



**TECHNICAL UNIVERSITY OF CRETE**

**SCHOOL OF ENVIRONMENTAL ENGINEERING**

## **Διπλωματική Εργασία - Diploma Thesis**

**Predicting the spatial distribution of the aquifer head using a  
Radial Basis Function network**

**Πρόβλεψη της χωρικής κατανομής υδραυλικού  
ύψους υδροφορέα με τη χρήση νευρωνικού δικτύου  
Ακτινικών Συναρτήσεων Βάσης**

**ΒΑΣΙΛΗΣ ΤΣΑΠΑΡΑΣ**

**(VASILIS TSAPARAS)**

Dissertation committee:

George Karatzas (Supervising)

Ioannis Nikolos

Emmanouil Varouchakis

## **ACKNOWLEDGEMENTS**

First, I would like to thank my supervising professor Dr. George Karatzas, who gave me the chance to select this demanding, but interesting subject.

Especially, I would like to thank professor Dr. Ioannis Nikolos from the School of Production Engineering and Management for the time and effort he dedicated for the entire duration of this time consuming work, to guide me and provide me the help and material needed.

Moreover I would like to thank very much Panagiota Theodoridou for supplying the data on which this dissertation is based, as well as for the help, the details and the material she gave me for them.

I would also want to thank Dr. Emmanouil Varouchakis for his participation in the dissertation committee.

## ABSTRACT

The purpose of this work is to estimate the hydraulic head of an aquifer in the regional unit of Drama using an artificial neural network called RBFN (Radial Basis Function Network).

Conventional modeling techniques are often time-consuming and costly and have limitations in data and knowledge. Artificial neural networks provide an alternative to these obstacles, as they can provide solutions without defining the relationship between the data and the results, by being trained by data and generalizing. Their function is based on the biology of the human brain and is a form of artificial intelligence.

Artificial neural networks are trained by modifying the interneuronconnection strengths, known as synaptic weights, between their artificial neurons. This is done by feeding the network with input-output examples, to give them a desired response to input data and, thus, to modify their synaptic weights according to a learning rule. After a large number of iterations, the artificial neural network has constructed an input-output mapping and has adapted to the problem.

In the study area we have data from 250 points from observation wells. For each point we have coordinates  $x$ ,  $y$  and hydraulic head measurements. The neural network was trained on the majority of data and its accuracy was tested on a small number of testing data. Tests were performed by dividing the data set into subsets and training the neural network with each of them, as well as tests with random data. The accuracy of the neural network was examined in each case and graphical representations of the results and deviations from the measurement values were made.

**CONTENTS**

ACKNOWLEDGEMENTS.....i

ABSTRACT..... ii

CONTENTS.....iii

1. INTRODUCTION.....1

    1.1 GROUNDWATER.....1

    1.2 MODELLING TECHNIQUES.....2

    1.3 THE PROBLEM AT HAND.....4

2. BACKGROUND AND LITERATUREREVIEW.....8

2.1 ARTIFICIAL NEURAL NETWORKS (ANN).....8

    2.1.1 Definition.....8

    2.1.2 Human brain and artificial neural networks (ANN).....9

    2.1.3 Fundamentals of artificial neural networks (ANN).....10

    2.1.4 Activation function.....11

    2.1.5 Artificial neural network (ANN) architectures.....14

    2.1.6 Types of learning.....15

    2.1.7 History of ANN .....15

    2.1.8 Advantages of artificial neural networks (ANN).....17

    2.1.9 Multilayer perceptron networks (MLP).....18

    2.1.10 Radial basic function networks (RBFN).....20

    2.1.11 Comparison of MLP and RBFN.....22

2.2 PAST CASES.....23

    2.2.1 Uses of Multilayer perceptron networks for spatial predictions.....23

    2.2.2 Uses of Radial basis function networks for spatial predictions.....25

3. METHODS AND APPLICATION .....33

3.1 THE ANN ALGORITHM.....33

3.1.1 Characteristics.....	33
3.1.2 The Input files' structure .....	34
3.1.3 The output files' structure .....	37
3.2 TESTING THE ANN ALGORITHM.....	40
3.2.1 Testing with the entire set of data.....	40
3.2.2 Testing with divided subset groups of data.....	42
3.2.3 Contour maps.....	53
3.2.4 Testing with a random group of data .....	57
3.2.5 Observations.....	59
4. CONCLUSIONS.....	61
REFERENCES.....	62

# CHAPTER 1: INTRODUCTION

## 1.1 GROUNDWATER

Groundwater consists of water in pore spaces and fractures in rock and sediment underneath the Earth's surface. It reaches the ground through rainfall and snow and moves through the soil and inside the groundwater system, from where it will make its way back to surface streams, lakes, or oceans. Although groundwater only makes about 1% of the water on earth (as most of it resides in oceans), it makes about 35 times the amount of water in lakes and streams (Nelson, 2015). Globally, more than a third of all water used by humans comes from groundwater. In rural areas the percentage is even higher, with more than half of all drinking water supplied by groundwater (Nelson, 2015).

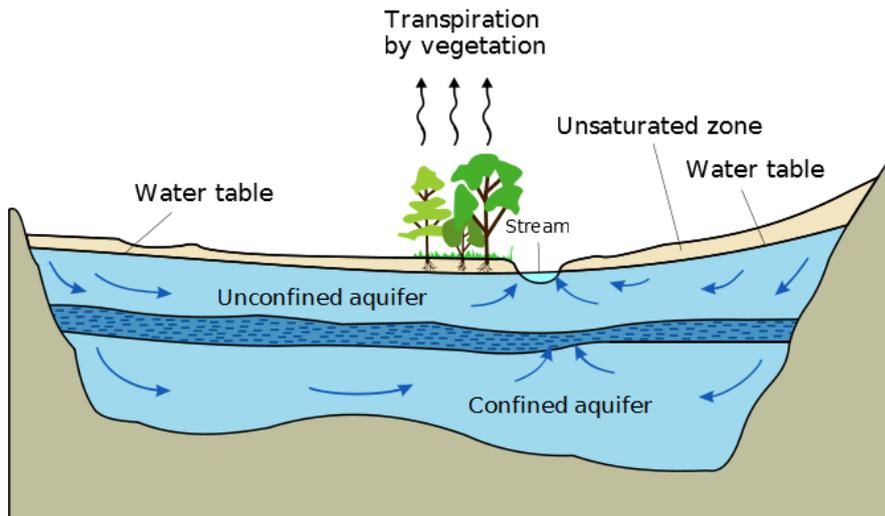
Aquifers are geologic formations from which large amounts of groundwater can be pumped for domestic, municipal, or agricultural uses. Sometimes, aquifers are separated vertically from each other by geologic formations that allow little or no water to flow in or out. If it is much less permeable than a nearby aquifer but still permits flow (e.g. sandy clay), such a formation is called an aquitard. When the water barrier is almost impermeable (e.g. clay) and forms a formidable flow barrier between aquifers it is called an aquiclude (Harter, 2003).

There are two types of aquifers:

- Unconfined aquifers: This is the most common type of aquifer, where the water table is exposed to the Earth's atmosphere through the zone of aeration.
- Confined aquifer: This type occurs an aquifer is confined between layers of impermeable strata (aquitards).

When rain falls on the surface it seeps down through the soil and into a zone called the unsaturated zone, where most of the pore spaces are filled with air. As it penetrates deeper it eventually enters a zone where all pore spaces and fractures are filled with water. This zone is the saturated zone. The surface beneath, which all openings in the rock are full of water, is called the water table (Harter, 2003).

The water table is everywhere beneath the Earth, even in desert regions but it rarely intersects the surface. In more humid regions it reaches the surface at streams and lakes and generally follows surface topography. The water depth changes accordingly with the amount of water flowing in and out of the saturated zone. For example during dry seasons, the depth to the water table increases, while during wet seasons the depth to the water table decreases (Nelson, 2015).



**Figure 1.1: Aquifer Diagram (USGS, 2016).**

The movement of groundwater is directed from higher elevations to lower elevations and also from higher pressure locations to lower pressure locations. Often, this movement is quite slow, because the water must pass between free spaces in the rock. In “groundwater hydraulics”, water pressure surface and water table elevation are described by the term “hydraulic head”. The hydraulic head is the driving force that powers groundwater movement as it is always directed downward of the hydraulic head gradient. If there is no hydraulic head gradient there can be no flow (Harter, 2003).

In order to intersect with the water table, a water well must be constructed. A water well is an excavation that can be used as a source of groundwater. If it is dug beneath the water table, water will fill the open space to the level of the water table and can be drawn out with a bucket or by pumping (Nelson, 2015). The groundwater elevation can also be measured at wells, which is important for this project. A groundwater monitoring well is a type of well-made for this specific purpose.

## **1.2 MODELLING TECHNIQUES**

Conventional modelling techniques are usually mathematical-analytical methods for an accurate modelling of the processes conducting detailed studies of physical, chemical and other dependencies. This is a time and cost consuming process, which requires much model calibration. In many cases this approach is not really applicable because of data and knowledge limitations

In hydrological processes, conventional modelling techniques include Statistical models and physically based hydrological models.

Statistical models: Hydrological phenomena are inherently random and complex by nature, making deterministic mathematical models that describe their physical

processes hard to be developed (Grimaldi et al., 2011). Therefore statistical approaches have been used to provide sufficient analyses. Statistical methods depend on data availability and quality, therefore they are strongly affected by conditions on the available dataset. Simulation, forecasting, uncertainty analysis, risk analysis and spatial interpolation are among the most important statistical models (Grimaldi et al., 2011).

Physically based hydrological modelling: This approach is based on our understanding of the physics of the hydrological processes and the physical equations that describe these processes (Seth, 2009). After the links between the model parts are determined, it can be solved with numerical methods. As the input data and computer requirements are enormous, their use is not practical for real time forecasting (Seth, 2009). Examples include watershed runoff models based on ST. Venant equations, infiltration models based on two phase flow theory of porous media and evaporation models based on theories of turbulence and diffusion (Xu, 2002).

Common hydrological modelling software includes MODFLOW and SWAT. MODFLOW originates from the US geological Survey and is used extensively worldwide for groundwater resource management (Akram et al., 2012). It is a three dimensional groundwater flow modeling environment that can be used for practical applications and simulation of contaminant transport. It functions by solving a system of equations describing the major flow using finite difference methods (Akram et al., 2012). SWAT is a continuous semi distributed watershed simulator that functions on a daily time step. Its components are hydrology, weather, erosion, plant growth, nutrients, pesticides, land management and stream routing (Arnold et al., 1998). ArcSWAT is a visual version of SWAT that is integrated with ArcGIS.

Multivariate approaches are methods that are considering different influencing factors and their relationships and allowing a certain generalization (Noack et al., 2014). Very common multivariate approaches include factor analysis, discriminant analysis and cluster analysis.

Factor Analysis: This technique is used when there are many variables in a research design and their reduction is necessary to a smaller set of factors (Richarme, 2002). There is no dependent variable and the researcher is looking for the underlying structure of the data matrix. It works optimally when the independent variables are normal and continuous with at least 3 to 5 variables for each factor (Richarme, 2002).

Discriminant analysis: This approach is used to describe characteristics that are specific to distinct groups and classify cases into preexisting groups based on

similarities between the case and the other case belonging to the groups. Mathematically this means that the objective is to weigh and linearly combine information from a set of dependent variables  $p$  so that the  $k$  groups are as distinct as possible (Brown et al., 2000). Its objective is most commonly to find a predictive equation to classify new cases or to interpret the predictive equation to better understand the relationships among the variables (Richarme, 2002).

Cluster Analysis: The purpose of this method is to divide large sets of data to subgroups that have more meaning (Richarme, 2002). This division is performed on the basis of similarity of the objects regarding certain characteristics. Outliers can be a problem and can be caused by too many irrelevant variables. It is better for the sample to be representative of the population (Richarme, 2002).

Compared to exact mathematical-analytical-numerical modelling, multivariate approaches offer several advantages (Noack et al., 2014):

- They are applicable even if relationships between the depending variable and the influencing factors are not really known.
- They consider many influencing factors.
- They work with available data.
- They are, comparably, quick and easy to use.

Artificial Neural Networks (ANNs) provide a multivariate data analysis. By training themselves from given examples they can learn the solution to complex problems without an explicit programming. This allows the user to produce a plausible result, even when the details of the relationship between the depending value and the controlling factor are unknown (Noack et al., 2014).

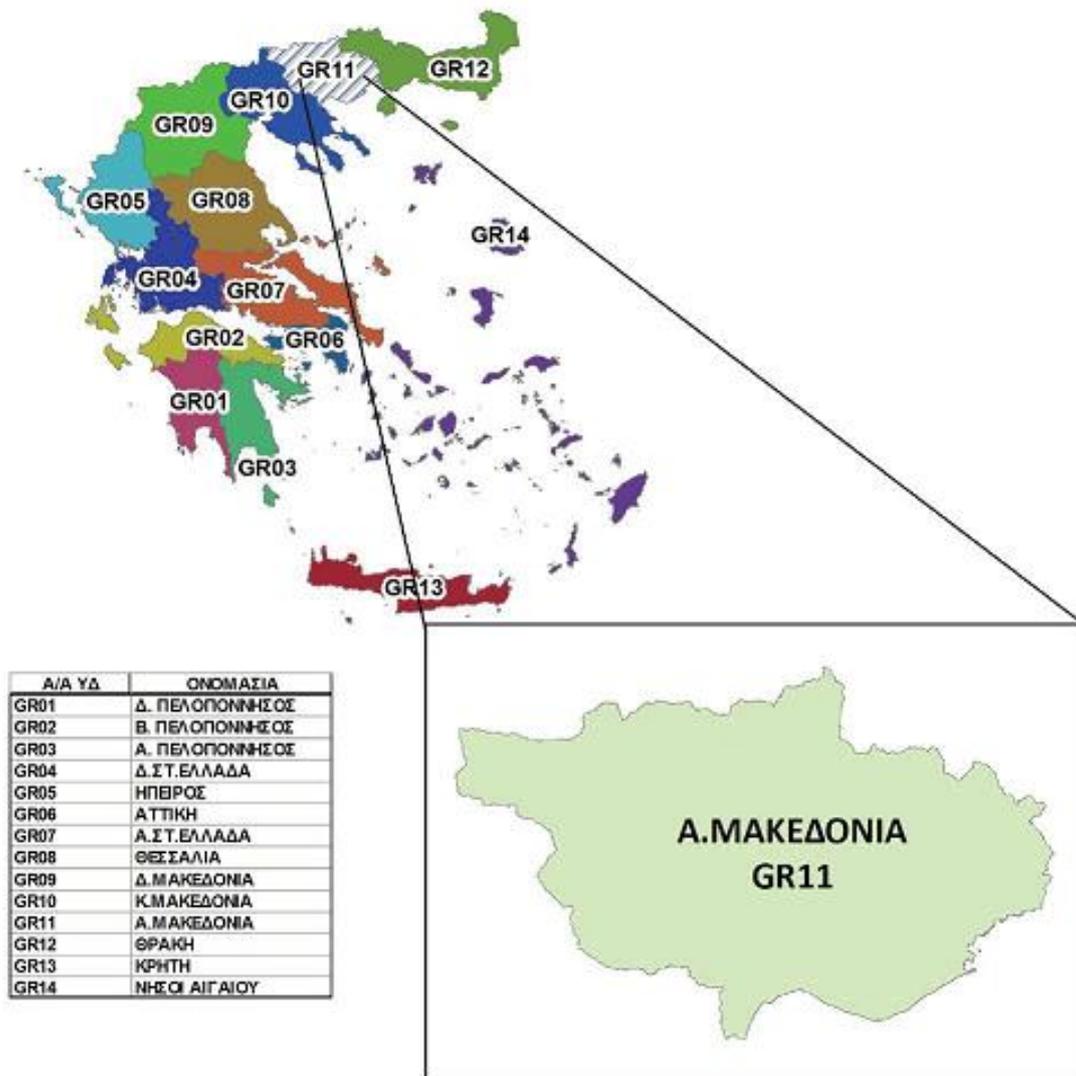
### **1.3 THE PROBLEM AT HAND**

#### Study area and available data

The study area is located at the regional unit of Drama in northern Greece. It was collected by the Water Administration of East Macedonia and Thrace and the Administration of Environment and Spatial Design of Drama. It belongs to the water region of East Macedonia (GR11) which covers an area of 7320 km<sup>2</sup>. The water region consists mostly of plains and a small amount of mountainous and semi mountainous terrain. The plains of Drama and Serres are the biggest part of the region.

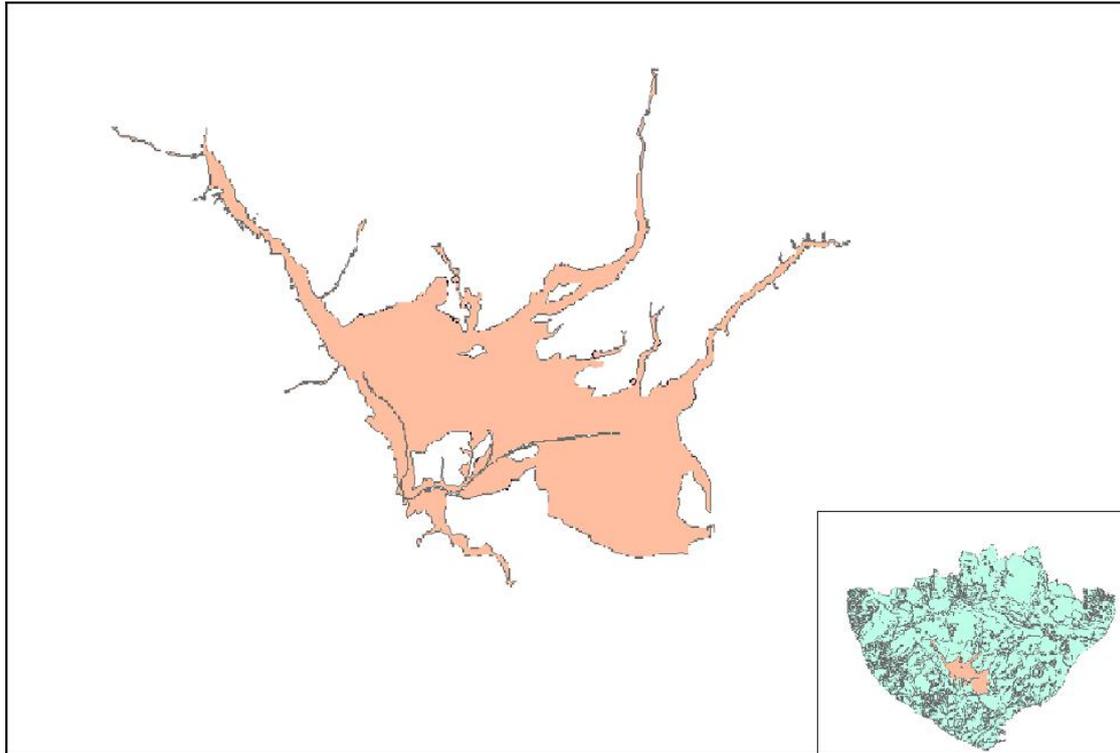
The climate of the region is semi-continental with an average temperature of 14.5°C to 16°C. The summers are hot and the winters are cold with January being the

coldest month and July the hottest. The average annual is between 508mm and 576 mm. The main use of the water in the water region is for irrigation (92%).



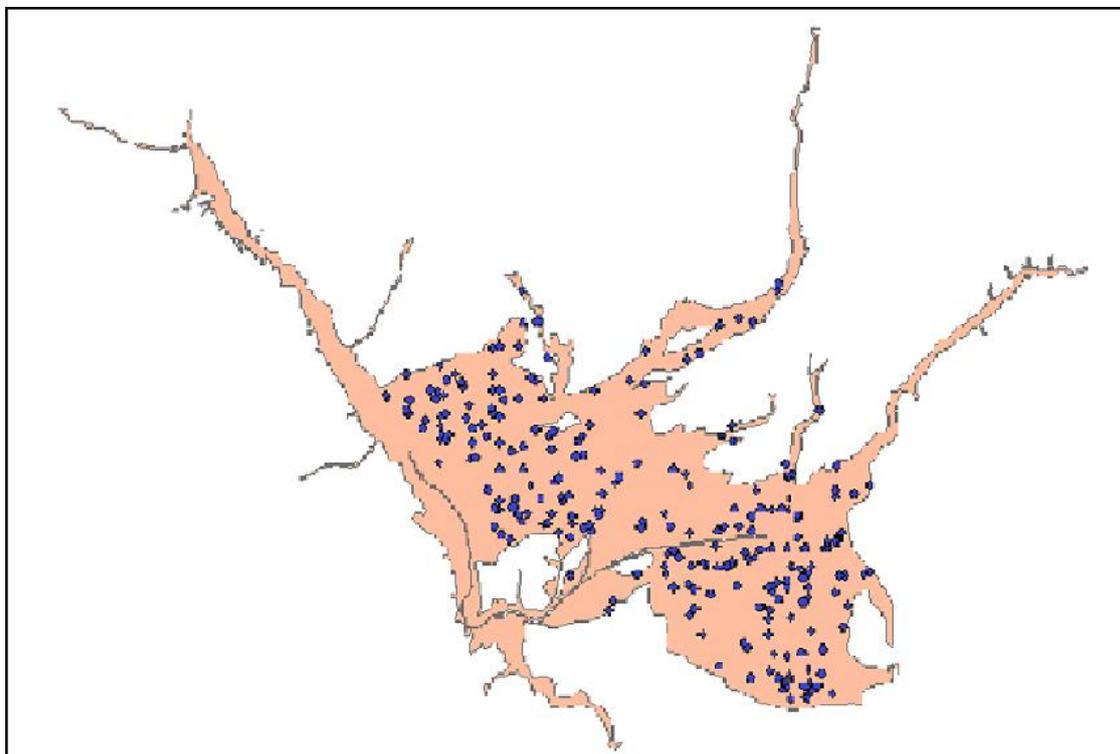
**Figure 1.1** The water regions of Greece and the water region of East Macedonia (GR11)

The area of interest, within which the data was collected, covers 210 km<sup>2</sup>. The aquifer is covered mainly by alluvium and is located under parts of the areas of Drama, Prosotsanis and Doksato.



