimental semivariogram definition [*Amini et al.*, 2005; *Rao and Prasad*, 1982].

2.7 Measuring model performance

Various techniques can be used to evaluate the performance of the methodology deployed. At a first level, the training and testing errors derived during ANN training and evaluation together with the error variance in kriging can provide good initial error estimation. However, in order to get a better insight into the modeling performance, the leave-one-out cross-validation method was applied. More specifically, for 10% of the time steps, well locations (for which field data are normally available) are treated as prediction points and the hydraulic head change is calculated using the kriging algorithm described above. For the rest 90% of the time steps, the simulation result is equal to output of the respective ANN. In this way, the error can be estimated for all methods applied. Different error indicators, defined in Table 2-4, are used in the analysis of the results.

Table 2-4: Cross-validation error indicators

Cross-validation error indicators	Equation	Optimal values
Root mean squared error (RMSE)	$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert z^*(s_i) - z(s_i)\rvert^2}$	0
Mean absolute error (MAE)	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert z^*(s_i) - z(s_i)\rvert$	0
Bias	$Bias = \frac{1}{N}\sum_{i=1}^{N}[z^*(s_i) - z(s_i)]$	0
Root mean square standardized error (RMSSE)	$RMSSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{z^*(s_i) - z(s_i)}{\sigma(s_i)}\right]^2}$	1

where *N*: number of samples, $z(s_i)$: measured value at point s_i, $z^*(s_i)$: predicted value at point s_i, $\sigma(s_i)$: the standard deviation at point s_i

In addition to the training and testing errors, the Nash-Sutcliffe Efficiency (NSE) coefficient was used to assess ANN efficiency (Equation 2.16):

$$NSE = 1 - \frac{\sum_{t=1}^{N}[q_{obs}(t) - q_{sim}(t)]^2}{\sum_{t=1}^{N}[q_{obs}(t) - \bar{q}_{obs}]^2} \tag{2.16}$$

where $q_{obs}(t)$ is the observed value at time step t, $q_{sim}(t)$ is the simulated by the model value at time step t and $\bar{q}_{obs}$ is the mean observed value over the entire simulation period of length N time steps.

The NSE coefficient can take values from $-\infty$ to 1. An NSE value of 1 (NSE = 1) indicates a perfect fit between the simulated and observed values of the target output. An NSE value of 0 (NSE = 0) indicates that the simulated values are as accurate as the mean of the observed values, whereas an NSE value less than 0 (NSE < 0) indicates a questionable choice of model [*Schaefli and Gupta*, 2007].

The methodologies described in this chapter have been used successfully in the past for the hydraulic head simulation of an aquifer. ANNs can provide good results for the temporal simulation of the hydraulic head, but only for point locations, while kriging has been constructed for the spatial interpolation of geological parameters. However, to the best of our knowledge the combination of the above methodologies has not been applied for the spatial and temporal simulation of groundwater parameters. In the present dissertation, ANNs, kriging and fuzzy logic are employed for the spatio-temporal simulation of groundwater hydraulic heads.

Chapter **3**

Methodology

The methodology followed in this work is described in this chapter. The first step is the development of a distinct ANN for every data location where hydraulic head data were available. If these results are going to be used for the spatial and temporal simulation of the hydraulic head, the next step is to develop a fuzzy logic system in order to define the appropriate neighborhood for every prediction point involved. When the fuzzy logic system is not used, the range of the corresponding variogram is used to define the wells that comprise the neighborhood. The final step is the implementation of the kriging algorithm and the visualization of the results. In this procedure several problems were encountered and solved in order to develop an integrated simulator suitable for the spatial and temporal prediction of the hydraulic head in an aquifer. The overall methodology is depicted in Figure 3-1.

The methodology developed and the parameters involved in the above mentioned steps will be analyzed in this chapter. The algorithms were created in Visual Basic .NET using Visual Studio 2010, while MATLAB R2013a was used for the visualization of the results through the .NET interoperability. This resulted in a standalone program, able to run on any computer without requiring a Visual Studio or MATLAB license, but only the installation of the appropriate MATLAB libraries (MATLAB Compiler Runtime) which are freely available.

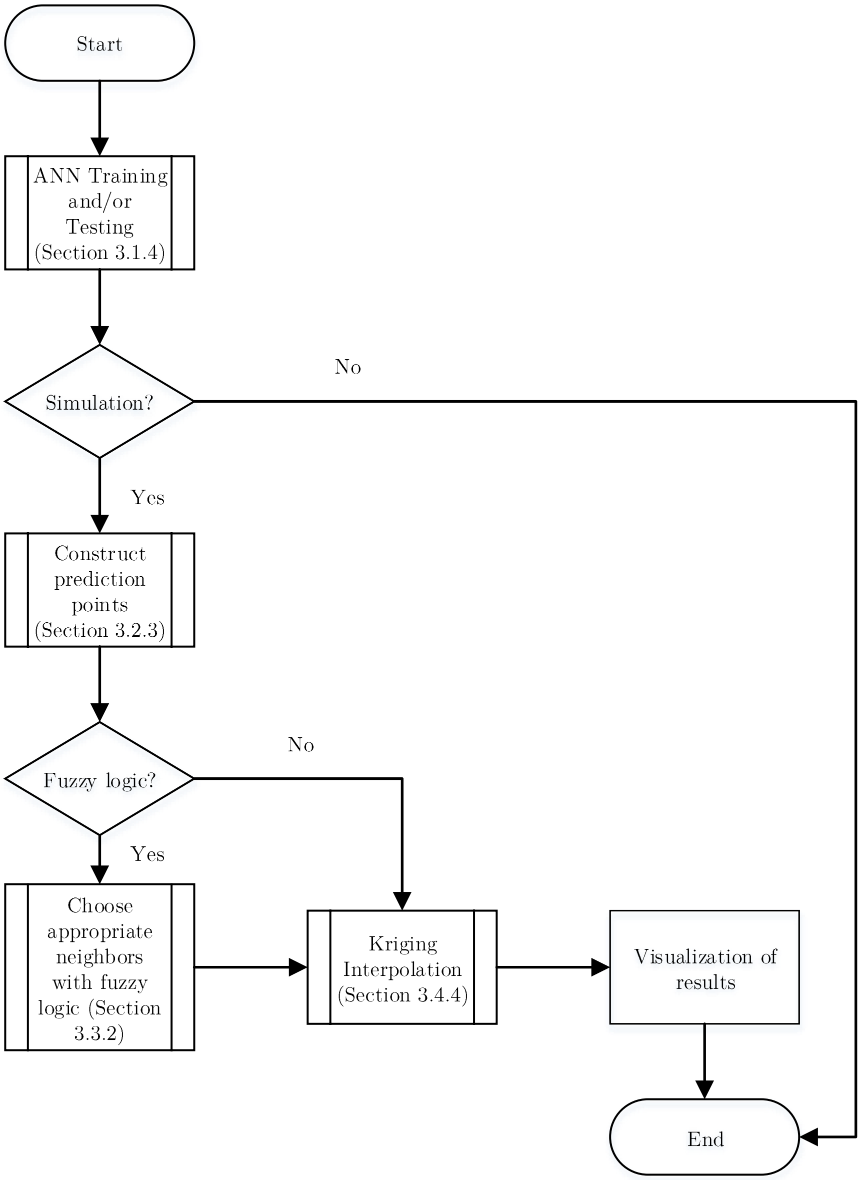

Start
ANN Training and/or Testing (Section 3.1.4)
Simulation?
No
Yes
Construct prediction points (Section 3.2.3)
Fuzzy logic?
No
Yes
Choose appropriate neighbors with fuzzy logic (Section 3.3.2)
Kriging Interpolation (Section 3.4.4)
Visualization of results
End

Figure 3-1: Overall methodology flowchart

3.1 ANN methodology

The first part of the algorithm includes the use of multiple ANNs. Since the capability of ANNs to simulate aquifer parameters temporally has already been established, only one ANN was trained and used to evaluate the output parameter at hand.

3.1.1. ANN architecture

The ANN used throughout this study is a fully connected Multi-layer Perceptron which includes an input and an output layer as well as one or two hidden layers. The use of more than two hidden layers is not advised due to the increased computational cost incurred. As mentioned in many cases in the past [*Haykin*, 1999; *Kröse and Van Der Smagt*, 1993], the optimal architecture for an ANN depends heavily on the application. There is no strict, universal rule that applies to every case regarding the optimal number of hidden layers or the number of hidden nodes within these layers. However, some rules have emerged that provide guidelines for choosing an architecture that will provide an adequately well-trainable network. *Mirchandani and Cao* [1989] proposed that the number of hidden nodes (H) in a feedforward neural network depend only on the number of input training patterns (T). There are conjectures that H is on the order of $T-1$ and $\log_2(T)$. According to *Fine* [1999], using three times as many training patterns as network parameters (weights) is adequate to achieve good generalization. The second rule was applied in this work as it resulted in an exact number of nodes for both cases of ANNs, with one or two hidden layers. It should be noted that this choice of number of nodes for an ANN does not yield the optimal results for the network, in terms of training and testing error, but results in a well-trained network.

The developed ANN was based on a previous network [*Trichakis et al.*, 2011b; *Trichakis et al.*, 2009] which was heavily modified in order to meet the requirements of the present application, as well as minimize the required simulation time. Converting the algorithm so as to be able to use multiple threads, both for the training and the evaluation process, was one major step in this direction. The input and output methods were also altered in order to accommodate the large volume of data and provide ease of use.

The available data were divided into two categories for training and testing. Eighty percent (80%) of the available data were used for the training process and for the estimation of the training error which reflects the ability of the network to simulate values similar to those it was trained with. The rest of the data (20%) were used to calculate the testing error between the simulation results obtained by the trained ANN and the observed field values. This error represents the network's ability to simulate values other than the ones used for training and provides an estimation of the network's generalization ability.

In order to minimize the possibility of entrapment in local minima, the learning rate was set equal to 0.01 and the momentum equal to 0.9 for all ANNs throughout this study. In order to avoid the overtraining of the network, an early stopping criterion with case-dependent numerical value was set. The sigmoid activation function was adopted, as it is considered best-suited in describing natural phenomena.

3.1.2. Input/output parameter definition

ANNs are black-box models, meaning that they do not take into consideration the entity represented by a model parameter. In order to be able to derive the optimal simulation output, parameters that have physical meaning should be used as inputs and outputs. Temporally varied input and output parameters can be used in an ANN. Parameters that have constant values in the aquifer, like hydraulic conductivity, will not be used. The influence of such constant parameters is incorporated to the weights of the ANN. In this way assumptions concerning, mainly, the geology of the study area, which are very common in classical numerical models, are avoided. The more input parameters and training sets are available the more accurate the simulation will be.

Water Budget

The water mass balance equation (Equation 3.1).is used as basis in order to determine the input and output parameters of the ANN architecture:

$$\mathit{\Delta S} = I - O + P - EPT \pm Q \tag{3.1}$$

where $\mathit{\Delta S}$ is the change in storage of the aquifer, I the inflow to the aquifer, O the outflow from the aquifer, P the precipitation, EPT the evapotranspiration and Q the pumping/recharge rate.

The inflow to the aquifer can be both from groundwater and surface water bodies. Parameters that can be used are the hydraulic head of the aquifer, as well as every parameter linked with the quantity and flow rate of any surface water, which are directly related to the aquifer hydraulic head. In surface water multiple parameters can be considered, from the discharge of a nearby river and the water height in a lake, to the surface runoff into the aquifer. All these parameters are directly linked to the hydraulic head of the groundwater with is the main parameter studied in this research.

The outflow from the aquifer can also be distinguished into surface water outflow and groundwater outflow. The negative sign can be omitted since it will be incorporated in the neural weights. Any parameter constant in time should be omitted since there is no pattern for the ANN to discover in such data.

Precipitation is one of the main parameters that affect an aquatic system. Any information regarding rainfall, snowfall or any other form of precipitation should be used and is vital for the design of a good simulation model. Data from multiple weather stations can be used for this purpose, as well as several time lags between the event and the reaction of the aquifer, as will be explained later in this chapter. The use of multiple weather stations is advised in the case of large study areas in particular, as weather conditions throughout the area may vary significantly. Using more than one weather stations in and around the study area ensures that the conditions at a prediction point are taken into consideration.

Evaporation and transpiration can affect the hydraulic head in an aquifer. It may include several other parameters like the air and soil humidity, the wind speed, hours of sunshine and air and soil temperature. Each parameter affects the result to a different degree. Therefore, the data available for every potential input parameter should be evaluated prior to being used in the final ANN which minimizes the error involved.

Pumping/recharge rate can be one of the main parameters in the water budget. It can affect significantly the hydraulic head in a region; in some cases, the simulation cannot be successful without the use of these data. However, these data are not always available, due to either illegal pumping or missing data. Moreover, the effect of pumping can be incorporated in the neural weights in the case where the pumping is constant. In any case, the use of these data, when available, can minimize the simulation error.

The use of the hydraulic head as input to the neural network has been extensively discussed in the past [*Trichakis et al.*, 2009]. In the present study, the hydraulic head is not considered as input parameter, because the aim is to use the algorithm to predict the hydraulic head for more than one day ahead. Both the hydraulic head and the hydraulic head change per time step are considered as potential ANN output parameters in this work.

Time lag definition

The water supply and water management systems, like regulated channels, are not the only factors affecting the spatial correlation of water level between monitoring wells. Natural factors can affect the spatial correlation of the data for the groundwater level. One such natural factor, rainfall, can be intense and extremely localised. Therefore, distant wells are less likely to show influence from the same rainfall events. Land use is another factor that could affect correlation. Rainfall-runoff in urbanised areas is different than in natural settings. In urban settings, pavement and storm sewers direct water into streams and channels; in natural settings, rainfall can readily infiltrate into the soil. Evapotranspiration in urban settings is also likely to be different to the one in a rural area. Because of the combined effect of all of these factors, correlation of the water level data in distant wells will generally be small.

Time lag is the time between an event happening (for example a rainfall occurance) and seeing its effect in the data. It is usually linked with the time that it takes for the water to penetrate the soil and reach the aquifer. The correlation coefficient between time series *A* for a parameter and time series *B* for the hydraulic head change can be used to determine the optimal lag between the parameter and the hydaulic head change (Equation 3.2):

$$Correl(A,B)=\frac{\sum(a-\bar{a})\cdot(b-\bar{b})}{\sqrt{\sum(a-\bar{a})^2}\cdot\sqrt{\sum(b-\bar{b})^2}} \tag{3.2}$$

where a denotes the value of the parameter, b denotes the value of the hydraulic head change and $\bar{a}$, $\bar{b}$ are the average values of time series *A* and *B*, respectively.
The highest correlation lagged parameters will be used as input to the ANN. Parameters considered to have time lag include precipitation (rainfall, snowfall etc.) and parameters related to surface water recharge or discharge (water level in a lake or river, flow rate etc.). More than one time lags can have approximately equal correlation coefficient between an input parameter and the hydraulic head change. Moreover, using consecutive days with high correlation coefficients can describe the full effect of the parameter to the hydraulic head.

List of possible input parameters

Based on the analysis above, input parameters of interest for the ANN simulation are:

- Pumping rates
- Recharge rates
- Surface water discharge/depth (various time lags)
- Precipitation (various time lags)
- Temperature
- Humidity
- Wind speed
- Hours of sunshine
- Hydraulic head at nearby locations or at different time steps

As mentioned before, the accuracy of the simulation improves when data are available for a great variety of parameters. In the case that sufficient data are not available for a parameter that significantly affects the water budget, the methodology must be applied with caution and the results must be cross-checked at every step. The overall error of the system provides an indication of the suitability of the methodology for a particular application.

3.1.3. ANN training and performance evaluation

After collecting the data and defining the appropriate time series to be used, the training and evaluation of the networks takes place.

An ANN was trained using the backpropagation algorithm for each of the spatial points with hydraulic head data available. The Root Mean Squared Error (RMSE) was used as an indicator of the error in the ANN training process:

$$E = \sum_{p} E^{p} = \frac{1}{2} \sum_{p} (d^{p} - y^{p})^{2} \qquad (3.3)$$

where E is the error, p the input pattern, d^p the observed hydraulic head change, and y^p the simulated hydraulic head change.

Following the training of all ANNs, the network performance is evaluated for unknown sets of input-output values. A set of spatially distributed time series for the hydraulic head change is generated by the simulation, one time series for every point location.

A maximum of 10,000 epochs was used for the training of every ANN, in order to achieve good training, and avoid overtraining. The final set of weights following training is used for the evaluation process.

3.1.4. ANN training and evaluation flowchart

The ANN training and evaluation algorithm is described by the flowchart presented in Figure 3-2.

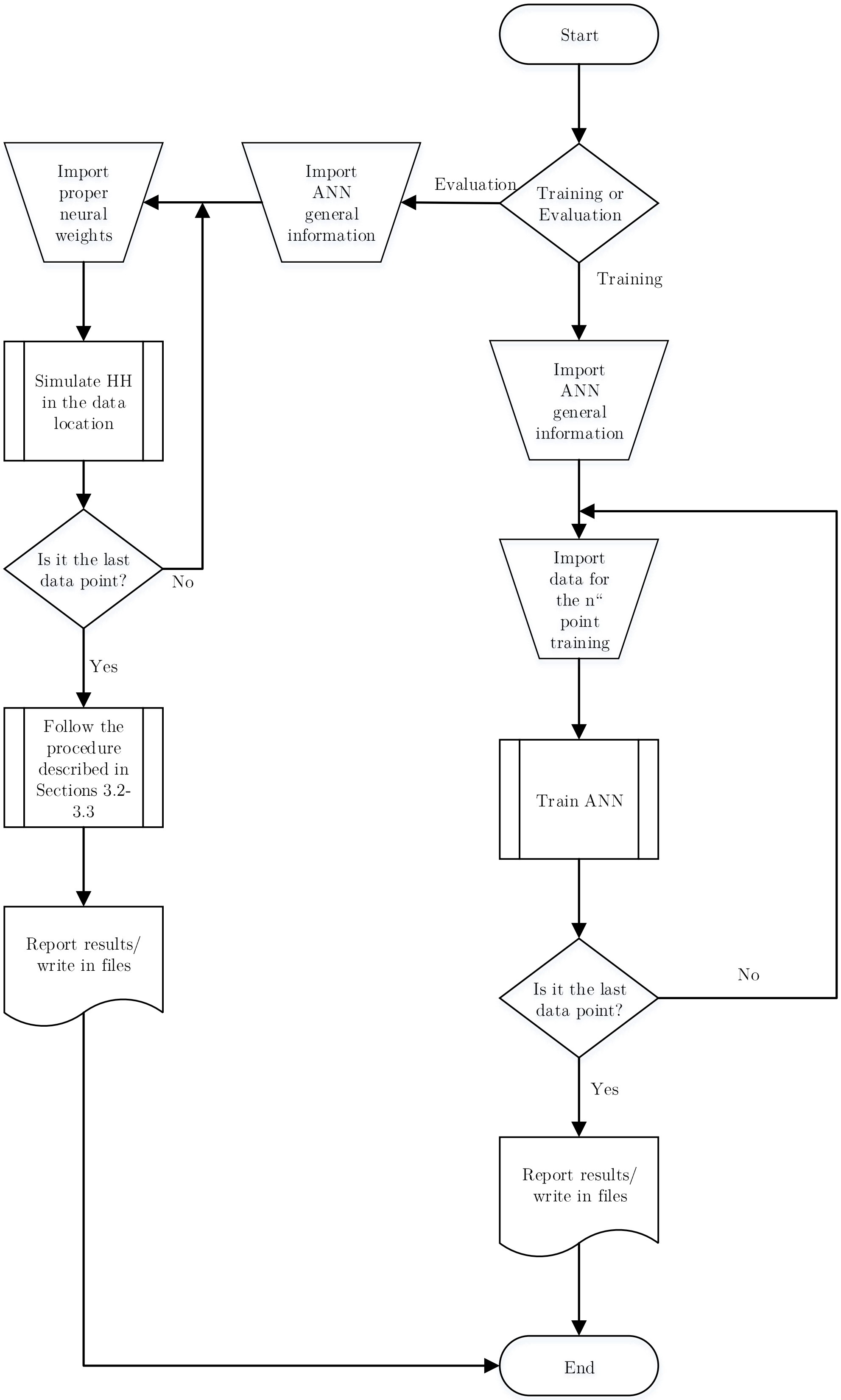

Start
Training or Evaluation
Evaluation
Training
Import ANN general information
Import proper neural weights
Simulate HH in the data location
Is it the last data point?
No
Yes
Follow the procedure described in Sections 3.2-3.3
Report results/ write in files
Import ANN general information
Import data for the n“ point training
Train ANN
Is it the last data point?
No
Yes
Report results/ write in files
End

Figure 3-2: ANN training and evaluation flowchart

The training and evaluation algorithms have been described in detail in Chapter 2. When training the ANNs, the first step is to import the general information for the ANNs construction, which include the number of wells/data points. For each one of the data points the appropriate input and output values are imported and then used for the ANN training. The final results (trained synaptic weights) and information concerning the quality of the training process (training and testing error) are reported n text files.

When the evaluation process takes place, again the first step is to import the general information concerning the construction of the ANNs. Next, the synaptic weights defined through the training process are imported and used for the hydraulic head simulation at each data location separately. The process succeeding network evaluation is described in Chapters 3.2 and 3.3. The final outputs of the ANN temporal simulation are saved in text files.

3.2 Determining prediction points

3.2.1. Delaunay triangles

A kriging algorithm is used for the spatial simulation of the hydraulic head a kriging algorithm is used. The kriging algorithm is iterated at each time step of the simulation, in order to evaluate the hydraulic head change at locations where no data are available. The specific locations at which predictions were evaluated are nodes of a grid constructed for the study area. This constructed grid is rectilinear grid, and all observation points fall exactly at grid nodes. The grid nodes at which observation data are not available are the prediction points. In order to exclude from the simulation grid nodes that fall far outside the study area and far from any observation point, Delaunay triangles were created from the known data locations. Nodes that fell outside the area covered by at least one triangle were not taken into account. The rule which is used for eliminating grid points is depicted in Figure 3-3. The red dots represent data points, form which the Delaunay triangles are drawn. Grid points that fall within these triangles (green dots) are considered prediction points, while grid points that does not belong to any triangle (blue dots) are excluded from the simulation.

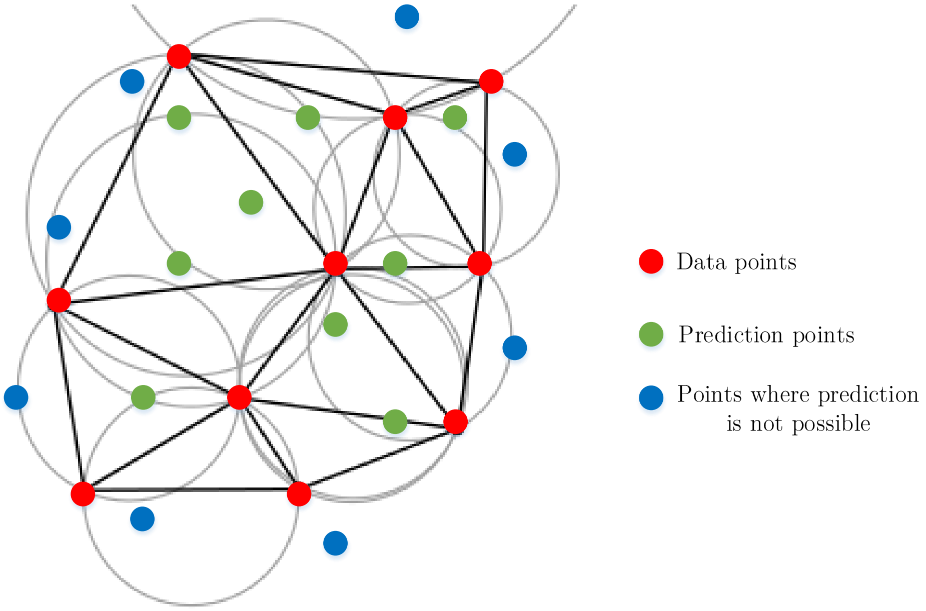


Figure 3-3: Prediction point construction

3.2.2. K-fold cross-validation

Cross-validation is a technique for assessing how a predictive model will generalize or perform to an independent data set. Cross-validation involves partitioning a dataset into complementary subsets, performing the analysis on one subset and validating on the other set. In k-fold cross-validation, the original dataset is randomly partitioned into k subsets. Of the k subsets, a single subset is used as validation data and the remaining k-1 subsets as training data. The cross-validation process is repeated k times, with a different subset used as validation data every time

In the cases under study in this dissertation, k-fold cross-validation was used to compare the observed values with the values obtained from the ANN simulation and the values obtained with kriging interpolation from neighboring data points. Specifically, for every data point, for 10% of the time steps, the observed in that point data were ignored, and the algorithm follows as if the data point does not exists. This means that ANN simulated values of the appropriate data locations are used and the predicted value is calculated following the kriging methodology. In this way three values are available for that data point and time step: the observed in the field, the ANN simulated value and the kriging interpolated. At the end of the process, the kriging interpolated values are compared to the observed and ANN simulated values, in order to derive an independent measure of performance. This means that the error values are not acquired or affected by the training process and the points where they came from are treated as nonexistent, hence providing an unbiased measurement of the error.

3.2.3. Determining prediction points flowchart

The procedure followed in order to determine the prediction points taking into account both Delaunay triangles and k-fold cross validation is described in Figure 3-4.

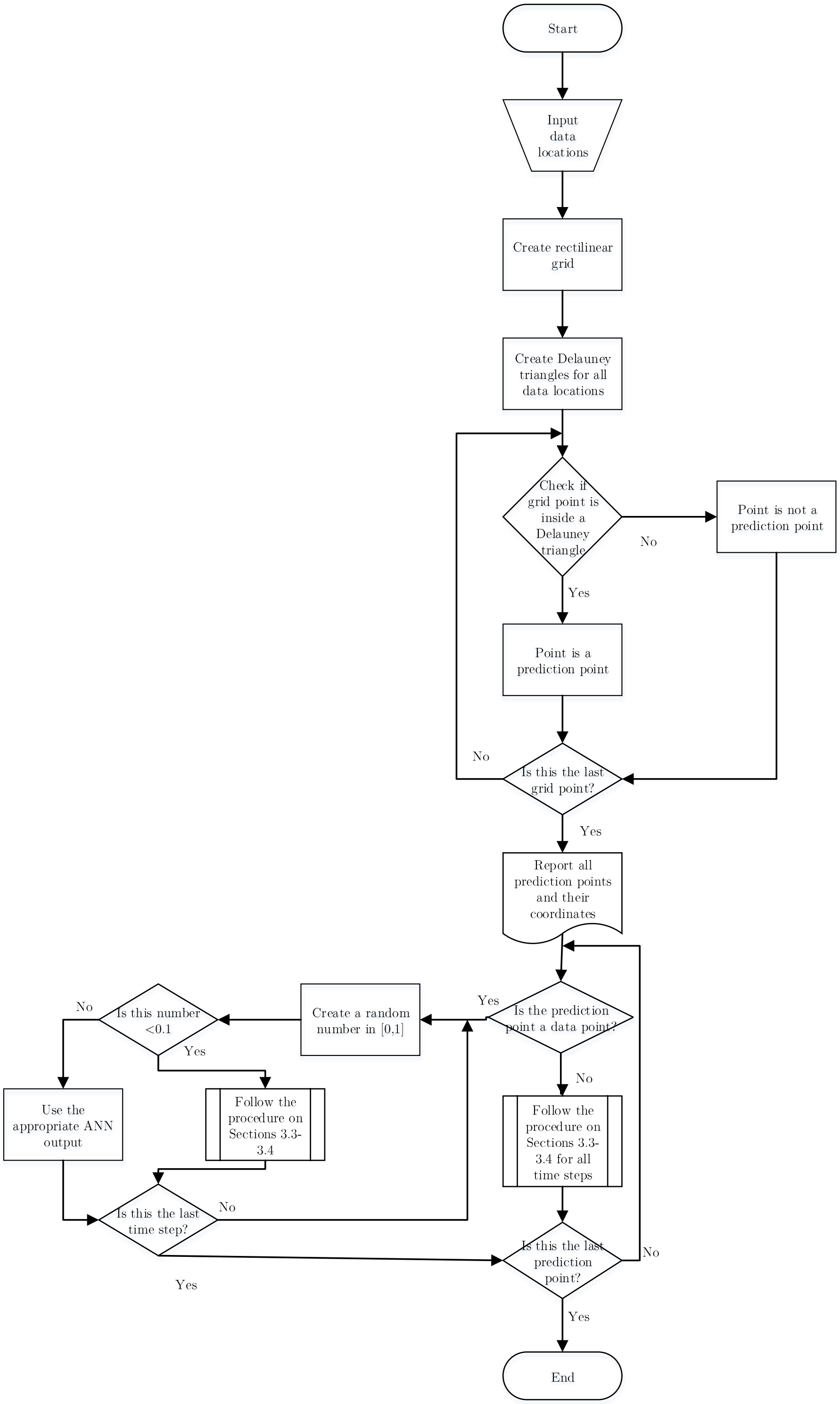

Start
Input data locations
Create rectilinear grid
Create Delauney triangles for all data locations
Check if grid point is inside a Delauney triangle
No
Point is not a prediction point
Yes
Point is a prediction point
No
Is this the last grid point?
Yes
Report all prediction points and their coordinates
Yes
Is the prediction point a data point?
Create a random number in [0,1]
No
Is this number <0.1
Yes
Use the appropriate ANN output
Follow the procedure on Sections 3.3-3.4
No
Follow the procedure on Sections 3.3-3.4 for all time steps
Is this the last time step?
No
Yes
Is this the last prediction point?
No
Yes
End

Figure 3-4: Flowchart for determining prediction points

In order to define the prediction points the following procedure is followed. First all the data points are imported to the algorithm. Using these points, a rectilinear grid is created, in which all data points falls exactly in a grid point. The next step is to create Delauney triangles correcting the data points. All grid points are then checked and only the ones that fall within a triangle are considered prediction points. At the end of this procedure all the prediction points are gathered and are saved, together with their coordinates.

In order to implement the cross-validation part of the algorithm, all prediction points are checked if they are a data point as well. If they are not then the procedure followed in Chapters 3.4 and 3.4 is followed. If a prediction point is also a data point then a random number between 0 and 1 is generated for each step. If this number is smaller than 0.1 (10% probability) then the output of the respective ANN is used, while otherwise the procedure in Chapters 3.3 and 3.4 is followed, as if the prediction point was not also a data point. The algorithm checks if it is the last time step and prediction point, and if this algorithm is terminated.

3.3 Fuzzy logic methodology

Ffuzzy logic was used determine the appropriate neighbors for each prediction point used in the kriging algorithm. To this end, the fuzzy logic system combines the ANN simulation error and the proximity of the sampling points to the prediction points.

3.3.1. Developing the fuzzy logic system

The hydraulic head change at each grid node at every time step can be calculated by the kriging algorithm. However, it is best if only wells near the prediction point and with small ANN error are used. A fuzzy logic algorithm was applied to combine the results for the distance between the prediction and observation points with the average ANN error ((training error + testing error)/2) of the specific ANN/data point. Parameter values were divided into three categories, small, average and large, and the membership functions were developed according to expert opinion for each of the two parameters.

The membership functions for distance used in the fuzzification process are shown in Figure 3-5 for the small, average and large categories, $\mu_{1s}(x)$, $\mu_{1a}(x)$ and $\mu_{1l}(x)$, respectively. The respective membership functions for the average ANN error, $\mu_{2s}(x), \mu_{2a}(x)$ and $\mu_{2l}(x)$, are also shown in Figure 3-5. Note that slightly greater emphasis is given to the distance compared to the error. This is apparent the ANN error fuzzification diagram in Figure 3 5, where (s) property value is 1 for a range of error values, while for the distance it is only 1 for a specific value.

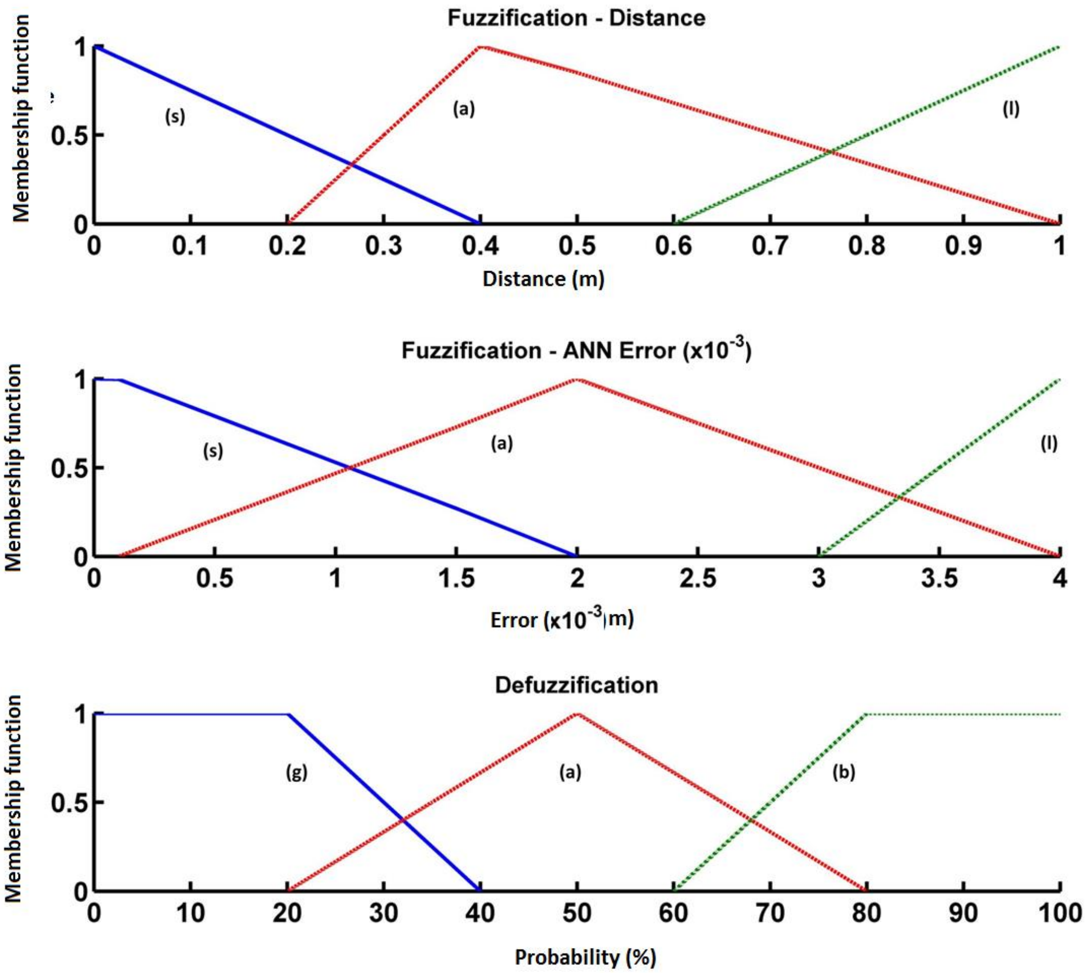


Figure 3-5: Examples of fuzzification and defuzzification diagrams

Next, a set of appropriate rules was applied in order to define the possibility of every set of parameters being good, adequate or bad. The set of three rules applied in this study is as follows:

- Rule 1: If the distance is small **and** the error is small, then the well is considered as good for the specific prediction point ($p_g(x) = \mu_{1s\cap 2s}(x) = minimum\{\mu_{1s}(x), \mu_{2s}(x)\}$)
- Rule 2: If the distance is average **or** the error is average, then the well is considered as adequate for the specific prediction point ($p_a(x) = \mu_{1a\cup 2a}(x) = maximum\ \{\mu_{1a}(x), \mu_{2a}(x)\}$)

- Rule 3: If the distance is large **or** the error is large, then the well is considered as bad for the specific prediction point ($p_b(x) = \mu_{1l \cup 2l}(x) = maximum\{\mu_{1l}(x), \mu_{2l}(x)\}$)

where $p_g(x)$, $p_a(x)$ and $p_b(x)$ represent the extent to which well x is considered good, adequate and bad, respectively. These rules are summarized in Table 3-1.