**Figure 1.2** The Study area and its location within the water district of East Macedonia (GR11).

Our data is collected by a groundwater monitoring network of 250 wells. Each well is defined by its x- and y-coordinates and by its hydraulic head  $z$  (in meters).



**Figure 1.3** Spatial distribution of the 250 wells.

## Purpose

The purpose of this work is to provide a spatial estimation of water table elevation in our area of interest using a type of ANN called Radial Basis Function Network (RBFN). It will determine if this type of network can be effectively used for accurate spatial estimators of water table elevation, and therefore if it can be effectively and efficiently used as an alternative to conventional modelling techniques (numerical solution of flow equations).

## CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

### 2.1 ARTIFICIAL NEURAL NETWORKS (ANN)

#### 2.1.1 Definition

As mentioned before, multivariate approaches like ANNs provide the ability to determine the missing data needed, without explicit knowledge of the physical parameters of the system under examination. In our case, the input data required is the measured water table elevation in specific known positions (wells with known coordinates) so as to initially train the ANN. After the training phase, the ANN will be able to predict the water table elevation in other positions (not used for its training).

An Artificial Neural Network (ANN) is a computational model designed to simulate the neurons of the human brain that are capable of performing specific information-processing tasks. It is a system composed of many simple processing nodes, whose function is determined by network structure and connection strength (Gopal, 2017).

Knowledge is acquired by the network from its environment through an initial learning process. The procedure used to perform the learning process is called a learning algorithm, the function of which is to modify the so-called synaptic weights of the network in an orderly fashion to attain a design objective. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge (Haykin, 1999). Each artificial neuron receives one or more input signals and sums them to generate an output. The sums of each node are weighted, and the values are passed through a nonlinear function known as an activation function or transfer function. The transfer function in classic Multi-layer Perception (MLP) networks is usually a sigmoid function (Gopal, 2017).

Thanks to their ability to learn from data ANNs can adapt themselves to changes in the environment and deal in a very effective way with incomplete information and noisy data. In contrast to numerical models, they can be useful in problems where it is not possible to define the rules or the steps that lead to the solution (Gopal, 2017). They have applications in a large number of fields like finance, engineering, robotics, simulation, control and their use continues to expand since the amount of data globally is rising along with the data processing speed of computers (Gopal, 2017).

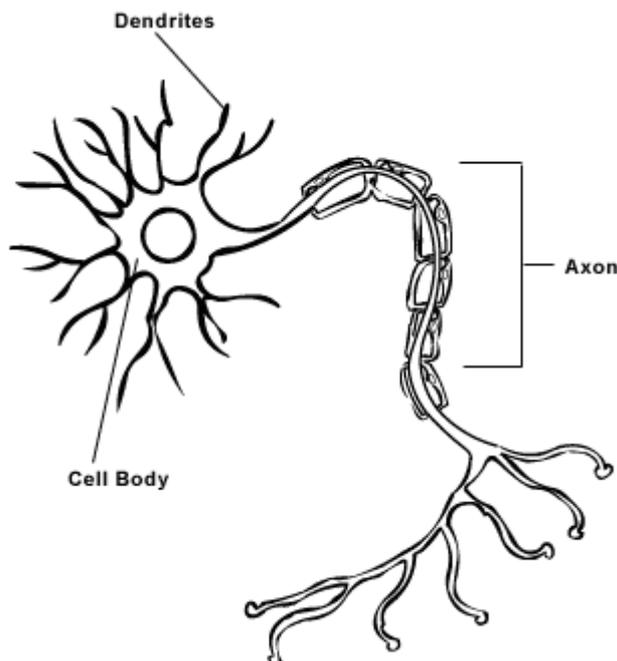
Simulating the way the human brain operates lies at the center of the notion of ANNs and the research that has been performed around it. Therefore, it would be helpful to initially make a brief reference to the structure and operation of the human brain.

### 2.1.2 Human brain and Artificial Neural Networks (ANNs)

The awareness that the human brain does not operate like a conventional computer but, on the contrary, in a very complex, nonlinear and parallel fashion, enabling it to process information with enormous speed and facility, has raised interest in and considerable development in this field of Artificial Intelligence (AI) (Haykin, 1999).

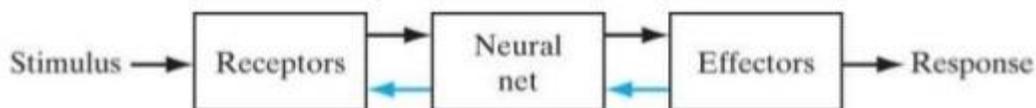
The brain is the central part of the human nervous system, consists of approximately hundreds of billions of neural cells, the neurons. Cooperation between these billions of neurons results in unified, coordinated and organized human behavior (Kalat, 2001). The idea of neurons as structural blocks of the human brain was introduced by Ramon y Cajal over a century ago and it facilitated the understanding of this complex matter in a considerable way (Haykin, 1999). To elaborate further, each neuron consists of (Haykin, 1999):

- **The cell body**, where all the genetic information is stored and where most of its metabolic activity is carried out.
- **The dendrites**, which branch out from the cell body and which incorporate the synapses, namely the connective units between neurons.
- **The axon**, a single and much longer filament that stems out from the cell body. The axon is responsible for transmitting 'action potentials' (brief voltage pulses) along its body towards another neuron. Each axon, in turn, also ends in synaptic terminals.



**Figure 2.1: A biological neuron (Woodford, 2018).**

In the above description the synapses hold a key position in the communication between neurons. Pioneering work in this area was carried out by Sherrington as early as 1906, who first introduced the term (Kalat, 2001). Synapses are the key connective and function units that enable interaction between neurons. This interaction is primarily of a chemical nature. A transmitter substance (chemical) is released through a presynaptic process. This substance diffuses across the synaptic junction between neurons and then initiates a postsynaptic process. Therefore, the synaptic junction between neurons is bridged in the following way: A presynaptic electrical signal is converted into a chemical signal and then back into a postsynaptic electrical signal (Haykin, 1999).



**Figure 2.2: Block diagram representation of the nervous system(Haykin, 1999).**

In the figure above the human nervous system is depicted as a three stage system. The brain is represented by the neural net which holds a central position in the system and which is constantly receiving information, processing it and making it appropriate decisions. The receptors convert stimuli (inputs) from the human body or the external environment into electrical impulses. These, in turn convey information to the neural net (the brain). Finally, the electrical impulses generated by the neural net are connected into responses (system outputs) by the effectors(Haykin, 1999).

### **2.1.3 Fundamentals of Artificial Neural Networks (ANNs)**

In this section, the fundamentals of ANNs and their basic building blocks will be presented. Afterwards, in the following sections, different types of architectures and neural networks will be examined, being variations of these basic elements.

The neuron is the basic information processing unit that is vital to the operation of the neural network. The model of the neuron forms the basis of designing artificial neural networks (Haykin, 1999).

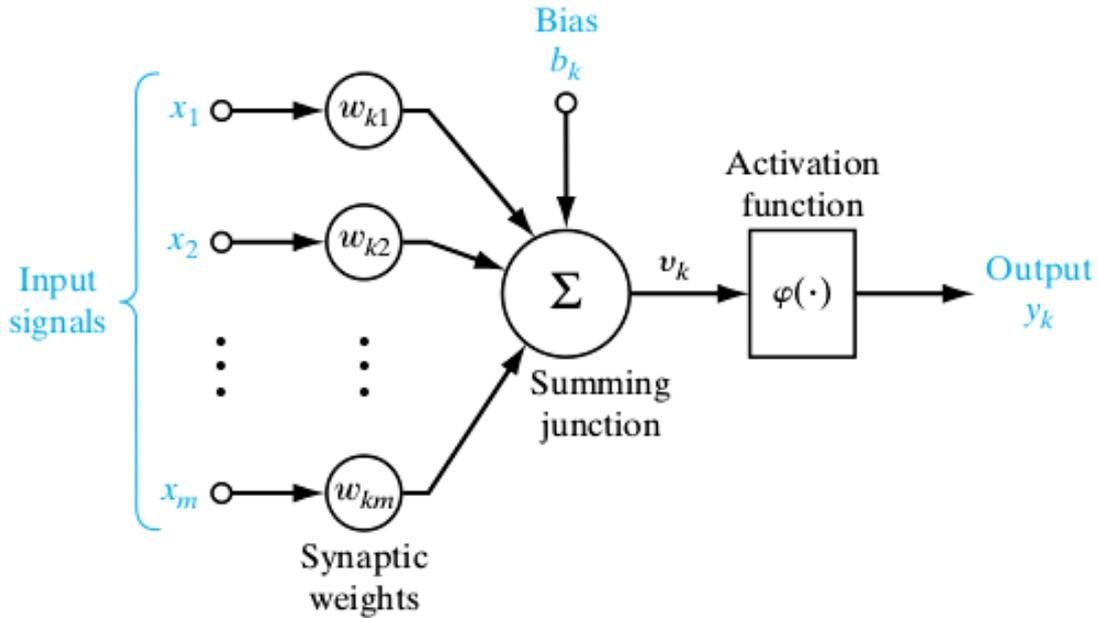


Figure 2.3: Nonlinear model of a neuron (Haykin, 1999).

The neuronal model consists of a set of connections between the neurons that are characterized by a weight. More specifically, an input signal  $x$  of synapse  $j$  connected to neuron  $k$  is multiplied by the weight  $w_{kj}$ . It also includes an adder for summing the input signals modified by the weights, in an operation that constitutes a linear combiner. An activation function reduces the amplitude of the neurons output to a certain range of values (Haykin, 1999). Finally, the neuronal model includes an externally added bias  $b_k$ . The bias is a constant that is independent from the networks input but is adapted by the learning procedure (Kroset al., 1996).

#### 2.1.4 Activation function

The activation function gives the effect of the total input on the activation of the unit  $k$  (the neuron). It is a function  $\phi$ , which takes the total input and produces a news value of the activation of the unit  $k$  (Kroset al., 1996). According to (Haykin, 1999) there are 3 types of activation functions (for MLP networks).

**1. Threshold function:** This type of activation function works as follows:

$$\varphi(u) = \begin{cases} 1 & \text{if } u_k \geq 0 \\ 0 & \text{if } u_k < 0 \end{cases} \quad (2.1)$$

Therefore, the output of neuron  $k$  is written as follows:

$$y_k = \begin{cases} 1 & \text{if } u_k \geq 0 \\ 0 & \text{if } u_k < 0 \end{cases} \quad (2.2)$$

The induced local field  $u_k$  of the neuron is,

$$u_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (2.3)$$

The threshold function's graph can be viewed in Figure 2.4.

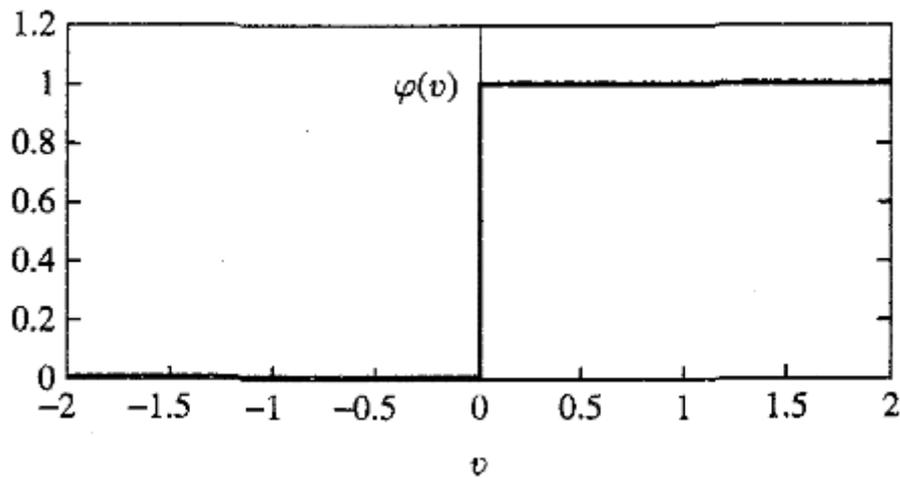


Figure 2.4: The threshold function (Haykin, 1999).

**2. Piecewise-Linear Function:** For this function we have this equation:

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

This is a piece-wise linear equation that functions as displayed in Figure 2.5.

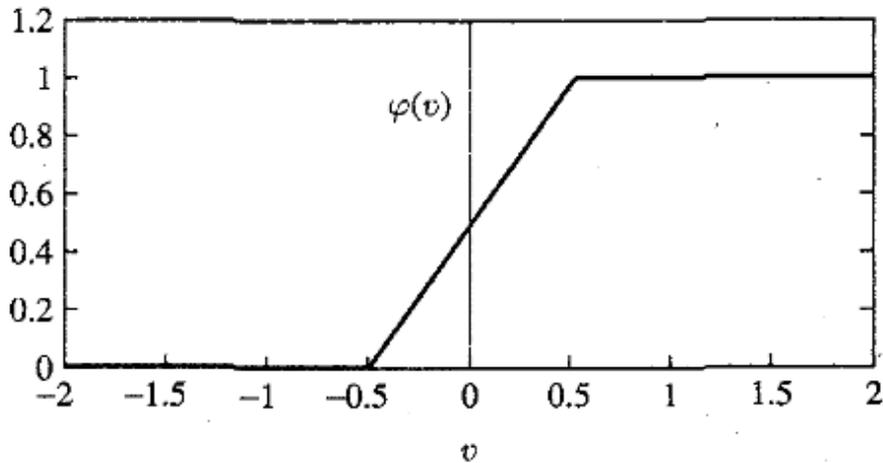


Figure 2.5: Piecewise-linear function (Haykin, 1999).

**3. Sigmoid function:** This function is the one that is predominantly used in the construction of MLP ANNs. It is defined as a strictly increasing function that displays a graceful balance between linear and nonlinear behavior. Therefore this makes the sigmoid function a very close model to the biological neurons function. The logistic function is an example of the sigmoid function and is defined by

$$\varphi(u) = \frac{1}{1 + e^{-au}}. \quad (2.5)$$

The parameter  $a$  is the slope parameter whose value produces sigmoid functions of different slopes. When  $a$  approaches infinity the sigmoid function turns into the threshold function. While the threshold function assumes values of 0 and 1, the sigmoid function assumes a continuous range of values from 0 to 1.

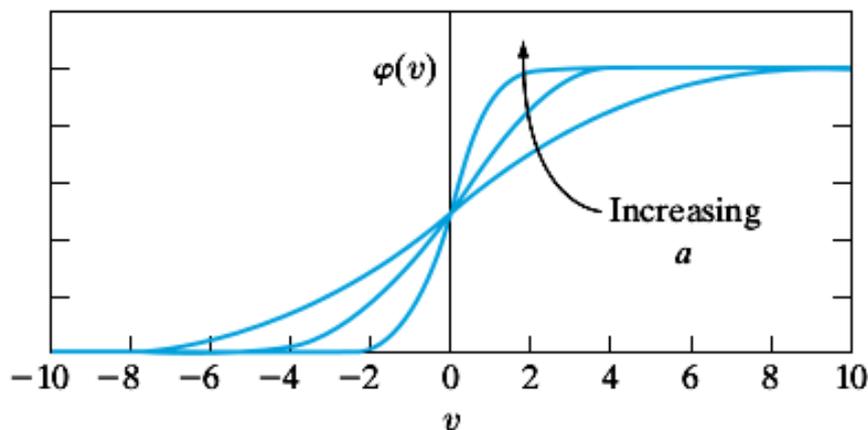


Figure 2.6: Sigmoid function for varying parameter  $a$  (Haykin, 1999).

In some cases, the hyperbolic tangent function can be used as a sigmoid function. This will produce output values in the range from -1 to 1

$$\varphi(u) = \tanh(u). \tag{2.6}$$

### 2.1.5 Artificial Neural Network (ANN) architectures

There are different architectures of ANNs, depending on the pattern of connections between the neurons. The main distinction is between (Krose et al., 1996):

- Feedforward networks: where we have a strictly feed forward data flow from input to output units. This has the possibility to extend over numerous layers of units but there is no feedback connections that allow data flow from output units to input units of the same or different layer.
- Recurrent networks: These types of networks contain feedback connections. They are complex networks, whose dynamic properties are important. Their activation values of their units change through a process, until they reach a stable state.

Depending on the number of layers, artificial neural networks can be categorized in:

- Single-Layer Feedforward Networks: This is the simplest form of a layered network, where we have an input layer of source nodes that feeds into an output layer of neurons but not vice versa (as it is feedforward). The term “single” refers to the output layer of computation nodes, as the input layers doesn’t perform any computations (Haykin, 1999).

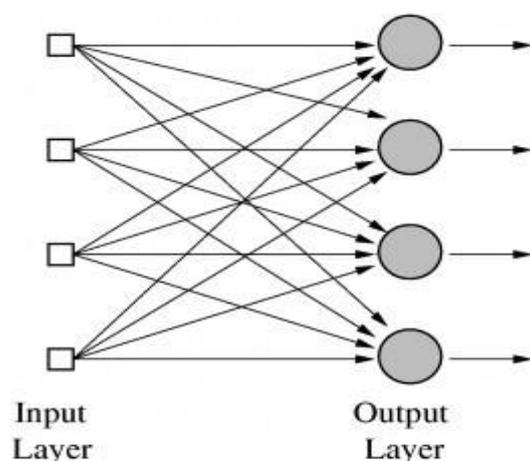


Figure 2.7: Single-layer feedforward network (Mas Flores, 2008).

- Multi-layer Feedforward Networks: This type of ANNs has at least one hidden layer of neurons, called hidden neurons or hidden units, between the input layer and the output layer. The function of the hidden units allows the

network to extract high order statistics from a large size of input nodes (Haykin, 1999).

### 2.1.6 Types of learning

An artificial neural network has to be trained in order for an application of a set of inputs to produce the desired set of outputs. There are many ways to set the strength of the connections between the units. One way is for the weights to be set explicitly, but the predominant way is for the neural network to be trained by feeding it with teaching patterns and letting it change its weights according to a learning rule (Kroset al, 1996). There are two main types of learning for ANNs, supervised and unsupervised learning:

- **Supervised Learning:** This form of learning is also called learning with a “teacher”. The term “teacher” refers to having knowledge of the environment in the form of a set of input-output examples. When exposed to a training vector (i.e. example), the neural network is provided with a desired response for it from the “teacher”. As a result, the neural network’s parameters are adjusted under the influence of the training vector and error signal (the difference between the desired response and the actual response). After a large number of repetitions of the adjustment the neural network can emulate the teacher and can deal with the environment on its own (Haykin, 1999).
- **Unsupervised Learning:** In this form of learning, in contrast with supervised learning, the training process consists only of input data. The neural network discovers some of the properties of the dataset and learns to display them in its output. Regarding what exactly these properties are, depends on the type of neural network and learning rule (Gopal, 2017). For example, a competitive learning rule may be used, which describes a neural network that consists of an input layer and a competitive layer. After the available data are received through the input layer, the competitive layer, which consists of neurons that compete with each other, responds to features contained in the input data (Haykin, 1999).

### 2.1.7 History of ANN

The following events and their dates are considered the major milestones in the history of the field of ANN.

In 1943 the pioneering work of McCulloch psychiatrist and neuroanatomist and Pitts a mathematician introduced a logical calculus of neural networks, uniting in this way

the studies of neurophysiology and mathematics(McCulloch et al., 1943). Their work was inspired by the structure and operation of the human biological nervous system and in particular by the attributes of the neuron. In reference to the complicated way the human brain works, they demonstrated that a network composed of a sufficient number of units operating in a parallel fashion through synaptic connections could perform any computable task.

In 1948, in his book “Cybernetics”, Wiener is the first to formulate the connection between statistical mechanics and learning systems(Wiener, 1948). Wiener’s book was then followed in 1949 by Hebb’s “The Organization of Behavior” (Hebb, 1949), in which he first presents a learning rule based on the capacity of the brain for constant change as it addresses various tasks. Drawing from Ramon y Calais early work, Hebb goes on to state that whenever connections of synapses between neurons are activated, these synapses between are reinforced. As a result, the network develops and adapts to the learning model that is introduced to it. Hebb’s book eventually led to the development of computational models of learning and adaptive systems (Haykin, 1999). A first attempt to use computer simulation to test Hebb’s neural theory of learning is presented in the paper by Rochester, Holland, Haibt, and Duda in 1956(Rochester et al., 1956).

In 1958, Rosenblatt introduced a novel method of supervised learning, the ‘perceptron’, which offered a new approach to pattern recognition. Rosenblatt first developed a network that could perform multiple tasks(Rosenblatt, 1958). In 1960Widrow and Hoff introduced the ‘least mean-square(LMS algorithm)’, which they used first to develop the Adaline network, involving an adaptive linear element and secondly the Madaline network, a multiple-adaline structure(Widrow et al., 1960).

In 1961, a paper by Minsky under the title ‘Steps toward Artificial Intelligence’ was published(Minsky, 1961). A large section of this paper refers to material that is now recognized as relating to neural networks. Later in 1967, in his book “Computation: Finite and Infinite Machines” (Minsky, 1967)Minsky places the work presented by McCulloch and Pitts in 1943 in the context of the theory of computation and automata theory. However, in 1969 Minsky and Papert set a limit to the expectations that had arisen in the area of neural networks, because of the work on the perceptron model. They used mathematics to demonstrate that single-layer perceptrons cannot perform certain computations(Minsky et al., 1969).

The above was one of the main factors that discouraged subsequent in the area of ANN. Interest in the field was revived in the 1980s. Hopfield demonstrated in 1982 that a neural network (dynamically stable networks) can be used to store information (Hopfield, 1982).

In 1986, Rumelhart, Hinton and Williams developed the ‘back-propagation algorithm’, which has proved to be the most popular learning algorithm for the training of multilayer perceptrons (MLPs) (Rumelhart et al., 1986). Their work represented a step beyond the perceptron model and even though the algorithm had been described earlier by Werbos in 1974 (Werbos, 1974), Rumelhart, Hinton and Williams have been accorded the credit for introducing it to machine learning and for demonstrating in strict mathematical terms how it could be used.

It is worth to be noted that after a long time there is resurgence and renewed interest in ANNs. The existing hardware allows much faster computations and GPUs have proved to be very successful in the training of ANNs. The combination of large amounts of stored data, high infrastructure speed and capable hardware have put ANNs in the forefront of Artificial Intelligence (AI) under the term deep learning (from deep neural networks).

### 2.1.8 Advantages of the Artificial Neural Networks (ANNs)

ANNs rely on their parallel distributed structure and ability to learn from data and generalize for their effectiveness. By generalizing, ANNs can produce outputs by inputs that were not part of their training. Therefore, they can solve complex and large scale problems that numerical models have big difficulty dealing with. More specifically, ANNs have the following benefits and capabilities as presented in (Haykin, 1999):

**Nonlinearity:** Artificial neurons can be nonlinear, making the entire network nonlinear. The ANN can then be particularly effective when the underlying mechanism of the input signal is inherently nonlinear (Haykin, 1999).

**Input-Output Mapping:** Neural networks train themselves through the concept of supervised learning, where their synaptic weights change values as the training process takes place. By reapplying the previous training examples through a different order the network constructs an input-output mapping for the problem. This constitutes a “nonparametric statistical inference” approach to the problem, which means that no prior assumptions are made on a statistical model for the input data.

**Adaptivity:** Neural networks adapt their synaptic weights to their environment. Even when there are changes in an environment the neural network has been trained in, it can be easily retrained. Furthermore, if the environment is non stationary and its parameters change with time, the neural network can be designed to change its synaptic weights continuously in real time.

**Evidential Response:** This refers to the ability of neural networks to provide information about the confidence in decisions made, mainly in classification tasks.

**Contextual information:** Since every neuron in the network is affected by the activity of the rest of the neurons, and knowledge is represented in its structure and activation state, neural networks can deal with contextual information naturally.

**Fault tolerance:** Neural networks have the potential to be inherently fault tolerant in the event of damage. The reason for this is when the neurons and their collective links are damaged, stored information is only impaired in quality because of the distributed nature of the network.

**VLSI Implementability:** Thanks to the massively parallel nature of neural networks certain tasks can be performed with incredible speed. This makes very large scale integrated technology a suitable way to implement neural networks in order to capture complex patterns in a very hierarchical fashion.

**Uniformity of Analysis and design:** It refers to neural networks ability to function universally as information processors. Neurons are a common ingredient to all neural networks. This means that it is possible to share learning algorithms between different fields and applications of neural networks.

**Neurobiological analogy:** Neuroscientists study ANNs in order to interpret neurobiological phenomena while computer scientists and AI engineers study neurobiology for new methods to solve complex problems.

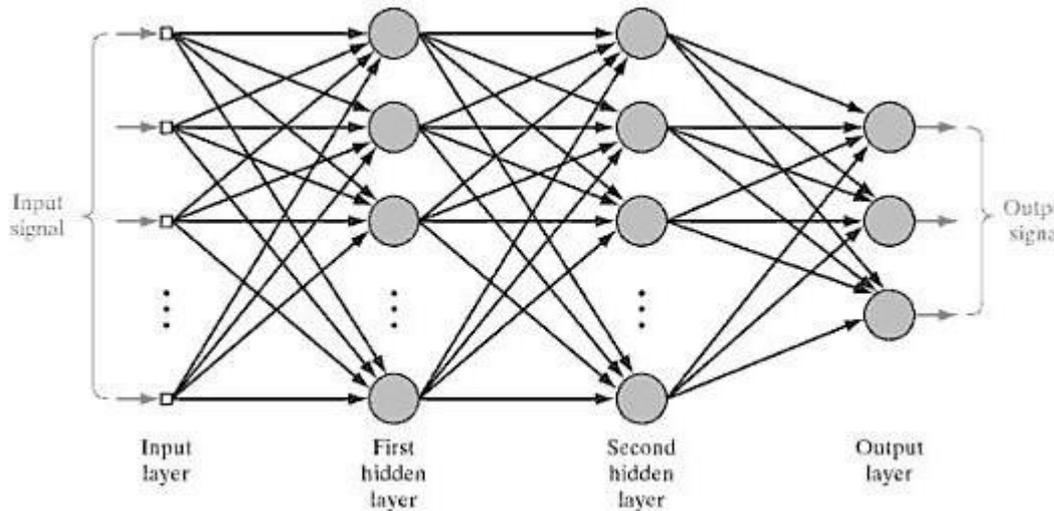
Next the two main types of ANNs will be analyzed and their advantages and disadvantages will be compared.

### 2.1.9 Multilayer Perceptron (MLP) networks

Multilayer perceptrons (MLPs) are the most commonly used ANNs. They have been applied successfully to solve complex and diverse problems by being trained in a supervised manner through the highly popular algorithm, the error back-propagation algorithm. Through it, learning occurs in the network by modifying connection weights after each training vector is processed, based on the amount of error in the output compared to the expected results (Gopal, 2017). There are three distinctive characteristics that define MLPs (Haykin, 1999):

1. The neuron model of the network contains a nonlinear activation function, most commonly a sigmoid logistic function. The use of the logistic function as mentioned before is biologically motivated, since it attempts to imitate the refractory phase of real neurons.

2. The neural network contains one or more layers of hidden neurons that are neither input nor output layers of the network. The hidden neurons allow the network to learn complex tasks by extracting progressively more meaningful features from the input data.
3. The neural network displays a high degree of connectivity.



**Figure 2.8: Multi-layer perceptron (Haykin, 1999).**

In the following paragraphs the basic idea behind the backpropagation algorithm will be presented.

The idea behind the procedure is that when a learning pattern is clamped, the activation values are propagated to the output units and the actual network output is compared with the desired output values. Through this we end up with an error  $e_o$  in each output unit  $o$ . The error value  $e_o$  must be brought to zero. The easiest way to achieve this is to change the connections in the neural network in such a way that in the next iteration the error  $e_o$  will be zero for this particular pattern zero (Kroset al, 1996).

But this step only addresses the hidden to output unit weights. In order to change the input to hidden unit weights there is a second phase which involves a backward pass through the network, during which the error signal is passed to each unit in the network and appropriate weight changes are calculated(Kroset al, 1996).

Since for our application we won't use multi-layer perceptron's, we won't go in depth regarding their function and equations. Instead, the properties and functions of Radial Basis Function Networks (RBFNs) will be presented in detail.

### 2.1.10 Radial basic function networks (RBFN)

RBFNs are artificial neural networks whose activation function is a radial basis function.

#### Radial basis functions

Radial basis functions consist a special class of functions. Their main feature is that their response decreases (or increases) monotonically with the distance from their central point (usually a Euclidian distance function is used). The center, the distance scale and the precise shape of the radial basis function are parameters of the model (all fixed if it is linear). A typical radial basis function is the Gaussian, which, in the case of scalar input is as follows:

$$h(x) = \exp\left[-\frac{(x - c)^2}{r^2}\right] \quad (2.7)$$

Its parameters are its center  $c$  and its radius  $r$  (Orr, 1996). A Gaussian radial basis function monotonically decreases with distance from the center. In contrast, a multi-quadric radial basis function monotonically increases with distance from its center, which in the case of a scalar input is given as:

$$h(x) = \frac{\sqrt{r^2 + (x - c)^2}}{r} \quad (2.8)$$

Gaussian-like radial basis functions are local (give significant response only in a neighborhood near the center) and are more commonly used than multi-quadric type radial basis functions, which have a global response. They are also closer to biology because their response is finite (Orr, 1996).

Radial basis function networks (RBFNs) are composed of a simple three layer architecture. The input layer propagates input values to a single hidden layer, which features neurons with radial basis activation functions. In the output layer, each neuron receives a linear combination of the output of hidden neurons. In the case of one output node, the global nonlinear function computed by the network can be expressed as a linear combination of  $m$  basis functions associated with each hidden layer neuron (Binaghi et al., 2013).

$$f(x) = \sum_i^m w_j h_j(x) \quad (2.9)$$

Here  $\mathbf{x}=[x_1,\dots,x_k]^T$  is the K-dimensional input vector,  $w_j$  are the weighing coefficients of the linear combination and  $h_j(\mathbf{x})$  represents the output of the Gaussian shaped basis function, with scale factor  $r_j$ , associated with the j-th neuron in the second layer. The response of j-th neuron decreases monotonically with the distance between the input vector  $\mathbf{x}$  and the center of each function  $c_j=\{c_{1j},\dots,c_{kj}\}$  (Binaghi et al., 2013).

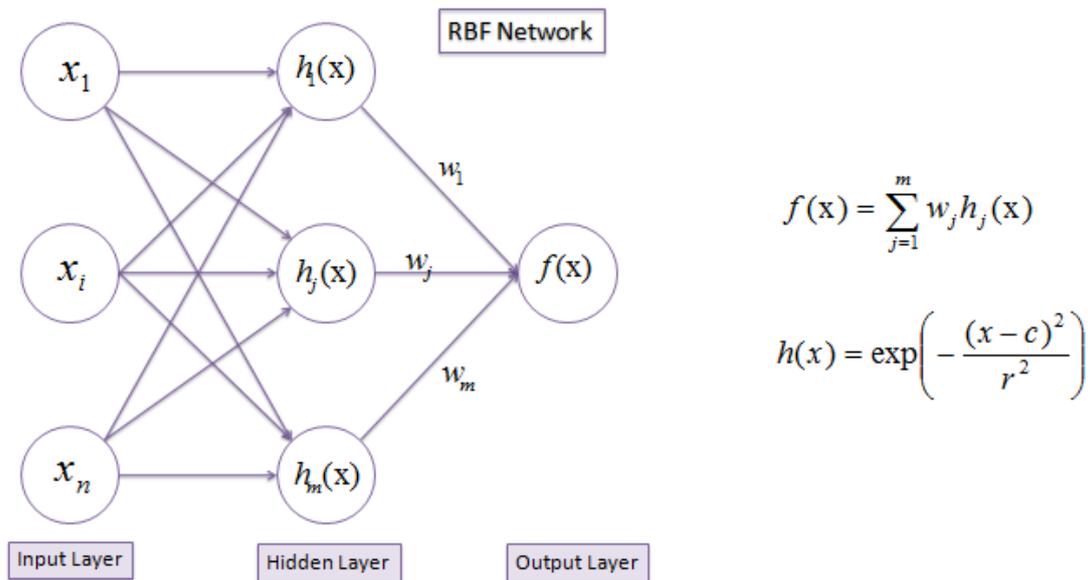


Figure 2.9: Radial basis function network architecture (Chandradevan, 2017).

As the radial basis function network (RBFN) is being trained, it learns an approximation for the true input-output relationship, based on a given training set of examples constituted by N input-output pairs  $[x_i, y_i], i = 1, \dots, N$ . Following Moody and Darken (1989), the training scheme is two phased:

1. Phase one is unsupervised and decides values for  $c_j, j = 1, \dots, M$
2. Phase two solves a linear problem to find values for  $w_j, j = 1, \dots, M$

The configuration of the model requires two parameters to be inserted.

1. The number  $M$  of the first level local processing units
2. The number  $p$  of the  $p$ -means heuristic used to determine the scale factor  $r_j, j = 1, \dots, M$  of radial basis functions associated with first level processing units.

For the second phase, having model parameters, for  $c_j, j = 1, \dots, M, r_j, j = 1, \dots, M$  known, the weights  $w_j, j = 1, \dots, M$  are computed, minimizing the difference between predicted output and real output by least mean squares, computed through the pseudo inverse

$$\mathbf{w} = (H^T H)^{-1} H^T \mathbf{y} = H^+ \mathbf{y} \quad (2.10)$$

where:

$$H = \begin{bmatrix} h_1(x_1) & \cdots & h_M(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_M(x_N) \end{bmatrix} \quad (2.11)$$

Here is the vector of output data:

$$\mathbf{y} = [y_1, \dots, y_N] \quad (2.12)$$

And the following are second level weights:

$$\mathbf{w} = [w_1, \dots, w_M]^T \quad (2.13)$$

The trained network is tested using a proper set of examples that haven't been used in training (Binaghiet al., 2013).

### 2.1.11 Comparison of MLP and RBFN

RBFNs and MLPs are examples of nonlinear layered feedforward networks. They are both universal approximators. It is, therefore, not surprising to find that there always exists an RBF network capable of accurately mimicking a specified MLP, or vice versa. However, there are differences between them (Haykin, 1999):

First of all, an RBFN has a single hidden layer, whereas an MLP may have more than one hidden layers. Next, the computation nodes of an MLP, located in a hidden or an output layer, share a common neuronal model, while on the other hand the computation nodes in a hidden layer of a RBF network are quite different and serve a different purpose from those in the output layer of the network. Furthermore, the hidden layer of a RBF network is nonlinear, whereas the output layer is linear. However, the hidden and output layers of an MLP, used as a pattern classifier, are usually nonlinear. When the MLP is used to solve nonlinear regression problems, a

linear layer for the output is usually the preferred choice. Another difference is the activation function in the hidden units, where RBFNs compute the Euclidean norm (distance) between the input vector and the center of that unit, while MLPs compute the inner product of the input vector and the synaptic weight vector of that unit. Finally MLPs construct global approximations to nonlinear input-output mapping while on the other hand RBFNs using exponentially decaying localized nonlinear functions construct local approximations to nonlinear input-output mappings. For this reason, for the approximation of nonlinear input-output mapping, MLPs may require a smaller number of parameters than the RBF network for the same degree of accuracy (Haykin, 1999).

For our application a RBFN will be used, as we consider the approach of local approximation better suitable to the parameters of our problem, while the spatial nature of the problem at hand is better-fitted to RBFNs.

## 2.2 PAST CASES

### 2.2.1 Uses of multilayer perceptron networks for spatial predictions

Multilayer perceptron networks have been in use for spatial predictions since the 1990s. A common method is combining the neural network with Geographical Information Systems (GIS). In general, the integration of neural networks and maps includes the following steps:

In order to train the MLP, the sample data for supervised training is often subdivided into *training*, *validation*, and *testing* sets. The distinctions between these subsets are crucial. Training data is a set of examples used for learning that is to fit the parameters [weights] of the classifier. Validation data is a set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network, while testing data is a set of examples used only to assess the performance [generalization] of a fully specified classifier (Ripley, 1996).

Determination of the networks structure: Most commonly, it will be a three layer (input, hidden, output) feed forward neural network. The input units in such an application are the input variables, so their number is the number of input units. The output units and their number are the output variables depending on the application (Gopal, 2017). The number of hidden units can be determined through trial and error experimentation. If the hidden units are too few the neural network will have low accuracy in its function. On the other hand a large number of hidden units can cause the neural network to over fit the training data, resulting in a reduction of generalization accuracy (Gopal, 2017).

Learning rule: Back-propagation (or other methodologies).

Transfer function: The activation function used is nonlinear, most commonly the tangent or sigmoid function.

### **Spatial predictions of groundwater pollution by integrating a MLP network and GIS**

#### Case 1: Groundwater simulation of Mazandaran plain, Iran

In this case (Gholami, 2016) the Ground Water Quality Index (GWQI) data from 85 drinking wells was divided into training data, cross validation data and testing data. The neural network structure was determined by trial and error and was evaluated using the testing data.

Then, the model was applied on the study area to calculate GWQI in sites without water quality testing. The study area which is 10000 km<sup>2</sup> is divided in 10000 pixels sized 1x1 km. Each pixel has the input values of groundwater quality factors. The pixel coordinates are added into the GIS medium and for each pixel we add its input values in the ANN to calculate the GWQI. The actual GWQI values from the 85 drinking water wells were overlain in GIS to evaluate the accuracy by the comparison. In this case the results were very accurate.

#### Case 2: Prediction of nitrate pollution in groundwater in a South Rhodope aquifer (Thrace, Greece)

In this case (Gemitzi, 2008) a MLP was used with a back-propagation algorithm for classification. Testing pixels correspond to sample points, where nitrate concentration in groundwater was measured. Every sampling location is a cluster of nine pixels, where the central one is the sampling point and the eight neighboring are assumed to be the same as the central pixel. A feature vector is associated to each pixel. So for the 214 boreholes that were used as sampling locations results a total of 1926 pixels where nitrate concentration was measured to be used for training and testing the network.

The GIS software Idrisi Andes was used with integration with the MLP. Each pixel was classified into one of the four nitrate pollution categories. Half of the data set was used for training and the rest for testing the network, to validate the results. The neural network was analyzed multiple times to determine the networks architecture and learning rate, until there was a higher than 70% accuracy. The results of this application showed great promise with the model being successful in 74% of the cases.

#### Conclusions – MLP for spatial predictions

We may conclude that MLPs provide a very good alternative to conventional modelling techniques, regarding spatial predictions. While dealing with nonlinear problems MLPs offer two principal advantages.

The first is their adaptive nature, as MLPs learn from past data and adjust their parameters (the connecting weights) automatically. The second advantage is that no physical model is needed, which allows them to estimate functions from a small amount of training data and then construct numerical estimators that can be applied to many physical problems. Furthermore, they have a low sensitivity to uncertain parameters.

A MLP that has been trained for a certain model can be applied to similar models without a big reduction in accuracy. However if there are considerable differences, it is better to be retrained.

### 2.2.2 Uses of radial basis function networks for spatial predictions

Radial basis function networks (RBFNS) are a very flexible method for spatial predictions. Similarly to MLPs, they provide good results and can be used by inexperienced users compared to conventional modelling and statistics techniques.

Determination of the networks structure: RBFNs for spatial predictions most commonly have 3 layers. The input layer has as many units as the number of input features. Depending on whether we have a regression or classification task, the RBFN has 1 output unit or multiple, according to the number of classes of the problem. The number of hidden units with radial basis functions is determined with trial and error, or through an optimization procedure.

Radial function: Usually RBFNs use the Gaussian radial basis function for spatial predictions. This is because the Gaussian radial basis function gives a significant response in the area close to its center compared to global radial basis functions whose response increases monotonically with the distance from the center.

#### Case 1: Snow cover thickness estimation using RBFNs

In this case (Binaghi et al., 2013) a RBFN is used to estimate snow thickness with both regression and classification functions. The model is based on a minimal set of topographic data from a limited number of stations in the Italian Central Alps. Eleven input variables were used (equaling to 11 input neurons for both tasks), 8 of which are climatic and 3 geographic ones.

For the regression task the architecture was determined by varying the temporal window  $T$  and testing with different numbers of hidden  $M$  units with radial basis functions. Values of 100, 250, 500, 600 and 750 were assumed. The optimal values were determined to be  $T=45$  with  $M=500$ . There is one output neuron for the regression task. The following table includes the regression results depending on the number of centroids  $M$  in the form of RMSE, NRMSE AND MAE.

#Centroids	RMSE	NRMSE	MAE
100	26.22	6.60%	15.18
250	22.31	5.61%	12.03
500	18.20	4.58%	9.58
600	17.63	4.44%	9.28
750	16.99	4.27%	8.74

**Table 2.1 Regression results(Binaghi et al., 2013).**

For the classification task, 4 classes were used depending on snow thickness. Several configurations were considered but again the optimal architecture was shown to be T=45 and M=500, but this time with 4 output neurons, one for each class. To produce a snow cover map, spatial interpolation of input climatic variables was used and then use of the RBF network to compute the corresponding predicted snow cover value for each input value.