Table 3-1: Set of rules used in fuzzy logic

		Distance		
		Small	Average	Large
Error	Small	Good	Adequate	Bad
	Average	Adequate	Adequate	Bad
	Large	Bad	Bad	Bad

It should be pointed out that these rules are treating each one of good/adequate/bad categories differently. This means that for a well to be considered as good, both the error and the distance should be small. On the other hand, if the distance or the error is large, then it is sufficient to practically minimize the probability of this well being used by the kriging algorithm.

The next step was the aggregation of the rule outputs, that is, the unification of the outputs of all rules into a single fuzzy set. The input of the aggregation process is the list of the clipped consequent membership functions and the output is one fuzzy set for each output variable. In this case the probability of well x being an appropriate neighbor.

Fuzzification helps evaluate the rules, the final output of the system should be a crisp value. This is achieved through the defuzzification process (third graph in Figure 3-5). The centroid technique appears to provide consistent defuzzification results. It is a well-balanced method, which is sensitive to the height and width of the total fuzzy region as well as to the sparse singletons [*Negnevitsky*, 2005].This way, only the best well locations, the wells with the highest x values, which can also be expressed as an appropriateness percentage, are used by the kriging algorithm.

Tuning the fuzzy system to derive the appropriate results may involve executing a number of actions [*Negnevitsky*, 2005]:

- Review model input and output variables and if required redefine their range.

- Review the fuzzy sets and if required define additional sets on the universe of discourse. The use of wide fuzzy sets can cause the fuzzy system to perform roughly.
- Provide sufficient overlap between neighboring sets. Although there is no method to determine the optimum amount of overlap, it is suggested that triangle-to-triangle and trapezoid-to-triangle fuzzy sets should overlap between 25 and 50% of their bases [*Cox*, 1999].
- Review the existing rules and if required add new rules to the rule base
- Examine the rule base for opportunities to write hedge rules to capture the pathological behavior of the system
- Adjust the rule execution weights
- Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation and thus a system can still behave well even when the shapes of the fuzzy set are not precisely defined.

3.3.2. Fuzzy logic flowchart

After the determination of the prediction points, the procedure described by the flowchart in Figure 3-6 takes place. When the fuzzy logic system is not used the procedure is bypassed. Otherwise, when the fuzzy logic system is used, the fuzzification, rule propagation and defuzzification processes take place.

First all distances between a prediction point and a data point is calculated. For each data point the fuzzification of its distance from the prediction point and its ANN training and testing error takes place, using the corresponding graphs. The application of the fuzzy rules in these values is the next step, followed by the defuzzification process. All data points' probability values are sorted and the best data points are reported back to the main part of the algorithm. This procedure is repeated for al prediction. When the last prediction point neighbors are defined, the process is terminated.

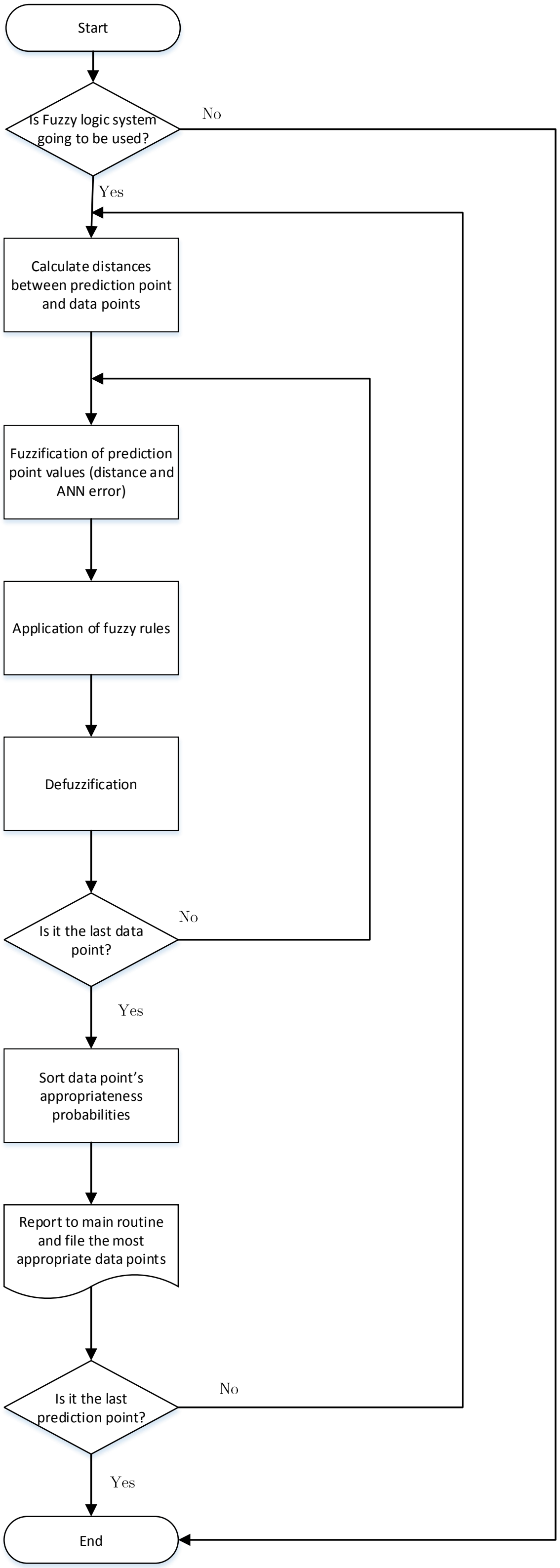

Start
Is Fuzzy logic system going to be used?
No
Yes
Calculate distances between prediction point and data points
Fuzzification of prediction point values (distance and ANN error)
Application of fuzzy rules
Defuzzification
Is it the last data point?
No
Yes
Sort data point's appropriateness probabilities
Report to main routine and file the most appropriate data points
Is it the last prediction point?
No
Yes
End

Figure 3-6: Fuzzy logic flow chart

3.4 Kriging methodology

3.4.1. Variogram determination

The first step in the applied kriging algorithm is the determination of the variogram. Since a large number of variograms are constructed, one for each prediction point and time step, it is important to automate the procedure. The following procedure was iterated at each algorithm step for the variogram determination.

First, the distances between all data points are calculated. The maximum and minimum distances are identified and ten classes are defined between these values. Then, taking all possible pairs of points i and j, the following are calculated for each class of distance:

1. the number of pairs present in the class,
2. the average distance in the class, and,
3. the average square increment $\frac{1}{2}\left(z_i - z_j\right)^2$.

The graph of the semivariance versus distance comprises the experimental variogram. Special attention should be given to the number of pairs within each class. If the number of pairs in any class is less than thirty, the classes should be rearranged so as to accommodate this rule. In our case we choose the neighborhood size accordingly so as to fulfill this rule, as explained in Section 3.4.2.

The next step is to fit the experimental variogram to an authorized variogram model. The variogram models that are going to be tested for performance are the ones listed in Table 2.1. The selected variograms are expected to provide good results in hydrological data [*Kitanidis*, 1997]. Non-linear Least Square Fitting was performed, using the AlgLib library for Visual Basic .NET [*Bochkanov and Bystritsky*, 2011]. Non-linear methods generally have more tunable parameters than linear ones and their convergence speed is problem-dependent, since they are iterative processes. With AlgLib it is possible to set various stopping criteria for the fitting process such as sufficiently small step between iterations, small function change between iterations and specified number of iterations. Moreover, it is possible to set secondary constraints like intervals of possible parameter values. In this way the final variogram parameters set the theoretical variogram model as close as possible to the experimental variogram. It is common practice not to use all classes created in the previous step in order to increase the accuracy of the fitted variogram. For this reason, only the first nine classes were used.

3.4.2. Neighborhood selection

The determination of the appropriate neighborhood depends on whether the fuzzy logic system is used or not. In the case where the fuzzy logic system is used, the procedure described in Section 3.3 applies and only a specific number of wells is used. This number should be chosen carefully so as to allow for the 30 pairs per class in the variogram determination process.

In the case where the fuzzy logic system is not used, then the selection of the appropriate neighbors depends on the distance between the prediction point and the data points. Only data points that fall within the range of the variogram are used.

3.4.3. Parameter value evaluation

After choosing the most appropriate neighbors, using either of the methods described above, the evaluation of the parameter value at the prediction point takes place. The first step in this process is to use the derived variogram model to compute the vector of distances between the prediction point and the data points within the neighborhood. This vector is often symbolized with D and has the form shown in Equation 3.4:

$$D = \begin{bmatrix} \gamma(h_{p1}) \\ \gamma(h_{p2}) \\ \vdots \\ \gamma(h_{pn}) \\ 1 \end{bmatrix} \tag{3.4}$$

where h_{pi} the distance between the prediction point and the i^{th} data point, n is the total number of data points in the neighborhood.

The covariance matrix is also constructed by a applying the variogram to the distances between the data points in the neighborhood. The matrix is often symbolized with C and the form described by Equation 3.5

$$C = \begin{bmatrix} 0 & \gamma(h_{12}) & & \gamma(h_{1n}) & 1 \\ \gamma(h_{21}) & 0 & \cdots & \gamma(h_{2n}) & 1 \\ \gamma(h_{31}) & \gamma(h_{32}) & & \gamma(h_{3n}) & 1 \\ \vdots & & \ddots & \vdots & \\ \gamma(h_{n1}) & \gamma(h_{n2}) & \cdots & 0 & 1 \\ 1 & 1 & & 1 & 0 \end{bmatrix} \tag{3.5}$$

The next step is to multiply the inverted matrix C with vector D, in order to obtain the vector λ of weight factors w, which is applicable to the current output only. The last row of the w vector it the Lagrange multiplier. Finally, the predicted value ($\hat{z}$) at a specific prediction point for a given time step is acquired by summing the products of the weighting factors and the parameter values at chosen data points (Equation 3.6).

$$\hat{z} = \sum_{i=1}^{n} (w_i \cdot z_i) \tag{3.6}$$

This process is repeated for every prediction point and every time step.

3.4.4. Kriging flowchart

In order to evaluate the hydraulic head in the prediction points using the kriging interpolation, the procedure described in Figure 3-7 was followed. After the prediction point construction and the definition of the appropriate neighbors for every point separately, either using the fuzzy logic system or not, for every time step and for every prediction point, the next procedure is followed. First the experimental variogram is constructed, followed by its fitting to a variogram model. Using equations 3.4 and 3.5 the covariance matrix C and vector D are constructed and the finally the parameter at hand is simulated. After simulating the hydraulic head in all the time steps and for all prediction points, the results are passed on to MATLAB for the final visualization.

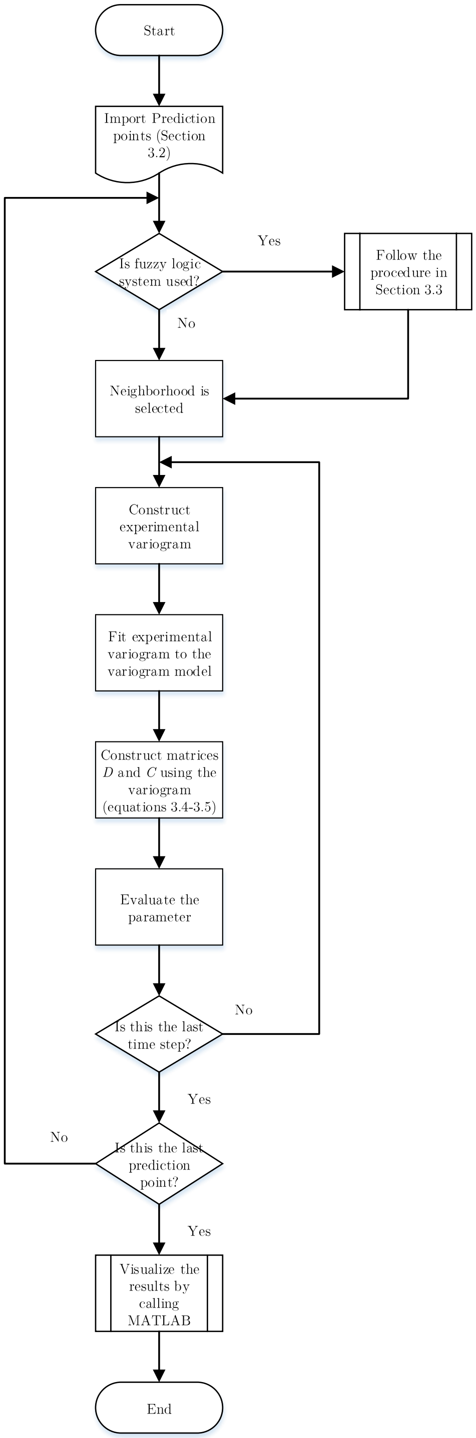

Start
Import Prediction points (Section 3.2)
Is fuzzy logic system used?
Yes
Follow the procedure in Section 3.3
No
Neighborhood is selected
Construct experimental variogram
Fit experimental variogram to the variogram model
Construct matrices D and C using the variogram (equations 3.4-3.5)
Evaluate the parameter
Is this the last time step?
No
Yes
No
Is this the last prediction point?
Yes
Visualize the results by calling MATLAB
End

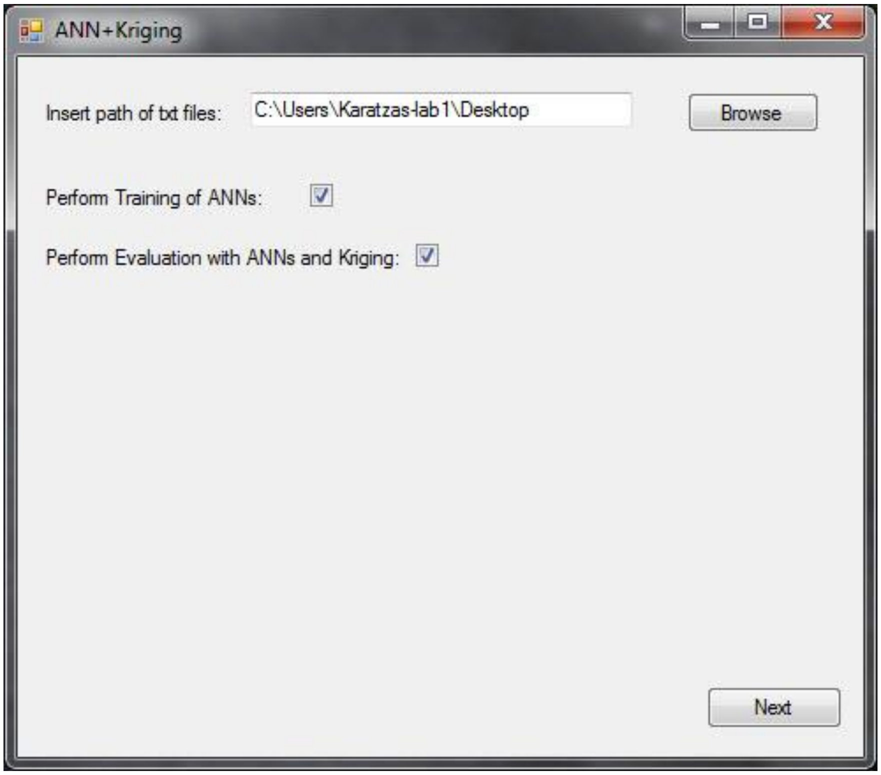
Figure 3-7: Kriging interpolation flowchart

3.5 Graphical User Interface

A Graphical User Interface (GUI) was designed and developed in order to facilitate the use of the algorithm. The use of the interface and the procedure followed to execute the algorithm are described and explained in the following sections.

3.5.1. Input files definition

The import of the files with the appropriate input data, depending on the problem is the first step in the execution of the algorithm. The path to the folder where all the input files are located is identified in the GUI window that appears when the execution of the program begins (Figure 3-8). The generated output files are also stored in the same folder. Next, the desired operation is selected from two options: ANN training and evaluation. In the case where the ANNs have already been trained, one can select to perform evaluation only, otherwise ANN training should be performed first.



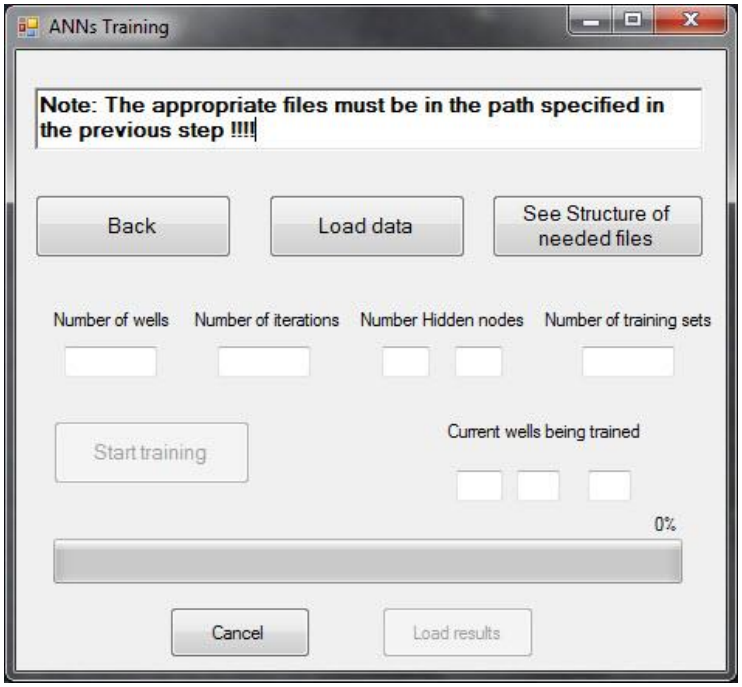
Insert path of txt files: C:\Users\Karatzas-lab1\Desktop

Perform Training of ANNs: ☑

Perform Evaluation with ANNs and Kriging: ☑

Figure 3-8: Introductory GUI window

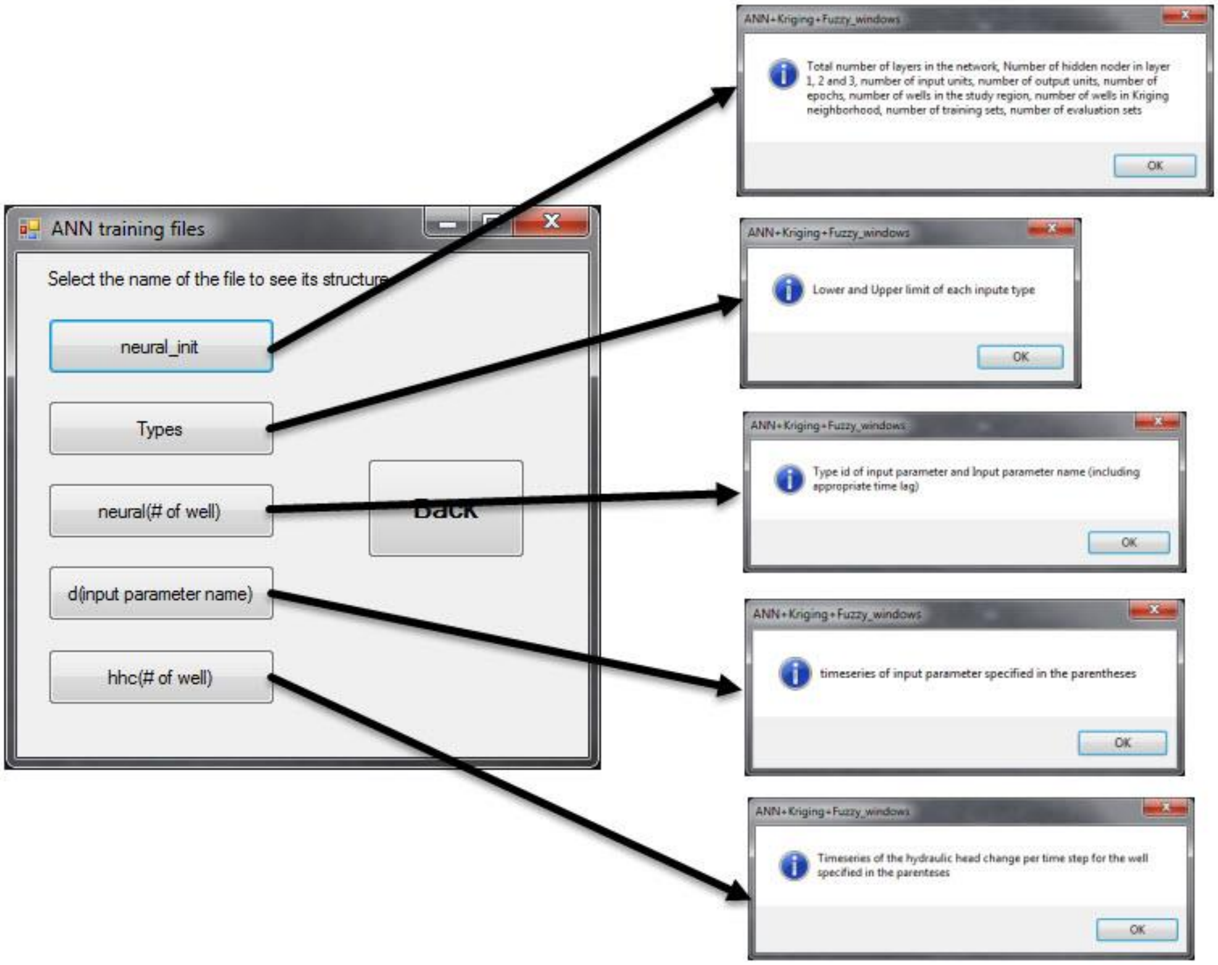
The next window in the program execution (Figure 3-9) allows for the loading the files necessary for ANN training.



Note: The appropriate files must be in the path specified in the previous step !!!!

Figure 3-9: ANN Training window

The ‘see structure of needed files’ button in the ANN training window provides assistance regarding the proper structure of the files. The window shown in Figure 3-10 appears for assistance with the structure of the different import files.



Total number of layers in the network, Number of hidden noder in layer 1, 2 and 3, number of input units, number of output units, number of epochs, number of wells in the study region, number of wells in Kriging neighborhood, number of training sets, number of evaluation sets

Lower and Upper limit of each inpute type

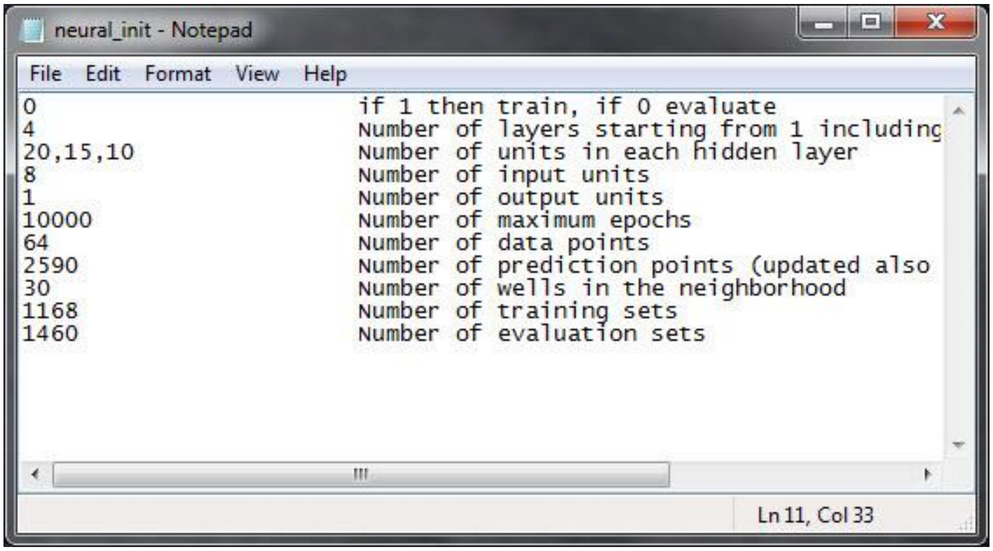
Type id of input parameter and Input parameter name (including appropriate time lag)

timeseries of input parameter specified in the parentheses

Timeseries of the hydraulic head change per time step for the well specified in the parenteses

Figure 3-10: Help in identifying the appropriate files

The first file that must be imported prior to ANN training is file ‘neural_init.txt’, which contains all the information about the structure of the ANNs and other general information.



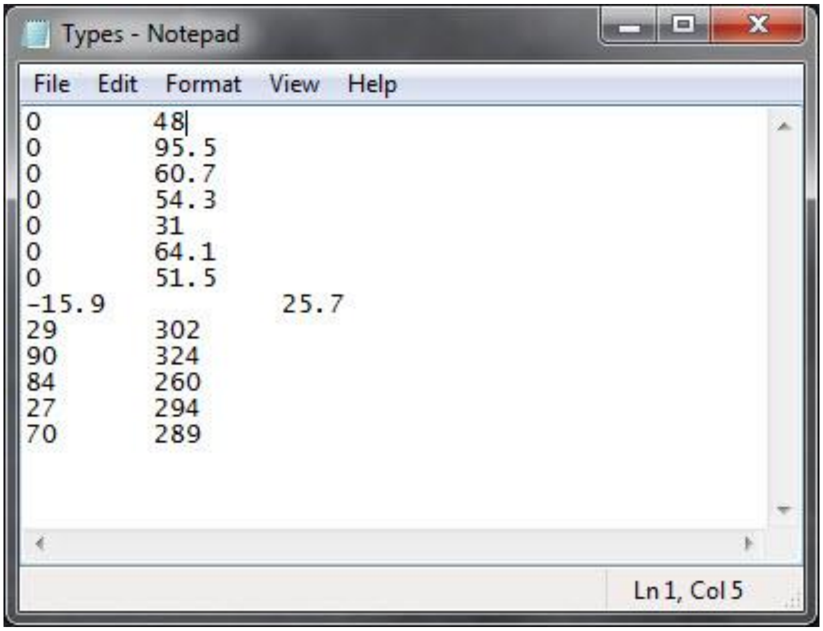
```
0                            if 1 then train, if 0 evaluate
4                            Number of layers starting from 1 includinc
20,15,10                     Number of units in each hidden layer
8                            Number of input units
1                            Number of output units
10000                        Number of maximum epochs
64                           Number of data points
2590                         Number of prediction points (updated also
30                           Number of wells in the neighborhood
1168                         Number of training sets
1460                         Number of evaluation sets
```

Figure 3-11: Neural_init.txt file structure

File Neural_init.txt has the structure shown in Figure 3-11. A value of 0 in the first line indicates that the ANNs are used for evaluation while a value 1 indicates that ANN training is performed first, followed by evaluation. This value is updated according to the selection made in the introductory window. The number of layers, including input, output and hidden layers, is set in the second file line, while the number of hidden nodes (units) per hidden layer is set in the third line. If data for more layers are written in the third line, they are ignored by the algorithm and only the data concerning the number of layers declared in the second line are used. The number of input and output units is declared in lines 4 and 5, respectively. The number of epochs, of data points and of prediction points is declared in lines 6-8. The number of prediction points is also updated after the first step of the evaluation process, and is not taken into account during the training process.

3.5.2. Training

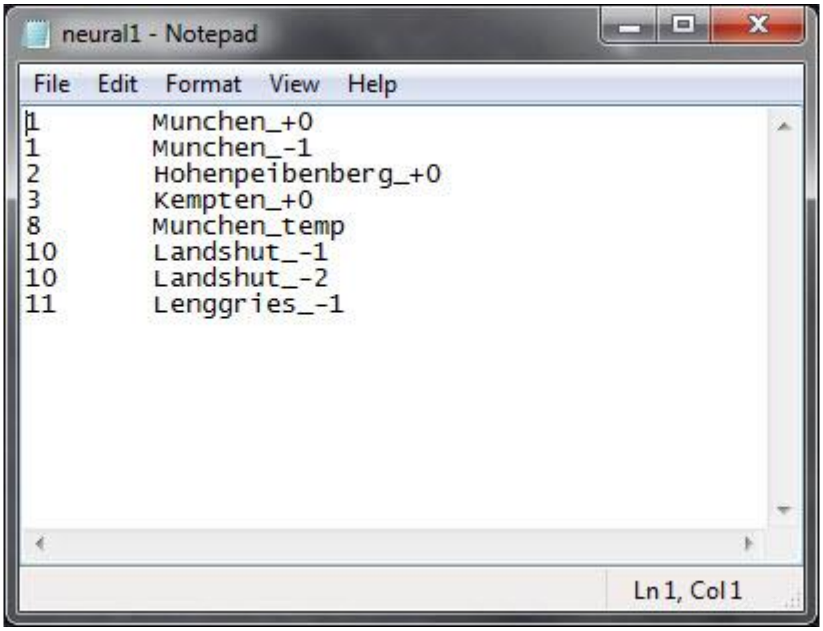
Multiple files are imported after the ANN training process begins. The first of these files is types.txt, which contains information regarding the range (maximum value - minimum value) of all possible input parameters (eg. rainfall from different meteorological stations, temperature etc.). A typical example of this file is presented in Figure 3-12.



```
0       48
0       95.5
0       60.7
0       54.3
0       31
0       64.1
0       51.5
-15.9           25.7
29      302
90      324
84      260
27      294
70      289
```

Figure 3-12: Types.txt input file

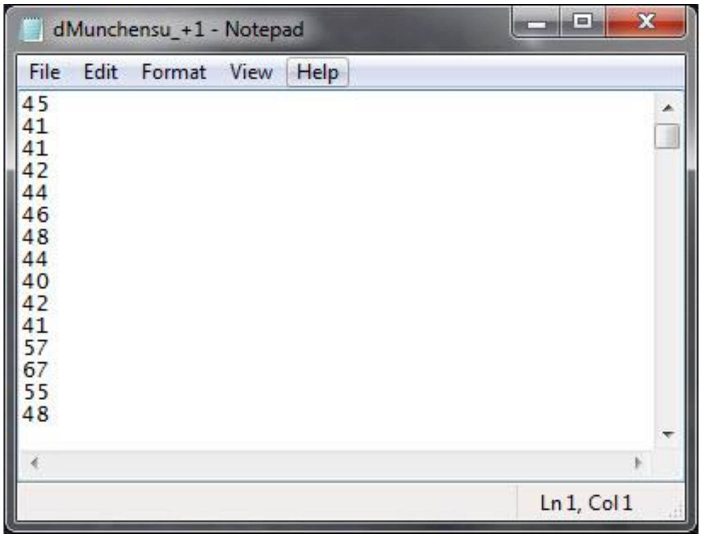
The next set of input files refers to the input parameters of each ANN separately. Each file is named neural followed by the ID number of the well corresponding to the ANN, starting from well No. 1. For example, the information in file neural1.txt shown in Figure 3-13 concerns the input parameters for the corresponding ANN. More specifically, the first column of the file records the ID number of the type of input and the second the name of the parameter along with the appropriate time lag.



```
1       Munchen_+0
1       Munchen_-1
2       Hohenpeibenberg_+0
3       Kempten_+0
8       Munchen_temp
10      Landshut_-1
10      Landshut_-2
11      Lenggries_-1
```

Figure 3-13: neural1.txt example file

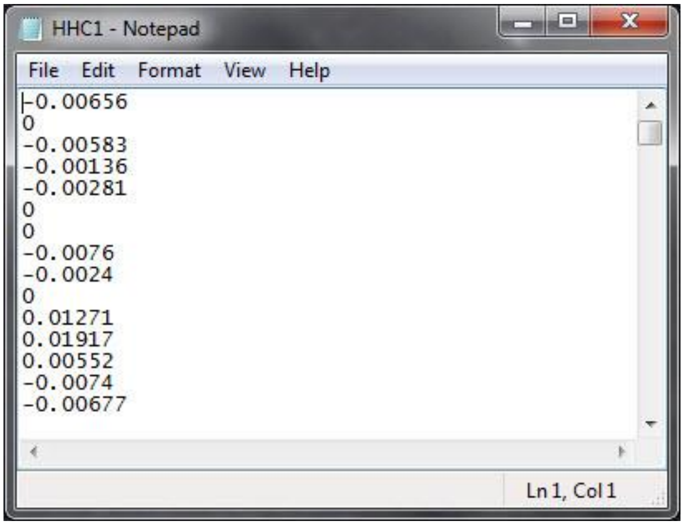
The next set of files essential for the execution of the algorithm is the set of files containing the input parameter time series. The name of these files starts with d followed by the name of the input parameter. For example, Figure 3-14 shows the structure of file dMunchensu_+1.txt containing the surface water level time series for the Munich station and for time lag +1 day.



```
45
41
41
42
44
46
48
44
40
42
41
57
67
55
48
```

Figure 3-14: Sample file containing the time series for the surface water level parameter (dMunchensu_+1.txt)

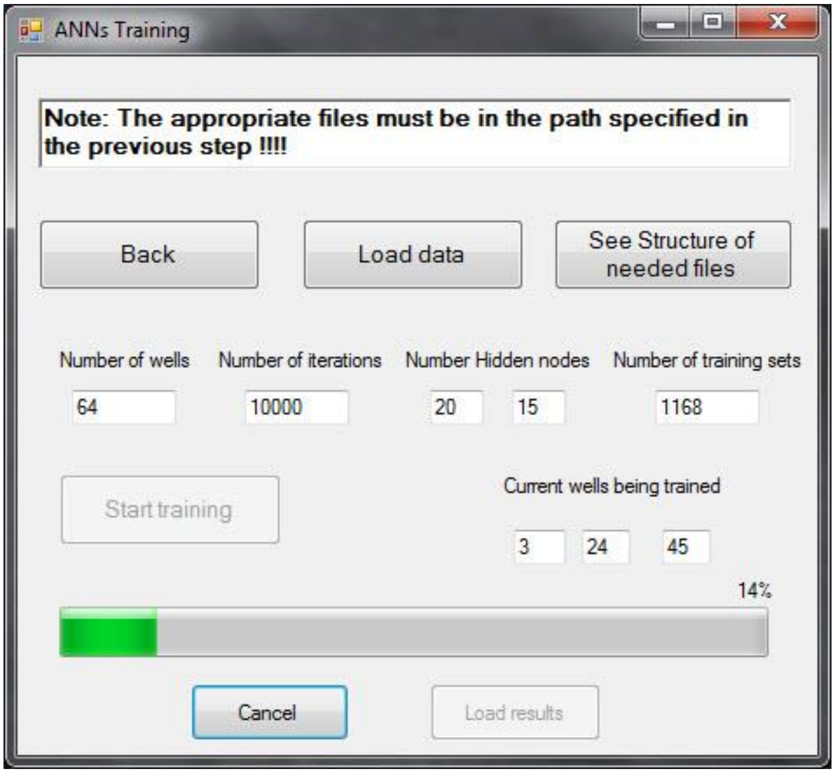
The last set of files necessary for the training process is the set of files containing the output parameter values, in this case hydraulic hear or hydraulic head change observed in the field. Note that hydraulic head change is the desired output for each ANN. Each file is named HHC followed by the ID number of the data point, as shown for the sample file for data point No.1 in Figure 3-15.



```
-0.00656
0
-0.00583
-0.00136
-0.00281
0
0
-0.0076
-0.0024
0
0.01271
0.01917
0.00552
-0.0074
-0.00677
```

Figure 3-15: HHC1.txt representing the desired output for well No. 1

After loading all the necessary files the main part of the training process can take place. The window displayed during the training process is shown in Figure 3-16. The code is multithreaded with three ANNs being trained simultaneously. The ID number of the wells being currently trained, the percentage of completed wells, as well as the values of some basic parameters set by the user are displayed in the window.