Date	Methods	Overall accuracy						Mean	Far
		0–1000	1000–1300	1300–1600	1600–1900	1900–2200	>2200		
11-Feb-03	RBF	99.67%	91.98%	93.60%	67.60%	60.77%	87.44%	83.51%	15.62%
	iDW	7.46%	12.90%	15.45%	29.50%	59.86%	85.58%	35.12%	31.10%
	Spline	85.44%	53.35%	45.38%	52.01%	67.17%	74.20%	62.92%	15.30%
25-Feb-03	RBF	99.98%	96.32%	98.79%	72.89%	46.92%	86.17%	83.51%	20.63%
	IDW	7.47%	9.68%	11.07%	17.95%	41.86%	80.64%	28.11%	28.66%
	Spline	84.64%	50.03%	42.46%	44.18%	53.45%	70.85%	57.60%	16.68%
29-April-03	RBF	99.97%	98.31%	99.92%	91.96%	61.96%	71.93%	87.34%	16.38%
	IDW	60.41%	35.13%	30.97%	27.23%	33.62%	58.42%	40.96%	14.56%
	Spline	49.92%	46.93%	40.46%	38.76%	39.39%	53.46%	44.82%	6.17%

**Table 2.2 Overall accuracy for 6 elevation ranges (Binaghi et al., 2013)**

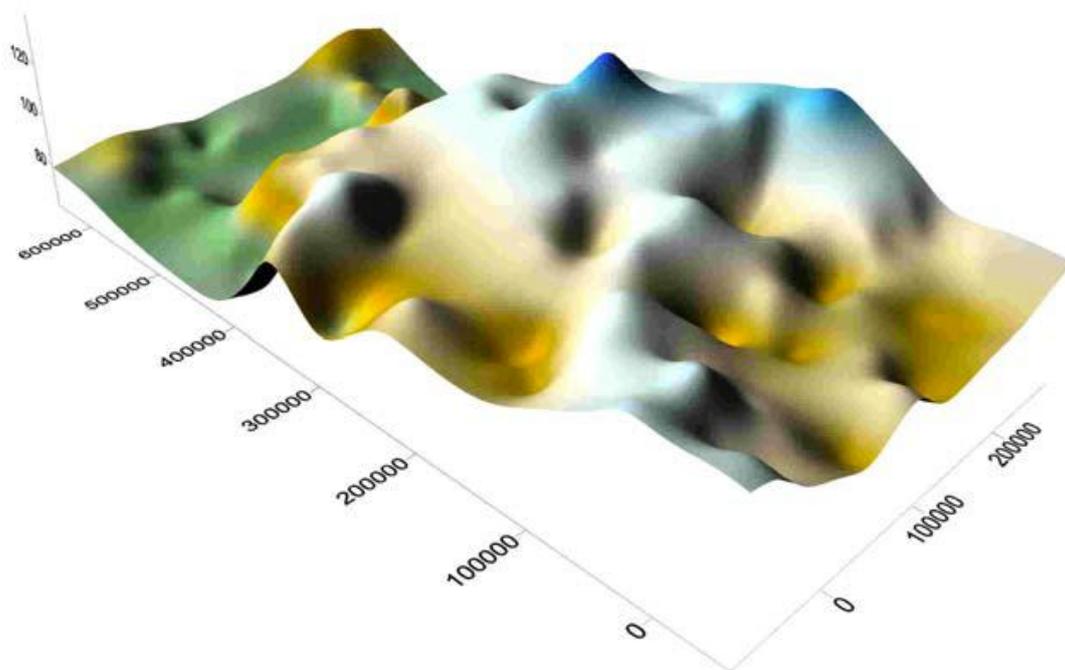
Compared to conventional deterministic methods, such as Spline and Inverse Distance Weighting IDW, the RBFN performed better with a 72% accuracy compared to 60% of the other 2 methods. However, regarding spatial predictions from climatic data RBFNs can overestimate values, because of the microclimatic conditions around the stations that provide data. For example, during the melting season and early autumn, when the snow distribution is extremely inhomogeneous.

### Case 2: SIC 2004

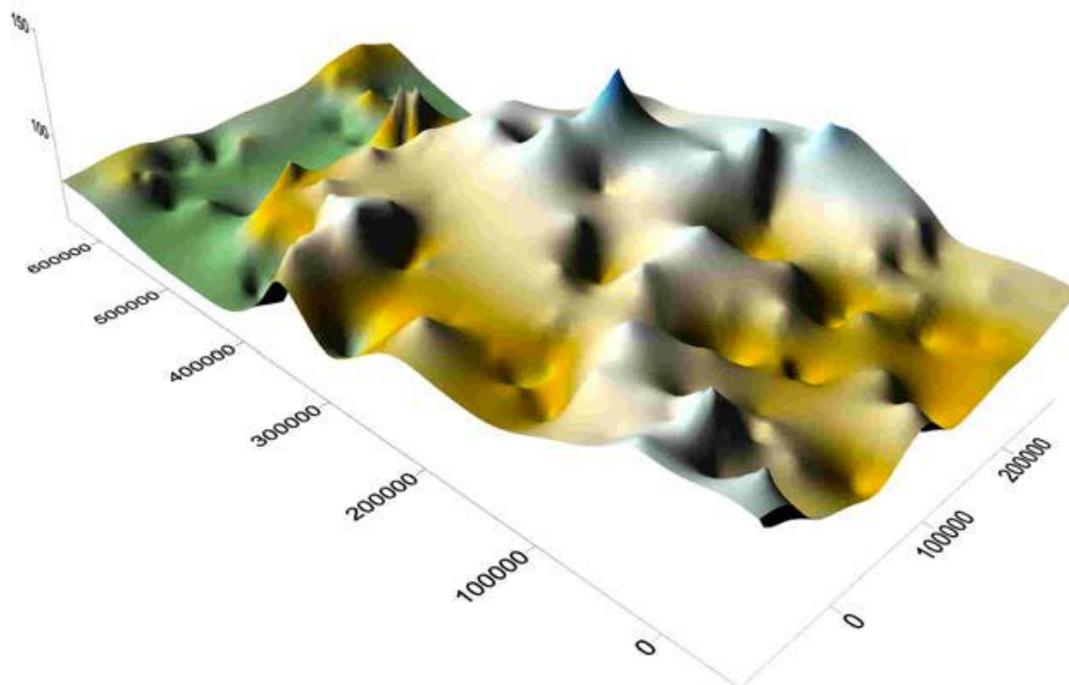
This case is a part of the Spatial Interpolation Comparison (SIC) exercises organized by the European Commission through the Joint Research Center (JRC). The case is presented in a paper comparing RBFs with geostatistics (Rusu et al., 2006).

Different methods were tested by estimating values of a variable at different locations with the help of a subset of observed measurements. After the measurements were made, the true values were given to compare the accuracy of each approach. The data of SIC 2004 were daily mean values of gamma dose rates measured in South West Germany in an area covering 400 kmx700km and 1008 monitoring stations.

The real 808 values were used for comparison with the interpolation results. The results of the RBFN and by kriging were presented in a 3D graph



**Figure 2.10 Modeling results calculated by the RBFN(Rusu et al., 2006).**



**Figure 2.11 Modeling results calculated by kriging (Rusu et al., 2006).**

The indicators (ME, MAE, PMAE, MIN, MAX, PMIN, PMAX, PEAR) of the comparison between the RBFN and kriging are different types of errors and were presented in a table as well as the comparison between different functions (MQ, MLOG, INVMQ, SPLINE) for the RBFN.

	ME	MAE	PMAE	MIN	MAX	PMIN	PMAX	PEAR
RBF	-1,41	9,15	9,2	-61,14	44,53	0	53,91	0,78
KRG	-1,31	9,28	9,34	-58,39	47,28	0,06	55,82	0,77

**Table 2.3 Statistics of RBF and kriging interpolation results (Rusu et al., 2006).**

	ME	MAE	PMAE	MIN	MAX	PMIN	PMAX	PEAR
MQ	-1,41	9,15	9,2	-61,14	44,53	0	53,91	0,78
MLOG	-1,19	9,82	9,94	-71,82	34,15	0	45,3	0,77
INVMQ	-12,18	199,4	190,78	-2017	1446	0,42	2004	-0,01
SPLINE	-2,94	53,23	54,59	-635,54	704,15	0,03	706,15	0,17

**Table 2.4 Statistics of interpolation results using various RBF types (Rusu et al., 2006).**

First, by examining the 3D models, the study concludes that kriging highlights more details, but the differences are minor and regard the different level of smoothness.

Referring to the results presented in the tables, the study concludes that for RBFNs the choice of function is important. Multiquadric-type radial functions are more suited for sparse data, because they offer a more global response compared to Gaussian type functions. More importantly the study concludes that although the results and time required to execute are similar for RBFNs and kriging, RBFNs are much easier and simpler to use. All that is required for RBFNs is the choice of function and smoothing parameter, whereas for kriging a complex variogram modeling is needed.

Case 3: RBF network for spatial mapping of wave heights

This paper (Kalra et al.,2005) describes the use of RBFNs for remote sensed information, and more specifically satellite based information. The application consists of mapping remote sensed deep water waves with coastal waves. The paper focuses on the comparison between RBFNs and traditional feed forward backpropagation (FFBP) as well as the adaptive neuro-fuzzy inference system (ANFIS), which combines the advantages of ANNs and fuzzy logic. We will focus on the comparison between the RBFN and the FFBP.

The purpose of this research was to estimate the daily significant wave heights at a coastal location, depending on the wave heights collected by a satellite along its tracks. The common approach to project the satellite data collected at deeper oceanic locations to a shallow water site would be statistical regression. But it would suffer from modeling limitations and numerical approximations as well as its requirement of considerable exogenous information, time, and computer storage. Therefore ANNs were considered in the three different architectures.

The study area is located in the western side of the Indian coastline. The remote sensed information was collected by a radar altimeter aboard a satellite. The number of input and output nodes for the FFBP and ANFIS was 21 and for the RBF network 21 and 1. The number of hidden units for the FFBP was decided by trials to be 9 and for the RBF network 13. To calibrate the RBF network, nearly 80 percent of the available data was used and the rest was used to test or validate it.

Error measure	RBF	ANFIS	FFBP(Algorithm-RP)
R	0.90	0.82	0.87
MAE	0.26	0.32	0.34
RMSE	0.40	0.56	0.48
SI	0.28	0.40	0.34
MSRE	0.11	0.16	0.12

**Table 2.5 Error measures (Kalra et al.,2005).**

The selection of the RBFN proved to be fitting for this application, despite the fact that in ocean engineering the FFBP is the first choice of most ANN applications. This

can be seen from the satisfactory error measures that were produced during the testing phase in the table of error measures. The RBF network has the highest magnitude of R and the lowest in the rest of the error measures.

The paper concludes that the RBFN was found to be more suitable for the mapping compared with ANFIS and the FFBP. The reason is because the satellite data need an unsupervised 'local tuning' before being used in the neural network calibration. The RBF network, as mentioned before in chapter 2.1.11, ensures localized functioning of the transfer function in contrast to the globalized one of the FFBP. The paper also highlights the significance of calibrating the network based on the specific dataset.

Case 4: Radial basis function neural network for hydrologic inversion: an appraisal with classical and spatial-temporal geostatistical techniques in the context of site characterization.

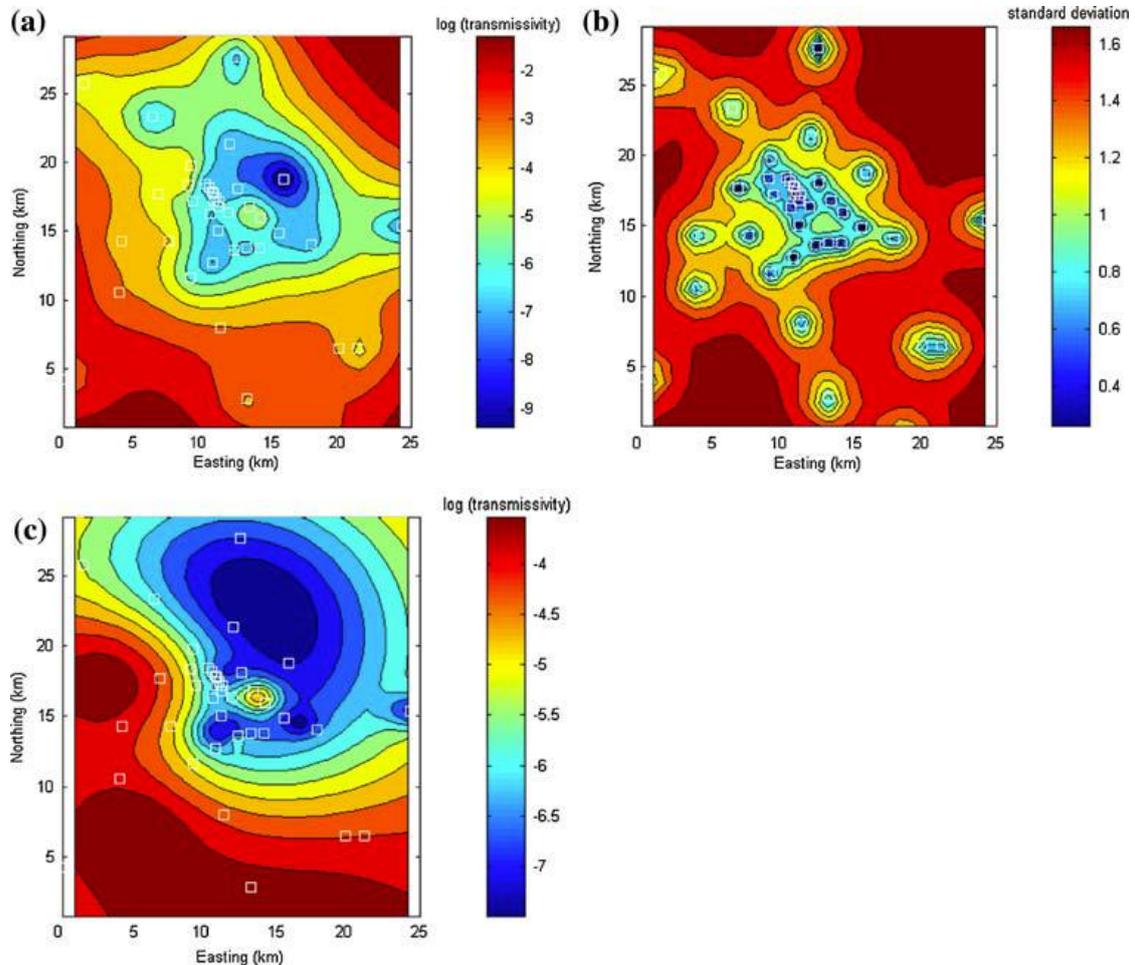
This work (Bagtzoglou et al., 2008) highlights the performance of RBF networks compared to classical and temporal geostatistical techniques, like Ordinary Kriging (OK) and Bayesian Maximum Entropy technique (BME) for mapping. The focus is around the use of three techniques (RBF, OK and BME) for spatial mapping and the consequential hydrologic inversion. The hydraulic conductivity (or transmissivity) is the hydraulic property of concern. In this work hydrological data are used, which makes it suitable also for our application. We will focus mainly on the parameters that optimize the RBF networks performance.

The study area is the Waste Isolation Pilot Plant (WIPP) facility located in the Chihuahuan Desert of southeastern New Mexico. In the WIPP area there are 21 data observations sites and by including the surrounding geophysical area of interest there are 41 data points. Hydraulic conductivity values were measured at all 41 sites and hydraulic head values at 32 sites. The hydraulic conductivity values were obtained from drill stem tests, slug tests and pumping tests of long periods. They are measured in ( $m^2/sec$ ) and vary by orders of magnitude so the  $\log_{10}$  of the raw values is used. The hydraulic conductivity data range from -10.1 to -2.9 and the hydraulic head values range from 905m to 938m. The values were drawn from a 27 km x 28 km area with the WIPP site being a square with a side of 6km.

The RBF network was trained using the 41 x and y direction spatial coordinates and the corresponding log-transmissivity data points. Convergence was obtained after 150-300 epochs, a fracture compared to back propagation ANNs with a tenth of the computational requirements. The k-means clustering approach was used to define the RBF centers. The RBF network produced results on 2287 structured grid notes.

These results are the data used for comparison of the RBF network to the geostatistical estimation technique (kriging). It was found that the RBF network can

produce a realistic model with limited data with very similar results to kriging and it is more accurate when the number of hidden nodes increases. The paper also tests coupling soft or inter dependent data with hard data for the RBF through the network output.



**Figure 2.12 Log-transmissivity interpolation at the WIPP site with: a) OK, b) kriging standard deviation, and c) RBF-ANN (Bagtzoglou et al., 2008).**

Regarding the parameters that optimize the RBF network performance, according to the paper, initializing the network with small random weights may be critical to convergence and repeatability. Furthermore it is important that the spreads of the basis functions to be established before training, because the spreads are altered only by training values that activate the function.

**Conclusions – RBFNs**

There are many advantages in using RBFNs for spatial predictions (Rusu et al., 2006). First, depending on the radial functions type, the RBF model may offer a localized response (therefore is able to identify the local characteristics of the surface to be modeled), or a global response (identifying this way the global characteristics of the surface to be modeled). RBFNs are exact interpolators, honoring the control points

when the point coincides with the grid node being interpolated. Furthermore, smoothing factors can be employed in order to reduce the effects of small-scale variability between neighboring data points (Rusu et al.,2006).

However, for good results the optimal architecture must be determined, which isn't always a straightforward procedure. It has to be tested with real data sets as seen in the cases studied.

## CHAPTER 3: METHODS AND APPLICATION

### 3.1 THE ANN ALGORITHM

#### 3.1.1 Characteristics

The algorithm used for the radial basis function network is written in FORTRAN programming language (Nikolos et al., 2007). In general a RBFN is a three layer, fully connected feed-forward network, which performs a nonlinear mapping from the input space to the hidden space ( $R^L \rightarrow R^M$ ), followed by a linear mapping ( $R^M \rightarrow R^1$ ) from the hidden to the output space ( $L$  is the number of input nodes,  $M$  is the number of hidden nodes, while the output layer has a single node).

The corresponding output  $yy(xx)$ , for an input vector  $xx = [xx_1, xx_2, \dots, xx_L]$  is given

$$yy(xx) = \sum_{i=1}^M w_i \cdot \varphi_i(xx) \quad (3.1)$$

where  $\varphi_i(xx)$  is the output of the  $i^{th}$  hidden unit

$$\varphi_i(xx) = G(\|xx - cc_i\|), \quad i = 1, \dots, M \quad (3.2)$$

The connections (weights) to the output unit ( $w_i, i = 1, \dots, M$ ) are the only adjustable parameters. The RBFN centers in the hidden units  $cc_i, i = 1, \dots, M$  are selected in a way to maximize the generalization properties of the network. The nonlinear activation function  $G$  in our case is chosen to be the Gaussian radial basis function

$$G(u, \sigma) = \exp\left(-\frac{u^2}{\sigma^2}\right) \quad (3.3)$$

where  $\sigma$  is the standard deviation of the basis function.

There are several strategies that can be adopted concerning the selection of the radial-basis functions centers in the hidden layer, while designing a RBFN: a) Random selection of fixed centers, which is the simplest approach and the selection of centers from the training data set is a sensible choice. b) Self-organized selection of centers. c) Supervised selection of centers, which is the most generalized form of a RBFN. The standard process is to select the input vectors in the training set as RBFN centers. In this case results  $M = NR$ , where  $NR$  is the number of training data. For large training sets (resulting in large  $M$  values) this choice is expected to increase storage requirements and CPU cost. Additionally, the  $M = NR$  choice could lead to

over-fitting and/or bad generalization of the network. The used solution is the selection of  $M < NR$  and consequently the search for sub-optimal solutions, which will provide a better generalizing capability to the network.

As far as training is concerned, there are two different approaches, the direct and the iterative learning. In our case the first approach was adopted. The direct learning process is based on a matrix formulation of the governing equations of RBF network. The presentation of the network with the  $NR$  input patterns allows the formulation of a  $(NR \times M)$  matrix  $H$ , which becomes square in the special case when  $NR = M$ . Each line in the interpolation matrix  $H$  corresponds to a learning example and each column to a RBFN center. The output unit values result in the form of the matrix product:

$$H(NR \times M) w(M \times 1) = yy(NR \times 1) \quad (3.4)$$

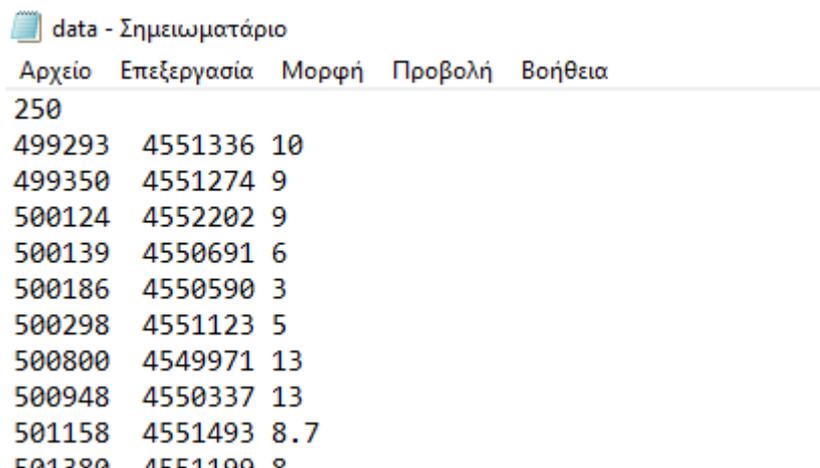
where  $yy$  is the desired output vector as provided by the training data set, and  $w$  is the synaptic weights vector, which consists of  $M$  unknowns to be computed. The inversion of  $H$  is accomplished through the Gram-Schmidt technique (Nikolos et al., 2007).