Note: The appropriate files must be in the path specified in the previous step !!!!

Number of wells: 64

Number of iterations: 10000

Number Hidden nodes: 20 15

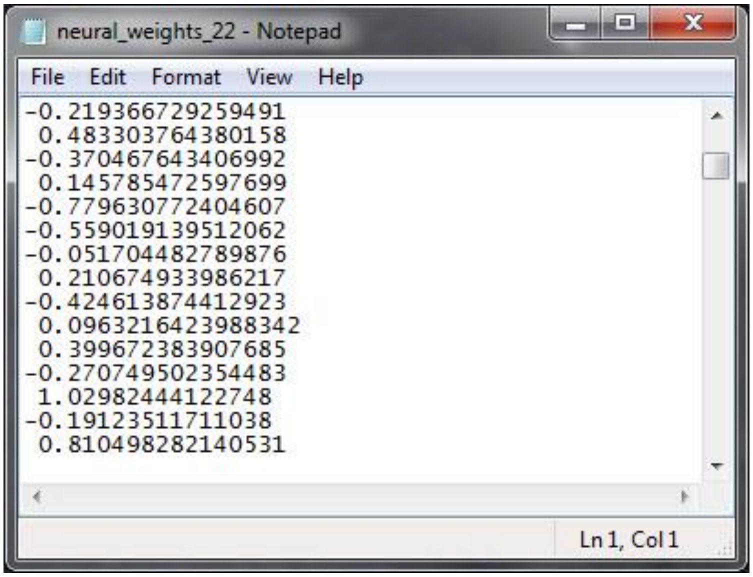
Number of training sets: 1168

Current wells being trained: 3 24 45

14%

Figure 3-16: Training process window

The files containing the synaptic weights of each ANN are generated automatically upon termination of the training process are named the neural_weights followed by the ID number of the data point. A sample file for data point No. 22 is shown in Figure 3-17.

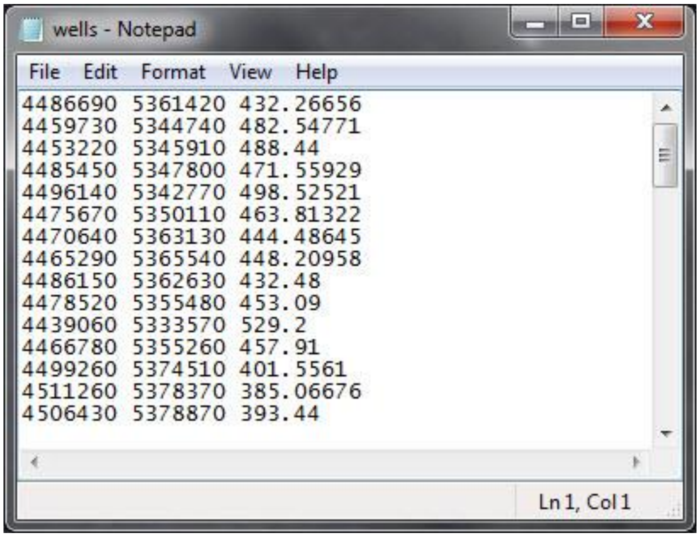


```
-0.219366729259491
 0.483303764380158
-0.370467643406992
 0.145785472597699
-0.779630772404607
-0.559019139512062
-0.051704482789876
 0.210674933986217
-0.424613874412923
 0.0963216423988342
 0.399672383907685
-0.270749502354483
 1.02982444122748
-0.19123511711038
 0.810498282140531
```

Figure 3-17: Typical form of file neural_weights_22.txt

3.5.3. Evaluation

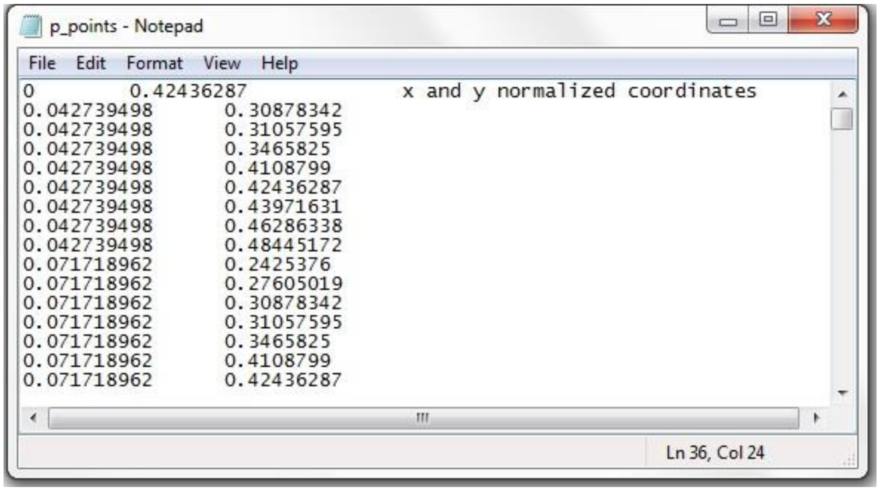
The files generated by the training process are necessary for the evaluation process One additional file, wells.txt, with records for the coordinates and the initial hydraulic head for each data point (well) must be created by the user (Figure 3-18).



```
4486690 5361420 432.26656
4459730 5344740 482.54771
4453220 5345910 488.44
4485450 5347800 471.55929
4496140 5342770 498.52521
4475670 5350110 463.81322
4470640 5363130 444.48645
4465290 5365540 448.20958
4486150 5362630 432.48
4478520 5355480 453.09
4439060 5333570 529.2
4466780 5355260 457.91
4499260 5374510 401.5561
4511260 5378370 385.06676
4506430 5378870 393.44
```

Figure 3-18: Typical structure of wells.txt

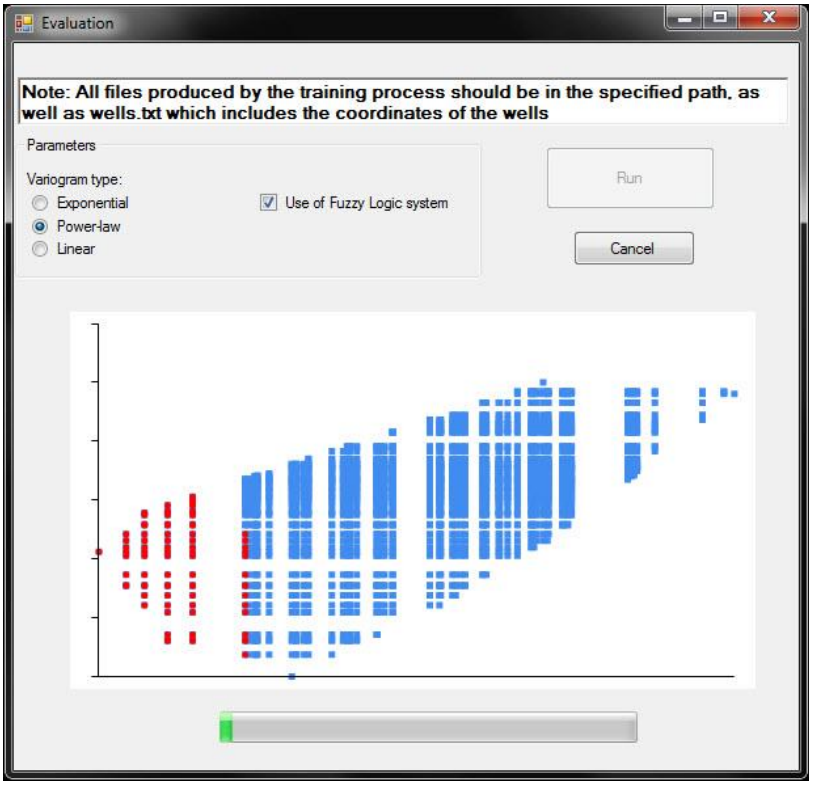
The next step in the process is to determine the prediction points. The x and y normalized coordinates for the prediction points are recorded in file p_points.txt. The structure of a typical such file is presented in Figure 3-19.



```
0            0.42436287           x and y normalized coordinates
0.042739498      0.30878342
0.042739498      0.31057595
0.042739498      0.3465825
0.042739498      0.4108799
0.042739498      0.42436287
0.042739498      0.43971631
0.042739498      0.46286338
0.042739498      0.48445172
0.071718962      0.2425376
0.071718962      0.27605019
0.071718962      0.30878342
0.071718962      0.31057595
0.071718962      0.3465825
0.071718962      0.4108799
0.071718962      0.42436287
```

Figure 3-19: Typical structure of p_points.txt

The evaluation process window is shown in Figure 3-20. Prior to the execution of the evaluation process, the user must select the type of variogram model to be used, as well as whether the fuzzy logic system will be used or not. The red dots on the window graph represent prediction points for which the hydraulic head has already been evaluated, while the blue dots represent prediction points for which the hydraulic head has yet to be evaluated.



Note: All files produced by the training process should be in the specified path, as well as wells.txt which includes the coordinates of the wells

Parameters

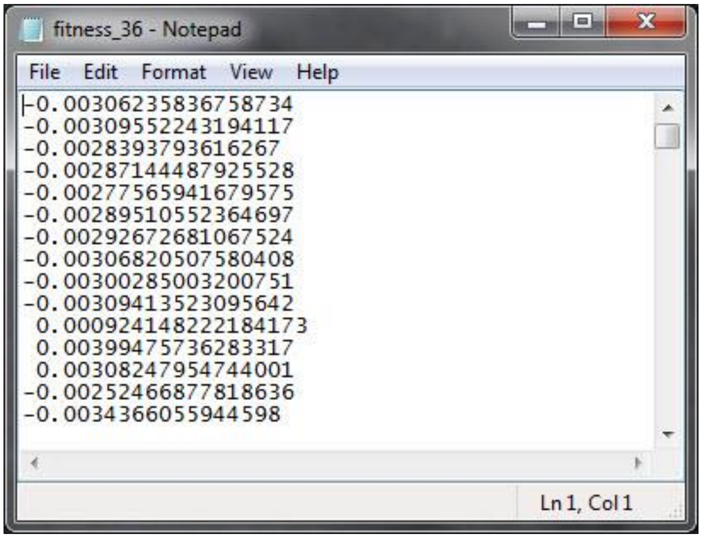
Variogram type:

○ Exponential
● Power-law
○ Linear

☑ Use of Fuzzy Logic system

Figure 3-20: Evaluation process window

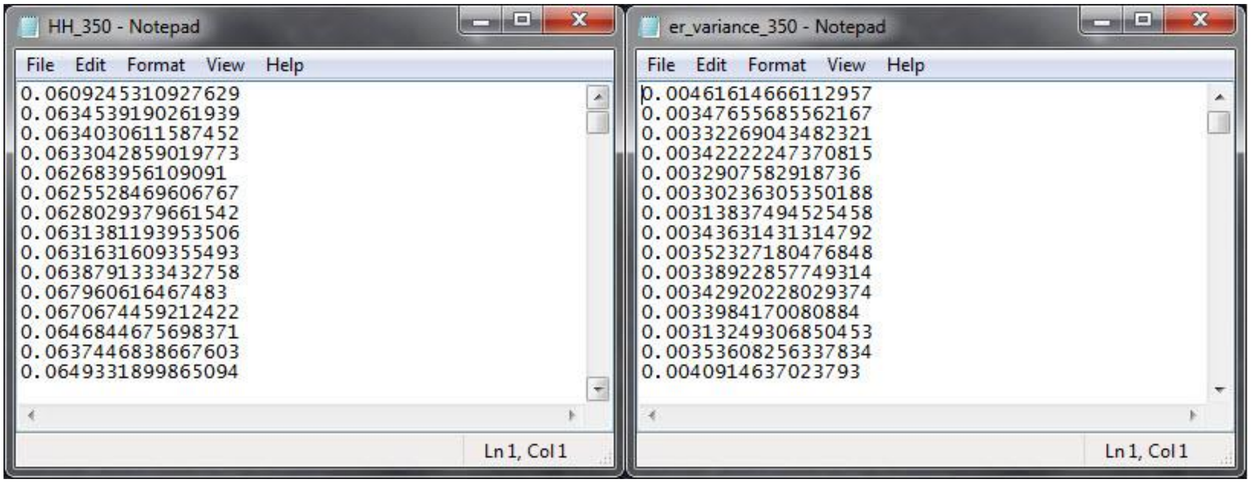
ANN evaluation at data points is performed first; the results in terms of hydraulic head or hydraulic head change are recorded in files named fitness_ followed by the data point ID number (Figure 3-21).



```
-0.00306235836758734
-0.00309552243194117
-0.0028393793616267
-0.00287144487925528
-0.00277565941679575
-0.00289510552364697
-0.00292672681067524
-0.00306820507580408
-0.00300285003200751
-0.00309413523095642
 0.000924148222184173
 0.00399475736283317
 0.00308247954744001
-0.00252466877818636
-0.0034366055944598
```

Figure 3-21: Fitness_36.txt file structure

Following the completion of the evaluation process, two types of files are generated for every prediction point: the file with the hydraulic head or hydraulic head change (named HH_ followed by the prediction point ID number) and the file with the corresponding error variance (named er_variance_ followed by the prediction point ID number) (Figure 3-22).

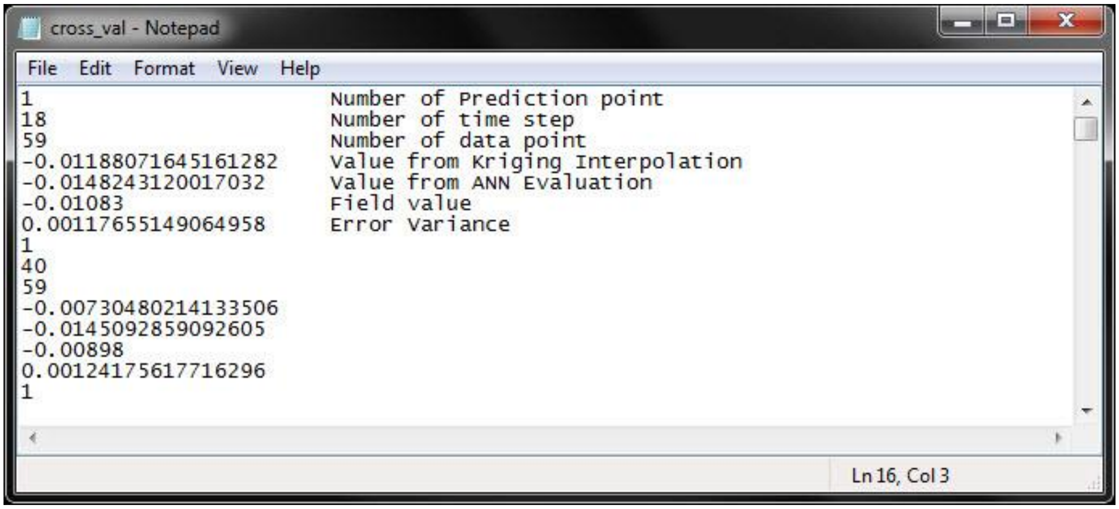

HH_350 - Notepad
File Edit Format View Help
Ln 1, Col 1
er_variance_350 - Notepad
File Edit Format View Help
Ln 1, Col 1

```
0.0609245310927629
0.0634539190261939
0.0634030611587452
0.0633042859019773
0.062683956109091
0.0625528469606767
0.0628029379661542
0.0631381193953506
0.0631631609355493
0.0638791333432758
0.067960616467483
0.0670674459212422
0.0646844675698371
0.0637446838667603
0.0649331899865094
```

```
0.00461614666112957
0.00347655685562167
0.00332269043482321
0.00342222247370815
0.0032907582918736
0.00330236305350188
0.00313837494525458
0.00343631431314792
0.00352327180476848
0.00338922857749314
0.00342920228029374
0.0033984170080884
0.00313249306850453
0.00353608256337834
0.0040914637023793
```

Figure 3-22: Hydraulic head change and error variance output files

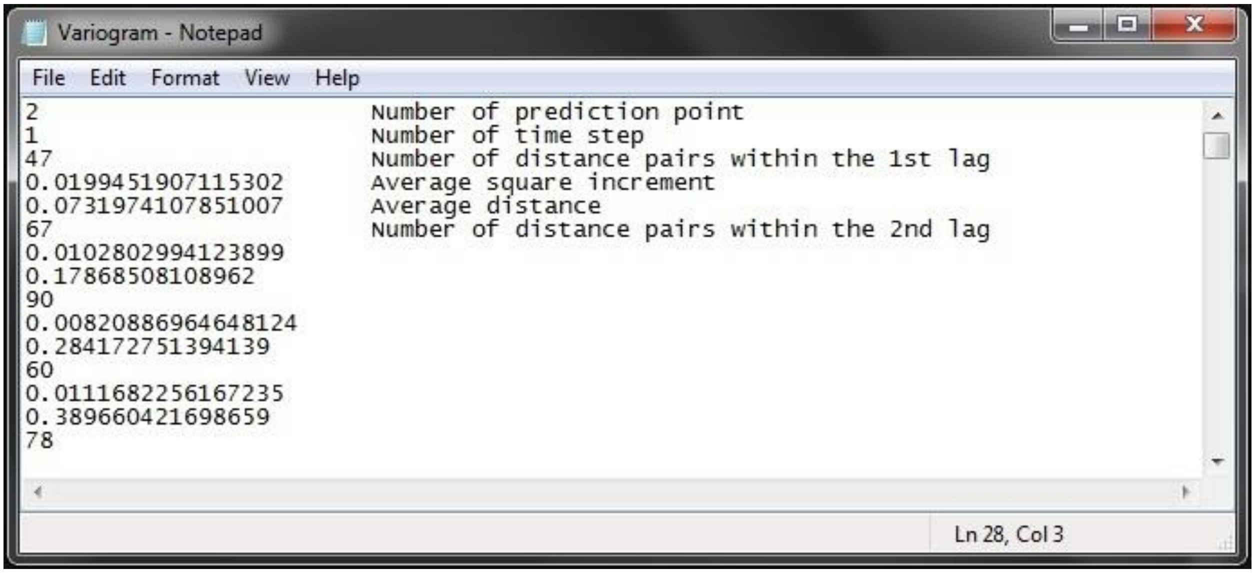
Finally, the cross-validation results are recorded in file cross_val.txt, which includes the prediction point ID number, the data point ID number, the time step at which cross-validation was performed, the simulation value obtained by the ANN, the simulation value obtained by kriging interpolation, the field measurement value and the error variance (Figure 3-23).



```
1                       Number of Prediction point
18                      Number of time step
59                      Number of data point
-0.0118807164516l282    Value from Kriging Interpolation
-0.014824312001 7032    Value from ANN Evaluation
-0.01083                Field value
0.00117655149064958     Error Variance
1
40
59
-0.00730480214133506
-0.0145092859092605
-0.00898
0.00124175617716296
1
```

Figure 3-23: cross_val.txt file structure

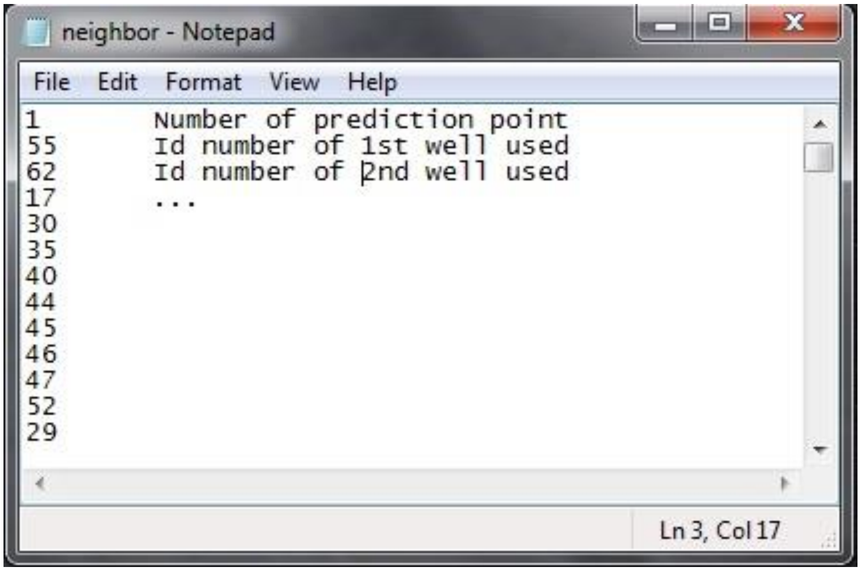
Information for the variograms constructed at time steps 1, 500 and 1000 is stored in file Variogram.txt. This information includes the variogram lags constructed, the number of distance pairs within each lags, the average square increment and the average distance value of the lag, as described in Figure 3-24.



```
2                       Number of prediction point
1                       Number of time step
47                      Number of distance pairs within the 1st lag
0.0199451907115302      Average square increment
0.0731974107851007      Average distance
67                      Number of distance pairs within the 2nd lag
0.0102802994123899
0.17868508108962
90
0.0082088696464812 4
0.284172751394139
60
0.0111682256167235
0.389660421698659
78
```

Figure 3-24: Variogram.txt file structure

In the case where the fuzzy logic system is used, another file, named neighbor.txt, becomes available. This file contains the ID number of the wells used for every prediction point and its structure can be seen in Figure 3-25.



```
1       Number of prediction point
55      Id number of 1st well used
62      Id number of 2nd well used
17      ...
30
35
40
44
45
46
47
52
29
```

Figure 3-25: neighbor.txt file structure

3.5.4. Results presentation

In the final step of the algorithm, Visual Studio communicates with MATLAB and passes through the vectors required for the visualization of the simulation results. When the simulation output is hydraulic head change, graph triplets similar to the one shown in Figure 3-26 are generated for every time step. The top graph represents hydraulic head (evaluated using the initial hydraulic head measurement values), the middle graph represents hydraulic head change and the bottom graph represents error variance. When the simulation output is hydraulic head, the graph for hydraulic head change is not illustrated in the figure. These consecutive figures are used by MATLAB to generate an .mp4 file with the temporal variations of the parameters.

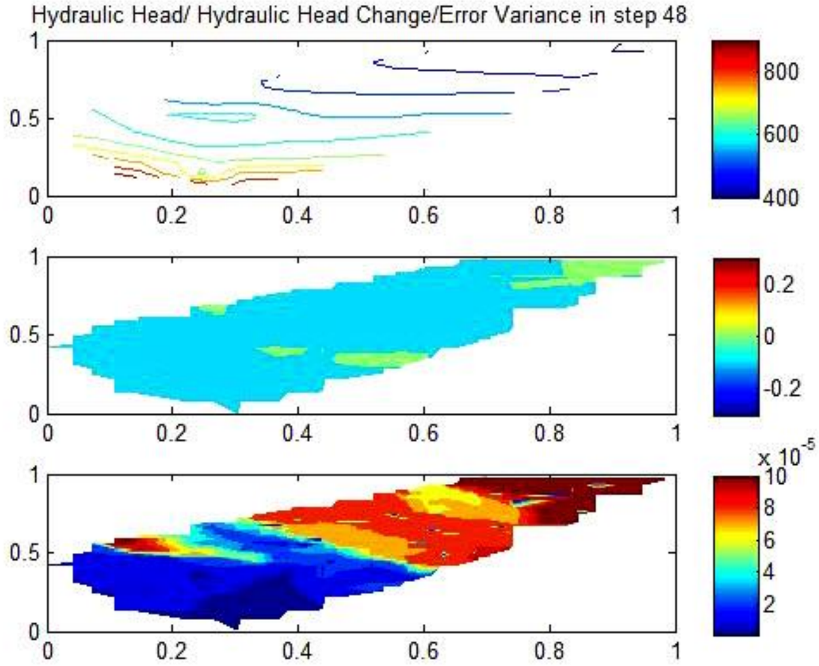
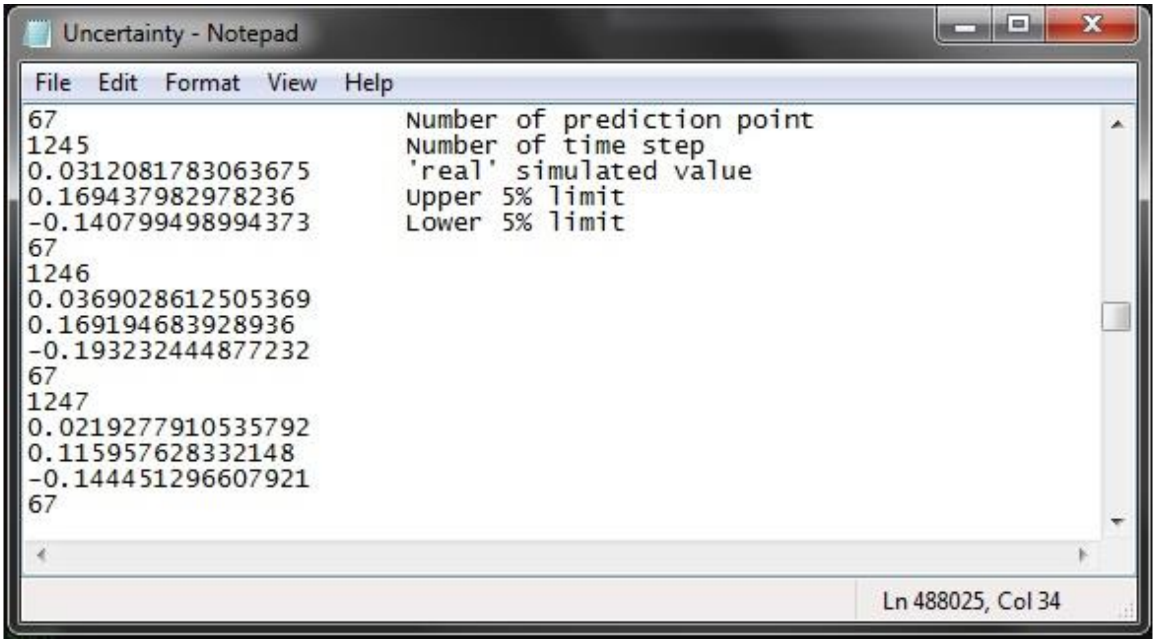


Figure 3-26: Visualization of results

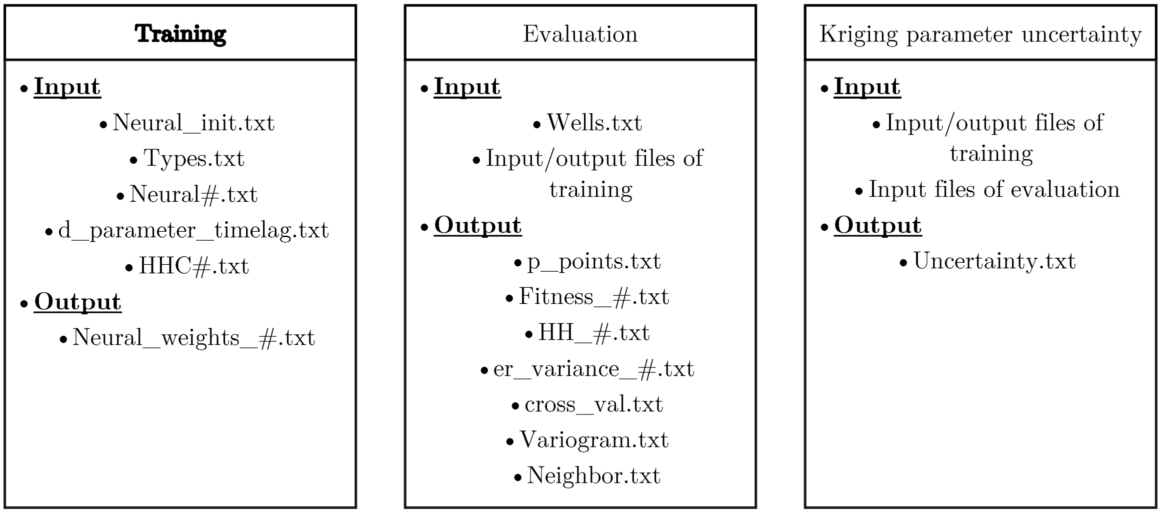
When the uncertainty of the system is examined with the Bayesian Kriging methodology, as described later in Section 3.6.2, the output file Uncertainty.txt is generated. In this case the upper and lower 95% limits of the prediction intervals are recorded, as shown in Figure 3-27.



```
67                              Number of prediction point
1245                            Number of time step
0.0312081783063675              'real' simulated value
0.169437982978236               Upper 5% limit
-0.140799498994373              Lower 5% limit
67
1246
0.0369028612505369
0.169194683928936
-0.193232444877232
67
1247
0.0219277910535792
0.115957628332148
-0.144451296607921
67
```

Figure 3-27: Uncertainty.txt file structure

In the most general case, the execution of the program implementing the proposed methodology involves three stages: ANN training, evaluation of simulation results and of other measures of interest, and uncertainty analysis. The files required as input at each stage and the files generated as output from each stage are summarized in Figure 3-28.



Training

- **Input**
  - Neural_init.txt
  - Types.txt
  - Neural#.txt
  - d_parameter_timelag.txt
  - HHC#.txt
- **Output**
  - Neural_weights_#.txt

Evaluation

- **Input**
  - Wells.txt
  - Input/output files of training
- **Output**
  - p_points.txt
  - Fitness_#.txt
  - HH_#.txt
  - er_variance_#.txt
  - cross_val.txt
  - Variogram.txt
  - Neighbor.txt

Kriging parameter uncertainty

- **Input**
  - Input/output files of training
  - Input files of evaluation
- **Output**
  - Uncertainty.txt

Figure 3-28: Summary of input and output files at different stages of the program execution

3.6 Uncertainty methodology

In order to evaluate the performance of the algorithm under different conditions, an uncertainty analysis was performed both for ANN training and for the kriging methodology.

3.6.1. ANN training

ANN predictions may suffer from uncertainty due to inaccuracies in the training set, such as noisy or incomplete, noncharacteristic time series. Uncertainty can also be attributed to the limitations of the model and of the training algorithm.

Conventionally, a single ANN is trained and, after training, the ANN parameters remain fixed. Such an ANN is a deterministic model where a given input data vector will always produce a specific output value. In order to evaluate the effect of the single ANN training performed to the output, two uncertainty calculation methodologies were implemented.

Monte Carlo method

The Monte Carlo methodology can be used for calculating of ANN model uncertainty to the output of ANNs. Monte Carlo methods are computer based techniques for brute force numerical simulation of probabilistic processes and can be summarized in the following steps:

- For the set of input variables, generate a large number of sets of random numbers that have statistical properties similar to those of the real-world variables. If there are actual measurements describing the variables, it is also possible to randomly pick among the measured values, as an approximation to the distribution underlying the particular values that were measured.
- Calculate the output variable for each set of values for the input variables generated in the first step.
- Examine the distribution of the simulated values of the output parameter to determine features of interest, e.g. prediction intervals
- The process is repeated several times to test whether the results are sensitive to the particular random values that have been generated. The number of repetitions necessary to achieve a particular level of precision is problem-dependent. The greater the number of iterations, the greater the accuracy of the prediction intervals; however, the computational costs may be enormous for a large number of iterations.

In order to investigate the uncertainty in the predictions of the ANNs, the algorithm was executed multiple times with random initial values for the neural weights and random training and testing sets. All the random number generators were set to choose a random integer number with uniform distribution. The derived results were used to determine the percentile confidence intervals for the output parameter meaning the hydraulic head or hydraulic head change per time step.

Percentile Confidence Intervals (CI)

In this approach, the large number of simulated values for the parameter of interest are ordered from smallest to largest. The 90% Confidence Interval (CI) is then computed by just identifying the lower confidence limit as the 5th percentile and the upper confidence limit as the 95th percentile. This leaves 90% of the simulated parameter estimates within this range while dividing the remaining 10% of the simulated values equally into the upper and lower tails. The advantage of this method is that it does not make any distributional assumptions; it does not even require the distribution to be symmetric. However, evaluating the CI using this methodology can be less accurate than it could be using other methodologies, but still it is one of the most commonly used ways to compute a CI.

In order to use this methodology to evaluate one ANNs output CIs, the difference between the real and the simulated output values is calculated for every time step. These values represent the range of the absolute error in the simulation results. After sorting the error values, the 5th and 95th percentile values are identified, representing the upper and lower 5% CI limits of the absolute error. By adding these values to all simulation results for the specific ANN, the 5% upper and lower limit of the timeseries prediction is calculated.

3.6.2. Kriging predictions

Two different methodologies were used to evaluate the uncertainty of the kriging algorithm predictions: one based on the Monte Carlo method and one based on Bayesian Kriging.

Monte Carlo method

The results of the Monte Carlo method applied to ANN training in the previous section were used in this case. More specifically, the multiple different training results (different neural weights) were used for the evaluation of the hydraulic head, resulting in multiple different values for every time step and for every prediction point. Using the 5th and 95th percentile we are able to determine the 90% prediction interval.

Bayesian Kriging

The process of interpolating spatial data using kriging can be realized in two steps. The covariance function/semivariogram is determined first, followed by kriging interpolation. In the classical approach, the assumption that the model parameters are known is made and the uncertainty in the calculation of these parameters is considered as minimal. The purpose of Bayesian Kriging is to quantify the uncertainty in the covariance estimation.

Using Bayes' theorem, the a posteriori distribution of a parameter can be obtained from the parameter a priori distribution and from parameter measurements. A prediction value for the parameter value can be obtained from the a posteriori distribution. In this way, a model "averaging" over the unknown model parameters is performed.

However, when using spatial data, the a priori distribution cannot be calculated easily. In Bayesian Kriging [*Pilz and Spöck*, 2008], the a posteriori distribution of the parameters involved (semivariogram and/or transformation parameters) is specified by means of simulations, reflecting the uncertainty of the parameters.

Bayesian Kriging method uses artificial data to test how different kriging parameters can affect the kriging interpolation results. The first in this approach is to calculate the variogram by fitting the theoretical variogram model to the experimental data, following the steps described in Section 3.4. Next, the covariance matrix C is constructed by using the derived kriging parameters and the covariance function of the variogram model. By decomposing, in this case Cholesky decomposition, the covariance matrix (Equation 3.7) and multiplying the triangular matrix U with the vector of random numbers R it is possible to construct a vector of random but correlated with the initial parameters values (Equation 3.8).

$$U^T \cdot U = C \tag{3.7}$$

$$R_c = R \cdot U \tag{3.8}$$

These correlated random values have the same statistical characteristics with the initial data observed in the field. These random values are then fitted to a variogram model and the parameter value at any location (for which parameter data is unavailable) is estimated using kriging. After constructing several correlated random value vectors, the 90% prediction intervals of the simulated parameter (either hydraulic head or hydraulic head change) can be calculated.

This methodology has in its roots in Bayesian statistics since the simulated ANN values are used for the initial variogram construction, acting as prior knowledge, while the posterior distribution of the simulated parameter reflects the uncertainty due to the covariance estimation and hence the kriging parameters [*Pilz and Spöck*, 2008].

3.6.3. Uncertainty flowcharts

The flowcharts followed in both cases of uncertainty analysis are shown in Figure 3-29.

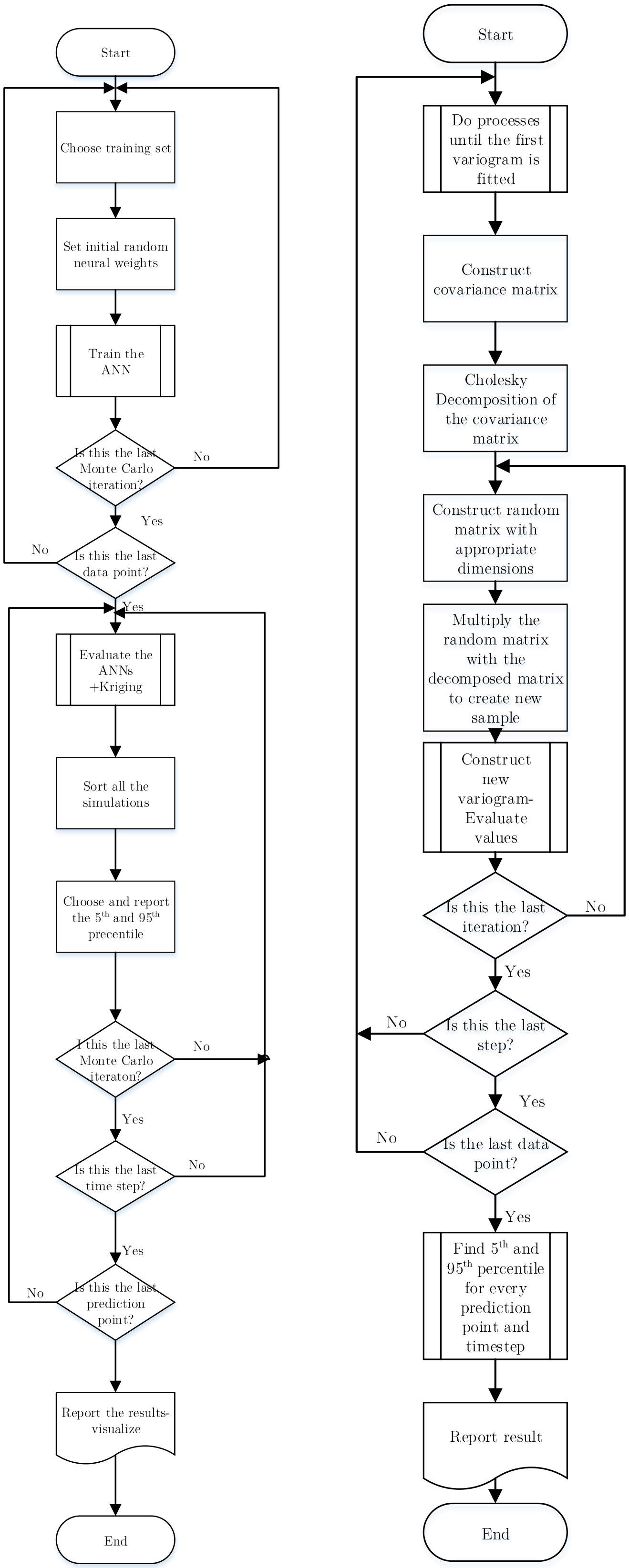

Start
Choose training set
Set initial random neural weights
Train the ANN
Is this the last Monte Carlo iteration?
No
Yes
Is this the last data point?
No
Yes
Evaluate the ANNs +Kriging
Sort all the simulations
Choose and report the 5th and 95th precentile
I this the last Monte Carlo iteraton?
No
Yes
Is this the last time step?
No
Yes
Is this the last prediction point?
No
Report the results-visualize
End
Start
Do processes until the first variogram is fitted
Construct covariance matrix
Cholesky Decomposition of the covariance matrix
Construct random matrix with appropriate dimensions
Multiply the random matrix with the decomposed matrix to create new sample
Construct new variogram-Evaluate values
Is this the last iteration?
No
Yes
Is this the last step?
No
Yes
Is the last data point?
No
Yes
Find 5th and 95th percentile for every prediction point and timestep
Report result
End

Figure 3-29: Flowcharts for Uncertainty analysis: (a) Monte Carlo and (b) Bayesian Kriging

In order to define the ANN training uncertainty the procedure followed is as follows. For a predefined number of realizations, the training dataset and the initial values of synaptic weights values are defined randomly. One ANN is trained for each monte carlo iterations. After completing all monte carlo iterations for all data points, the simulation using one set of trained ANNs and Kriging takes place. The simulation results are the sorted and the 95th and 5th percentiles are stored. After completing this procedure for all monte carlo simulations, for all time steps and prediction points, the results are passed to the MATLAB for visualization.

In Bayesian Kriging interpolation, an initial simulation, as described in Figure 3.7 is followed. After the first initial variogram fitting, using the parameters (sill and range) calculated and the covariance function the corresponding matrix is calculated. This matrix is decomposed and multiplied with a matrix containing random values. A new variogram is then fitted to the new samples of the matrix. This process is repeated for all iterations for all time steps and for all data points. The results for every time step and prediction point are sorted and the 95th and 5th percentiles are identified. The results are reported back to the main algorithm and saved in text files.

Chapter **4**

Study area

The proposed methodology was applied for the simulation of the hydraulic head change/hydraulic head in two different case studies. The two study areas were selected with diverse characteristics in order to illustrate the flexibility of the proposed methodology. In the first case the methodology is applied to a rural, large area in Bavaria, Germany, and in the second case to an urban, small scale area in Miami, Florida, USA. Data for different parameters were used in each case based on the particular characteristics of the study area and depending on data availability. In both cases a daily time step was used to simulate the water lever in the aquifer. In this chapter, the geology and climate of the two study areas are presented. Moreover, the available meteorological and hydrogeological data were analyzed in order to determine the appropriate ANN structure for every case.

4.1 First Case Study – Bavaria, Germany

4.1.1. General information

The first study area (Figure 4-1) where the proposed algorithm was tested is located in the southern part of Germany, in Bavaria, and covers approximately 7800 km^2. The study area includes a large rural part as well as the city of Munich. The elevation in the study area varies from 500 m in the northern part to 950 m in the southern part. The Isar River flows through the study area, as well as its tributaries Amper and Wurm. The Isar River has a total length of 291 km and a mean discharge of 173 m^3/s. Various studies on both the quantity and the quality of groundwater in the region have been conducted over the years and a spring monitoring network has been developed [Bender et al., 2001; Heinrichs and Udluft, 1999]. Moreover, the Bavarian State Office for the Environment (Bayerische Landesamt für Umwelt - BLfU) has participated in numerous studies concerning both surface and ground water in the region. For example, Blomenhofer et al. [2012] studied the effects of climate change on water balance and groundwater while also investigating the long-term behavior of groundwater levels and spring discharges [Blomenhofer et al., 2011].

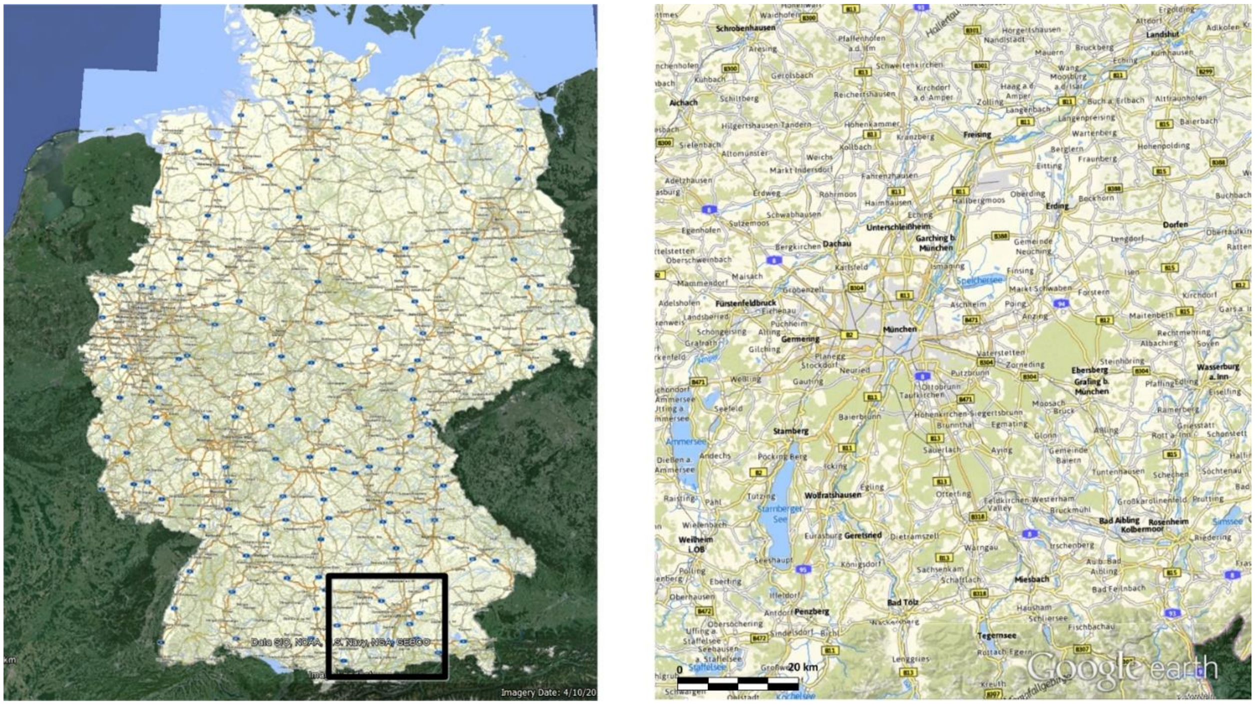


Figure 4-1: Area of Study *[Google Earth]*

Geology

The geology of the study area is dominated by loose sediments, like gravel, sands and clays, which accumulated in continental environments over the past 15 million years. Deposition took place in fluvial as well as glacial sedimentary settings. Floodplains, alluvial fans and glacial sander terraces were formed by and subjected to repeated phases of weathering and erosion. More specifically, Molasse deposits (Neogene) and Glacial and post-glacial gravel deposits (Quaternary) are the main geological formations in the region. The stratigraphy around Munich is presented in Figure 4-2 and the geology of the study area in Figure 4-3 [*Bauer et al.*, 2009].

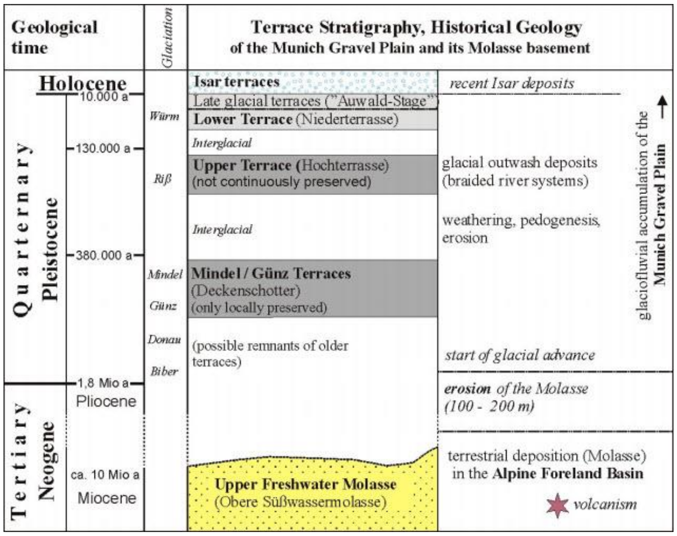


Figure 4-2: Stratigraphy in Munich [*Bauer et al.*, 2009]

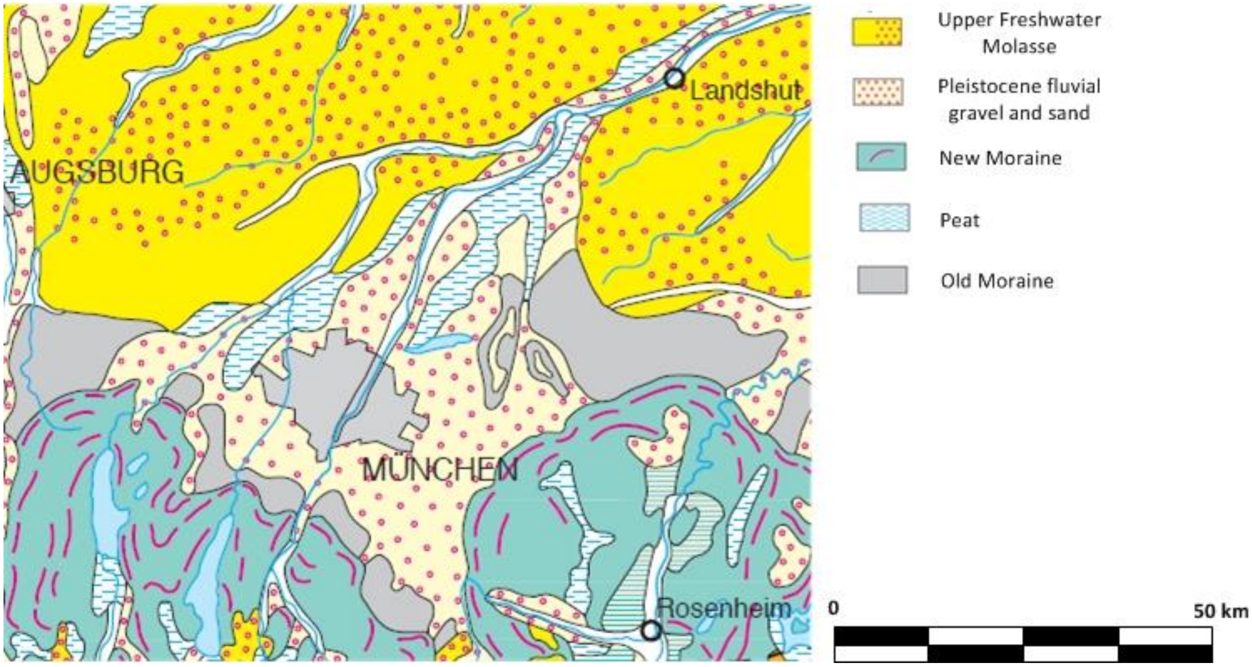


Figure 4-3: Geology of the 1st study area [*Bauer et al.*, 2009]

The subsurface of Munich represents a multiple aquifer system that is closely related to the lithostratigraphy. Basically, an upper rather homogenous aquifer is situated within the Quaternary gravel deposits which must be distinguished from groundwater occurring in granular soils of the Neogene Molasse. Furthermore, karstic groundwater occurs in Jurassic limestones that underlay the Molasse in more than 2 km depth. A simplified sketch of the alluvial ground is presented in Figure 4-4. The permeability in the study area can be characterized as low to medium, ranging from 10^{-6} to 10^{-4} cm^{2}.

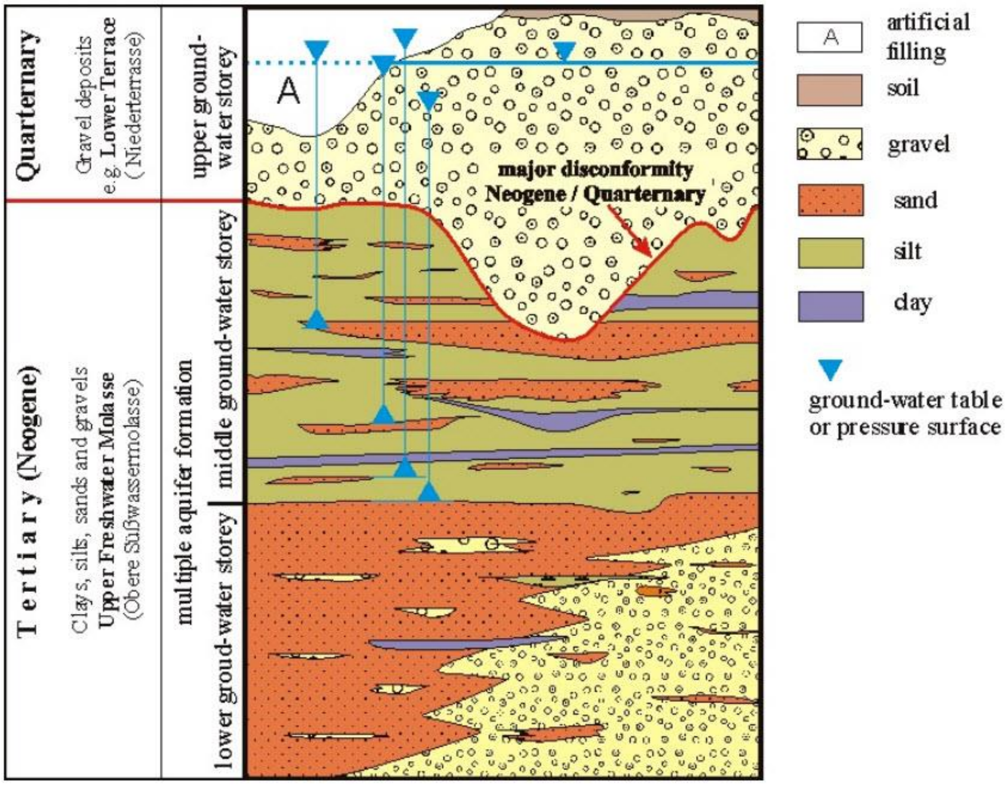


Figure 4-4: Simplified sketch of the alluvial ground in the study area [*Bauer et al.*, 2009]

Climate

Deutscher Wetterdienst (DWD), Germany's National Meteorological Service, is the source for the climate data used in this case study. The reference period for the data is from 1981 to 2010. The data were obtained from the MUENCHEN-STADT (WST) meteorological station at latitude 48^{o} 10', longitude 11^{o} 33' and elevation 515.2 m. The selected station, located in the center of the study area, is ideal to provide an overview of the climatic conditions in the study area.

The annual temperature ranges from -5^{o} C to 23^{o} C. Only in extreme conditions the temperature falls below -12^{o} C or rises above 29^{o} C. During winter, the temperature is usually below 0^{o} C, while January is characterized as the coldest month with an average temperature of 0.3^{o} C. In spring the temperature rises gradually from March until late May. Summer generally runs from late May through August with maximum temperatures above 20^{o} C. The hottest month is the month of July, with an average temperature of 19.4° C (Figure 4-5).

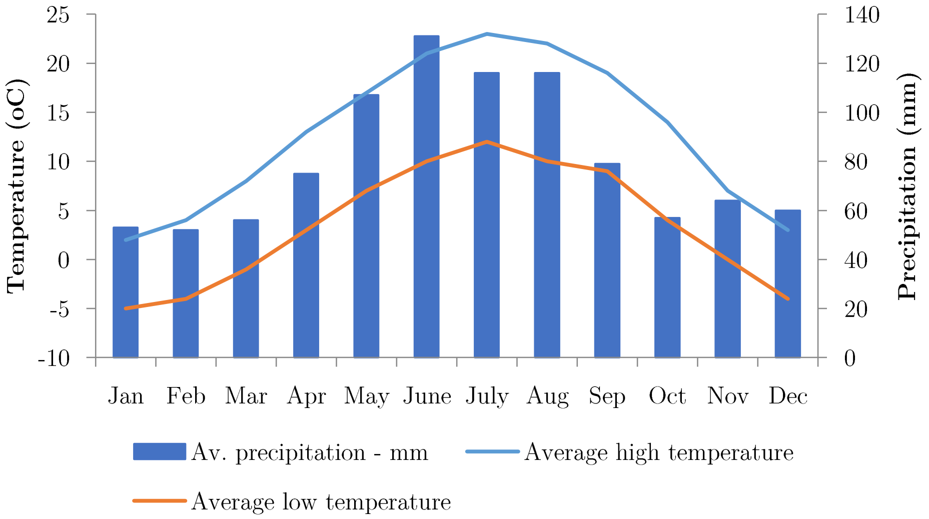


Figure 4-5: Average precipitation and average high and low temperature at Munich weather station

The average annual rainfall in Munich is 944 mm. Showers and thunderstorms that occur in late spring and summer contribute by far the largest amount of precipitation than the rest of the time periods. This behavior is typical for areas with humid continental climate. The highest average monthly precipitation of 131 mm occurs in June, while the lowest average monthly precipitation of 52 mm occurs in February. The month having the most rainfall is June with 13.8 mm on average, while the month with the less rainfall is February (8.6 mm). The average monthly precipitation (mm) is depicted in Figure 4-5.

4.1.2. Available data

The available data for this study area included the daily time series for temperature, precipitation, wind speed, relative humidity, snowfall and the hours of daily sunshine for nine (9) meteorological stations in or around the study area. Daily discharge and water level across the Isar River for fifteen (15) monitoring stations is also available. Daily hydraulic head measurements were available for sixty-four (64) wells in the study area, while incomplete hydraulic head time series were available for two more wells. The data cover the time period from 1/11/2008 to 31/10/2012 (1460 days - time steps). The data regarding water levels used in this study were acquired through the Bavarian State Office for the Environment (Bayerische Landesamt für Umwelt – BLfU), while the meteorological data from the German National Meteorological Service (Deutscher Wetterdienst – DWD).

Some parameters, such as relative humidity, wind speed and daily sunshine, were disqualified from being input parameters from the outset. Due to the very low correlation coefficient between these parameters and hydraulic head change, they are expected to increase the complexity of the ANNs without improving the results. In previous work [*Chatzakis et al.*, 2014], it was shown that snowfall as an input parameter does not improve the results of the ANNs, due to long sequences of consecutive time steps with zero value and due to high values of time lags involved. For this reason, snowfall was also not considered as ANN input parameter in this study. The rest of the parameters were examined in order to determine the most appropriate monitoring stations and the optimal time lags, using the correlation coefficient. The data from some of the monitoring stations, both meteorological and surface water, were not used as ANN inputs due to the poor correlation between hydraulic head and the data parameter. Finally, data from seven (7) meteorological stations (München, Hohenpeibenberg, Kempten, Augsburg, Regensburg, Oberstdorf, Stötten), from five (5) surface water monitoring stations (Freising, Landshut Birket, Lenggries, München, Puppling) and from sixty-four (64) data points (wells) were used. The locations of the final input parameter stations and the data points (outputs) are depicted in Figure 4-6. The two data points with incomplete data time series were used for the validation of the algorithm.

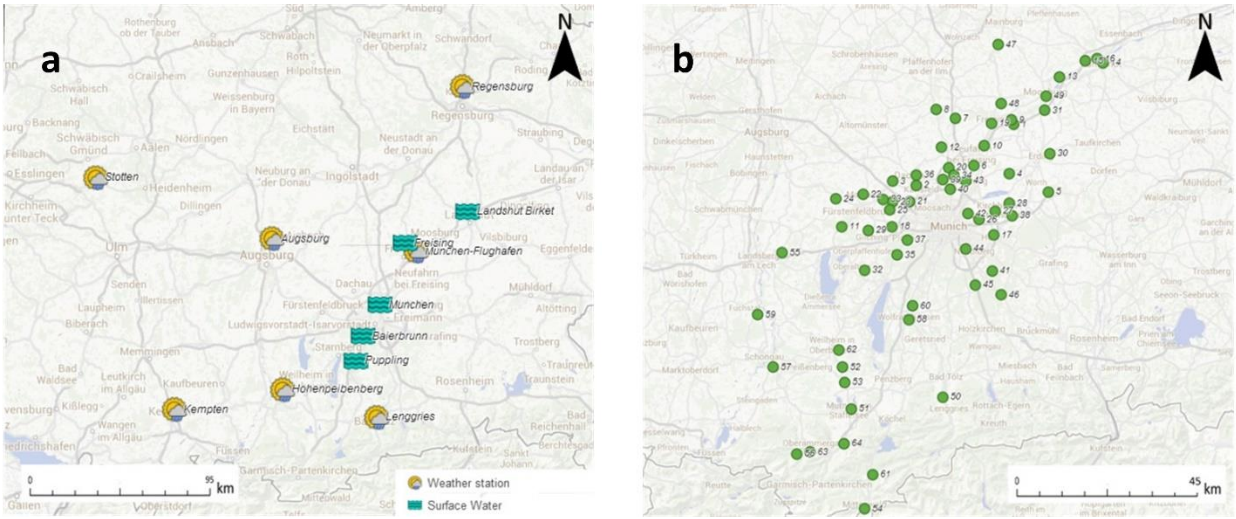


Figure 4-6: (a) Weather and surface water monitoring stations (b) Data points

4.1.3. ANN structure/Kriging implementation

The most appropriate input parameters for every well/ANN are determined by taking into consideration the well's proximity to the monitoring station, and the statistical characteristics of the parameter data. Some hydrological input parameters appear to have a time lag with respect to the hydraulic head change which is the output parameter. To determine the optimal time lag for each input parameter, various values of time lag between the parameter and the hydraulic head change were tested. Finally, the input parameters to the artificial neural network were selected from the parameters with the best correlation coefficient. More specifically, for temperature, the time series from the Munich weather station was used because it was the best correlated for all wells. As for rainfall, for each ANN, data from the two meteorological stations with the best correlation co4fficient were used. In total, four inputs for rainfall were used for rainfall, the two best correlated time lags (consecutive days) for each meteorological station [*Chatzakis et al.*, 2014; *Tapoglou et al.*, 2014].

The same procedure was followed for the surface water level parameter and its monitoring stations. The stations with the three best correlation coefficients were used as input parameters for each case. In summary, each ANN has eight (8) inputs parameters:

- four (4) rainfall inputs (two best lags for each of the two stations with the best correlation)
- one (1) temperature input from the Munich station
- three (3) surface water level inputs (two stations, best lags)

The hydraulic head change in the well per time step is the output parameter for every ANN.

As mentioned in Section 3.1.1, 80% of the available data are used for ANN training and the remaining 20% for testing the network. Two possible architectures were tested based on the rule by*Fine* [1999] stating that using three times as many training samples as network parameters is adequate to achieve good generalization. For a data set consisting of 1168 training samples, an ANN with approximately 390 weights can achieve good generalization. The weights are calculated during the training process. The number of weights in the network depends on the network architecture, more specifically, on the number of layers and nodes. If the nodes are arranged in a single hidden layer, then a network with 43 hidden nodes (Architecture 1), as shown in Figure 4-7, has the desired total number of weights. If the nodes are arranged in two hidden layers, 20 hidden nodes in the first layer and 11 hidden nodes in the second layer (Architecture 2) will ensure an acceptable total number of weights.

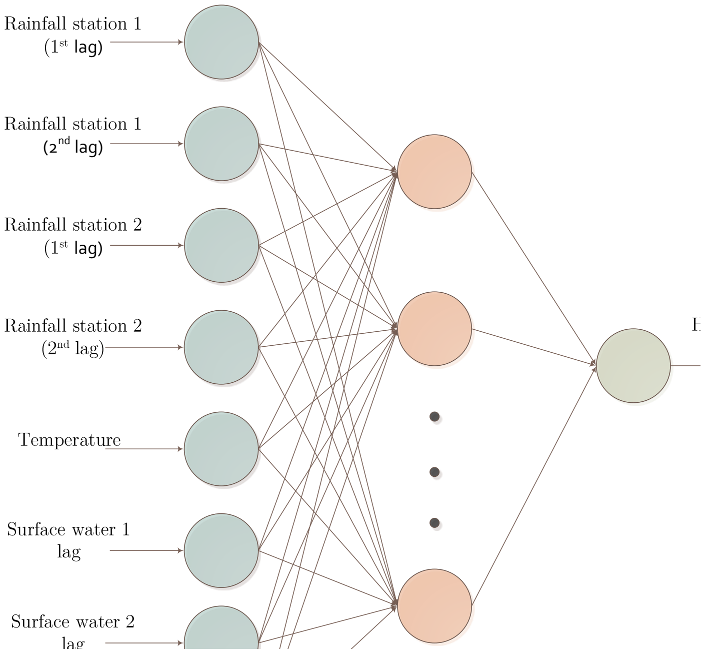


Figure 4-7: ANN structure with a single hidden layer for the first study area

As far as the kriging implementation is concerned, the number of neighbors for each prediction point was set to thirty, in order to satisfy the rules discussed in Chapter 3. In the case where fuzzy logic is not used, the neighborhood for each prediction point was selected based on the distance between the prediction point and the sampling points. Otherwise, when fuzzy logic is used, the appropriate neighborhood is determined based on both this distance and the ANN simulation error, as described in Section 3.1. Furthermore, three different variogram models, linear, exponential and power-law, were tested in order to determine the most appropriate variogram for the particular study area.

4.2 Second Case Study – Miami Dade County, Florida, USA

4.2.1. General information

The second study area is located in Miami-Dade County, Florida, USA. In contrast to the previous study area, this is an urban area with most of the wells inside or around the city of Miami. The study area covers approximately 1460 km^2 (Figure 4-8).

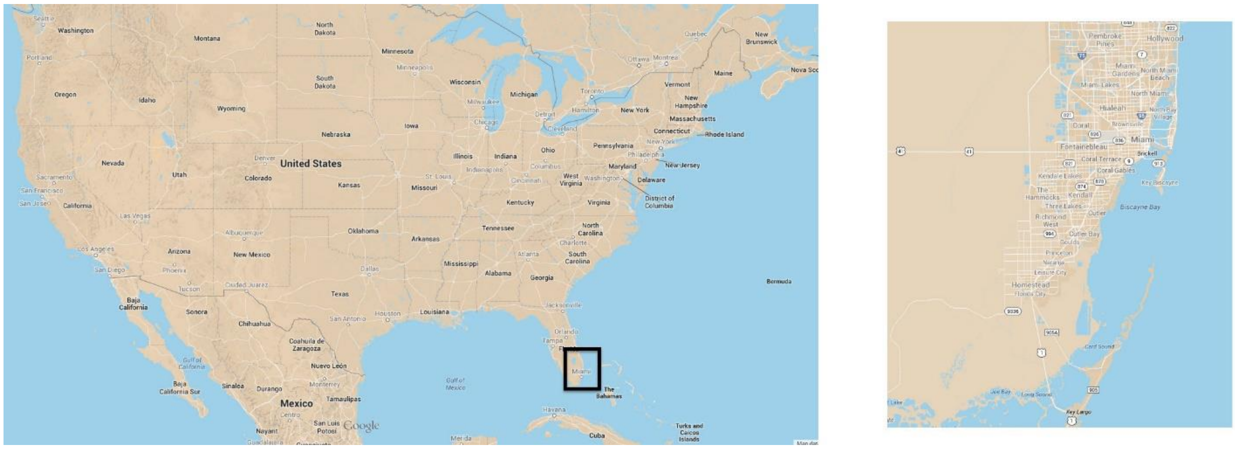


Figure 4-8: Second study area, Miami Dade county (Google maps Engine)

It is a coastal area with elevation that varies from 0 m in the east to 5 m in the west and it is part of the Biscayne aquifer. The Biscayne aquifer is one of the most productive aquifers in the world; it is unconfined and shallow with the average hydraulic head often being less than 1 m. It extends from table water a couple meters thick at the west to as much as 45 m thick near the east coast. It consists of very porous limestone with large pores created by secondary dissolution (Figure 4-9). Due to characteristics like large permeability, proximity to the landscape and its location (in an urban area with industrial growth) the Biscayne aquifer is very susceptible to pollution [*Fish and Stewart*, 1991].

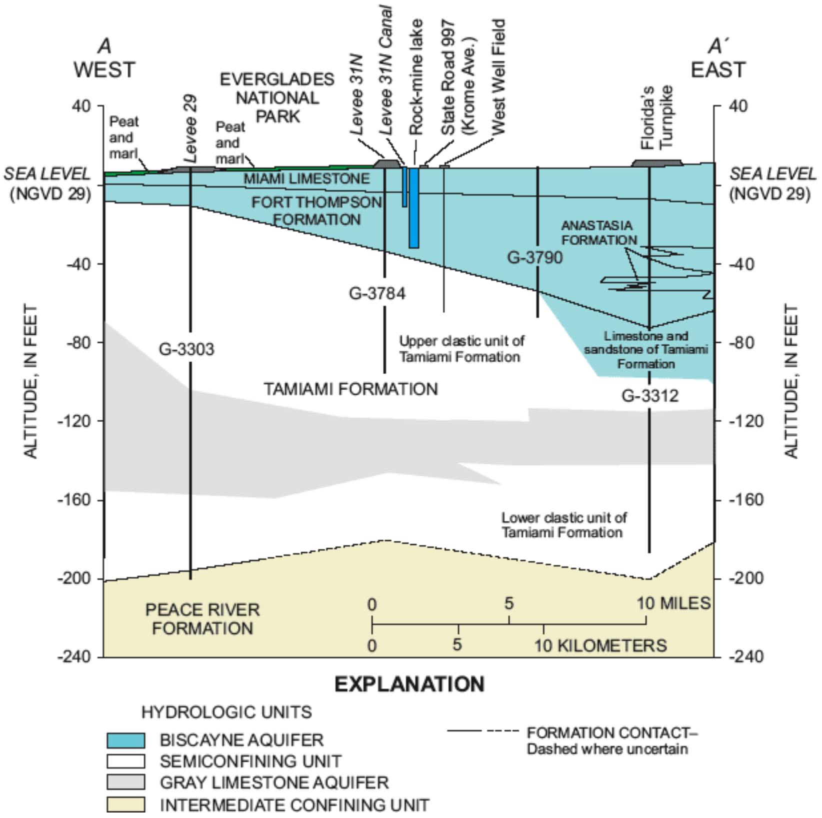


Figure 4-9: Hydrogeological section showing schematics of geological formations, aquifer and semipermeable units [*Cunningham et al.*, 2003]

Geology

As analyzed by USGS [2014], geologically, the study area is dominated by Miami limestone and Key Largo Limestone. The highly porous and permeable Miami Limestone forms much of the Biscayne Aquifer of the surficial aquifer system. Key Largo Limestone is highly porous and permeable and is part of the Biscayne Aquifer of the surficial aquifer system. The dissolution of these rocks has also formed Karst in the Pleistocene limestone. The geology of the southern Florida peninsula is depicted in Figure 4-10.

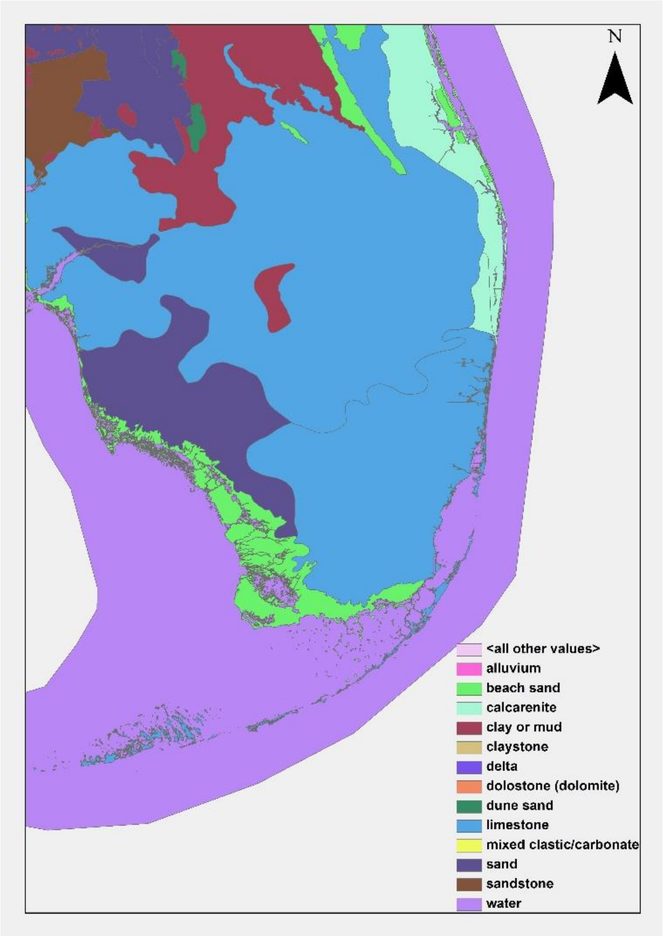


Figure 4-10: Geology of Southern Florida Peninsula (data provided by USGS)

Climate

The climatological information mentioned below is acquired through the National Climatic Data Center of the National Oceanic and Atmospheric Administration (NOAA) of the USA. The region where the study area is located has tropical monsoon climate with hot and humid summers and short warm winters. The annual temperature ranges from 60° F (15.5° C) in the winter to around 90° F (32.2° C) in the summer. The wet season begins early in May and usually ends mid-October. During this period, the temperature is in the mid-80s to low 90s, accompanied with high humidity. Thunderstorms, hurricanes and rainfalls are common during this period.