### 3.1.2 The Input files' structure

The ANNs configurations are modified through the following input files "case.txt", "data.txt", "evaluate.txt", "initialize.txt", "NeuralCombo.txt".

#### Input file "data.txt"

In this file, the size of the available dataset is inserted in the first line and, below that, the corresponding data (in the form of  $x, y, z$  coordinates) is inserted in three-column-format. The first two columns are the variables  $x$  and  $y$  corresponding to the geographic coordinates and the third column is the known measurement of the height of the aquifer  $z$  for the specific geographic position (Figure 3.1).



The screenshot shows a text editor window titled "data - Σημειωματάριο". The window contains a table with five columns: "Αρχείο", "Επεξεργασία", "Μορφή", "Προβολή", and "Βοήθεια". Below the header, the first line of the data file is "250". The following lines contain three columns of data: x, y, and z coordinates. The last line of data is "501158 4551493 8.7".

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
250				
499293	4551336	10		
499350	4551274	9		
500124	4552202	9		
500139	4550691	6		
500186	4550590	3		
500298	4551123	5		
500800	4549971	13		
500948	4550337	13		
501158	4551493	8.7		

**Figure 3.1: Input file “data.txt”.**

Input file “evaluate.txt”

Here, the x, y geographic coordinates, used for evaluation of the ANN, are inserted in two columns. The size of the dataset is in the first line. As it will be described later in this work, for some cases the same coordinates that were used for training will be also used for the evaluation procedure (Figure 3.2).

```

evaluate - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
|250
499293 4551336
499350 4551274
500124 4552202
500139 4550691
500186 4550590
500298 4551123
500800 4549971
500948 4550337
501158 4551493
501280 4551100
    
```

**Figure 3.2: Input file “evaluate.txt”.**

Input file “case.txt”

In this file, the architecture and the training characteristics of the neural network are defined, as it is described in Figure 3.3.

```

case - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
| Test Case main data
| =====
| popsize      =      250
| nvars        =         2
| NeuralTrain=      198
| NeuralTest   =         52
| globmax      = 25.00000000000000
| globmin      = 0.000000000000000E+000
| =====
| RBF_NEURAL
| NUM_LAYERS=         3
| Units(      0 )=         2
| Units(      1 )=      198
| Units(      2 )=         1
    
```

**Figure 3.3: Input file “case.txt”.**

- popsize: The size of the dataset.
- nvars: The number of input variables.
- NeuralTrain: Number of data that will be used for training.
- NeuralTest: Number of data that will be used for testing.
- Globmax: The maximum value of the ANN output.
- Globmin: The minimum value of the ANN output.
- NUM\_LAYERS: The number of layers of the radial basis function network.
- Units(0): Number of neural units of the first layer (Input Layer).
- Units(1): Number of neural units of the second layer.
- Units(2): Number of neural units of the third layer (Output Layer).

Input file “initialize.txt”

In this input file we define the number of input variables, the range of the single output variable and the range of input variables (Figure 3.4):

nvars: number of input variables

globmax: the maximum expected value of the single output variable (in our case the z-coordinate).

globmin: the minimum expected value of the single output variable (in our case the z-coordinate).

lobound(1), upbound(1): minimum and maximum values of the first input variable (in our case the x-coordinate).

lobound(2), upbound(2): minimum and maximum values of the second input variable (in our case the y-coordinate).

```

initialize - Σημειωματάριο
Αρχείο  Επεξεργασία  Μορφή  Προβολή  Βοήθεια
1
2  25.0  0.0          !nvars, globmax, globmin [zmax, zmin]
499293  519877        !lobound(1), upbound(1) [xmin, xmax]
4538828 4555898       !lobound(2), upbound(2) [ymin, ymax]
    
```

**Figure 3.4: Input file “initialize.txt”.**

### Input file “NeuralCombo.txt”

This file defines some options concerning the nature and training process of the utilized artificial neural network (Figure 3.5).

```

NeuralCombo - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
TRUE !NEURALBOOL
FALSE !MLPBOOL
TRUE !RBFBOOL
300 !MaxTrain1
300 !MaxTest1
0.8000 !Train_perc
1.0 !Centers_perc
    
```

**Figure 3.5: Input file “NeuralCombo.txt”.**

NEURALBOOL: always set as “TRUE”.

MLPBOOL: TRUE if the MLP network is used, FALSE otherwise.

RBFBOOL: TRUE if the RBF network is used, FALSE otherwise.

MaxTrain1: The maximum expected number of available training data.

MaxTest1: The maximum expected number of available testing data.

Train\_perc: Percentage of data used for training.

Centers\_perc: Percentage of training data used as RBF centers (usually set equal to 1.0).

### **3.1.3 The output files’ structure**

The results of the process are displayed at the output files “evalfit.txt” and “RBF\_error.txt” (or at “MLP\_error.txt”, when a multilayer perceptron network is used).

### Output file “evalfit.txt”

This textfile contains the results of the evaluation procedure (using the input data contained in evaluate.txt input file). The first two columns are the two input variables contained in evaluate.txt input file (the x- and y- coordinates where the network is evaluated) and the third column is the artificial neural network prediction of the single output variable at the corresponding positions (the aquifer height z at these coordinates) (Figure 3.6).

evalfit - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
	250			
	499293.000000000		4551336.00000000	8.27193182121846
	499350.000000000		4551274.00000000	8.37243501664489
	500124.000000000		4552202.00000000	6.09304553145193
	500139.000000000		4550691.00000000	8.76794195064576
	500186.000000000		4550590.00000000	8.90733081905637
	500298.000000000		4551123.00000000	7.97876479336992
	500800.000000000		4549971.00000000	9.07895026466576
	500948.000000000		4550337.00000000	8.71162906041718
	501158.000000000		4551493.00000000	8.40888767197612
	501380.000000000		4551199.00000000	8.71239374173456
	501416.000000000		4551427.00000000	8.73566983646015
	501467.000000000		4548607.00000000	9.13955109499511
	501542.000000000		4550078.00000000	8.63720524721430
	501543.000000000		4552518.00000000	8.63089402773767
	501554.000000000		4549517.00000000	8.55840435106074

Figure 3.6: Output file “evalfit.txt”.

Output file “RBF\_error.txt”

In this output file the errors of the radial basis function network predictions are listed. These errors are computed using the differences between the measured values and the corresponding predicted values of the aquifer height z. The first value is the mean training error (using the training data set), while the second value is the mean testing error (using the testing data set) (Figure 3.7).

RBF\_error - Σημειωματάριο

Αρχείο	Επεξεργασία	Μορφή	Προβολή	Βοήθεια
	3.261853760668297E-003		1.049440636440963E-002	

Figure 3.7: Output file “RBF\_error.txt”.

### Graphical presentation of the results

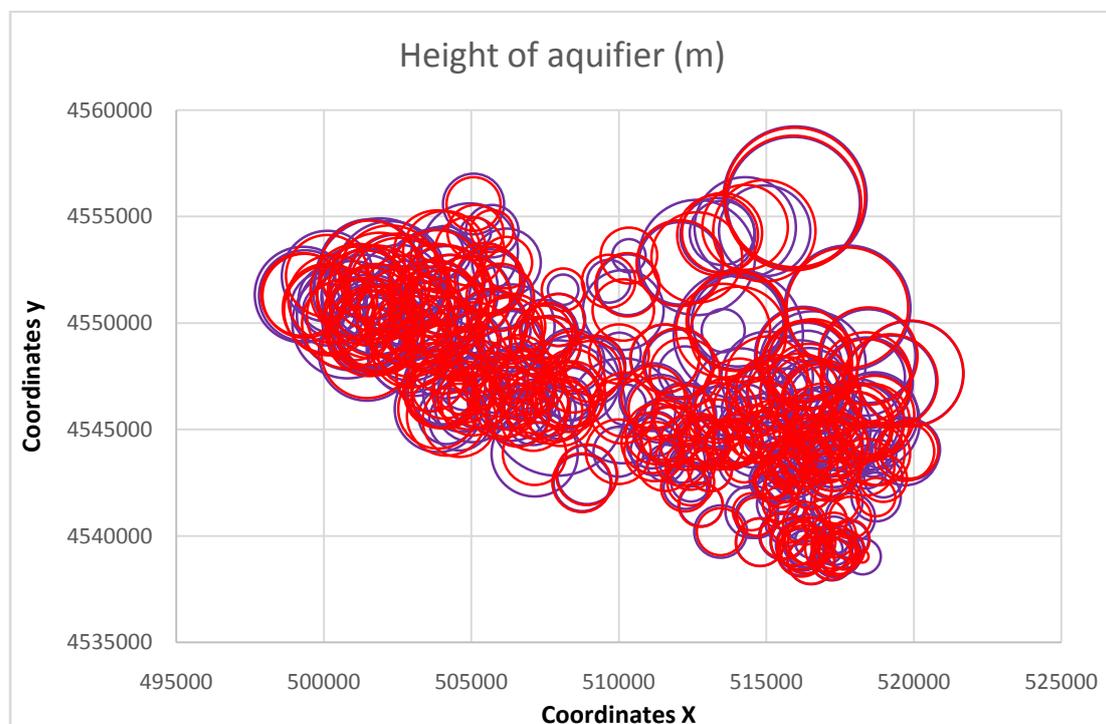
For the graphical presentation of the data and the results, Microsoft Excel was used. The x- and y-coordinates correspond to axis x and y respectively. The height of the aquifer z is displayed as the radius of a circle having its center in the position of the measurement (or prediction). The bigger the radius of the circle, the larger the height. Blue circles correspond to the measured aquifer height z, while red circles correspond to the predicted aquifer height, as it is calculated by the radial basis function network. The more the 2 types of color circles fit, the more accurate is the prediction of the neural network at the corresponding coordinates.

### 3.2 TESTING THE RBFN ALGORITHM

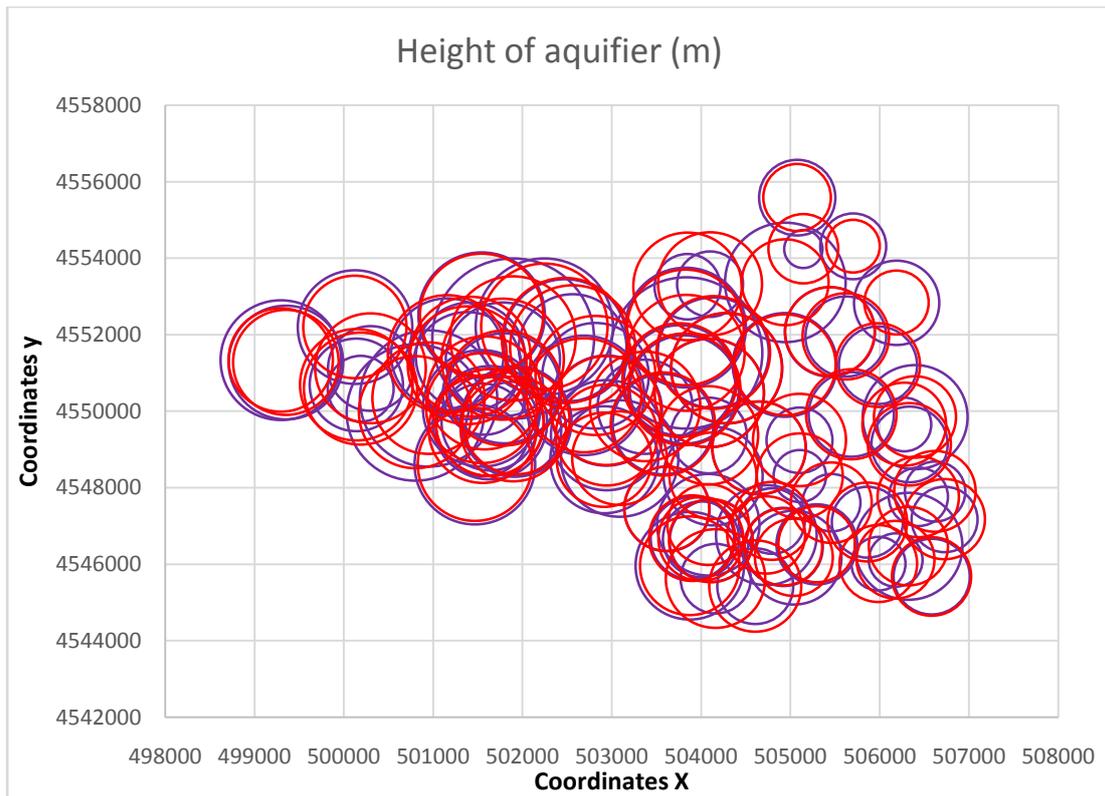
In this part, we will use the available measured data to train the ANN and evaluate its accuracy. On each test most of the data will be used to train the ANN and the rest are going to be used as testing data. On the first part we will use the entire set of data. Afterwards the data is split into two subsets at  $x=510000$  to determine whether the accuracy increases if we test the ANN on each subset separately.

#### 3.2.1 Testing with the entire set of data

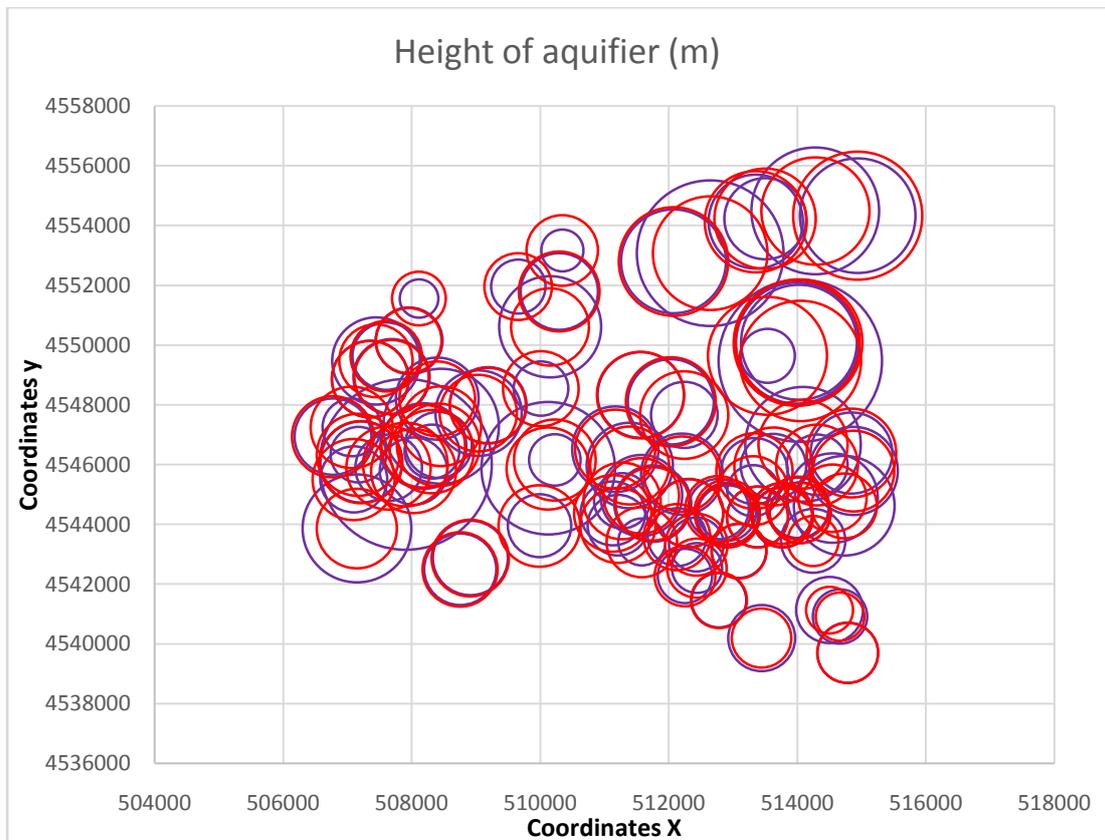
In our 250 size dataset, 98% of the data have been used for training and the rest 2% for testing. The corresponding training and testing errors are  $3.26 \cdot 10^{-3}$  and  $1.05 \cdot 10^{-2}$ . The prediction results for the entire data set are visualized at the following graph (Figure 3.8). In this graph the radius of each circle visualizes the aquifer's height at the corresponding position (at the center of the circle). Blue circles correspond to the measured aquifer's height, while red circles correspond to the predicted aquifer's height, as it is calculated by the radial basis function network (after being trained). As it can be seen in Figures 3.8 to 3.11, a good approximation of aquifer's height has been achieved for most of the available wells. We can also observe that the available wells' positions can be divided into two discrete groups, left and right of the axial position  $x=510000$  approximately (Figure 3.8).



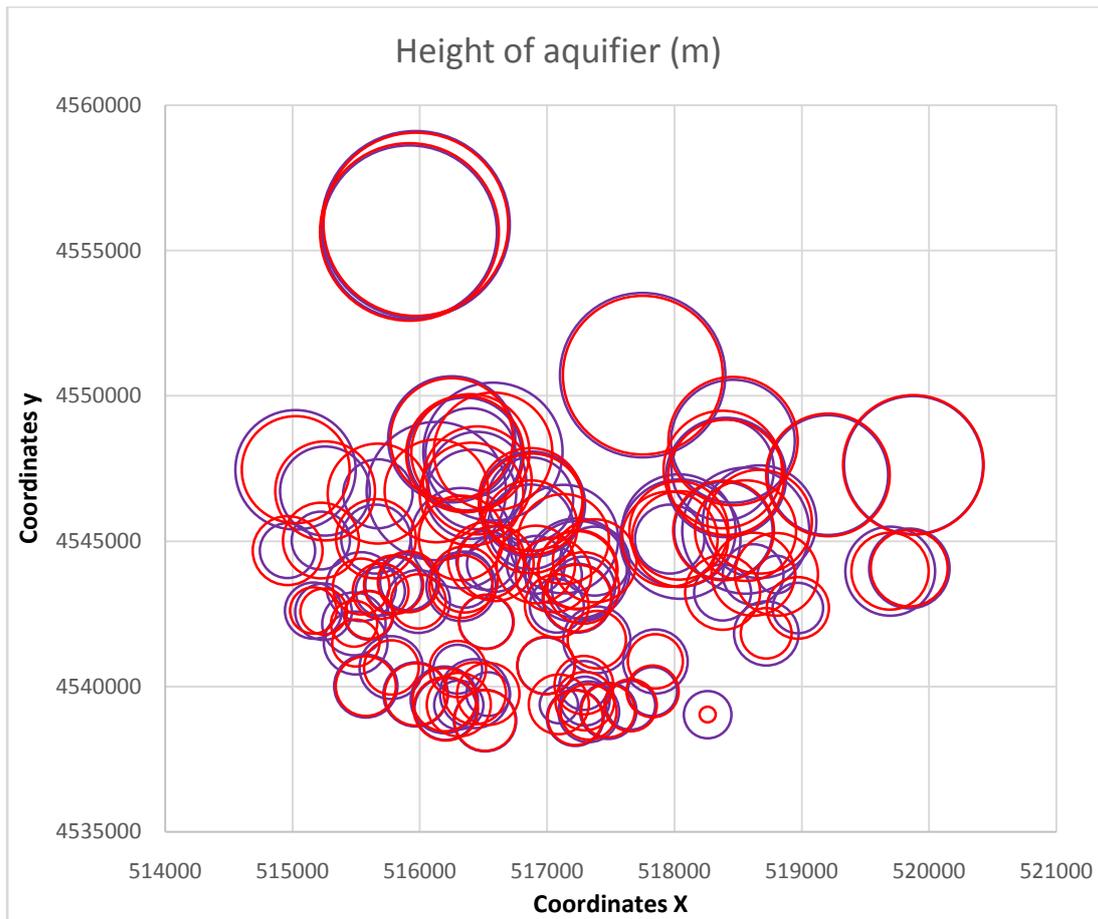
**Figure 3.8: Test with the entire dataset.**



**Figure 3.9:** Test with the entire dataset (detail - left third).



**Figure 3.10:** Test with the entire dataset (detail - middle third).



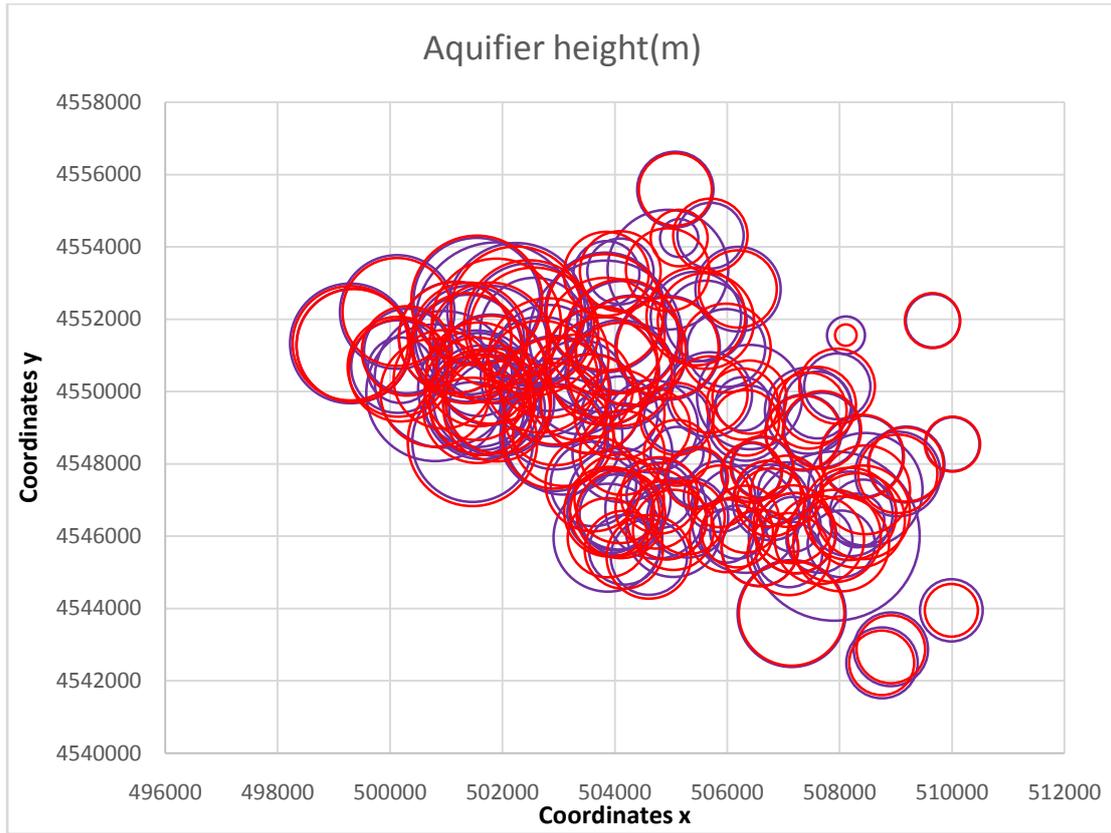
**Figure 3.11: Test with the entire dataset (detail - right third).**

### 3.2.2 Testing with a divided group of data

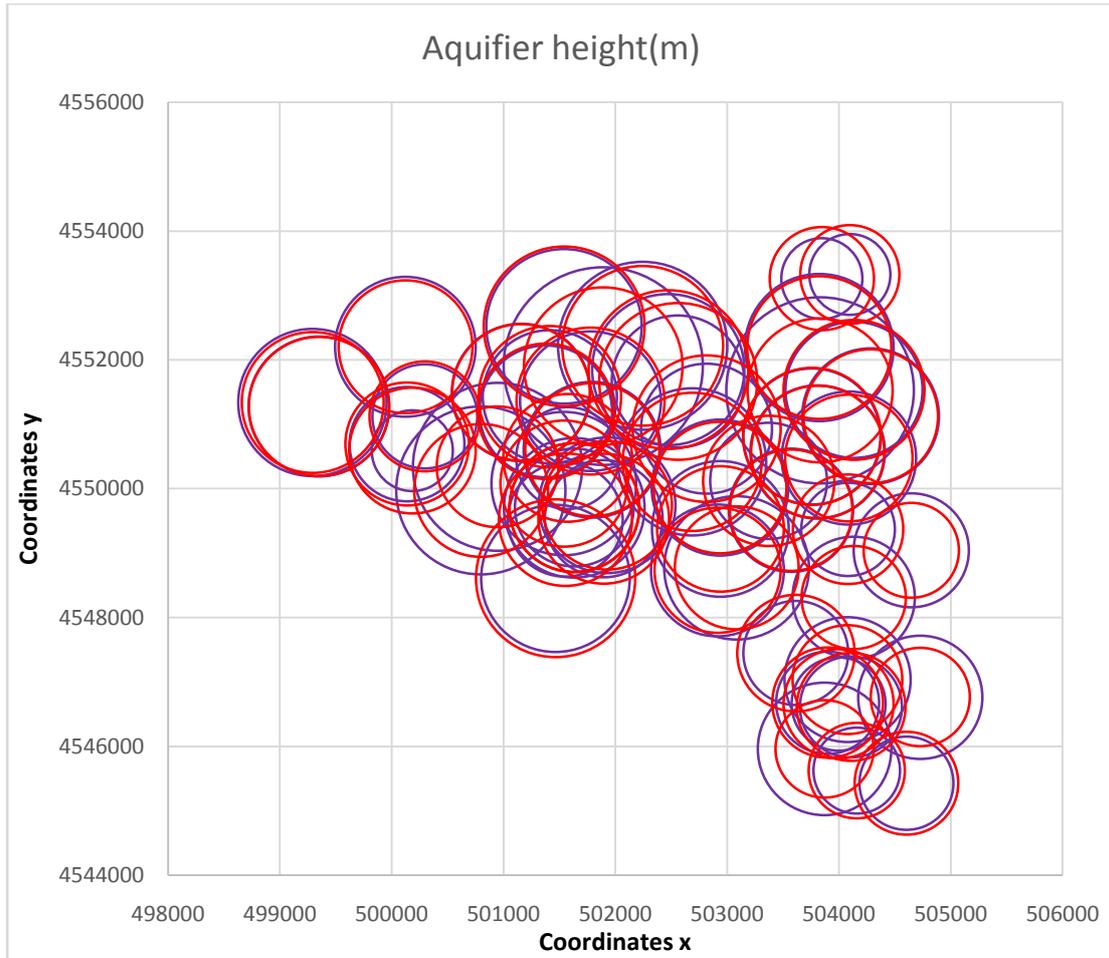
As it was previously mentioned, the wells' positions can be divided into two discrete groups. This observation will be used in order to increase the predictive capability of the utilized RBFN. To this end, we trained two different RBFNs, with the data of the two discrete groups of wells, and we compared the corresponding results with those of the prediction based on a single group.

#### Testing with the first group of data

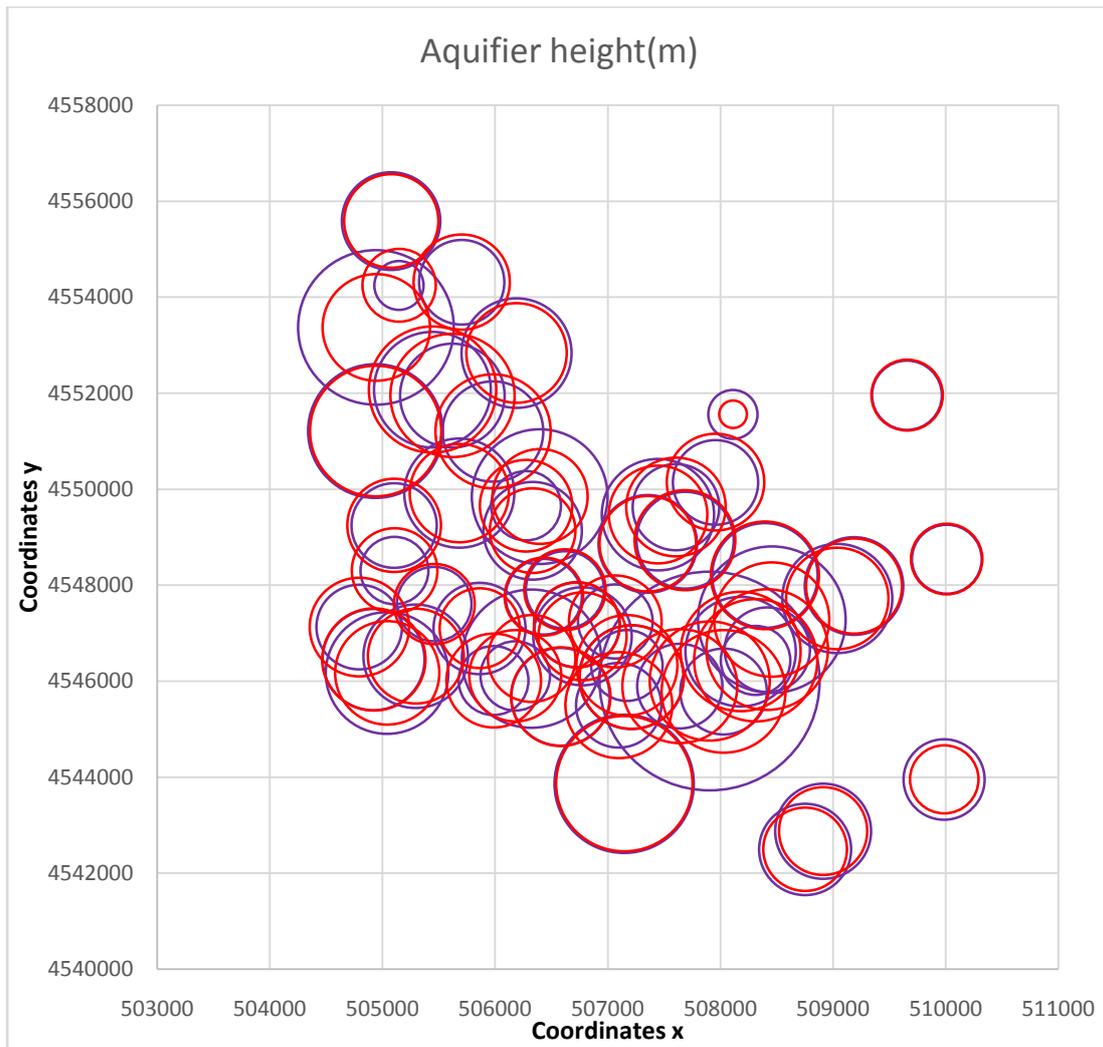
The first (left) group of data has a size of 110 units, from which 99%(108 units) will be used for training and 1% for testing (2 units). The resulting training and testing errors are  $3 \cdot 10^{-4}$  and  $4.7 \cdot 10^{-5}$ , accordingly. The corresponding comparison between the measured and predicted aquifer's height is presented in Figure 3.12 and more zoomed in on each left and right halves, in figures 3.13 and 3.14.



**Figure 3.12: Test with first group of dataset.**



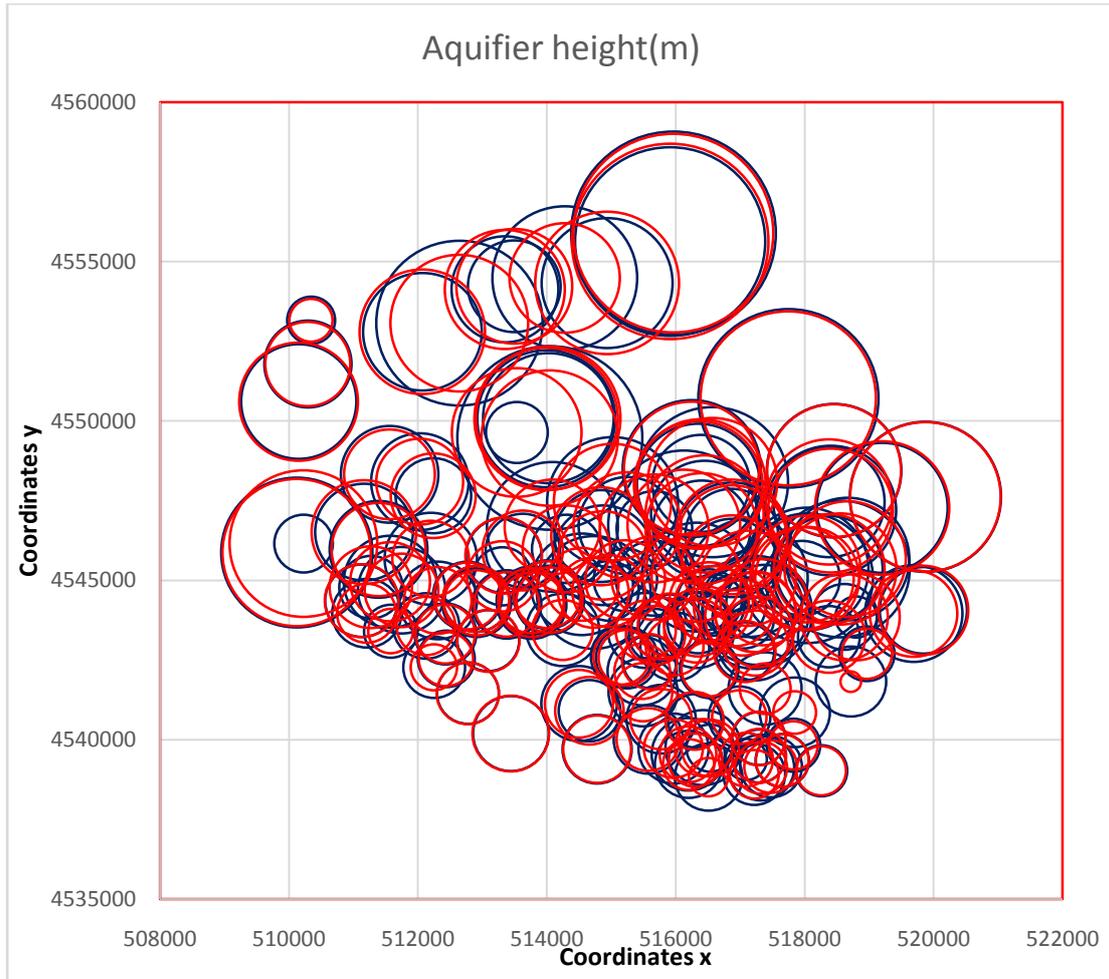
**Figure 3.13: Test with first group of dataset (detail - left half).**



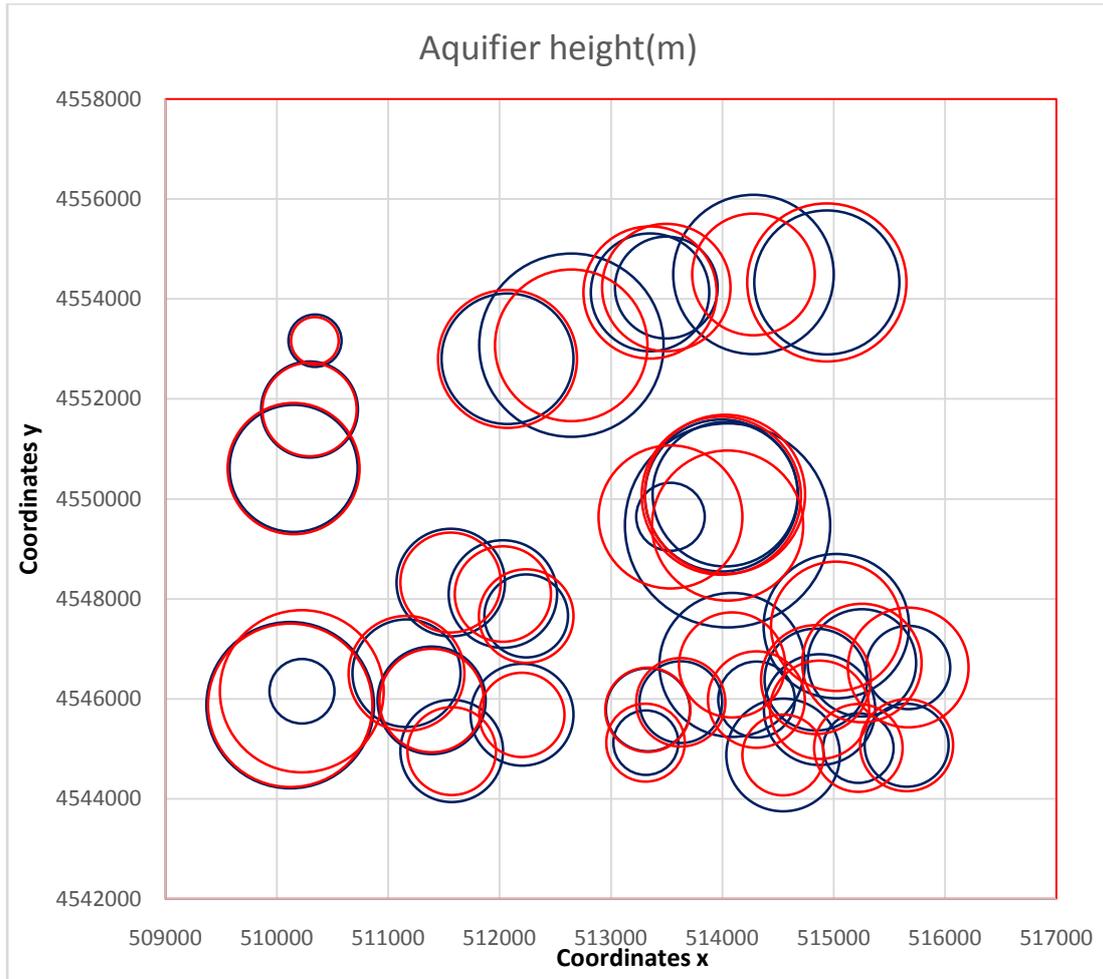
**Figure 3.14: Test with first group of dataset (detail - right half).**

Testing with the second group of data

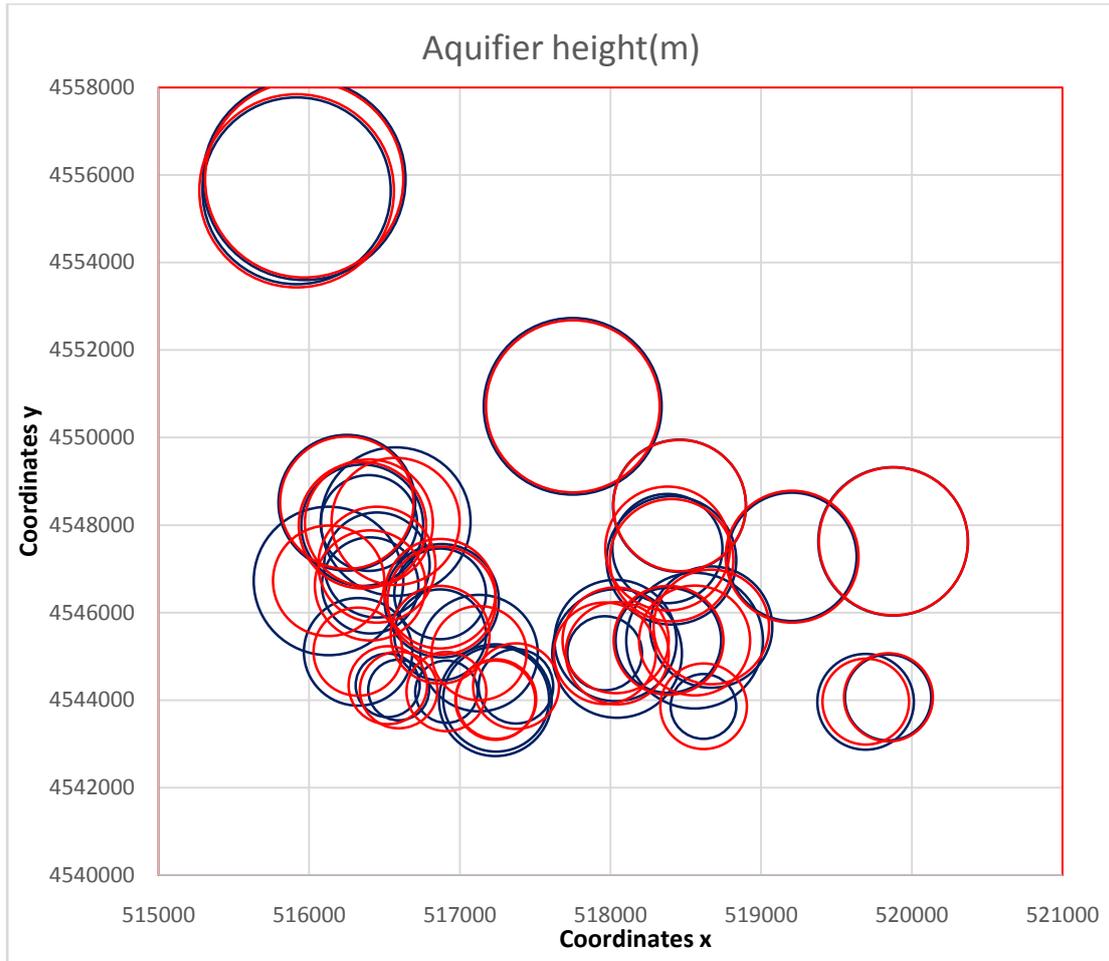
The second (right) group of data has a size of 140 units, from which 99%(138) have been used for training and 1%(2) for testing. The resulting training and testing errors are  $2.49 \cdot 10^{-3}$  and  $4.4 \cdot 10^{-2}$  accordingly. The corresponding comparison between the measured and predicted aquifer’s height is presented in Figure 3.15 and in more detail for each quarter in figures 3.16 to 3.19.