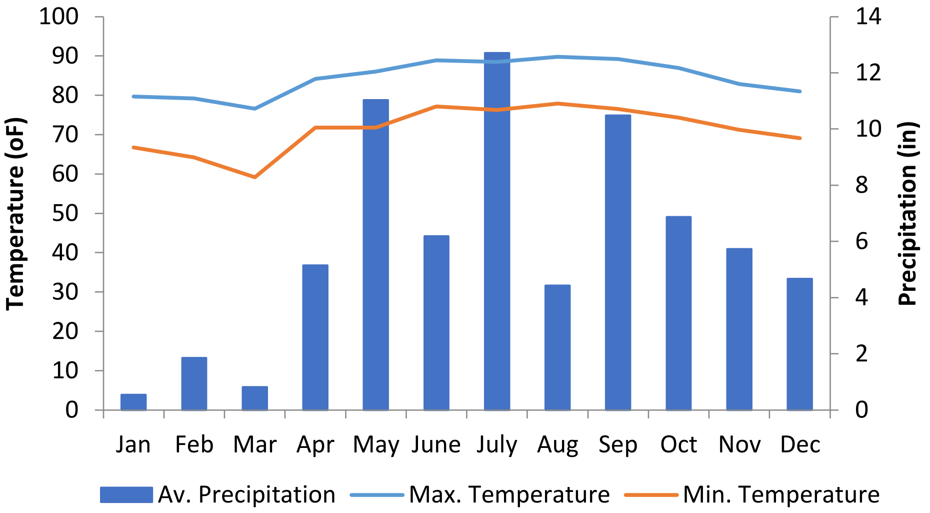


Figure 4-11: Average maximum and minimum Temperature and precipitation in Florida for 2013

The distribution of the average monthly maximum and minimum temperatures, along with the average monthly rainfall for the year 2013 is displayed in Figure 4-11. May and June, followed by September, were by far the wettest months of the year. The temperature is all year round at the same level, with the exceptions on February and March when they drop for about 10^o F.

4.2.2. Available data

The data available for this study area included the daily time series for maximum, minimum and average soil and air temperatures, precipitation at 2 m, wind speed, humidity and hours of daily sunshine from two (2) meteorological stations in or near the study area. Daily discharge and water level in water bodies from four (4) monitoring stations were also available. Daily hydraulic head measurements were available for 30 wells in the study area. The data cover the time period from 25/4/2010 to 22/2/2014 (1400 days - time steps). The data regarding water levels were acquired through United States Geological Survey (USGS), while the meteorological data from the National Climatic Data Center of the National Oceanic and Atmospheric Administration (NOAA).

As in the first case study, not all parameters for which data were available were considered in the ANN development. Wind speed and hours of daily sunshine as well as the maximum and minimum soil and air temperatures were considered of minimum importance and were not used. Regarding surface water, data from only three of the monitoring stations were available; two of the stations provided the data for the water level and the third for the river discharge. The locations of the surface water monitoring stations (a), of the hydraulic head monitoring stations (b) and of the meteorological stations (c) are depicted in Figure 4-12.

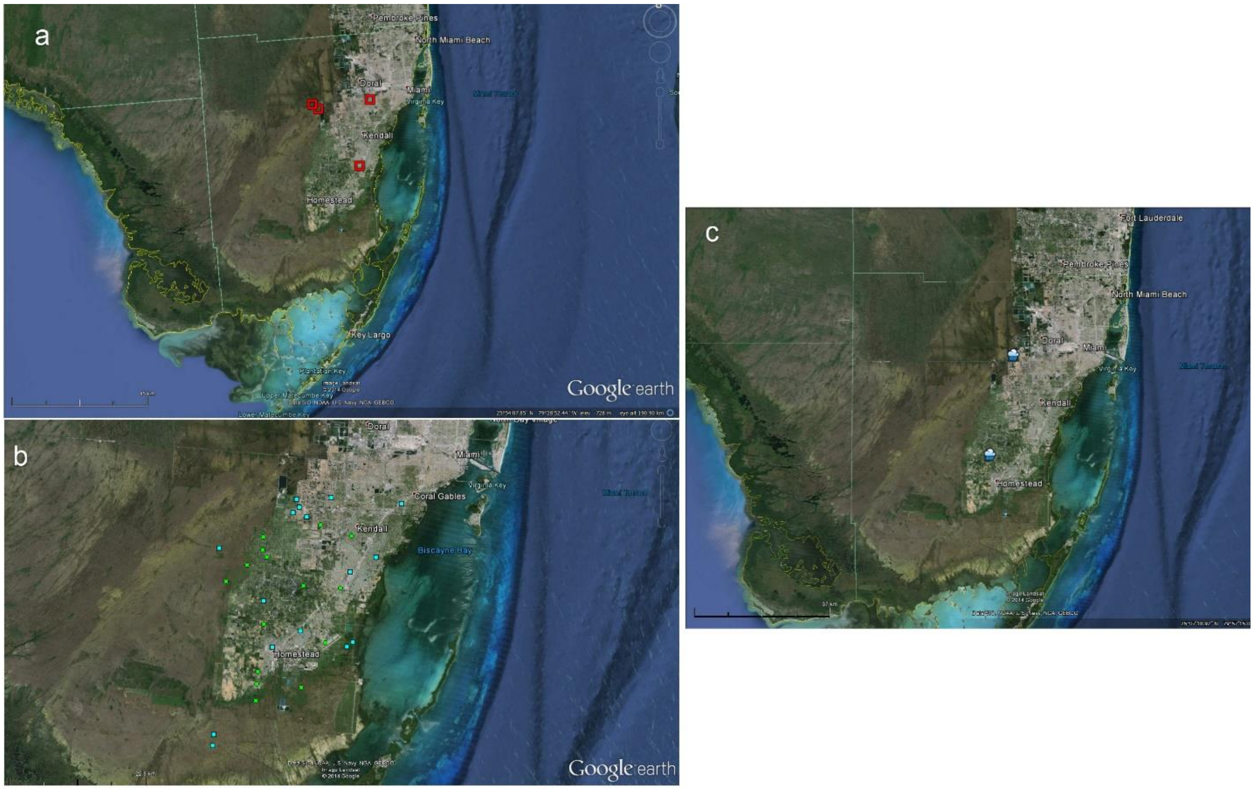


Figure 4-12: (a) Surface water monitoring stations, (b) Hydraulic head monitoring stations and (c) Meteorological stations

4.2.3. ANN structure/Kriging implementation

Following the same procedure as in the first study area, we concluded in the following input parameters for the ANNs:

- Precipitation from two meteorological stations (three lags)
- Soil temperature
- Air temperature
- Humidity
- Surface water level from two stations (two lags)
- Discharge from one station (best lag)

Since the aquifer is shallow, some parameters which were not considered in the case of the first study area, such as humidity and soil temperature, have greater significance in this case. Moreover, due to the smaller size of this second study area, there was no need to use data from more than two stations, even though data may have been available.

Contrary to the first case study, the output parameter for each ANN is the hydraulic head in the well per time step, instead of the hydraulic head change considered in the first case. This is due to the small values of the hydraulic head which result in very small changes in the hydraulic head per time step.

Taking into account the nine (9) input parameters and one output parameter, the 1100 available training sets and the rule of Fine [1999] two architectures were examined: the first of a single hidden layer with 32 hidden nodes (Architecture 1) and the second of two hidden layers with 17 and 12 hidden nodes in the first and second layer, respectively (Architecture 2). ANN Architecture 2 for the study area is shown in Figure 4-13.

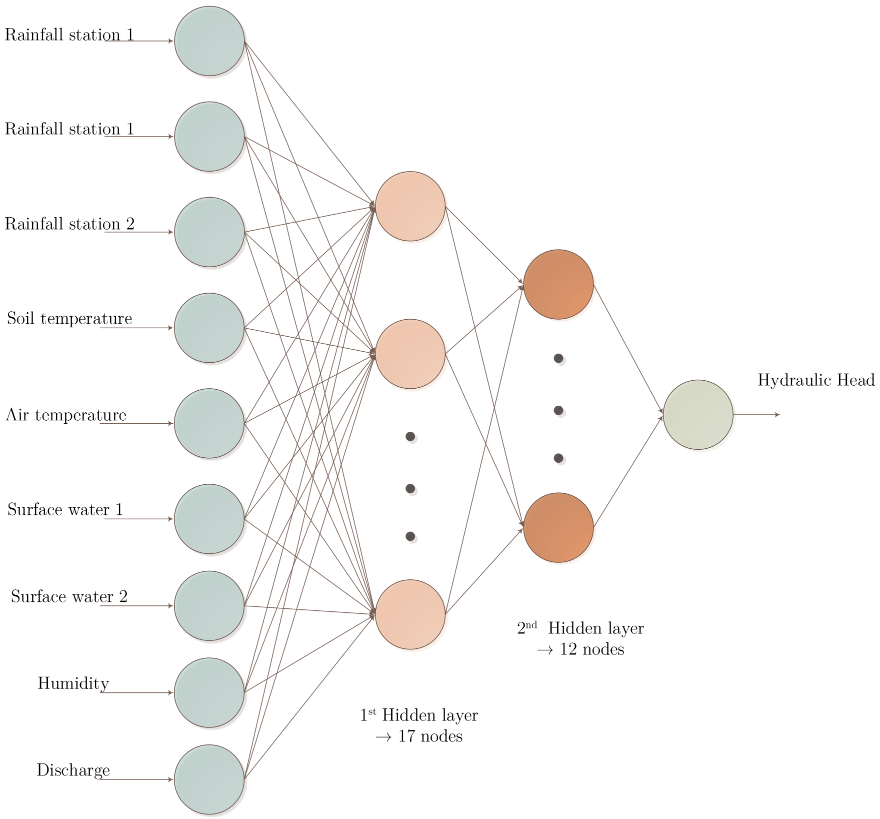


Figure 4-13: ANN structure with two hidden layers for the second study area

For this case study, the neighborhood size was also tested when using the fuzzy logic system. Taking into consideration the limitations described in Chapter 3, three neighborhood sizes were considered for 15, 20 and 25 neighbors.

Similarly to the first case study, the linear, exponential and power-law variogram models were tested in order to determine the most appropriate for the data at hand.

Chapter **5**

Results

The developed methodology was applied to two case studies with the appropriate modifications necessary to accommodate the specific characteristics of the different study areas. The simulation results for each study area, as well as various tests conducted in order to achieve the optimal results, are presented in this chapter.

For the case study in Bavaria, the dataset included information for a considerable number of observation wells (64) and meteorological data. The results concerning ANN training are presented first, followed by the results for fuzzy logic and kriging. Uncertainty results for ANN training and for the Kriging parameters are presented last.

For the case study in the Miami-Dade County, data for fewer observation wells (30) were available, but for a greater variety of parameters. This study area presents a higher degree of complexity due to the karstic aquifer present and due to the fact that it is entirely in an urban area. In the case of this study area, the fuzzy logic system was tested based on the improvements observed as a result of its use in the previous case study. Results for ANN training and kriging as well as the uncertainty results are presented.

5.1 First Case Study – Bavaria

The time lags for the rainfall and surface water level parameters must be studied first. The time lags for these parameters were computed for each data point separately. For rainfall, time lags between 0 and 15 days were tested, while for the surface water level, the time lags tested were between -3 and 3 days. Negative time lags were tested in the case of surface water level, because the river may supply the aquifer with water, but it is also possible that the aquifer supplies the river with water. In the end, the time lags used varied from 0 to 5 days for rainfall and from -1 to 1 for surface water level, while the average value for time lag was 1 day for rainfall and 0 days for surface water level. Due to the large extent of the study area and the difference in land use and geology between the monitoring stations and each data point, data from several monitoring stations were used, due to the large dimension size and the difference and use and geology structures between the measuring station and the data point. The meteorological data from the Munchen, Hohenpeibenberg, Kempten and Augsburg stations were used in most of the cases, while for the surface water data, the Landshut Birket and Munchen stations were the most commonly used. The correlation coefficient for the data time series used varied from 0.2 to 0.8.

Results are presented in the sequel for five representative wells. The selected wells represent five error categories, as explained in Section 4.1.3. For these representative wells, the most appropriate time lags for precipitation are presented in Table 5-1 and for surface water level in Table 5-2. For precipitation, the best correlation coefficient value for every well and the respective monitoring station, which determine the data time series used by the ANN are presented in Figure 5-1. The same information for surface water level is presented in Figure 5-2.

Table 5-1: Time lags used for precipitation for the five representative wells

Well number	1st station			2nd station			3rd station		
	Name	Time lag	Correlation coefficient	Name	Time lag	Correlation coefficient	Name	Time lag	Correlation coefficient
60	Augsburg	0	0.4	Kempten	0	0.5	Oberstdorf	0,1	0.46,0.45
55	Stotten	1	0.42	Augsburg	0,1	0.62,0.57	Kempten	1	0.33
58	Augsburg	0	0.44	Kempten	0,1	0.49, 0.45	Oberstdorf	0	0.47
64	Hohenpeibenberg	0,1	0.55,0.49	Kempten	0	0.54	Oberstdorf	0	0.61
44	Munchen	0,1	0.44,0.39	Hohenpeibenberg	0	0.39	Augsburg	0	0.36

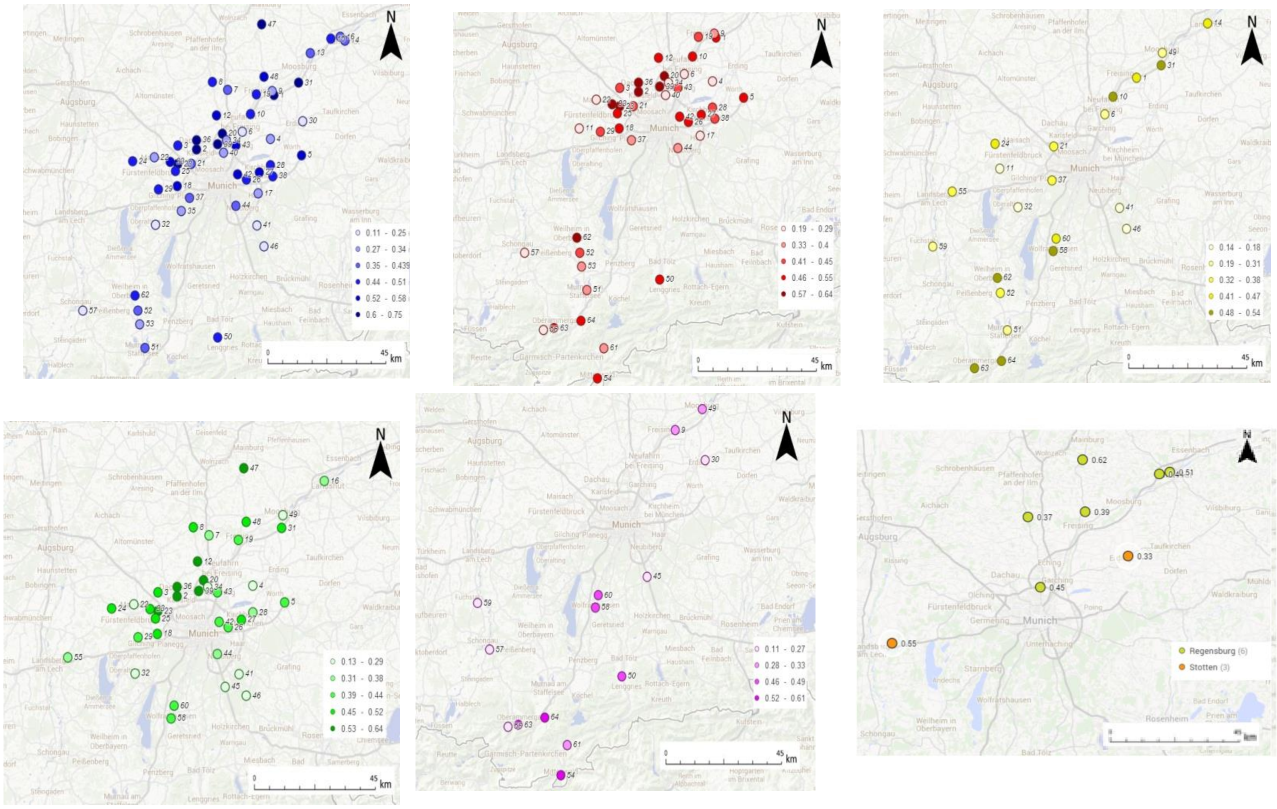


Figure 5-1: Best correlation coefficient values for used meteorological stations. From left to right and top to bottom: Munchen, Hohenpeibenberg, Kempten, Augsburg, Regensburg and Oberstdorf and Stotten

Table 5-2: Time lags used for surface water level for the five representative wells

Well number	1st station			2nd station			3rd station		
	Name	Time lag	Correlation coefficient	Name	Time lag	Correlation coefficient	Name	Time lag	Correlation coefficient
60	Landshut Birket	-1	0.4	Lenggries	0,-1	0.42-0.39			
55	Landshut Birket	1,0,-1	0.44-0.41-0.36						
58	Landshut Birket	-1	0.51	Lenggries	0	0.46	Munchen	-2	0.48
64	Landshut Birket	-1	0.35	Lenggries	-1	0.2	Munchen	0	0.29
44	Landshut Birket	0,-1,-2	0.70-0.69-0.63						

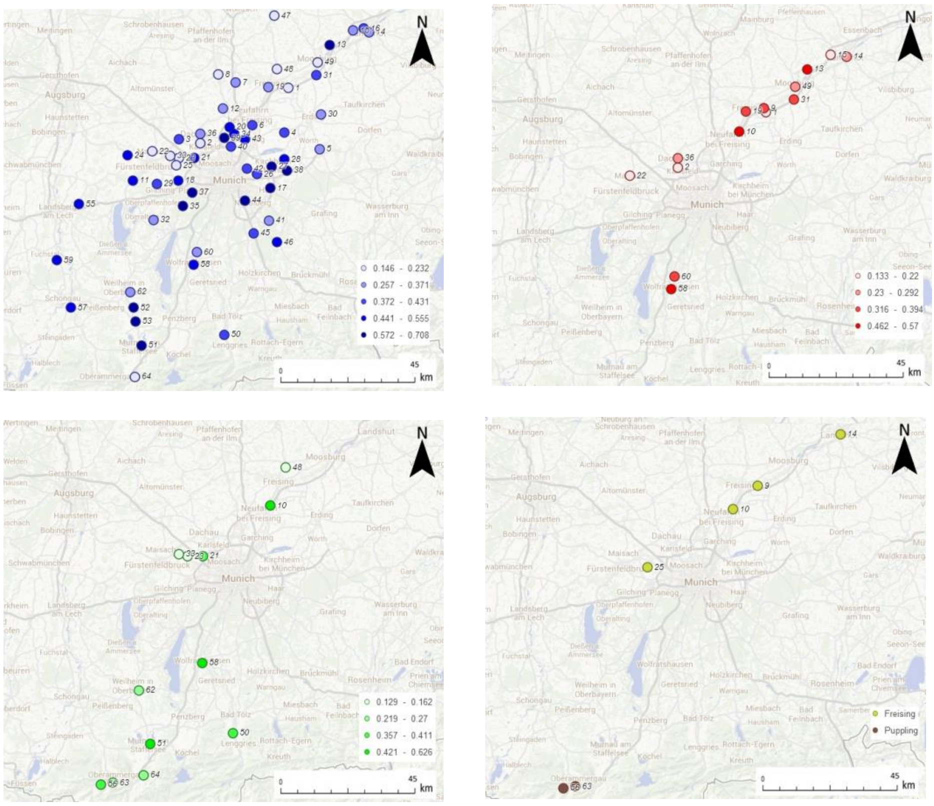


Figure 5-2: Best correlation coefficient values for used surface water level monitoring stations. From left to right and top to bottom: Landshut Birket, Munchen, Lenggries and Freising-Puppling

5.1.1. ANN results

Sixty-four (64) ANNs, one for each data point, were implemented using the input parameters described in Section 4.1 for each of the two different architectures. Recall that Architecture 1 has a single hidden layer with 43 nodes and Architecture 2 has two hidden layers, the first with 20 and the second with 11 nodes. Results for the training and testing RMSE values (average, maximum and minimum values) for the two architectures are presented in Table 5-3

Table 5-3: Statistical characteristics for the two ANN architectures under consideration

RMSE (m)	Architecture 1	Architecture 2
Average Training Error	$1.503 \cdot 10^{-4}$	$1.492 \cdot 10^{-4}$
Average Testing Error	$1.872 \cdot 10^{-4}$	$1.832 \cdot 10^{-4}$
Maximum Training Error	$7.754 \cdot 10^{-4}$	$7.601 \cdot 10^{-4}$
Maximum Testing Error	$7.195 \cdot 10^{-4}$	$7.045 \cdot 10^{-4}$
Minimum Training Error	$1.545 \cdot 10^{-5}$	$1.636 \cdot 10^{-5}$
Minimum Testing Error	$4.747 \cdot 10^{-5}$	$4.708 \cdot 10^{-5}$

Architecture 2 yielded better results than Architecture 1 in terms of training and testing errors, except for the case of minimum training error. The performance of Architecture 2 was overall better and will be used henceforth.

Results for the training and testing errors for every ANN, that is for every data point (well), are presented in Table 5-4.

Table 5-4: ANN training and testing error results (Architecture 2)

Well #	Training Error (m)	Testing Error (m)
1	$1.43 \cdot 10^{-4}$	$2.08 \cdot 10^{-4}$
2	$1.20 \cdot 10^{-4}$	$1.15 \cdot 10^{-4}$
3	$1.15 \cdot 10^{-4}$	$1.21 \cdot 10^{-4}$
4	$7.52 \cdot 10^{-5}$	$9.38 \cdot 10^{-5}$
5	$1.06 \cdot 10^{-4}$	1.2710^{-4}
6	$7.39 \cdot 10^{-5}$	$2.11 \cdot 10^{-4}$
7	$9.40 \cdot 10^{-5}$	$1.28 \cdot 10^{-4}$
8	$4.96 \cdot 10^{-5}$	$6.17 \cdot 10^{-5}$
9	$1.56 \cdot 10^{-4}$	$5.19 \cdot 10^{-4}$
10	$6.00 \cdot 10^{-5}$	$7.27 \cdot 10^{-5}$
11	$4.96 \cdot 10^{-4}$	$4.28 \cdot 10^{-4}$
12	$1.15 \cdot 10^{-4}$	$1.11 \cdot 10^{-4}$
13	$6.19 \cdot 10^{-5}$	$8.93 \cdot 10^{-5}$
14	$9.01 \cdot 10^{-5}$	$7.22 \cdot 10^{-5}$
15	$8.91 \cdot 10^{-5}$	$1.38 \cdot 10^{-4}$
16	$1.38 \cdot 10^{-4}$	$2.34 \cdot 10^{-4}$
17	$1.56 \cdot 10^{-4}$	$2.90 \cdot 10^{-4}$
18	$8.06 \cdot 10^{-5}$	$7.02 \cdot 10^{-5}$
19	$8.58 \cdot 10^{-5}$	$1.26 \cdot 10^{-4}$
20	$6.32 \cdot 10^{-5}$	$9.07 \cdot 10^{-5}$
21	$4.66 \cdot 10^{-5}$	$1.54 \cdot 10^{-4}$
22	$1.15 \cdot 10^{-4}$	$1.92 \cdot 10^{-4}$
23	$7.43 \cdot 10^{-5}$	$1.02 \cdot 10^{-4}$
24	$8.98 \cdot 10^{-5}$	$1.04 \cdot 10^{-4}$
25	$7.41 \cdot 10^{-5}$	$1.62 \cdot 10^{-4}$
26	$9.23 \cdot 10^{-5}$	$6.82 \cdot 10^{-5}$
27	$1.83 \cdot 10^{-4}$	$2.09 \cdot 10^{-4}$
28	$1.67 \cdot 10^{-4}$	$2.18 \cdot 10^{-4}$
29	$2.51 \cdot 10^{-5}$	$6.34 \cdot 10^{-5}$
30	$5.35 \cdot 10^{-4}$	$7.05 \cdot 10^{-4}$
31	$9.69 \cdot 10^{-5}$	$1.63 \cdot 10^{-4}$
32	$4.37 \cdot 10^{-4}$	$4.07 \cdot 10^{-4}$
33	$1.11 \cdot 10^{-4}$	$1.31 \cdot 10^{-4}$
34	$1.46 \cdot 10^{-4}$	$5.97 \cdot 10^{-5}$
35	$2.63 \cdot 10^{-4}$	$3.82 \cdot 10^{-4}$
36	$1.03 \cdot 10^{-4}$	$1.26 \cdot 10^{-4}$
37	$8.94 \cdot 10^{-5}$	$7.77 \cdot 10^{-5}$
38	$1.43 \cdot 10^{-4}$	$1.62 \cdot 10^{-4}$
39	$8.76 \cdot 10^{-5}$	$1.44 \cdot 10^{-4}$
40	$4.12 \cdot 10^{-4}$	$3.95 \cdot 10^{-4}$
41	$8.96 \cdot 10^{-5}$	$8.71 \cdot 10^{-5}$
42	$6.27 \cdot 10^{-5}$	$1.12 \cdot 10^{-4}$
43	$1.39 \cdot 10^{-4}$	$1.30 \cdot 10^{-4}$
44	$3.01 \cdot 10^{-4}$	$3.28 \cdot 10^{-4}$
45	$7.60 \cdot 10^{-4}$	$5.46 \cdot 10^{-4}$
46	$4.48 \cdot 10^{-4}$	$5.75 \cdot 10^{-4}$
47	$2.37 \cdot 10^{-4}$	$2.67 \cdot 10^{-4}$
48	$1.90 \cdot 10^{-4}$	$2.36 \cdot 10^{-4}$
49	$1.25 \cdot 10^{-4}$	$1.53 \cdot 10^{-4}$
50	$9.56 \cdot 10^{-5}$	$1.25 \cdot 10^{-4}$
51	$1.66 \cdot 10^{-4}$	$1.39 \cdot 10^{-4}$
52	$7.42 \cdot 10^{-5}$	$2.17 \cdot 10^{-4}$
53	$1.74 \cdot 10^{-4}$	$1.89 \cdot 10^{-4}$
54	$6.72 \cdot 10^{-5}$	$7.85 \cdot 10^{-5}$
55	$1.64 \cdot 10^{-5}$	$1.01 \cdot 10^{-4}$
56	$8.76 \cdot 10^{-5}$	$1.52 \cdot 10^{-4}$
57	$3.04 \cdot 10^{-4}$	$2.55 \cdot 10^{-4}$
58	$1.72 \cdot 10^{-4}$	$1.56 \cdot 10^{-4}$
59	$7.95 \cdot 10^{-5}$	$8.77 \cdot 10^{-5}$
60	$3.35 \cdot 10^{-5}$	$4.71 \cdot 10^{-5}$
61	$6.35 \cdot 10^{-5}$	$5.29 \cdot 10^{-5}$
62	$4.60 \cdot 10^{-5}$	$5.55 \cdot 10^{-5}$
63	$5.45 \cdot 10^{-5}$	$1.03 \cdot 10^{-4}$
64	$1.08 \cdot 10^{-4}$	$2.01 \cdot 10^{-4}$

Since the presentation of the results for all 64 wells may not be very practical and insightful, the results were organized according to the testing error performance in five categories and results will be presented for one representative well from each category. The five categories and the wells in each category are distinguished by color Table 5-4. The training and testing errors as well as the Nash-Sutcliffe Efficiency coefficient for the representative well of each category are presented in Table 5-5.

Table 5-5: Statistical characteristics for the representative wells

Well #	Training RMSE (m)	Testing RMSE (m)	Nash–Sutcliffe Efficiency
60	$3.35 \cdot 10^{-5}$	$4.71 \cdot 10^{-5}$	0.59
55	$1.64 \cdot 10^{-5}$	$1.01 \cdot 10^{-4}$	0.58
58	$1.72 \cdot 10^{-4}$	$1.56 \cdot 10^{-4}$	0.52
64	$1.08 \cdot 10^{-4}$	$2.01 \cdot 10^{-4}$	0.48
44	$3.01 \cdot 10^{-4}$	$3.28 \cdot 10^{-4}$	0.61

The testing error follows the training error closely in most cases, which means that the ANNs are not overtrained and that they have good overall generalization ability.

In theory, one would expect for the NSE coefficient to have values close to 1, which indicates a perfect fit between observed and simulated values. However, a model intended to simulate the fluctuations around a relatively constant mean value can only achieve high NSE values if it explains small time-scale fluctuations very well [*Schaefli and Gupta*, 2007]. Since the available data exhibit a constant fluctuation around zero, it is not possible to achieve high values for NSE. The derived NSE values can be characterized as satisfactory according to the classification of *Moriasi et al.* [2007]. It is also worth pointing out that in most cases the NSE decreases as the testing error increases, as expected, except for well No. 44. This can be attributed to the statistical characteristics of the hydraulic head change time series at well No. 44. More specifically, the small hydraulic head changes and the low mean value, compared to the rest of the wells, resulted in a smaller fraction in the second part of the NSE equation and hence in larger NSE value. Moreover, this dataset did not contain any outliers, as opposed to the rest of the time series examined. From previous studies, outliers are known to result in small NSE values [*Criss and Winston*, 2008; *Legates and McCabe*, 1999].

The observed and simulated hydraulic head change values versus time (one day time step) for the five representative wells are shown in Figure 5-3. The simulated versus observed values for the training and testing datasets are also presented in Figure 5-4.

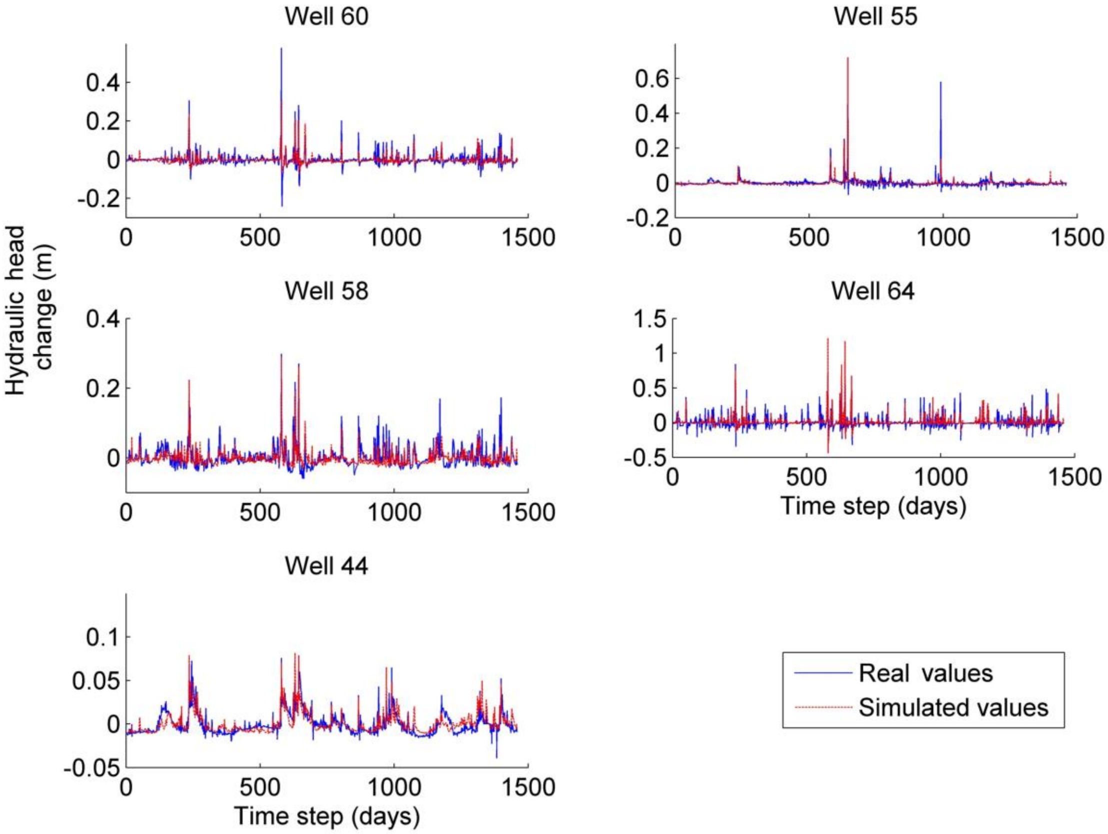


Figure 5-3: Observed and simulated hydraulic head change versus time for the five representative wells

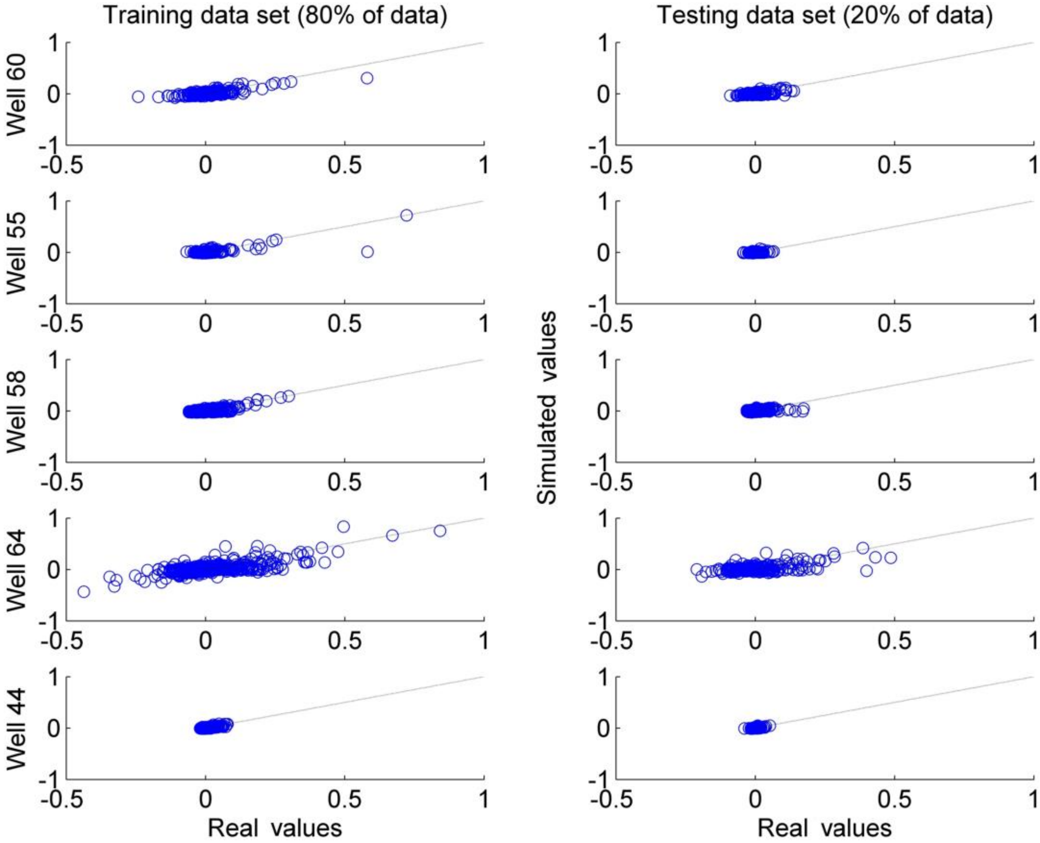


Figure 5-4: Simulated versus observed hydraulic head change values for the training and testing datasets

In both cases and for all five representative wells, the simulated values follow the observed values closely. For the five representative case, the average difference between observed and simulated values varies from $6.24 \cdot 10^{-5}$ to $4.13 \cdot 10^{-4}$ m. However, in some cases some outlier points tend to be underestimated.

5.1.2. Kriging results

The next step in the methodology is the implementation of the kriging algorithm. The use (or not) of the fuzzy logic system must be studied first. In both cases, three variogram models were tested; the linear, the exponential and the power-law. As described in Chapter 3, for each prediction point and for every time step, a new experimental variogram is fitted to the theoretical model. A typical fitting of experimental data to different theoretical variogram models is presented in Figure 5-5.

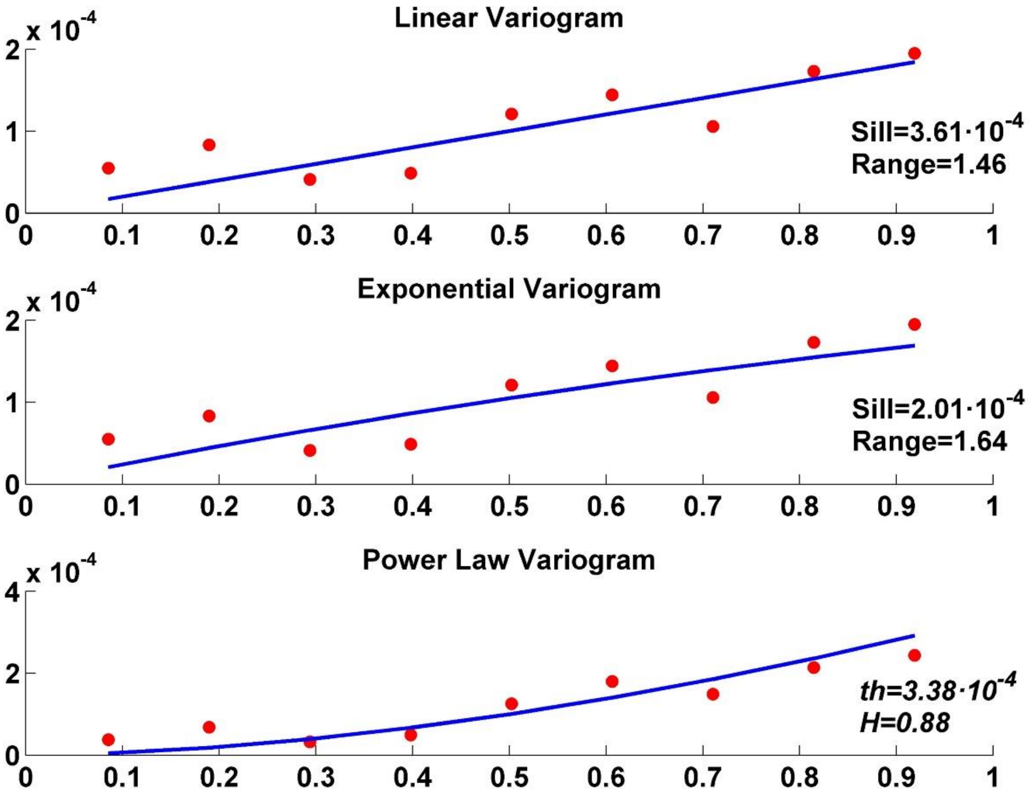


Figure 5-5: A typical case of experimental data fitting to three different theoretical variogram models

The data points for which complete hydraulic head data were not available were used for the verification of the model. The results for these two data points are presented in Figure 5-6

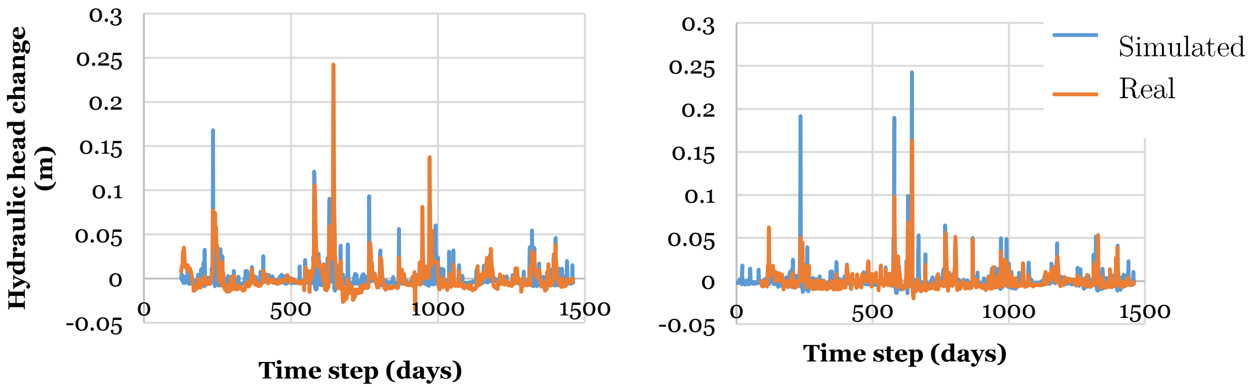
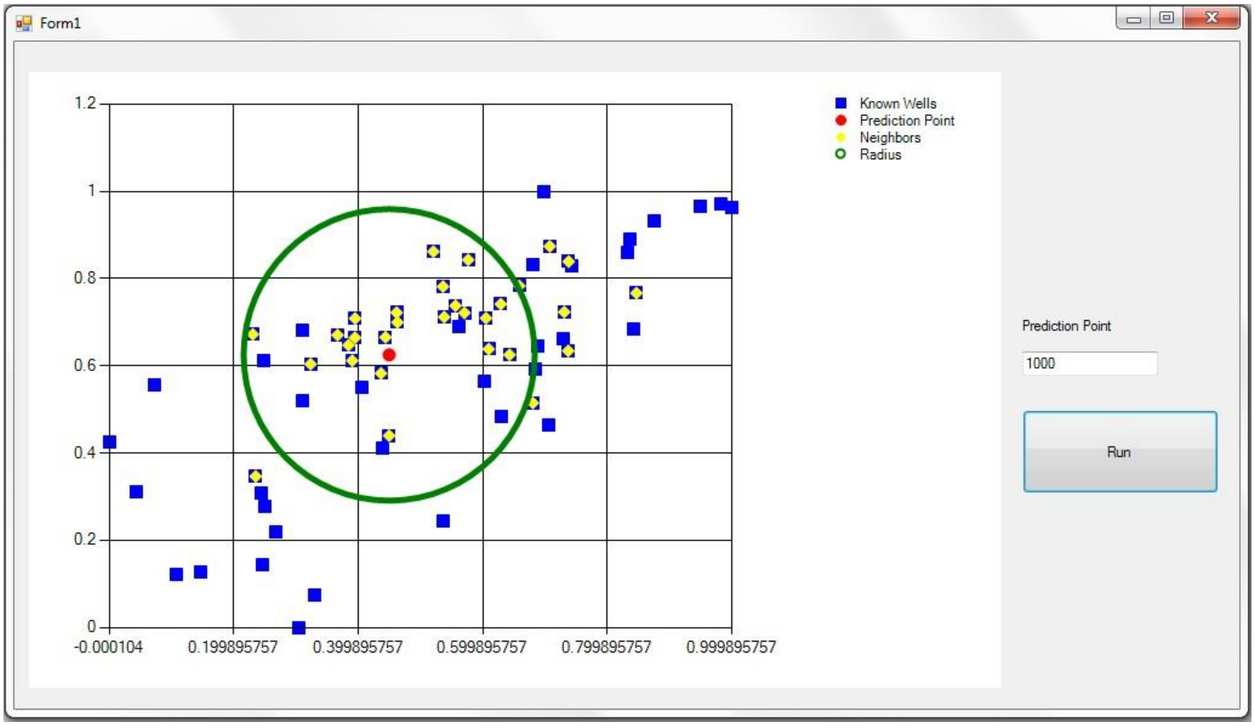


Figure 5-6: Observed and simulated hydraulic head change values for locations with incomplete data

In the first case, the average difference between observed and simulated values was equal to $1.04 \cdot 10^{-4}$ m. The minimum difference between observed and simulated values was equal to $1.66 \cdot 10^{-6}$ m and the maximum equal to 0.206 m, with this order of error magnitude occurring only once. The average error variance was equal to $4.71 \cdot 10^{-4}$ m. In the second case, the average difference between observed and simulated values was equal to $5.62 \cdot 10^{-5}$ m, the minimum difference equal to $1.41 \cdot 10^{-7}$ m and the maximum equal to 0.14 m. The average error variance in this case was $2.58 \cdot 10^{-4}$ m.

In order to evaluate the appropriateness of the fuzzy logic system, another tool was developed that illustrates, for a given prediction point, the neighborhood defined by the fuzzy logic system and the neighborhood defined when the fuzzy logic system is disabled. The tool makes use of the information contained in output file neighbor.txt. The two neighborhoods for the 1000th prediction point are illustrated in Figure 5-7.



Prediction Point
1000

Figure 5-7: Neighborhood for the 1000th prediction point, with and without the use of the fuzzy logic system

5.1.3. Cross-validation results

Leave-one-out cross-validation was performed in all cases, by assuming that an observation point was a prediction point for the 10% of the time steps. When the fuzzy logic system was used, the data from 30 neighbors were used to compute the value of hydraulic head change at a prediction point with kriging interpolation. When the fuzzy logic system was not used, the neighbors were determined by the variogram range. Three error indicators, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Bias, were calculated in order to determine the most appropriate variogram model.

Without fuzzy logic inference

The values of the error indicators for the case without fuzzy logic inference are summarized in Table 5-6.

Table 5-6: Cross-validation error for three variogram models without the use of fuzzy logic

Variogram type	RMSE (m)	MAE (m)	Bias (m)
Linear	$5.38 \cdot 10^{-2}$	$2.01 \cdot 10^{-2}$	$-1 \cdot 10^{-3}$
Exponential	$3.00 \cdot 10^{-1}$	$3.50 \cdot 10^{-2}$	$-8.44 \cdot 10^{-3}$
Power-law	$2.24 \cdot 10^{-2}$	$1.30 \cdot 10^{-2}$	$-8.85 \cdot 10^{-4}$

According to all three error indicators, the power-law variogram is the most appropriate for the data at hand. The linear variogram was the second best and the exponential was last.

With Fuzzy logic inference

The values of the error indicators for the case with fuzzy logic inference are presented in Table 5-7.

Table 5-7: Cross-validation error for three variogram models with the use of fuzzy logic

Variogram type	RMSE (m)	MAE (m)	Bias (m)
Linear	$2.18 \cdot 10^{-2}$	$1.41 \cdot 10^{-2}$	$1.64 \cdot 10^{-4}$
Exponential	$5.57 \cdot 10^{-2}$	$1.94 \cdot 10^{-2}$	$-2.46 \cdot 10^{-3}$
Power-law	$1.95 \cdot 10^{-2}$	$1.33 \cdot 10^{-2}$	$-2.05 \cdot 10^{-5}$

The power-law variogram is again the most appropriate variogram, as was in the case without fuzzy logic inference.

The cross-validation error variance for the three variogram models is presented in Table 5-8.

Table 5-8: Cross-validation error variance for different variogram models

Variogram type	Error variance		
	Minimum	Maximum	Average
Linear	$3.37 \cdot 10^{-7}$	$8.13 \cdot 10^{-2}$	$2.41 \cdot 10^{-4}$
Exponential	$4.37 \cdot 10^{-7}$	$8.53 \cdot 10^{-2}$	$3.70 \cdot 10^{-4}$
Power-law	$2.26 \cdot 10^{-7}$	$4.12 \cdot 10^{-2}$	$1.90 \cdot 10^{-4}$

The power-law variogram with the fuzzy logic system yielded the best result for all the cases studied. In similar studies, non-differentiable variograms such as the power-law performed better than differentiable variograms [*Varouchakis and Hristopulos*, 2013a]. This can be attributed to the water deposition-removal process that takes place in the aquifer. This process can be characterized as a fractional Brownian motion-like random field which has a power-law variogram function. Therefore, the power-law variogram was able to simulate better the aquifers motion.

Results for Kriging interpolated versus ANN simulated cross-validation values are presented in Figure 5-8 for two cases of the ANN simulation model, with and without kriging interpolation. It should be noted that in the case of kriging the results summarize two different errors, one introduced by ANN simulation and one by kriging interpolation. This is also the reason why these results are slightly worse than the results of ANN simulation alone.

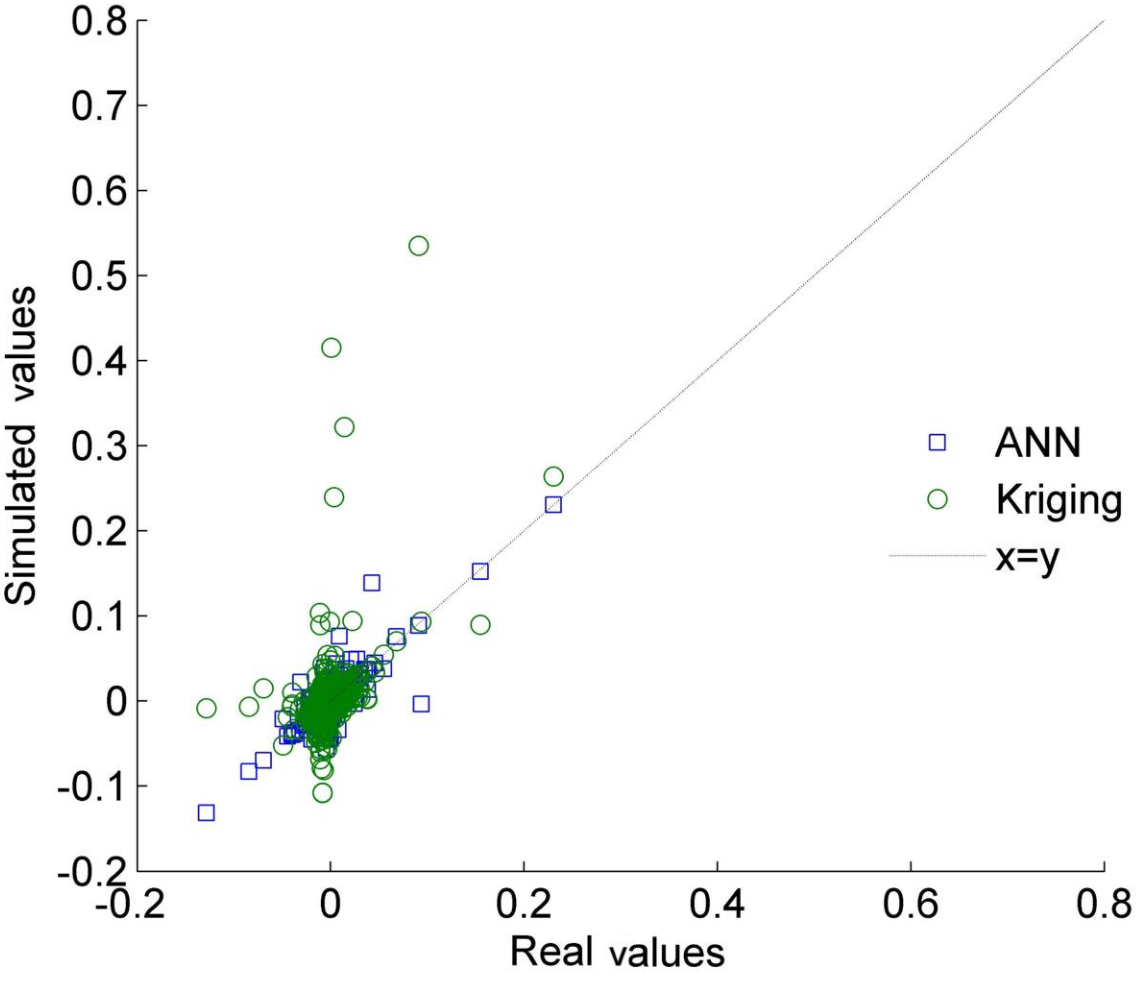


Figure 5-8: Kriging interpolated versus ANN simulated cross-validation values

The final spatial output of the model for five different time steps (1, 365, 730, 1095, and 1460) is presented in Figure 5-9. Three graphs are presented for every time step corresponding to hydraulic head (top), hydraulic head change (middle) and error variance (bottom). For the conversion of the hydraulic head change to hydraulic head, the simulated hydraulic head change per day was added to the initial hydraulic head value observed in the field on the first day of simulation. The initial hydraulic head for the prediction points was determined by kriging from the known initial values at data points.

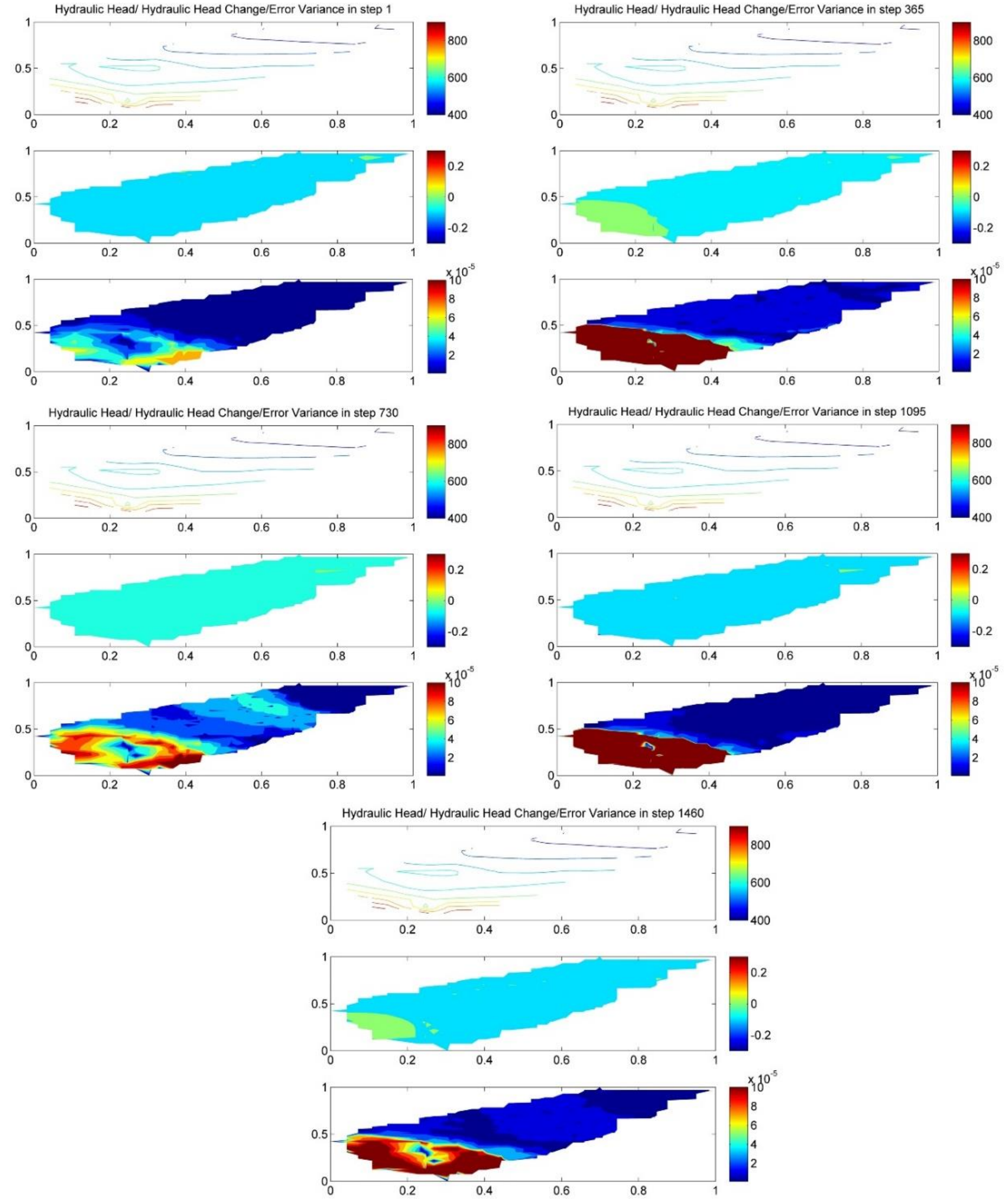


Figure 5-9: Hydraulic head, hydraulic head change and error variance for five different time steps

It should be noted that the higher values for error variance in the southern part of the study area are due to the lack of observation points in this area. Due to the high initial hydraulic head values, varying from 900 m in the southern part of the study area to 400 m in the northeastern part, it is not possible to observe the variability of the hydraulic head between the different time steps. This simulation can be characterized as successful, since it was possible to depict the spatial and temporal variations of hydraulic head and hydraulic head change having at the same time small cross-validation error and error variance [*Tapoglou et al.*, 2014].

5.1.4. Uncertainty results

In In order to assess the performance of the developed algorithm, uncertainty analysis was performed using the procedures described in Section 3.6. The first part of this section concerns uncertainty in ANN training and its effect on the model results, while the second part concerns uncertainty in the variogram parameters.

ANN prediction confidence intervals

The 90% confidence intervals for the hydraulic head change were obtained as follows. The first step in this methodology is to calculate, the difference between the real and the simulated output values for every time step. These values represent the range of the absolute error in the simulation results. After sorting the error values the 5th and 95th percentile values are identified, representing the upper and lower 90% CI limits of the absolute error. By adding these values to all simulation results for the specific ANN, the 5% upper and lower limit of the timeseries prediction is calculated. The 90% confidence interval of the predictions together with the simulated and real values of the hydraulic head for the five representative wells are presented in Figure 5-10.

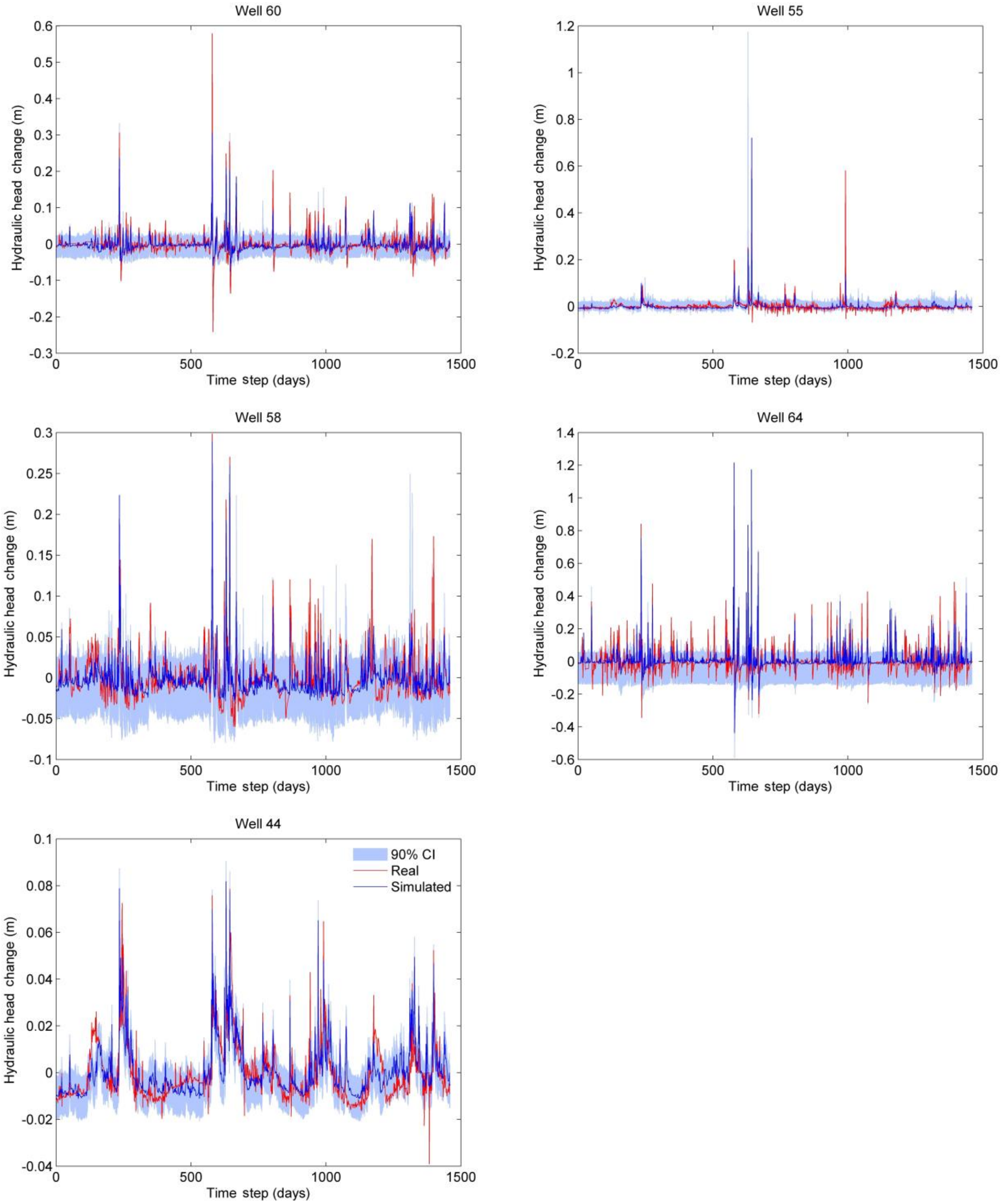


Figure 5-10: 90% Confidence intervals for hydraulic head change ANN simulation at representative wells

The width of the confidence intervals for the five representative wells is presented in Table 5-9.

Table 5-9: Percentile methodology confidence intervals for five representative wells

Well number	Confidence interval width (m)
60	0.049
55	0.031
58	0.052
64	0.137
44	0.023

Uncertainty in hydraulic head change due to ANN training

To examine the uncertainty in ANN training and its effect on the simulation results, 150 simulation runs were performed using different training and testing datasets as well as different initial random values for the neural network weights. The range of training and testing errors derived from the 150 simulation runs for each of the 64 ANNs (wells) is presented in Figure 5-11.

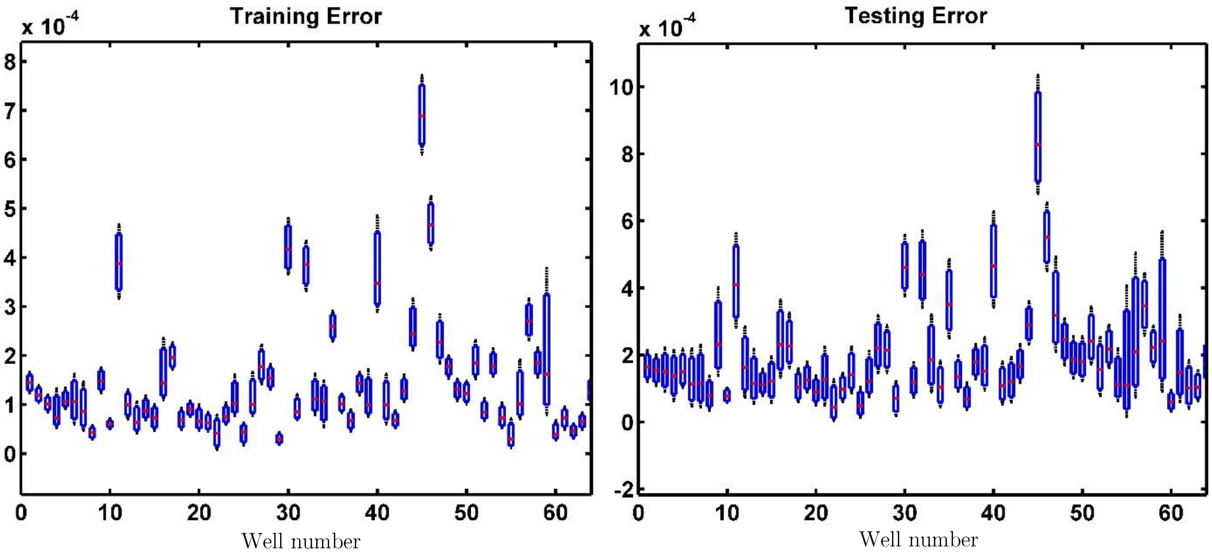


Figure 5-11: Training and testing error range for uncertainty analysis of training in all 64 wells

The range of the training error is small in most cases; however, there are cases where the range is relatively large, e.g., wells no. 40, 59. The testing error range is wider than the training error range for all wells.

The uncertainty in results for the hydraulic head change for the five representative wells are demonstrated in Figure 5-12.

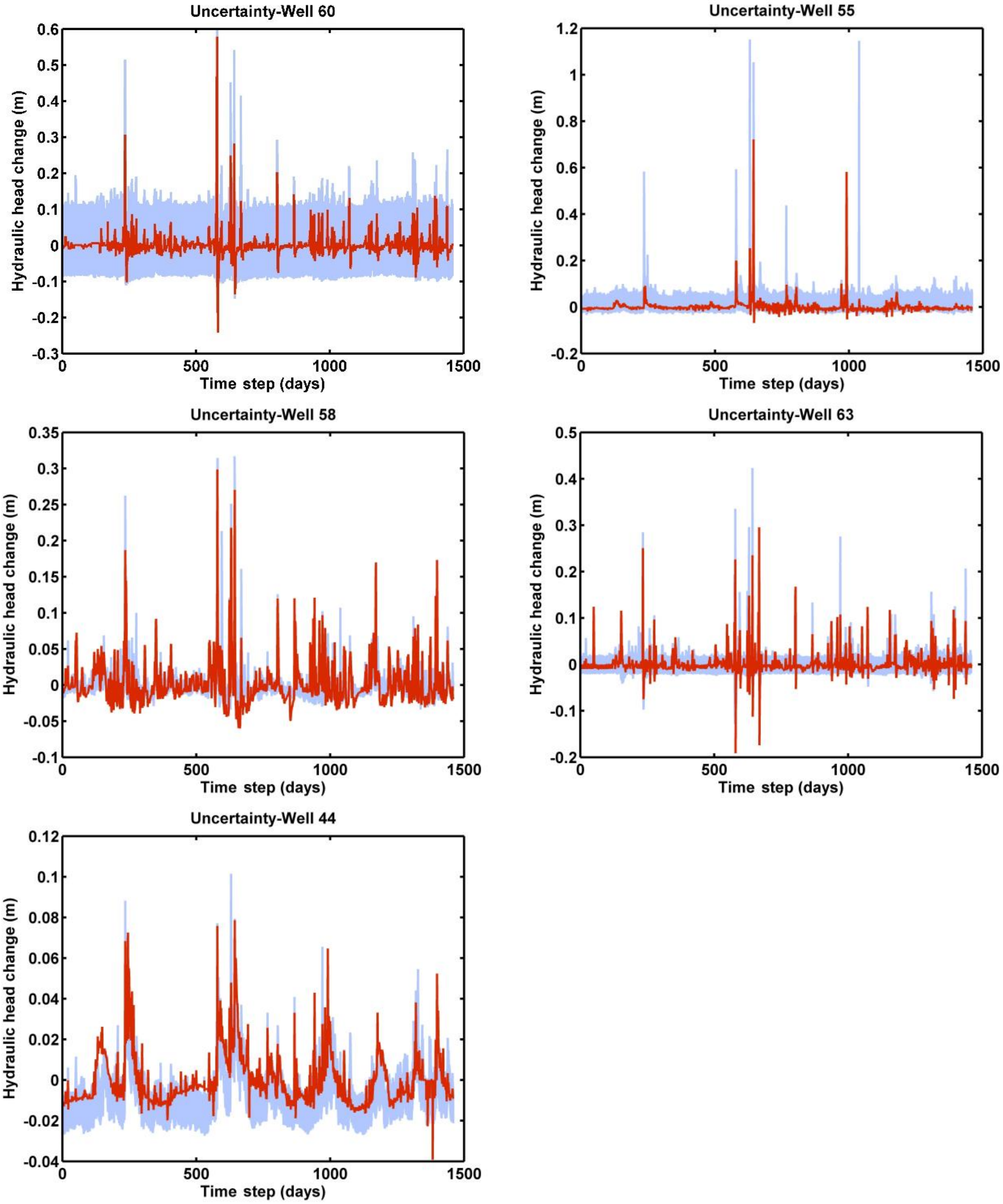


Figure 5-12: Uncertainty results for the five representative wells (the observed values are marked in red and the 90% prediction intervals in blue)

As can be observed from the results in Figure 5-12 , in most cases, the observed values fall within the 90% prediction interval for the hydraulic head change obtain with the ANN simulation. As the training error increases, the percentage of observed values that fall outside the 90% prediction interval grows. This percentage for the five wells representing different training error categories is reported in Table 5-10.

Table 5-10: Percentage of observed values outside the 90% prediction interval for representative wells

Well number	percentage of observed values outside 90% prediction interval
60	0.8
55	5
58	7.3
64	9.1
44	17

The results from the 150 simulation runs performed to test ANN training were used with kriging interpolation to determine the overall model uncertainty for the power-law variogram in the case where fuzzy logic is used. The kriging part of the algorithm was executed 150 times, with randomized initial values each time, in order to derive the overall model uncertainty. Results for the 90% prediction intervals for five prediction points are presented in Figure 5-13.

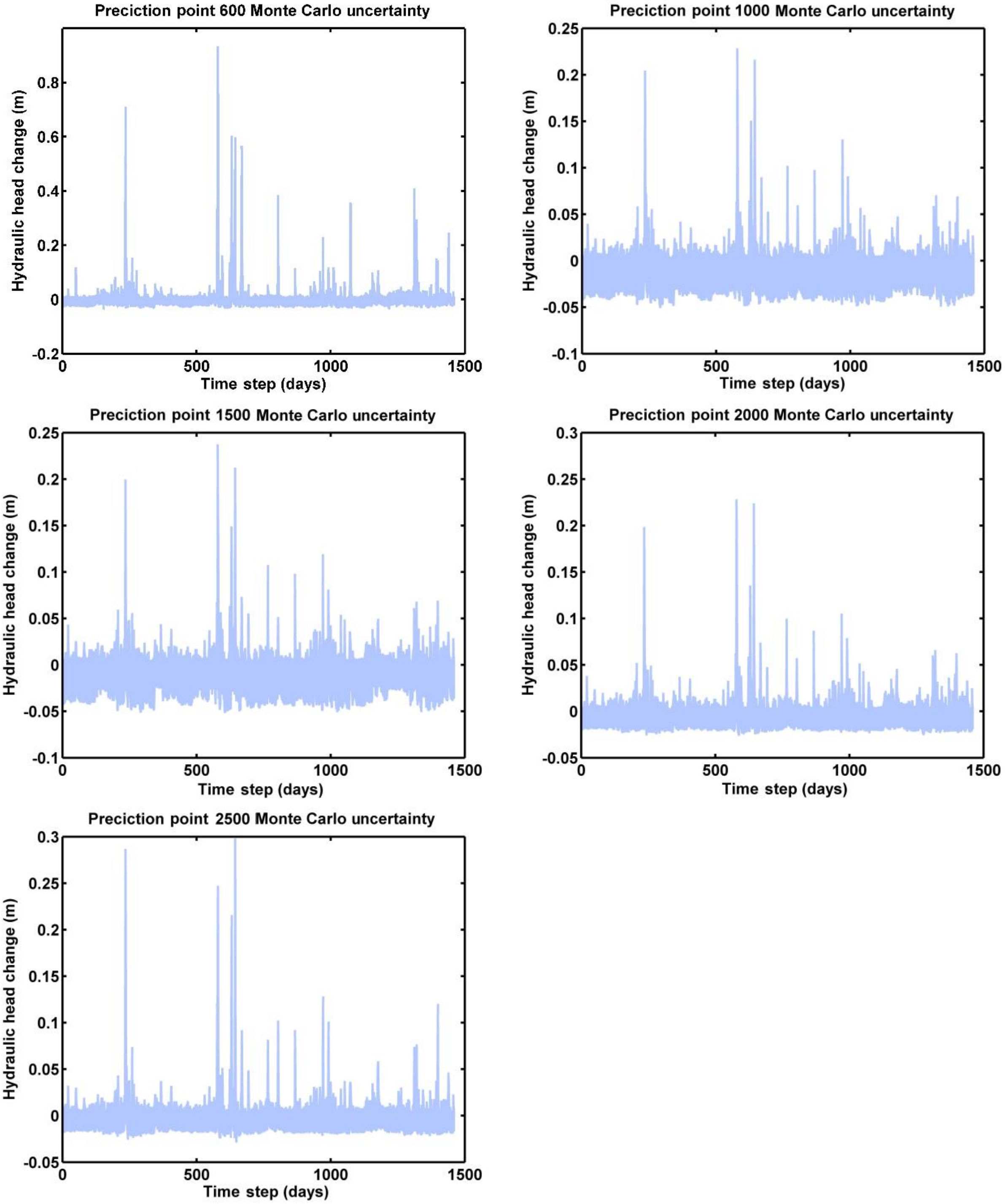


Figure 5-13: Hydraulic head uncertainty after Kriging interpolation at five prediction points

The average prediction interval width for prediction points 500, 1000, 1500, 2000 and 2500 is presented in Table 5-11.