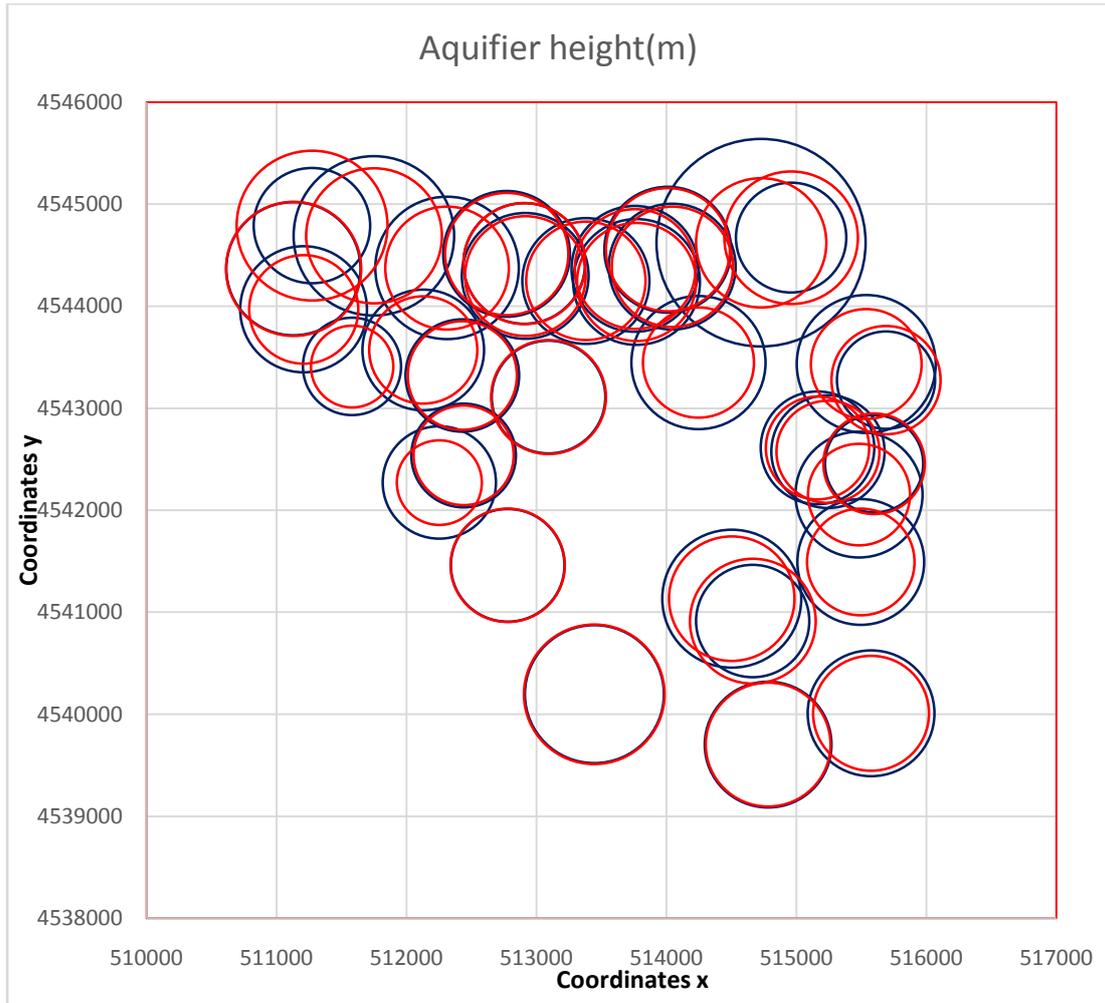
**Figure 3.15: Test with second group of dataset.**



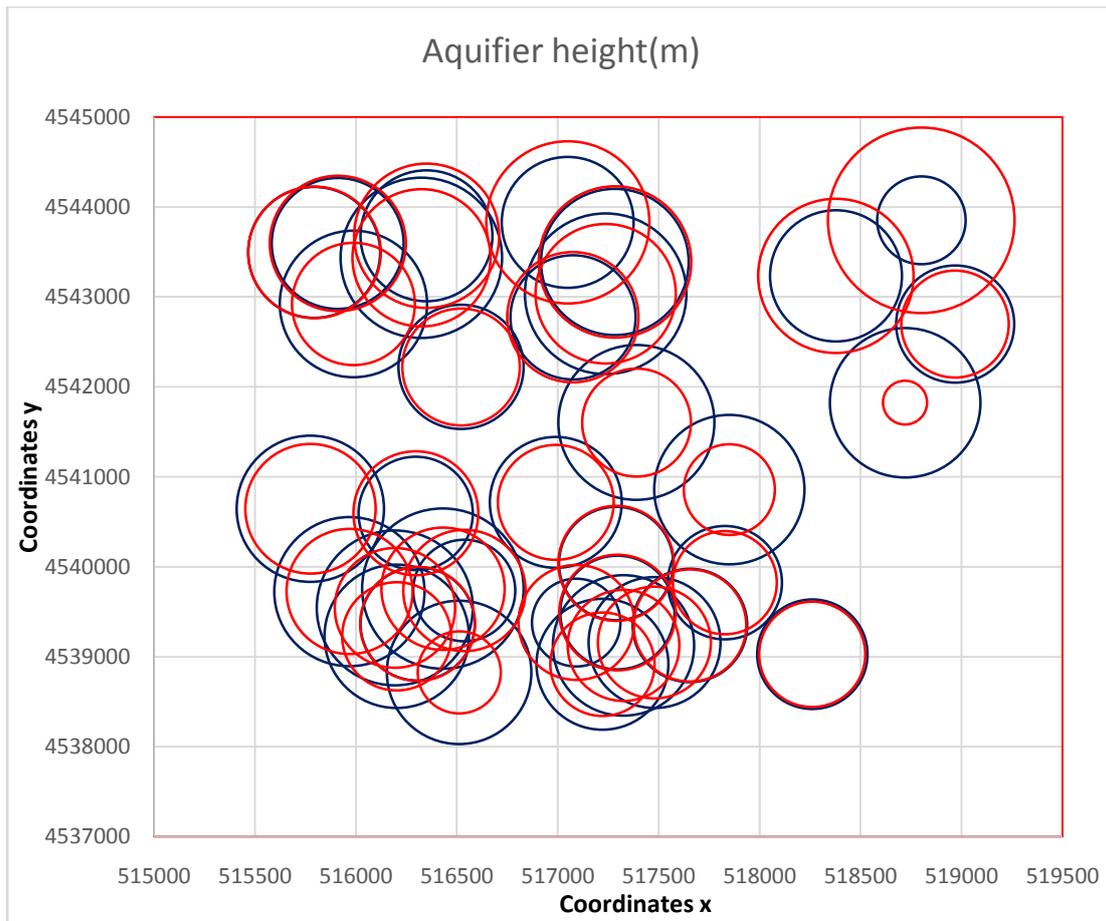
**Figure 3.16: Test with second group of dataset (detail - top left quarter).**



**Figure 3.17: Test with second group of dataset (detail - top right quarter).**



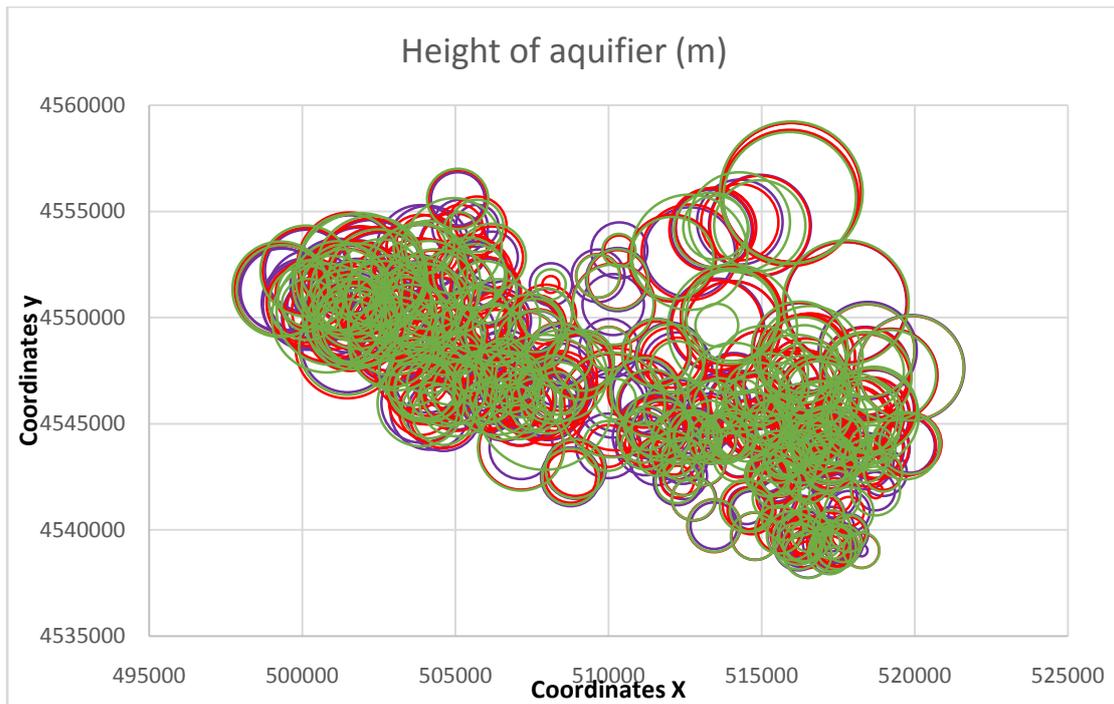
**Figure 3.18: Test with second group of dataset (detail - bottom left quarter).**



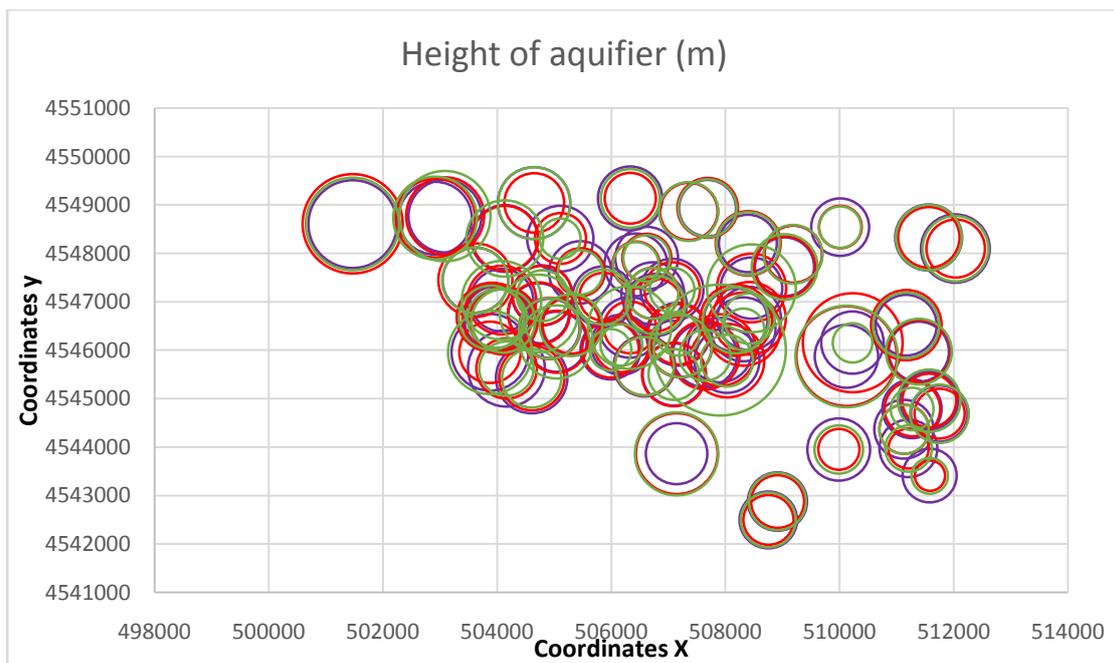
**Figure 3.19: Test with second group of dataset (detail - bottom right quarter).**

Divided dataset training compared to entire dataset training.

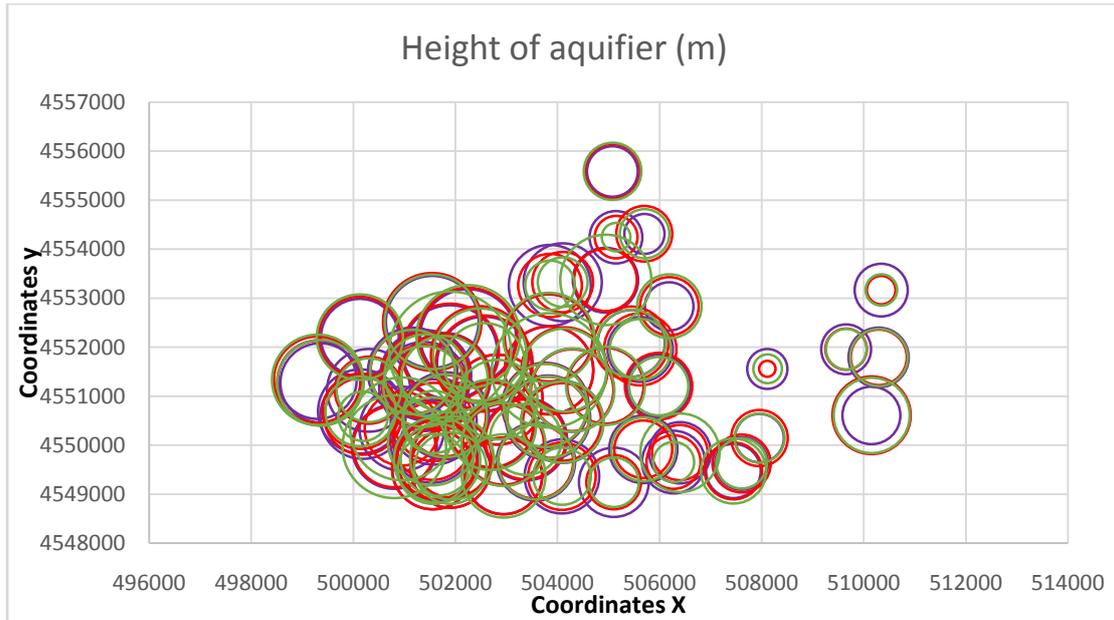
The following graph includes the predicted results for each of the 250 locations for both ways of training (entire dataset and divided dataset) compared to the measured data. The red circles correspond to the divided dataset training, the blue circles to the entire dataset training and the green circles to the measured data. The results are presented in Figure 3.20 and in more detail for each quarter in Figures 3.21 to 3.24.



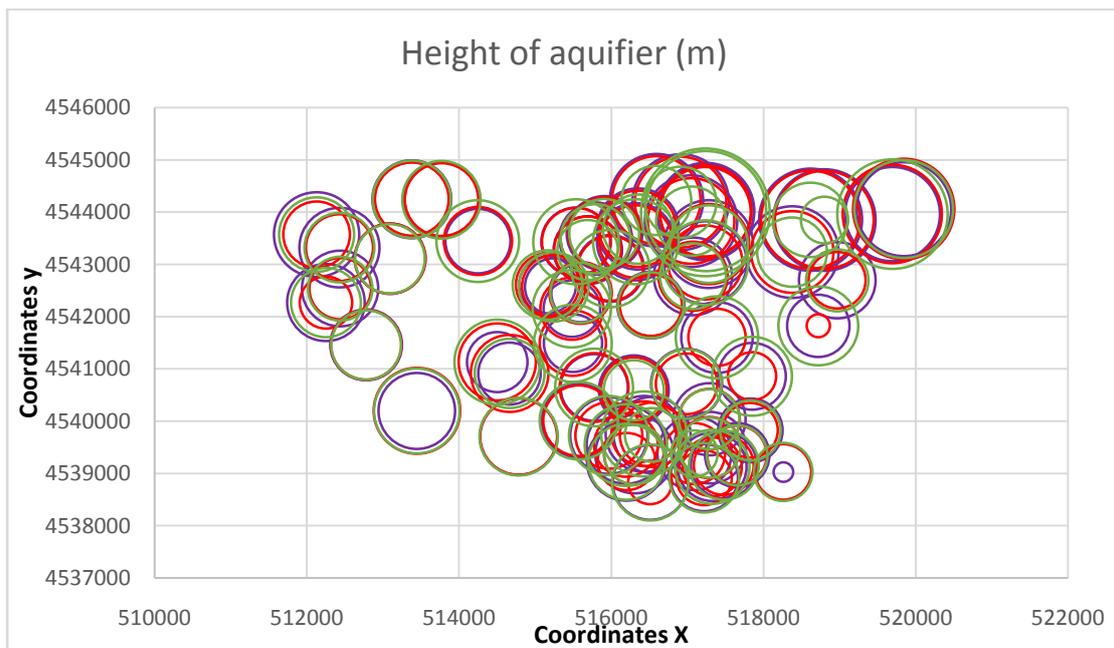
**Figure 3.20: Divided dataset training compared to entire dataset training and measured data**



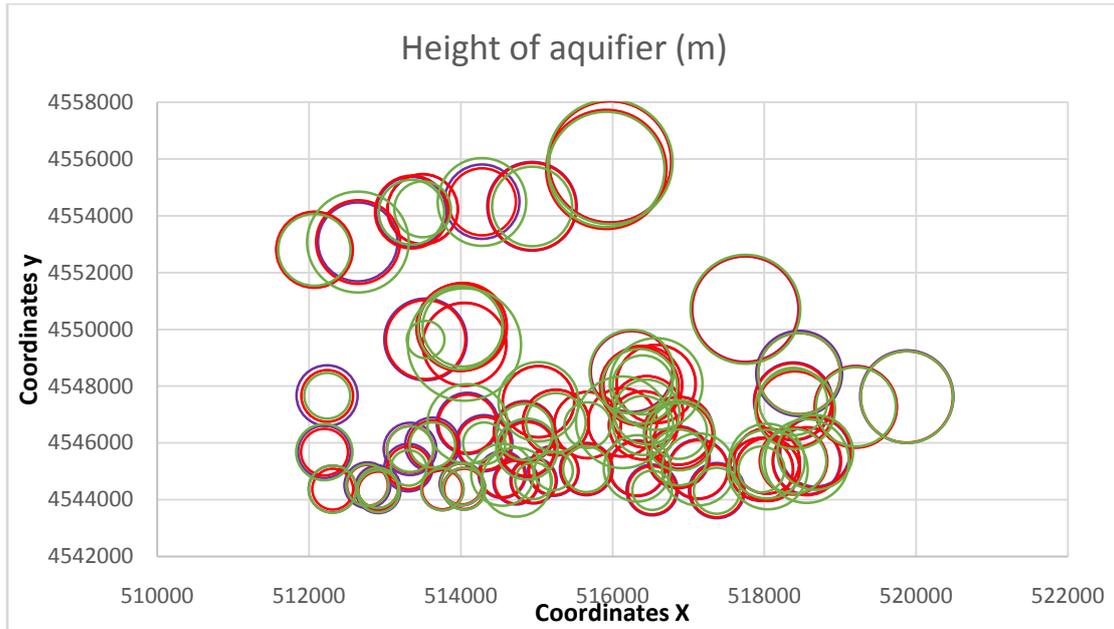
**Figure 3.21: Divided dataset training compared to entire dataset training and measured data(detail - bottom left quarter).**



**Figure 3.22: Divided dataset training compared to entire dataset training and measured data(detail - top left quarter).**



**Figure 3.23: Divided dataset training compared to entire dataset training and measured data(detail - bottom right quarter).**



**Figure 3.24:** Divided dataset training compared to entire dataset training and measured data(detail - top right quarter).

### 3.2.3 Contour maps

The following figures, 3.25 to 3.30 display a map of the study area with the isolines of the aquifer’s height. These maps have been constructed in order to better demonstrate the prediction accuracy of the RBF network, for the two different cases. First, the isolines for the measured data are presented. As it can be clearly observed, the results of the split data set provide more details of the aquifer’s height distribution, compared to the ones obtained with the entire data set.