Table 5-11: Average prediction width for five different prediction points

Prediction point	PI width (m)
500	0.05
1000	0.083
1500	0.077
2000	0.056
2500	0.062

Uncertainty in hydraulic head change due to kriging parameters

The effect of kriging parameter uncertainty was also studied following the procedure described in Section 3.6. Five hundred (500) different data samples and, hence, 500 different variograms were constructed for every prediction point and time step. Uncertainty results were derived for the linear variogram model and with use of fuzzy logic. Even though the linear model was second best choice to the power-law model, the latter was not used due to the fact that it does not have a fixed covariance function. The 90% prediction intervals together with the predicted value using the initial ANN simulation results for five prediction points are presented Figure 5-14.

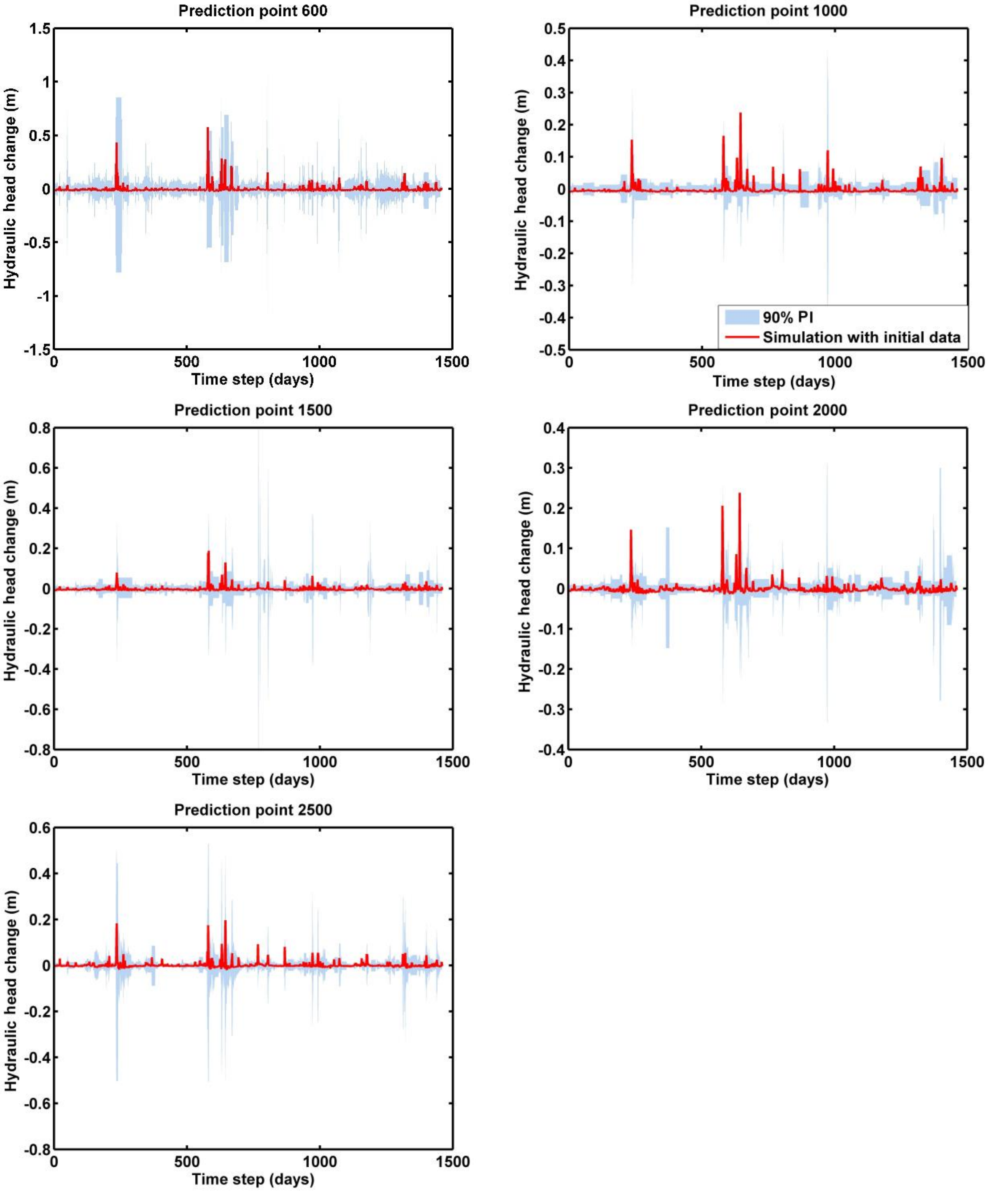


Figure 5-14: Uncertainty in hydraulic head change due to uncertainty in kriging parameters for five prediction points using Bayesian kriging

The use of the developed algorithm can be characterized as successful, since both cross-validation error had small values compared to the nature of the results and the uncertainty of the results was within reasonable range. In Table 5-12 the average range for every prediction point is presented.

Table 5-12: Prediction intervals for hydraulic head change due to uncertainty in kriging parameters

Prediction point	Average PI width (m)
600	0.14
1000	0.04
1500	0.012
2000	0.037
2500	0.012

5.2 Second Case Study – Miami-Dade County

The purpose of the developed algorithm's implementation in a second study area had as a purpose to verify the applicability of the model. The second study area was in a shallow aquifer in Miami-Dade County, Florida, USA. The geology of this study area is also complicated, as well as the land use, making the simulation of the aquifer challenging.

5.2.1. ANN results

The first step of the methodology is to determine the appropriate time lags for meteorological and surface water input parameters. In the case of precipitation, the most appropriate time lags varied from 0 to 3 days with the most common being 0 and 1 day. In the case of surface water, only 0 and 1 day lags were used both for surface water level and discharge. The correlation coefficient for the data time series used varied from 0.3 to 0.7. For the rest of the input parameters, soil and air temperature and humidity, time lag is not apparent and the value of the simulation day was used.

Data were available for thirty (30) wells in the study area. Results will be presented for three wells (no. 11, 3, 18), each representative of a category of wells with similar behavior. For the three representative wells, the time lags for the precipitation data, as well as the corresponding correlation coefficients are summarized in Table 5-13.

Table 5-13: Time lag and corresponding correlation coefficient values for precipitation for three representative wells

Well number	Station 420		Station 440	
	Time lag	Correlation	Time lag	Correlation
11	0	0.46	0,1	0.36,0.4
3	1	0.37	0,1	0.39,0.33
18	1,2	0.32,0.33	1	0.3

The most appropriate time lags for surface water level were 0 and 1 days from station USGS 02289500 for all wells, while for discharge 1 day time lag from station USGS 022907647 was the most appropriate for all wells. The correlation coefficient varied from 0.2, in the worst case, to 0.8 in limited best cases.

Recall that two ANN architectures were considered in this case study; Architecture 1 with one hidden layer of 32 nodes and Architecture 2 with two hidden layers, the first with 17, the second with 12 nodes (Section 4.2.3). The average, maximum and minimum RMSE values for the ANN training and testing errors are summarized in Table 5-14.

Table 5-14: Training and testing RMSE for the two ANN architectures under consideration

RMSE (m)	Architecture 1	Architecture 2
Average Training Error	$1.16\cdot10^{-3}$	$6.63\cdot10^{-4}$
Average Testing Error	$1.20\cdot10^{-3}$	$7.92\cdot10^{-4}$
Maximum Training Error	$2.19\cdot10^{-3}$	$1.78\cdot10^{-3}$
Maximum Testing Error	$3.42\cdot10^{-3}$	$2.98\cdot10^{-3}$
Minimum Training Error	$2.46\cdot10^{-4}$	$2.23\cdot10^{-4}$
Minimum Testing Error	$2.63\cdot10^{-4}$	$1.26\cdot10^{-4}$

The training and testing RMSEs are smaller for Architecture 2, not only in terms of average, maximum and minimum values, but also when comparing the results for each well individually. Thus, Architecture 2 with the two hidden layers will be used henceforth. The average value of the Nash-Sutcliffe efficiency coefficient is equal to 0.62, while the maximum value is equal to 0.71 and the minimum to 0.56. These values are considerably higher than the corresponding values derived for the first case study in Bavaria, due to the fact that, in this case, it is the hydraulic head that is simulated and not the hydraulic head change. However, the fact that it is impossible to achieve high NSE values when the simulated parameter takes small values still applies in this case, hence, it is not possible to achieve as high values as would have been expected.

The ANN training and testing RMSE errors for all wells are presented in Table 5-15. The wells are divided into three categories, depending on the training error. The wells are divided into three categories, according to training error. The three categories and the wells in each category are distinguished by color in Table 5-15.

Table 5-15: ANN training and testing error results (Architecture 2)

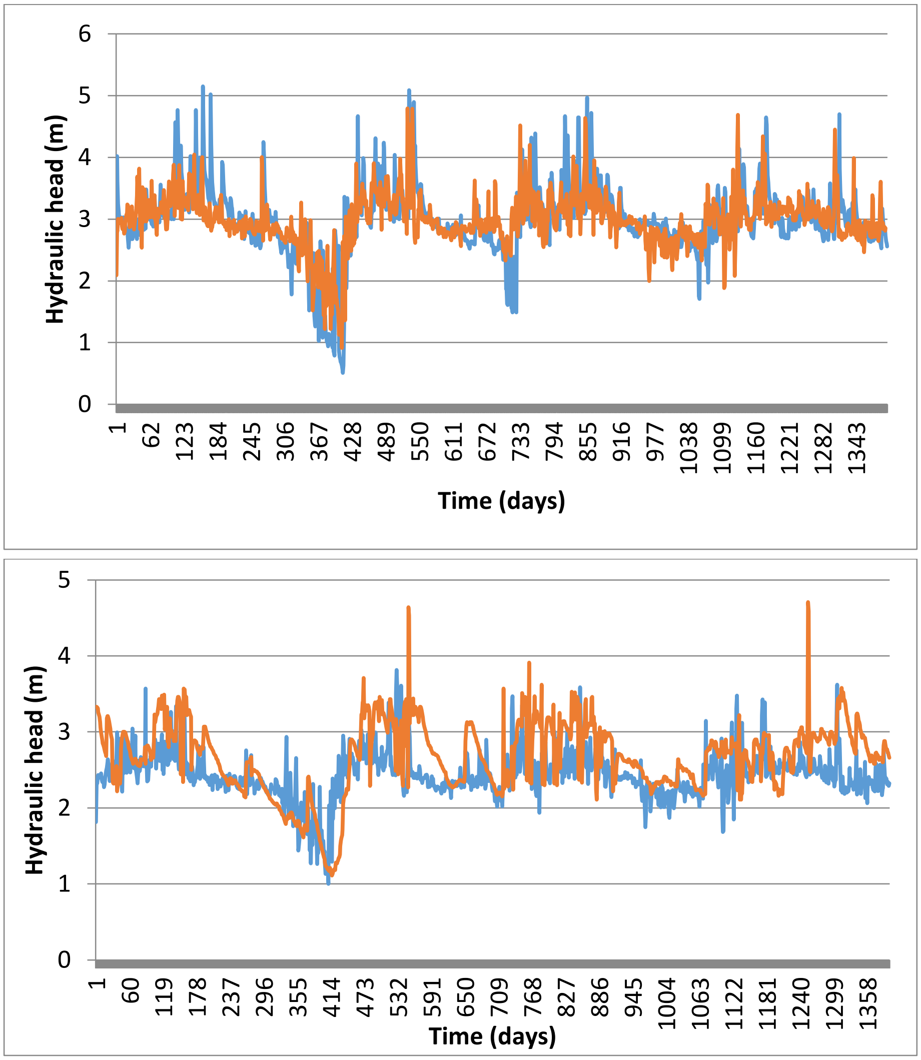
Well number	Training error (m)	Testing Error (m)
1	$6.70 \cdot 10^{-4}$	$4.34 \cdot 10^{-4}$
2	$1.08 \cdot 10^{-3}$	$1.14 \cdot 10^{-3}$
3	$5.68 \cdot 10^{-4}$	$4.19 \cdot 10^{-4}$
4	$3.19 \cdot 10^{-4}$	$3.70 \cdot 10^{-4}$
5	$2.74 \cdot 10^{-4}$	$1.26 \cdot 10^{-4}$
6	$7.53 \cdot 10^{-4}$	$2.30 \cdot 10^{-3}$
7	$5.69 \cdot 10^{-4}$	$6.64 \cdot 10^{-4}$
8	$5.41 \cdot 10^{-4}$	$5.29 \cdot 10^{-4}$
9	$2.23 \cdot 10^{-4}$	$3.32 \cdot 10^{-4}$
10	$8.98 \cdot 10^{-4}$	$1.30 \cdot 10^{-3}$
11	$2.77 \cdot 10^{-4}$	$3.13 \cdot 10^{-4}$
12	$6.78 \cdot 10^{-4}$	$5.96 \cdot 10^{-4}$
13	$3.28 \cdot 10^{-4}$	$4.14 \cdot 10^{-4}$
14	$6.98 \cdot 10^{-4}$	$5.60 \cdot 10^{-4}$
15	$1.03 \cdot 10^{-3}$	$8.20 \cdot 10^{-4}$
16	$1.78 \cdot 10^{-3}$	$2.98 \cdot 10^{-3}$
17	$3.15 \cdot 10^{-4}$	$4.55 \cdot 10^{-4}$
18	$1.11 \cdot 10^{-3}$	$6.23 \cdot 10^{-4}$
19	$3.92 \cdot 10^{-4}$	$6.29 \cdot 10^{-4}$
20	$1.31 \cdot 10^{-3}$	$1.80 \cdot 10^{-3}$
21	$4.71 \cdot 10^{-4}$	$4.28 \cdot 10^{-4}$
22	$2.91 \cdot 10^{-4}$	$3.70 \cdot 10^{-4}$
23	$5.08 \cdot 10^{-4}$	$5.63 \cdot 10^{-4}$
24	$5.29 \cdot 10^{-4}$	$6.27 \cdot 10^{-4}$
25	$1.09 \cdot 10^{-3}$	$1.41 \cdot 10^{-3}$
26	$7.41 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$
27	$2.31 \cdot 10^{-4}$	$3.60 \cdot 10^{-4}$
28	$1.23 \cdot 10^{-3}$	$9.39 \cdot 10^{-4}$
29	$5.48 \cdot 10^{-4}$	$5.54 \cdot 10^{-4}$
30	$4.27 \cdot 10^{-4}$	$6.32 \cdot 10^{-4}$

The training and testing errors as well as the Nash-Sutcliffe Efficiency coefficient for the three representative wells are presented in Table 5-16.

Table 5-16: Statistical characteristics for the representative wells of the second case study

Well number	Training Error (m)	Testing Error (m)	NSE
11	2.77E-04	3.13E-04	0.69
3	5.68E-04	4.19E-04	0.65
18	1.11E-03	6.23E-04	0.58

For the three representative wells, the observed and simulated hydraulic head values are plotted against time (daily time step) in Figure 5-15.



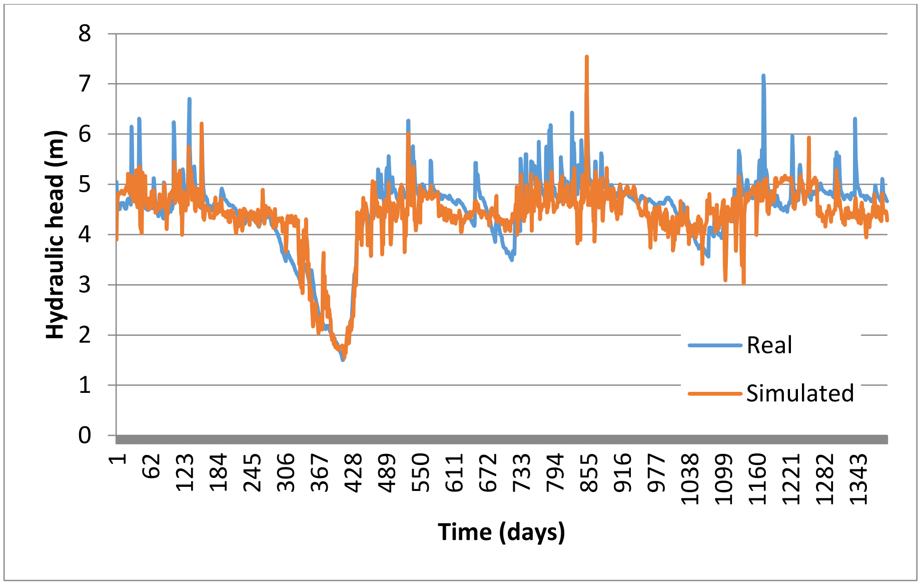
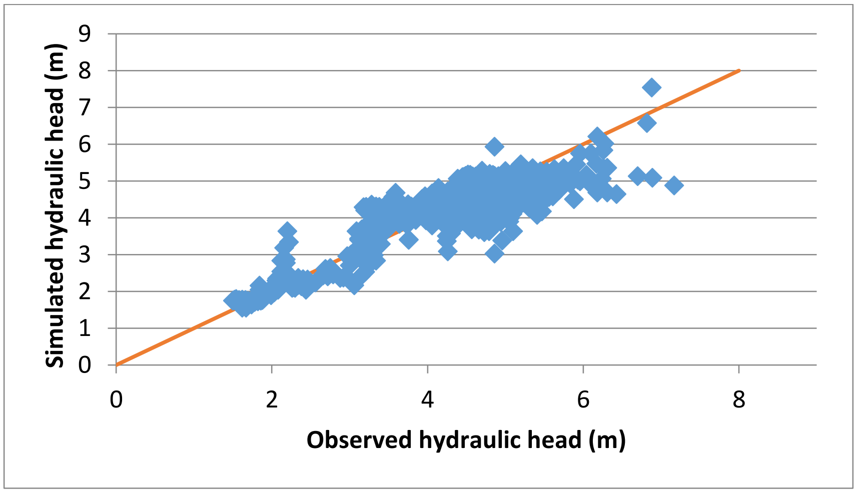


Figure 5-15: Observed and simulated hydraulic head versus time for the representative wells (from top to bottom: wells no.11, 3 and 18)

For all three well categories, the algorithm was capable of simulating adequately the real hydraulic head values observed in the field. However, there were some cases where extreme values could not be simulated accurately, and simulation results could only capture the general trend of the physical system. The simulated versus observed hydraulic head for the three representative wells is presented in Figure 5-16.



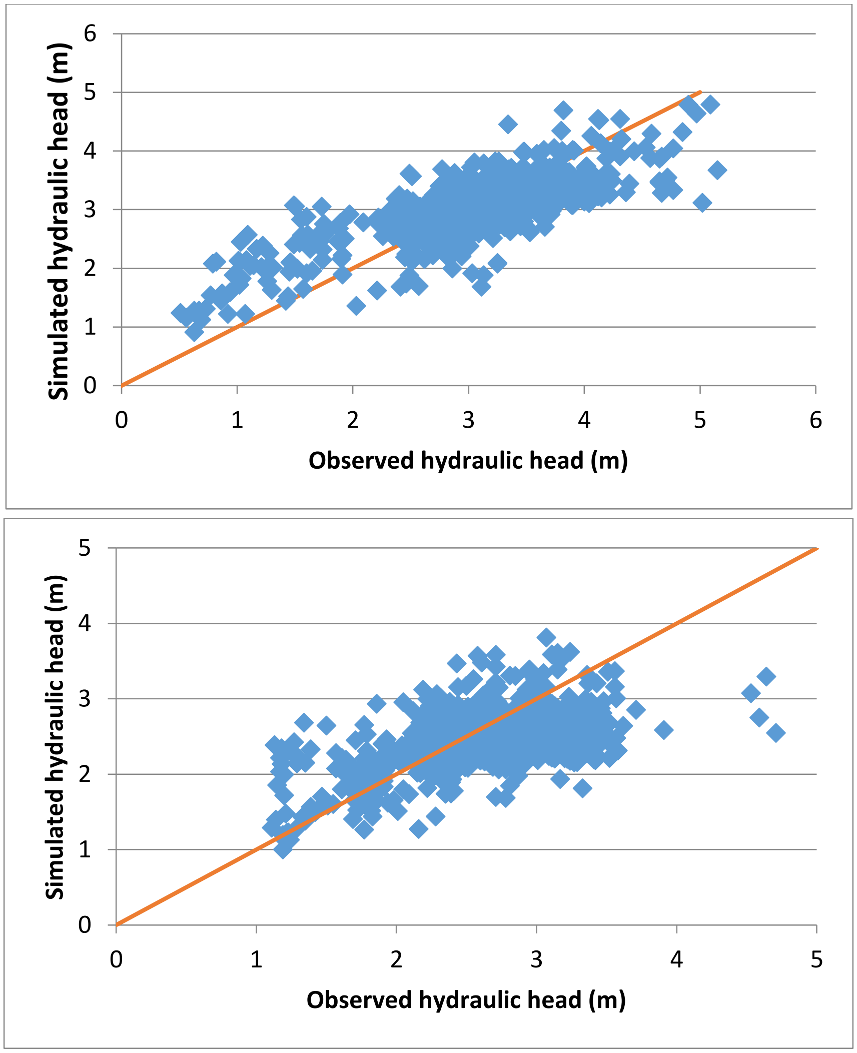


Figure 5-16: Simulated versus observed hydraulic head for three representative wells (from top to bottom: wells no. 11, 3 and 18)

As would be expected, there is a noticeable difference in performance between the ANN categories (Figure 5-16). For the first category, the results are very close to the y=x line which represents the perfect fit. For the second category, more values are further away from the y=x line, and for the last category, even more simulated values part from the observed, while it was not possible for four values to be adequately simulated, since they are positioned far for the y=x line. These values represent four outliers found in the observed data.

5.2.2. Kriging results

After training one ANN for each data point, the implementation of the kriging part of the algorithm takes place. As described in Chapter 3, a new experimental variogram is fitted to the theoretical model for each prediction point and for every time step. In this second study area, only the case with the use of fuzzy logic was considered and the size of the neighborhood was put to test. The initial neighborhood was set to 20 data points and for the variogram model with the best results this assumption was tested. The number of 20 neighbors was initially chosen so as to be able to construct a dependable variogram, which will follow the rule stated in Chapter 3 that at least 30 distance pairs must be apparent for every distance lag.

The variogram models fitted to the data at hand and then evaluated for their performance was the linear, the exponential and the power-law. The evaluation was performed through leave one out cross-validation, as described in Chapter 3, for the 10% of the data points and time steps.

5.2.3. Cross-validation results

The fuzzy logic system was used in this case study. The values of the error indicators for three different variogram models are presented in Table 5-17. A neighborhood size equal to 20 neighbors was considered in all variogram cases.

Table 5-17: Cross-validation error indicators

	RMSE (m)	MAE (m)	Bias (m)
Linear	1.14	0.89	-0.0485
Exponential (20)	0.962	0.7047	0.0499
Power	0.98	0.7422	0.0446

The values of the error indicators are some orders of magnitude larger than the values of the same indicators computed for the first case study. This is due to the fact that, in this case, the hydraulic head is simulated and not the hydraulic head change. Of the three variogram models, the exponential variogram had the best results for the error indicators and the linear the worst.

In order to study the effect of the neighborhood size on the simulation results, the algorithm was also executed for exponential variogram and for the cases of 15 and 25 neighbors in the kriging neighborhood. The results for the error indicators for the exponential variogram and different neighborhood sizes are presented in Table 5-18.

Table 5-18: Error indicators for different neighborhood sizes

Number of neighbors	RMSE (m)	MAE (m)	Bias (m)
15	1.2262	0.95	-0.079
20	0.962	0.7047	0.0499
25	1.15	0.86	-0.056

Values of neighborhood size smaller than 15 neighbors were not tested because the constraint of having at least 30 pairs per distance class was often violated at variogram construction. For neighborhood size greater than 25 neighbors, the advantage of using the fuzzy logic system was eliminated, so the case of 20 neighbors was adopted in the rest of the results computations.

Results for the average error variance are presented in Table 5-19 for the variogram models and kriging neighborhoods discussed above.

Table 5-19: Error variance for different variogram models and neighborhood sizes

Variogram type	Error variance
Exponential (15)	0.76
Exponential (20)	0.21
Exponential (25)	0.39
Linear	0.86
Power-law	0.97

The cross-validation results for ANN and Kriging versus observed data are presented in Figure 5-17.

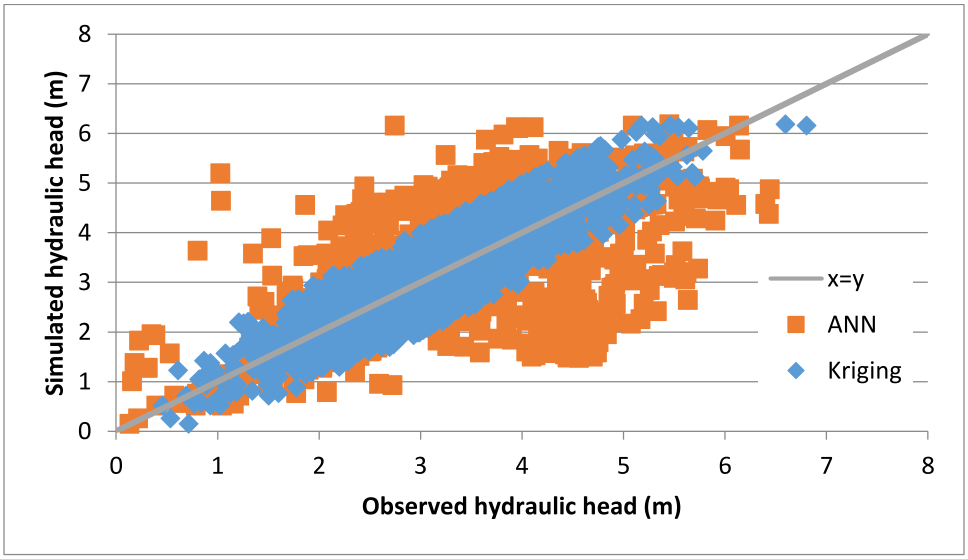


Figure 5-17: Cross validation Kriging and ANN results versus Observed data

In Figure 5-17 , note that the results for ANN simulation with kriging are closer to the y=x line than the ANN results alone, implying that it is possible for the kriging algorithm to improve the ANN results. The improvement of the ANN results is also reflected on the error indicators. More specifically, The RMSE, NSE and MAE values of the data used for the cross validation, both only using ANNs and ANNs together with kriging are presented in Table 5-20.

Table 5-20: Error indicators for cross-validation values

	ANN	ANN and kriging
NSE	0.58	0.8
MAE (m)	0.59	0.41
RMSE (m)	0.77	0.49

The simulation results for the hydraulic head in the study area are presented in Figure 5-18 for four different time steps: 1, 500, 1000, 1396 days.

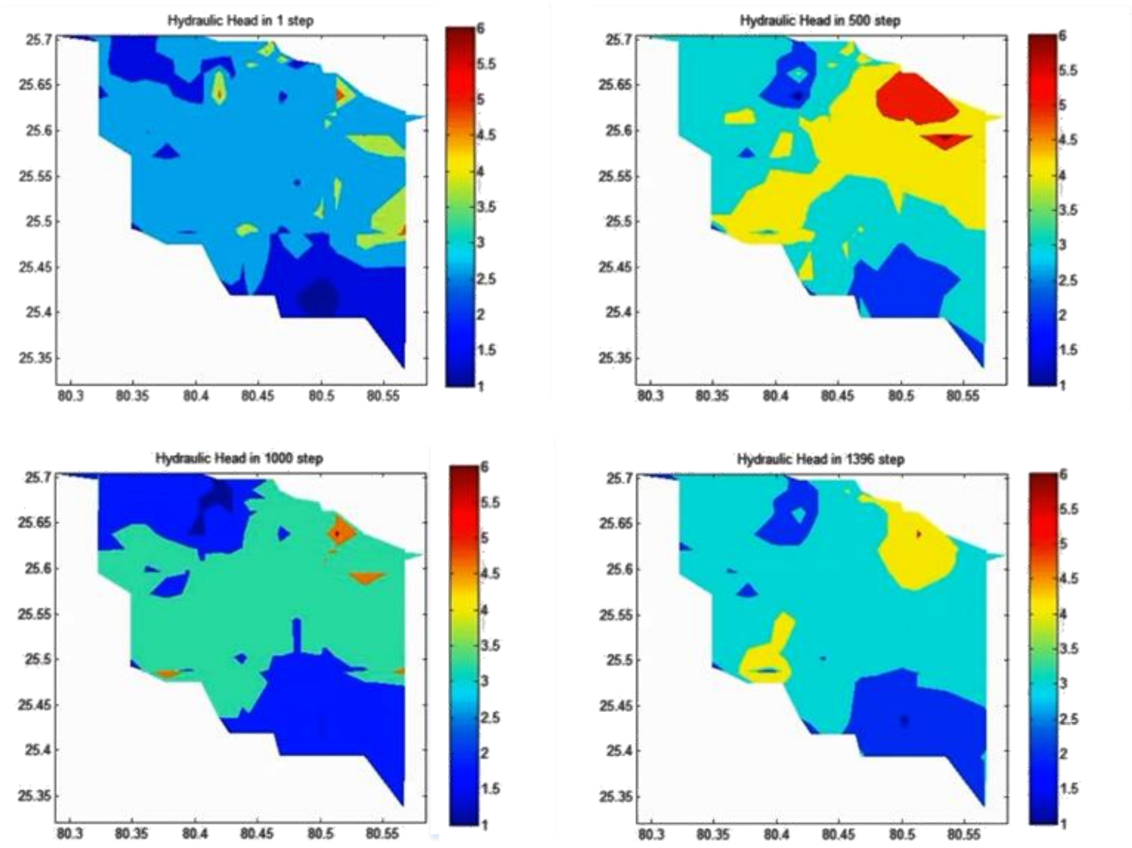


Figure 5-18: Simulated hydraulic head for time steps 1, 500, 1000, 1396

Characteristic statistical measures for the simulation results are summarized in Table 5-21.

Table 5-21: Statistical measures for the simulation results for the second study area

Category	Average	Maximum	Minimum
Training Error (m)	$6.63 \cdot 10^{-4}$	$1.7 \cdot 10^{-3}$	$2.23 \cdot 10^{-4}$
Testing Error (m)	$7.92 \cdot 10^{-4}$	$2.98 \cdot 10^{-3}$	$1.26 \cdot 10^{-4}$
Kriging Error Variance	0.2	1.6	$8.24 \cdot 10^{-6}$
Cross-Validation Error (RMSE)		0.962	

The simulation can be characterized as successful, especially taking into consideration the complexity of the natural system.

Temporal changes in the water-level in the Biscayne aquifer can be caused by numerous factors, such as the El Nino and the La Nina events, changes in the canal operation, the installation of new infrastructure, changes in water management practices and changes in land use. Since the aquifer is shallow, these parameters affect the hydraulic head heavily and are usually unpredictable making the simulation of the hydraulic head a difficult process.

5.2.4. Uncertainty results

ANN prediction confidence intervals

The 90% confidence intervals for the hydraulic head were obtained in a manner similar to the one described in Section 3.6.1 for the first case study. The results for the three representative wells are presented in Figure 5-19. The percentiles' values together with the confidence interval width are presented in Table 5-22.

Table 5-22: Lower and upper 5% percentiles for three characteristic wells

	Lower 5% (m)	Upper 5% (m)	PI width (m)
Well 11	-0.43	0.75	1.18
Well 3	-0.59	0.56	1.15
Well 18	-0.58	0.71	1.29

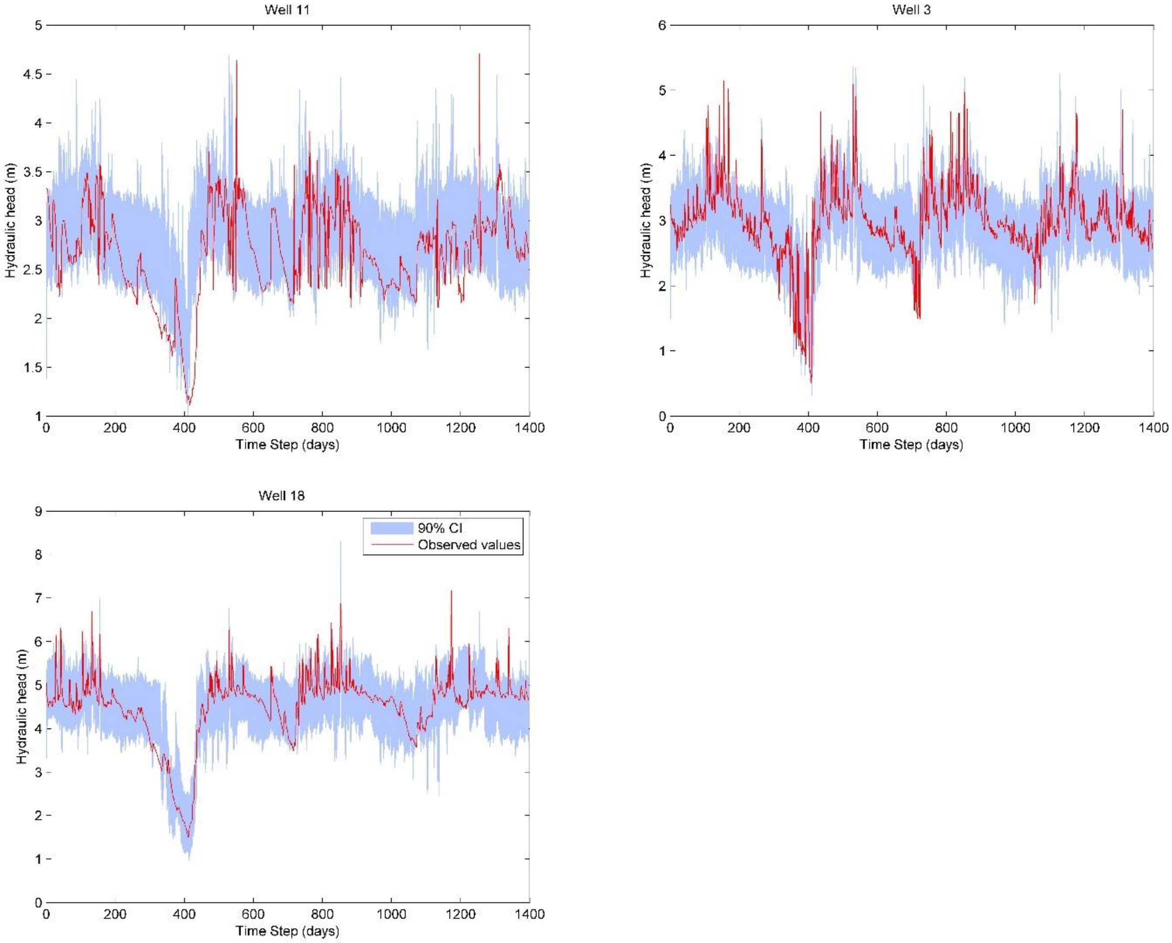


Figure 5-19: 90% confidence intervals for hydraulic head ANN simulation

While the 90% confidence intervals were constructed using the training dataset, this doesn't necessarily mean that the testing data will always fall within these intervals as well. In most cases, the observed values fall within the 90% confidence interval of the ANN prediction for hydraulic head, especially in the case of wells no. 3 and no. 18. There are a few exceptions in the case of well no. 11, and more specifically for the lowest observed values, which do not fall within the prediction. The coverage of these intervals with respect to the observed values was 82% for well 11, 75% for well 3 and 79% for well 18.

Uncertainty in hydraulic head change due to ANN training

In order to evaluate the uncertainty in ANN training, 300 simulation runs were performed for different training and testing datasets, as well as different initial random neural weights. The range of the training and testing errors for each of the thirty wells in the study area is presented in Figure 5-20.

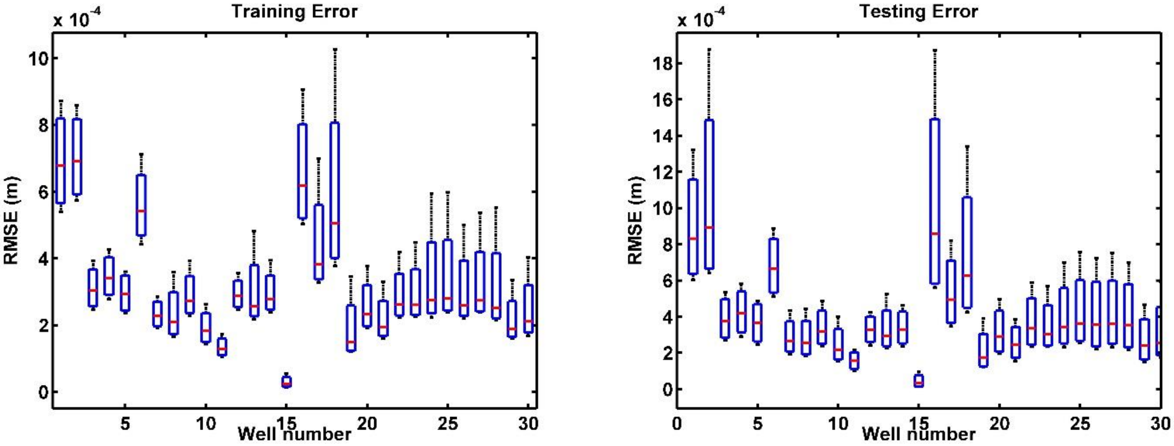


Figure 5-20: Training and testing error range for 30 ANNs (wells)

In most cases, the training error range is smaller than the testing error range, as would have been expected. The uncertainty results for the hydraulic head for the three representative wells, along with the observed hydraulic head values, are presented, in Figure 5-21.

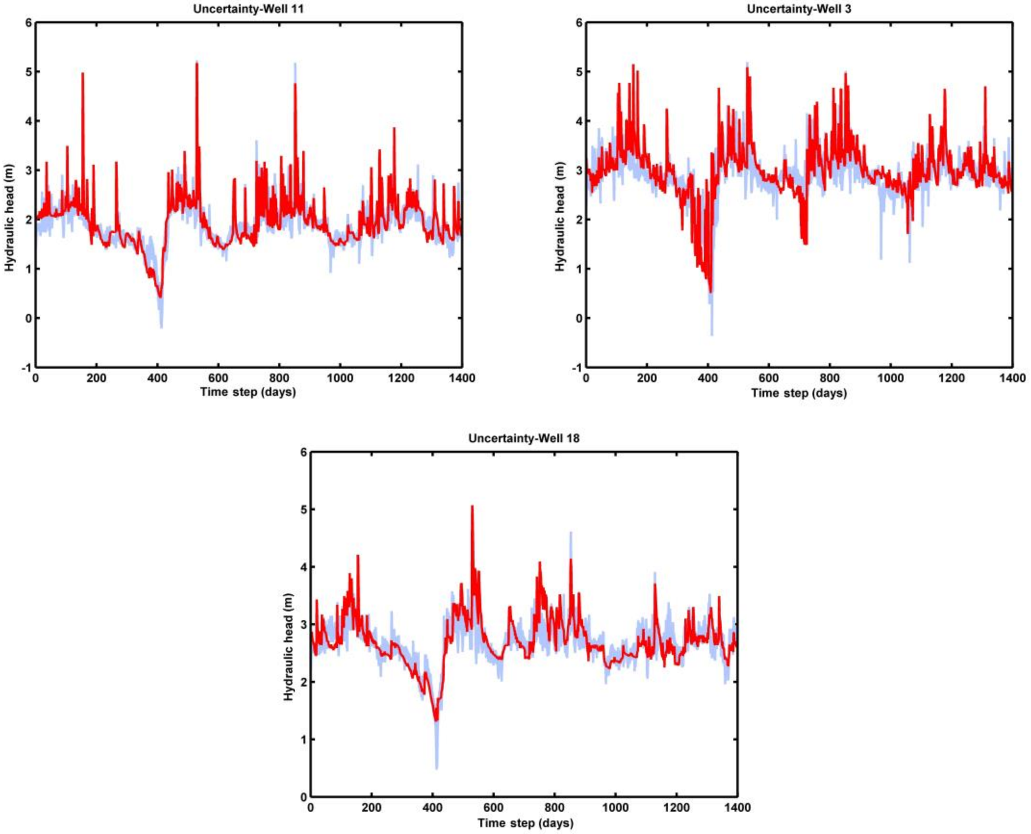


Figure 5-21: Uncertainty in hydraulic head due to uncertainty in ANN training for three representative wells (the observed values are marked in red and the prediction range in blue)

In most cases, the observed values fall within the prediction interval calculated by the Monte Carlo simulation of ANN training. A small percentage of observed values falls outside the prediction interval limits (Table 5-23). This percentage is small for the well category represented by well no. 11 and increases for the other two well categories, without, however, compromising the reliability of the model.

Table 5-23: Percentage of observed values outside the 90% prediction interval

Well Number	percentage of observed values outside the 90% prediction interval
11	5.5
3	9.1
18	11.2

The average width of the prediction intervals for each one of the representative wells are presented in Table 5-24.

Table 5-24: Average prediction intervals width for three representative wells

Well number	Average PI width (m)
11	0.153
3	0.243
18	0.452

By comparing the prediction intervals widths for the two different methodologies (Table 5-24and Table 5-22)is can be noted that the uncertainty derived for the training of the ANNs is only a small fraction of the model uncertainty calculated through the percentile methodology. This may also be attributed to the coarse nature of calculations in the percentile methodology.

The kriging part of the algorithm was executed for each of the 300 networks trained and prediction intervals were derived for every prediction point. The prediction intervals for prediction points 150, 300, 450 and 600 are presented in Figure 5-22.

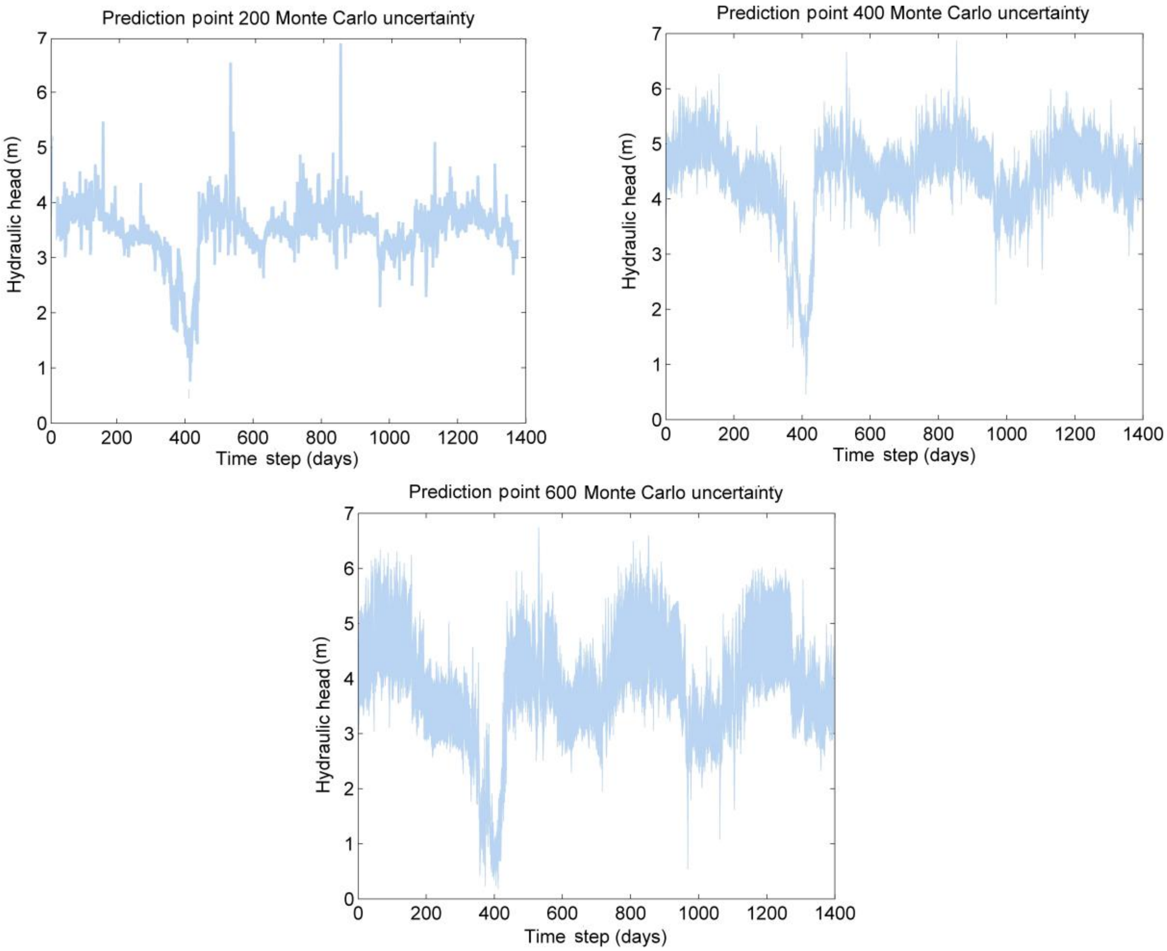


Figure 5-22: Hydraulic head prediction interval for four prediction points

More specifically, for the prediction points presented in Figure 5-22 the average width of the 90% prediction intervals are presented in

Table 5-25.

Table 5-25: Average prediction intervals width for three representative prediction points-ANN training

Prediction point	Average PI width (m)
200	0.25
400	0.86
600	1.424

Uncertainty in hydraulic head change due to kriging parameters

The effect of kriging parameter uncertainty was studied, similarly to the first case study. As in the previous case, 500 different data samples and 500 different variograms were constructed for every prediction point and time step. Uncertainty results were derived for the algorithm with use of fuzzy logic and for the exponential variogram model. The 90% prediction intervals together with the prediction derived from the ANN simulated values for three prediction points are presented in Figure 5-23 (simulation results from observed data are marked in red, while the range of the predictions from constructed data is marked in blue).

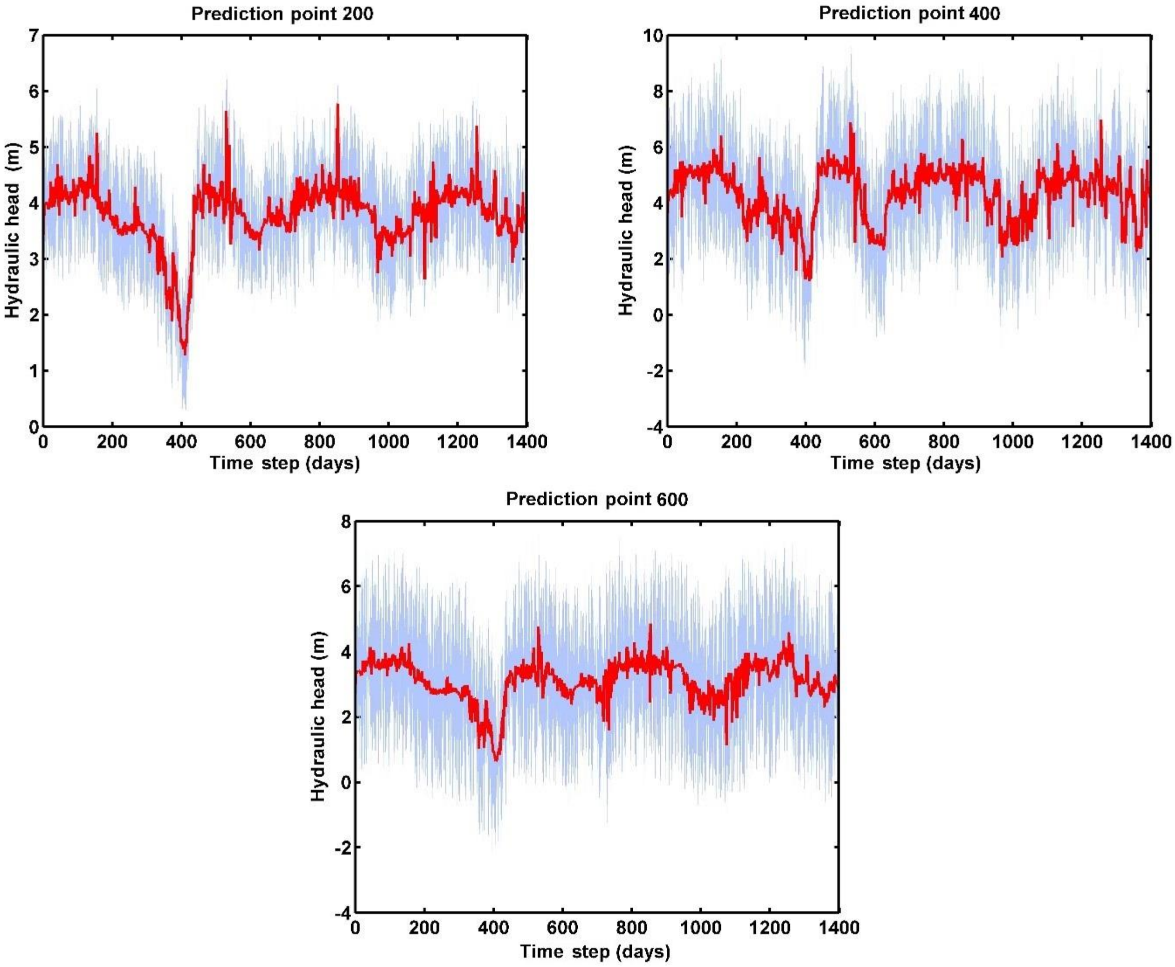


Figure 5-23: Uncertainty in hydraulic head change due to kriging parameters for three prediction points

The average prediction interval width for each one of the three representative prediction points is reported in Table 5-26.

Table 5-26: Average prediction intervals width for three representative prediction points-Kriging parameter uncertainty

Prediction point	Average PI width (m)
200	1.35
400	2.32
600	3.5

In all cases the uncertainty of the results derived from using the described methodology can be characterized as adequate, hence the model is reliable and can be used successfully for the hydraulic head simulation in an aquifer.

Chapter **6**

Conclusions

In the present dissertation, artificial intelligence tools and geostatistical methodologies were combined in order to simulate the hydraulic head in an aquifer both spatially and temporally. The first part of the algorithm that was developed includes the use of artificial neural networks for the temporal simulation of the hydraulic head. Artificial neural networks have been proven to be effective for this task in the past by using only readily available data such as meteorological data, hydraulic head values and surface water data. Any parameter linked to the water budget and not being constant throughout the simulation time can be used. The main advantage of ANNs over conventional modelling is the fact that they do not require geological data, which can be hard and costly to acquire. Various ANN architectures were tested in order to determine the most appropriate network, which was different for the two case studies. At the same time, the time lag between some parameters and their effect on the aquifer was also considered in order to optimize the results.

For the spatial simulation of the hydraulic head, the kriging interpolation methodology was applied. Different theoretical variogram models were tested for kriging in order to select the most appropriate one. In order to evaluate the performance of each variogram cross-validation was performed for the performance evaluation of each variogram. For 10% of the time steps, at prediction points which were also data points, the real measurements at the point were not taken into consideration, but the simulation continued and the value at the point was calculated as if it were not a data point. In this way, three values were available for this specific point; the observed (real) value, the value obtained through ANN simulation and the value obtained through ANN simulation and kriging. Various error indicators were calculated using these data in order to provide for the best possible choice of variogram model.

It became apparent, in the early stages of the algorithm development, that data points modeled by ANNs with large training and testing errors affected significantly the simulation results at nearby prediction points. For this reason, a fuzzy logic system was developed to determine the appropriate neighbors for every prediction point and tested. The fuzzy logic system takes into consideration both the ANN training and testing errors, and also the distance between a prediction point and the data points. The use of this system was proven to be helpful and to improve the cross-validation results.

The algorithm was developed in Visual Studio 2010/2013 in VB .NET, while for the visualization of the results MATLAB was used through the .NET interoperability. In order to reduce the runtime necessary for each simulation the algorithm was designed with multithreading capabilities. A Graphical User interface was developed in order to be user friendly. The final results of the simulation are provided to the user in the form of hydraulic head/hydraulic head change and error variance in an .mp4 file. Several text files containing all other intermediate simulation results are also provided.

The model was implemented for two case studies; one in Bavaria, Germany and one in Miami-Dade County, in Florida, USA. There are numerous differences between these two study areas, from the overall size of the area, to the geology and to the land use. Using the algorithm for two such diverse cases can put to test the general applicability of the algorithm.

The results for both case studies in terms of root mean squared error can be characterized as good. The lowest RMSE value was equal to $7.6 \cdot 10^{-3}$ m for the first case study and equal to 0.962 m for the second case study. A key difference leading to this deviation in error values was that hydraulic head change was simulated in the first case and hydraulic head in the second. This difference in output parameters was due to the dataset involved in every case. In the first study area they were large variations in the hydraulic head, while at the second case study, the values of the dataset were more normalized. In both cases, the performance of the algorithm was considered adequate, especially when taking into consideration the complexity of the second case study.

In order to evaluate the overall performance of the algorithm, uncertainty analysis was performed from various aspects. Firstly, the simulation results of the ANN were analyzed and the 90% confidence intervals for the predicted values were calculated using the percentile bootstrap methodology. Secondly, the training uncertainty of the ANNs was studied. Multiple executions of training were performed for all ANNs using each time different training and testing datasets and different initial random values for the neural weights. The results of every execution were then used in kriging interpolation and the final prediction interval at every prediction point was calculated. The last output uncertainty calculated was the one attributed to the kriging parameters. In this case synthetic data were created, having the same statistical characteristics of the ANN simulation results. By creating numerous variogram models using these data, the uncertainty in hydraulic head change due to the kriging parameters can be calculated.

For both study areas, all the calculated prediction intervals were within an acceptable range, hence the simulation model can be characterized as successful. Percentile methodology resulted in wide confidence intervals for the parameters studied, due to the coarse nature of the methodology involved. The uncertainty attributed to the ANN training, both for the ANN simulation and the kriging results was within reasonable range. Finally, the uncertainty in hydraulic head change due to the kriging parameters was low, hence the model can be characterized as reliable.

The developed model is a user friendly algorithm for the hydraulic head simulation. Using easy to acquire data, it can successfully simulate the output parameter, hydraulic head change or hydraulic head per time step, having small simulation cross-validation error and narrow uncertainty intervals for the output parameter. The results derived for both cases studies proved that the developed model can be successfully in different geological backgrounds and different study area sizes. The applicability of the model is not limited to the hydraulic head simulation, but it can be extended to other groundwater parameters that vary spatially and temporally, such as the concentration of a pollutant in groundwater.

Chapter 7

Further research

For the further verification of the model, it would be advisable to implement it in a study area with more spatially distributed validation points. Moreover, the use of the developed algorithm for the simulation of parameters other than the hydraulic head can be an interesting application. The concentration of a chemical substance can be an example of such application. Any parameter that varies spatially and temporally can be simulated by the algorithm, provided that long time series of data are available.

In order to further improve the results, various modifications can be realized. The training process of the ANNs can be altered, optimizing their architecture. This can be done by clustering the wells with available data into categories with same optimal ANN architecture, instead of using one common architecture for all the wells. In recent literature, it is stated that Support Vector Machines can yield better results than ANNs, without having at the same time any parameters to adjust. Since a large volume of data is necessary for the spatial and temporal simulation, it would be very helpful to link the developed code with database functionality. A good example would have been the use of SQL since its functionality can be linked to Visual Studio. Finally, the correlation analysis can be incorporated to the developed algorithm. As mentioned in previous chapters, the definition of the appropriate time lag for some parameters is of great importance. Incorporating this procedure and using its results, after a revision from the user concerning the final parameters used, as a preparatory step of the algorithm can reduce the overall time necessary for the model to run and can make the use of the program easier for inexperienced users.

Another possibility could be the determination of the velocity field in the study area, through the application of natural laws (Dupuit and Darcy's law, depending on the aquifer) to the simulation results. The use of the developed algorithm for the inverse problem is also an interesting possibility. In this case the algorithm can assist conventional models in the definition of aquifer parameters such as hydraulic conductivity and porosity.

References

Abbott, M. B., and J. C. Refsgaard (1996), *Distributed Hydrological Modelling*, Springer Netherlands.

Aboufirassi, M., and M. Mariño (1983), Kriging of water levels in the Souss aquifer, Morocco, *Journal of the International Association for Mathematical Geology*, *15*(4), 537-551.

Abrahart, R. J., A. J. Heppenstall, and L. M. See (2007), Timing error correction procedure applied to neural network rainfall—runoff modelling, *Hydrological Sciences Journal*, *52*(3), 414-431.

Ajmera, T. K., and A. Rastogi (2009), Artificial neural network application on estimation of aquifer transmissivity, *Journal of Spatial Hydrology*, *8*(2).

Amini, M., M. Afyuni, N. Fathianpour, H. Khademi, and H. Flühler (2005), Continuous soil pollution mapping using fuzzy logic and spatial interpolation, *Geoderma*, *124*(3–4), 223-233.

ASCE (1990a), Review of Geostatistics in Geohydrology. II: Applications, *Journal of Hydraulic Engineering*, *116*(5), 633-658.

ASCE (1990b), Review of Geostatistics in Geohydrology. I: Basic Concepts, *Journal of Hydraulic Engineering*, *116*(5), 612-632.

ASCE (2000), Artificial Neural Networks in Hydrology. I: Preliminary Concepts, in *Journal of Hydrologic Engineering*, edited, pp. 115-123, ASCE, ASCE Task Committee on Application of Artificial Neural Networks in Hydrology.

Aziz, K., A. Rahman, G. Fang, and S. Shrestha (2014), Application of artificial neural networks in regional flood frequency analysis: a case study for Australia, *Stoch Environ Res Risk Assess*, *28*(3), 541-554.

Baalousha, H. (2010), Assessment of a groundwater quality monitoring network using vulnerability mapping and geostatistics: A case study from Heretaunga Plains, New Zealand, *Agricultural water management*, *97*(2), 240-246.

Banerjee, P., V. S. Singh, K. Chatttopadhyay, P. C. Chandra, and B. Singh (2011), Artificial neural network model as a potential alternative for groundwater salinity forecasting, *Journal of Hydrology*, *398*(3–4), 212-220.

Bauer, M., K. Thuro, M. Scholz, and P. Neumann (2009), The geology of Munich (Germany) and its significance for ground modelling in urban areas, *Engineering geology of tomorrow's cities. Geological Society, London, Engineering Geology Special Publication 22*

Bhattacharya, B. (2005), *Learning from data for aquatic and geothenical environments*, UNESCO-IHE, Institute for Water Education.

Bochkanov, S., and V. Bystritsky (2011), ALGLIB-a cross-platform numerical analysis and data processing library, *ALGLIB Project. Novgorod, Russia.*

Bogardi, I., L. Duckstein, and F. Szidarovszky (1982), Bayesian analysis of underground flooding, *Water Resources Research*, *18*(4), 1110-1116.

Chatzakis, A., E. Tapoglou, and G. P. Karatzas (2014), Groundwater Simulation Using Artificial Neural Networks (ANNs) and Princeton Transport Code (PTC)- Performance Comparison, in *12th International Conference "Protection and Restoration of the Environment – PRE12"*, edited, pp. 2-8, Skiathos.

Chen, L., L. Ye, V. P. Singh, J. Zhou, and S. Guo (2013), Determination of Input for Artificial Neural Networks for Flood Forecasting Using the Copula Entropy Method, *Journal of Hydrologic Engineering*.

Coppola, E. A., A. J. Rana, M. M. Poulton, F. Szidarovszky, and V. W. Uhl (2005), A neural network model for predicting aquifer water level elevations, *Ground Water*, *43*(2), 231-241.

Cox, E. (1999), *The fuzzy systems handbook: a practitioner's guide to building, using, and maintaining fuzzy systems*, AP Professional.

Criss, R. E., and W. E. Winston (2008), Do Nash values have value? Discussion and alternate proposals, *Hydrological Processes*, *22*(14), 2723-2725.

Cruz, G. O., F. Ballesteros Jr, and A. Blanco (2014), Coupling Artificial Neural Networks and Genetic Algorithms in Redesigning Existing Cities for Flood Resistance, paper presented at ACHI 2014, The Seventh International Conference on Advances in Computer-Human Interactions.

Cunningham, K., M. Wacker, E. Robinson, C. Gefvert, and S. Krupa (2003), Hydrogeology and ground-water flow at Levee-31N Miami-Dade County, Florida, *US Department of the Interior, US Geological Survey*.

Delhomme, J. P. (1978), Kriging in the hydrosciences, *Advances in Water Resources*, *1*(5), 251-266.

Desbarats, A. J., C. E. Logan, M. J. Hinton, and D. R. Sharpe (2002), On the kriging of water table elevations using collateral information from a digital elevation model, *Journal of Hydrology*, *255*(1–4), 25-38.

Engelbrecht, A. P. (2007), *Computational Intelligence: An Introduction*, 2nd ed., John Wiley & Sons.

Fine, T. L. (1999), *Feedforward Neural Network Methodology*, Springer.

Fish, J. E., and M. T. Stewart (1991), *Hydrogeology of the surficial aquifer system, Dade County, Florida*, US Department of the Interior, US Geological Survey.

Gandin, L. S. (1963), Objective analysis of meteorological fields., *Translated from Russian in 1965 by Israel Program for Scientific Translations, Jerusalem* p. 242.

Gómez, I., L. Franco, J. Subirats, and J. Jerez (2006), Neural Network Architecture Selection: Size Depends on Function Complexity, in *Artificial Neural Networks – ICANN 2006*, edited by S. Kollias, A. Stafylopatis, W. Duch and E. Oja, pp. 122-129, Springer Berlin Heidelberg.

Govindaraju, R. S. (2000), Artificial neural networks in hydrology. I: Preliminary concepts, *Journal of Hydrologic Engineering*, *5*(2), 115-123.

Gundogdu, K. S., and I. Guney (2007), Spatial analyses of groundwater levels using universal kriging, *Journal of earth system science*, *116*(1), 49-55.

Haan, C. T. (2002), Statistical methods in hydrology, *Wiley.*

Haykin, S. S. (1999), *Neural networks: a comprehensive foundation*, Prentice Hall.

Hengl, T. (2007), A practical guide to geostatistical mapping of environmental variables, *Geoderma*, *140*(4), 417-427.

Hu, T., F. Wu, and X. Zhang (2007), Rainfall-runoff modeling using principal component analysis and neural network, *Nordic Hydrology*, *38*(3), 235-248.

Hundecha, Y., A. Bardossy, and H.-W. WERNER (2001), Development of a fuzzy logic-based rainfall-runoff model, *Hydrological Sciences Journal*, *46*(3), 363-376.

Kitanidis, P. K. (1997), *Introduction to Geostatistics: Applications in Hydrogeology*, Cambridge University Press.

Krige, D. (1951), A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand, *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, *52*(6), 119-139.

Kröse, B., and P. Van Der Smagt (1993), An introduction to neural networks, *University of Amsterdam.*

Kumar, V. (2006), Kriging of groundwater levels–a case study, *Journal of Spatial Hydrology*, *6*(1).

Lallahem, S., J. Mania, A. Hani, and Y. Najjar (2005), On the use of neural networks to evaluate groundwater levels in fractured media, *Journal of Hydrology*, *307*(1–4), 92-111.

Legates, D. R., and G. J. McCabe (1999), Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation, *Water Resources Research*, *35*(1), 233-241.

Loukas, A., and L. Vasiliades (2004), Probabilistic analysis of drought spatiotemporal characteristics inThessaly region, Greece, *Natural Hazards and Earth System Science*, *4*(5/6), 719-731.

Maier, H. R., and G. C. Dandy (2000), Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, *Environmental Modelling & Software*, *15*(1), 101-124.

Marinoni, O. (2003), Improving geological models using a combined ordinary–indicator kriging approach, *Engineering Geology*, *69*(1–2), 37-45.

Matheron, G. (1963), Principles of geostatistics, *Economic Geology*, *58*(8), 1246-1266.

McCulloch, W., and W. Pitts (1943), A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, *5*(4), 115-133.

Minsky, M., and S. Papert (1969), Perceptrons: an introduction to computational geometry, *The MIT Press, Cambridge MA*, 105-110.

Mirchandani, G., and W. Cao (1989), On hidden nodes for neural nets, *Circuits and Systems, IEEE Transactions on*, *36*(5), 661-664.

Mohanty, S., M. Jha, A. Kumar, and K. P. Sudheer (2010), Artificial Neural Network Modeling for Groundwater Level Forecasting in a River Island of Eastern India, *Water Resources Management*, *24*(9), 1845-1865.

Mohanty, S., M. K. Jha, A. Kumar, and D. K. Panda (2013), Comparative evaluation of numerical model and artificial neural network for simulating groundwater flow in Kathajodi–Surua Inter-basin of Odisha, India, *Journal of Hydrology*, *495*, 38-51.

Moriasi, D., J. Arnold, M. Van Liew, R. Bingner, R. Harmel, and T. Veith (2007), Model evaluation guidelines for systematic quantification of accuracy in watershed simulations, *Transactions of the ASABE*, *50*(3), 885-900.

Nayak, P., Y. Rao, and K. Sudheer (2006), Groundwater Level Forecasting in a Shallow Aquifer Using Artificial Neural Network Approach, *Water Resources Management*, *20*(1), 77-90.

Nayak, P. C., K. P. Sudheer, D. M. Rangan, and K. S. Ramasastri (2004), A neuro-fuzzy computing technique for modeling hydrological time series, *Journal of Hydrology*, *291*(1–2), 52-66.

Negnevitsky, M. (2005), *Artificial Intelligence: A Guide To Intelligent Systems*, Addison Wesley Publishing Company Incorporated.

Nguyen, H. T., N. R. Prasad, C. L. Walker, and E. A. Walker (2010), *A First Course in Fuzzy and Neural Control*, Taylor & Francis.

Pilz, J., and G. Spöck (2008), Why do we need and how should we implement Bayesian kriging methods, *Stoch Environ Res Risk Assess*, *22*(5), 621-632.

Rao, S. V. L. N., and J. Prasad (1982), Definition of kriging in terms of fuzzy logic, *Journal of the International Association for Mathematical Geology*, *14*(1), 37-42.

Rasmussen, C. E., and C. K. I. Williams (2006), *Gaussian Processes for Machine Learning*, the MIT Press.

Reddy, D. R., P. Bhramara, and K. Govindarajulu (2014), Application of Soft Computing Techniques for Analysis of Vapour Compression Refrigeration System, *International Journal of Current Engineering and Technology 2277 – 4106*, 368-373.

Rizzo, D. M., and D. E. Dougherty (1994), Characterization of aquifer properties using artificial neural networks: Neural kriging, *Water Resources Research*, *30*(2), 483-497.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1985), Learning internal representations by error propagation, DTIC Document.

Samani, N., M. Gohari-Moghadam, and A. A. Safavi (2007), A simple neural network model for the determination of aquifer parameters, *Journal of Hydrology*, *340*(1–2), 1-11.

Schaefli, B., and H. V. Gupta (2007), Do Nash values have value?, *Hydrological Processes*, *21*(15), 2075-2080.

Schalkoff, R. J. (1997), *Artificial neural networks*, McGraw-Hill.

See, L., D. Solomatine, R. Abrahart, and E. Toth (2007), Hydroinformatics: computational intelligence and technological developments in water science applications—Editorial, *Hydrological Sciences Journal*, *52*(3), 391-396.

Snepvangers, J. J. J. C., G. B. M. Heuvelink, and J. A. Huisman (2003), Soil water content interpolation using spatio-temporal kriging with external drift, *Geoderma*, *112*(3–4), 253-271.

Solomatine, D., and A. Ostfeld (2008), Data-driven modelling: some past experiences and new approaches, *Journal of hydroinformatics*, *10*(1), 3-22.

Stytz, M. R., and R. W. Parrott (1993), Using kriging for 3d medical imaging, *Computerized Medical Imaging and Graphics*, *17*(6), 421-442.

Ta'any, R., A. Tahboub, and G. Saffarini (2009), Geostatistical analysis of spatiotemporal variability of groundwater level fluctuations in Amman–Zarqa basin, Jordan: a case study, *Environ Geol*, *57*(3), 525-535.

Taormina, R., K.-w. Chau, and R. Sethi (2012), Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon, *Engineering Applications of Artificial Intelligence*, *25*(8), 1670-1676.

Tapoglou, E., G. P. Karatzas, I. C. Trichakis, and E. A. Varouchakis (2014), A spatio-temporal hybrid neural network-Kriging model for groundwater level simulation, *Journal of Hydrology*, *519, Part D*(0), 3193-3203.

Tapoglou, E., I. C. Trichakis, Z. Dokou, I. K. Nikolos, and G. P. Karatzas (2013), Groundwater-level forecasting under climate change scenarios using an artificial neural network trained with particle swarm optimization, *Hydrological Sciences Journal*, *59*(6), 1225-1239.

Tettamanzi, A., and M. Tomassini (2001), *Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems*, Physica-Verlag.

Trichakis, I., I. Nikolos, and G. P. Karatzas (2011a), Comparison of bootstrap confidence intervals for an ANN model of a karstic aquifer response, *Hydrological Processes*, *25*(18), 2827-2836.

Trichakis, I., I. Nikolos, and G. Karatzas (2011b), Artificial Neural Network (ANN) Based Modeling for Karstic Groundwater Level Simulation, *Water Resources Management*, *25*(4), 1143-1152.

Trichakis, I. C., I. K. Nikolos, and G. P. Karatzas (2009), Optimal selection of artificial neural network parameters for the prediction of a karstic aquifer's response, *Hydrological Processes*, *23*, 2956-2969.

Tu, J. V. (1996), Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, *Journal of Clinical Epidemiology*, *49*(11), 1225-1231.

USGS (2014), Geological units in Miami-Dade county, Florida, mrdata.usgs.gov/geology/state/fips-unit.php?code=f12086 accessed: 2014, edited.

Varouchakis, E. A., and D. T. Hristopulos (2013a), Improvement of groundwater level prediction in sparsely gauged basins using physical laws and local geographic features as auxiliary variables, *Advances in Water Resources*, *52*, 34-49.

Varouchakis, E. A., and D. T. Hristopulos (2013b), Comparison of stochastic and deterministic methods for mapping groundwater level spatial variability in sparsely monitored basins, *Environ Monit Assess*, *185*(1), 1-19.

Yang, F.-g., S.-y. Cao, X.-n. Liu, and K.-j. Yang (2008), Design of groundwater level monitoring network with ordinary kriging, *Journal of Hydrodynamics, Ser. B*, *20*(3), 339-346.

Zadeh, L. A. (1965), Fuzzy sets, *Information and Control*, *8*(3), 338-353.

Zadeh, L. A. (1994), Fuzzy logic, neural networks, and soft computing, *Commun. ACM*, *37*(3), 77-84.

Zounemat-Kermani, M., and M. Teshnehlab (2008), Using adaptive neuro-fuzzy inference system for hydrological time series prediction, *Applied Soft Computing*, *8*(2), 928-936.