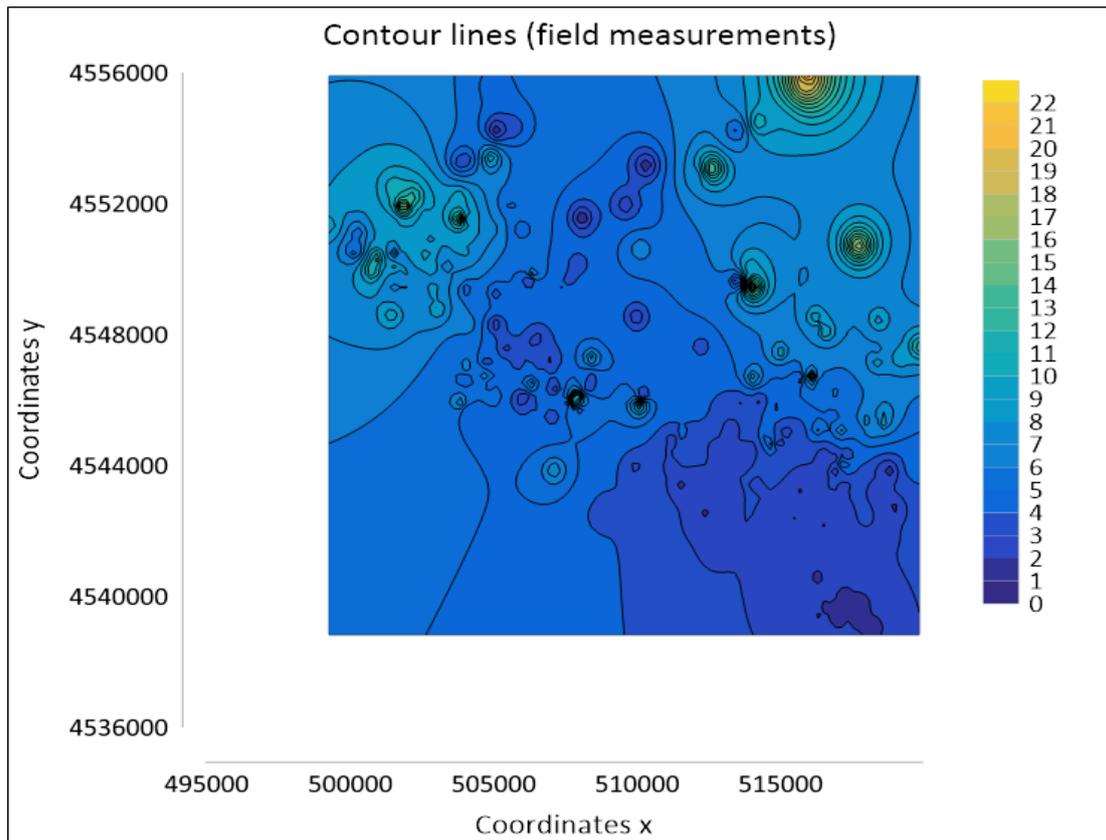


Figure 3.25: Contour lines of the measured aquifer height.

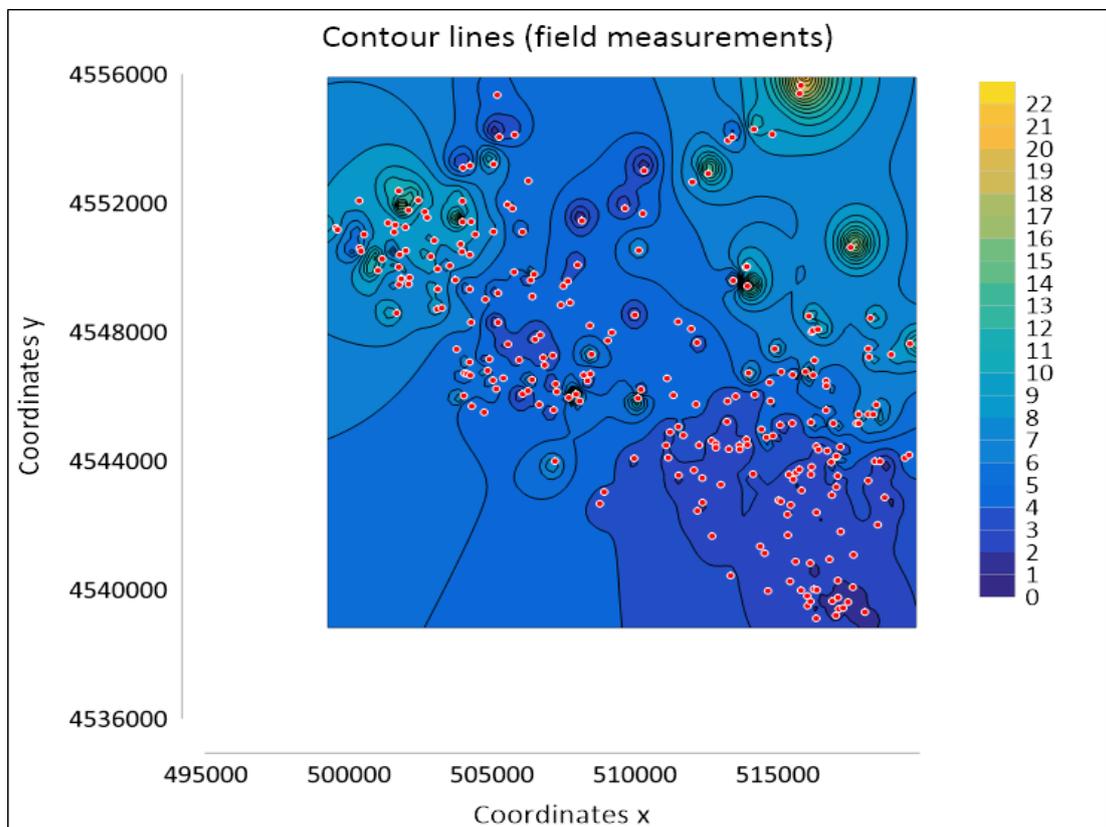
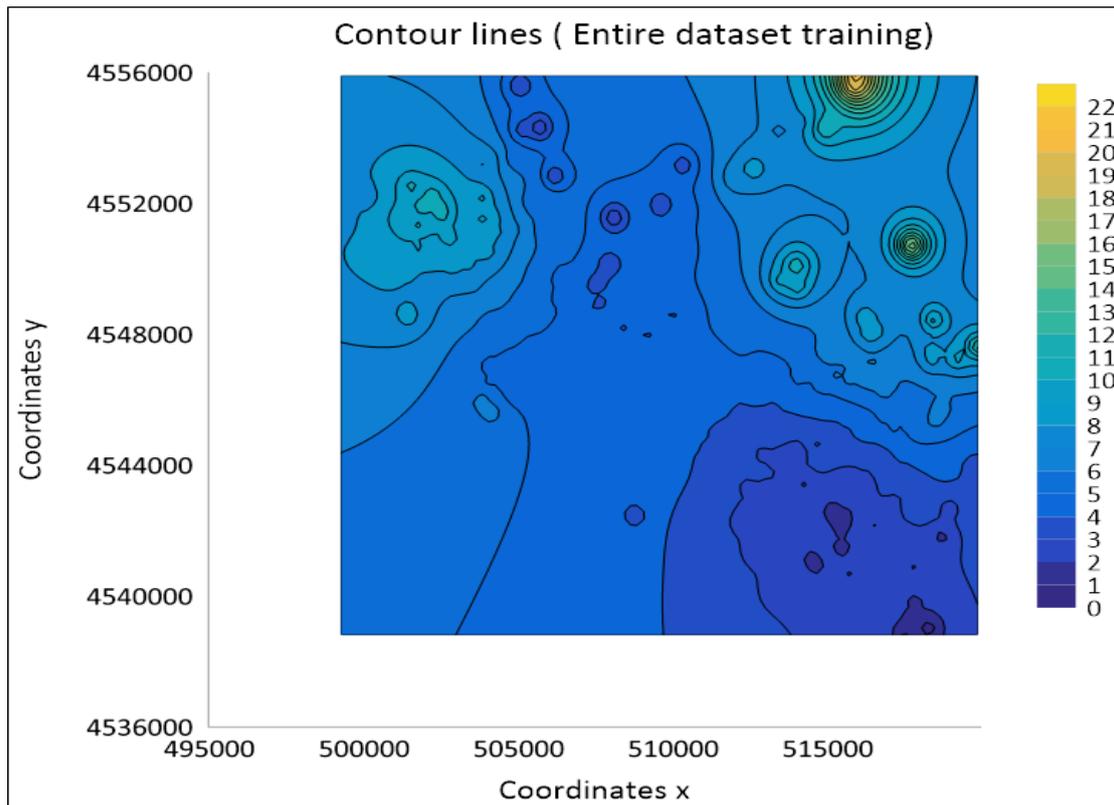
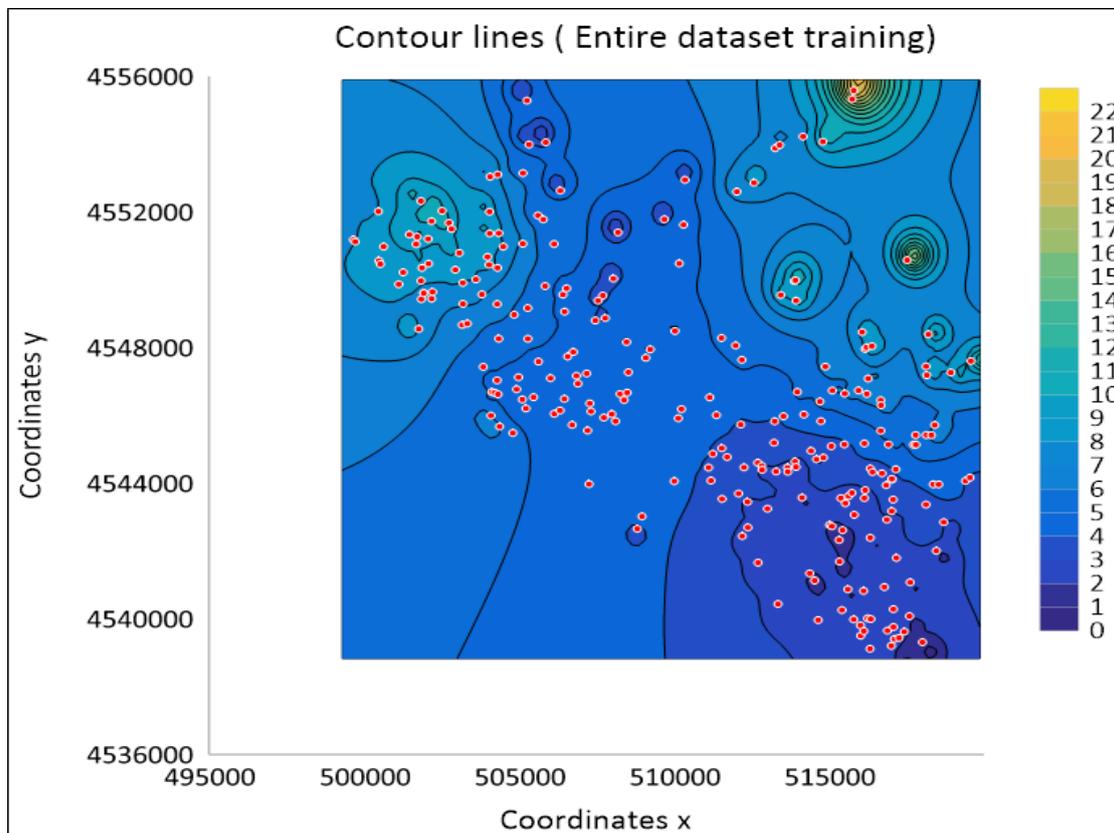


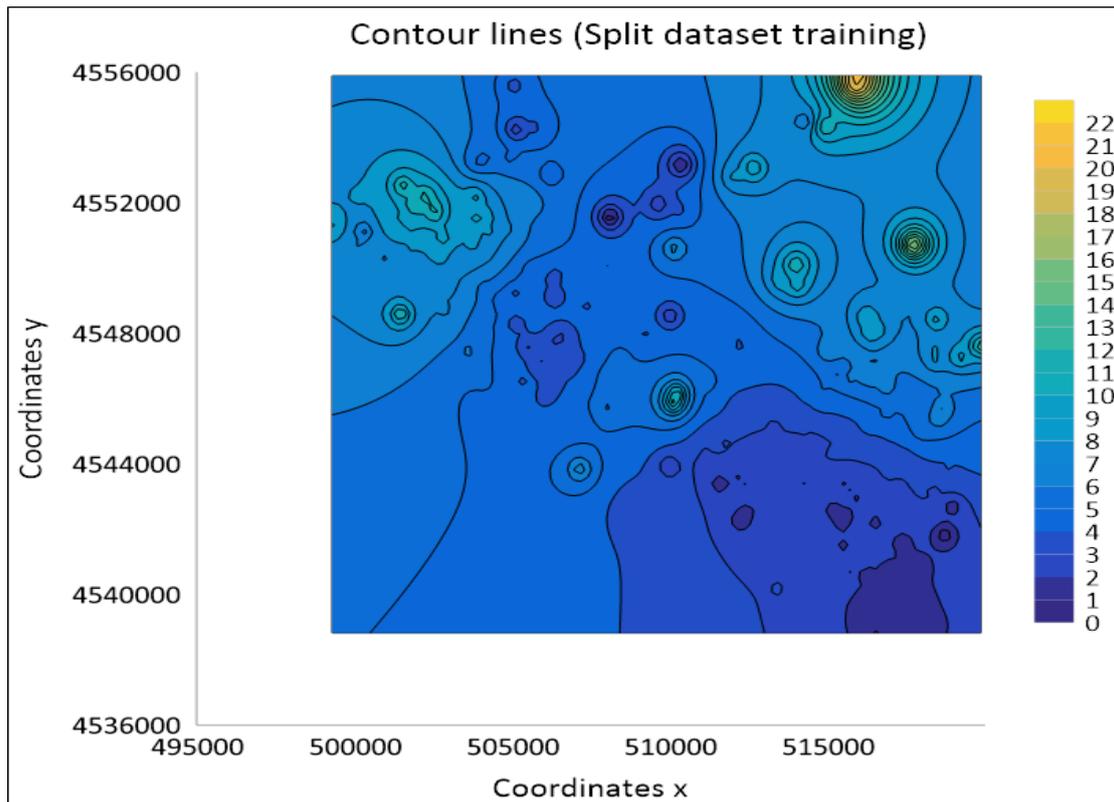
Figure 3.26 Contour lines of the measured aquifer height, along with the 250 measurement sites displayed with red circles.



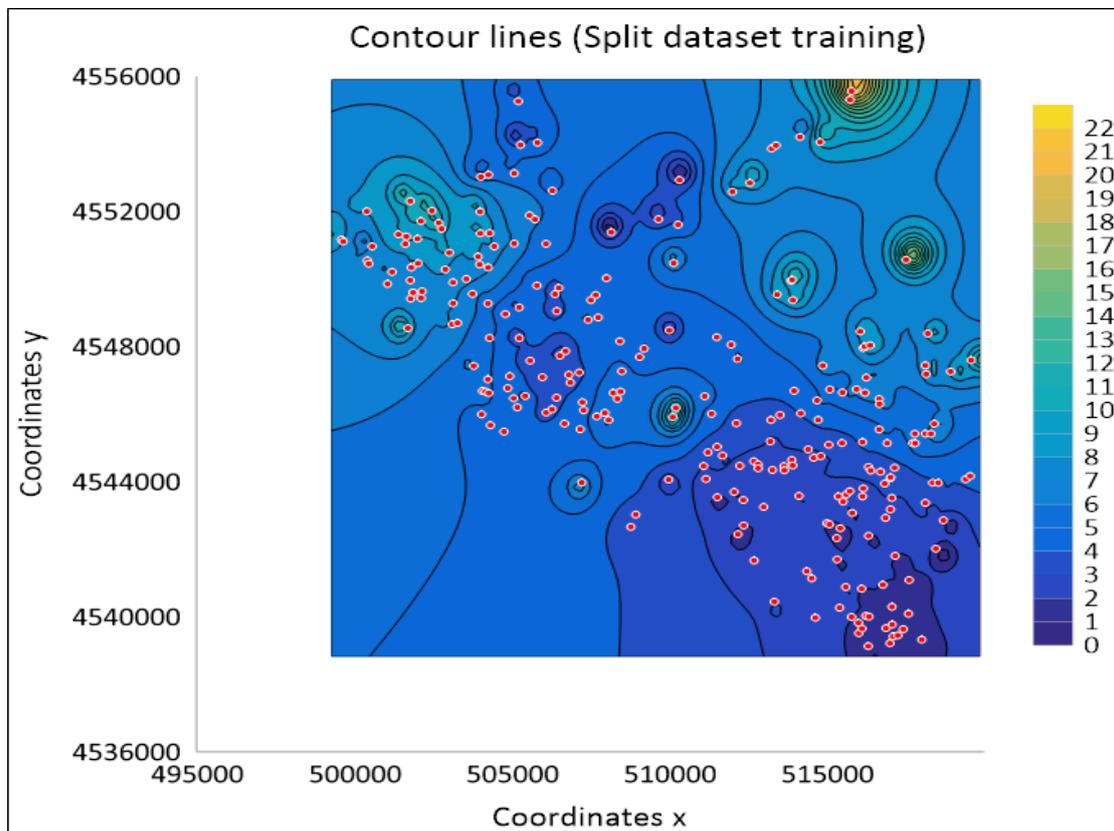
**Figure 3.27:** Contour lines of the predicted aquifer height, using the entire dataset for training the RBFN.



**Figure 3.28:** Contour lines of the predicted aquifer height, using the entire dataset for training the RBFN, along with the 250 measurement sites displayed with red circles.



**Figure 3.29:** Contour lines of the predicted aquifer height, using the splitted dataset for training the RBFN.



**Figure 3.30:** Contour lines of the predicted aquifer height using the splitted dataset for training the RBFN, along with the 250 measurement sites displayed with red circles.

### 3.2.4 Testing with a random group of data

The neural network was tested using a mathematical function to produce the “measured data”, while the spatial positions for the “measurements” have been produced randomly in a range from 0 to 100 for columns x and y. The function producing the z coordinate of the “measured data” is the following:

$$z = 2 * \sin\left(\frac{x}{10}\right) + 3 * \cos\left(\frac{y}{10}\right) + 6$$

The produced data was used for training the RBF network, using 99% (247) of the data as training data and 1% (3) as testing data. The corresponding results show training and testing errors of  $3.33 \cdot 10^{-4}$  and  $3.87 \cdot 10^{-4}$ . The results can be viewed in Figure 3.25 and in more detail for each quarter in Figures 3.31 to 3.35. As it can be seen, there is a very good agreement between the “measured” and the predicted data. The reason is that the spatial distribution of the available training data is very uniform, which results in a very uniform spatial distribution of the Radial Basis Functions’ centers. This results in a better approximation capability of the network.

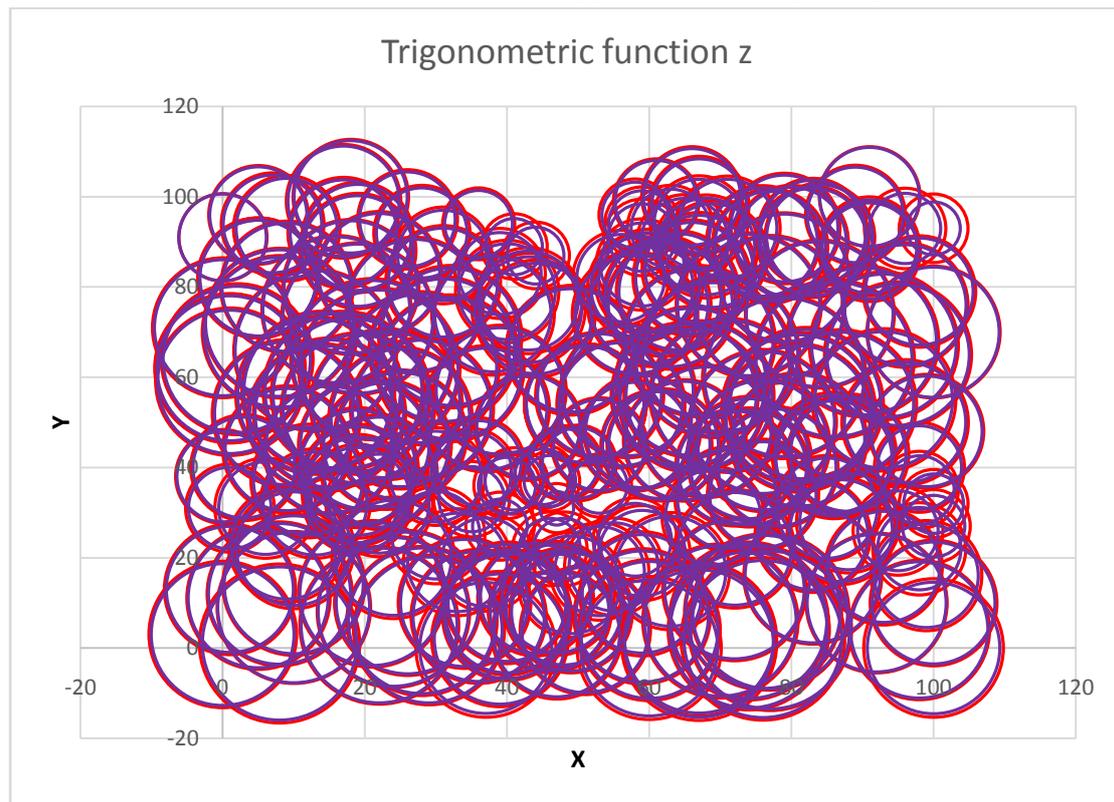
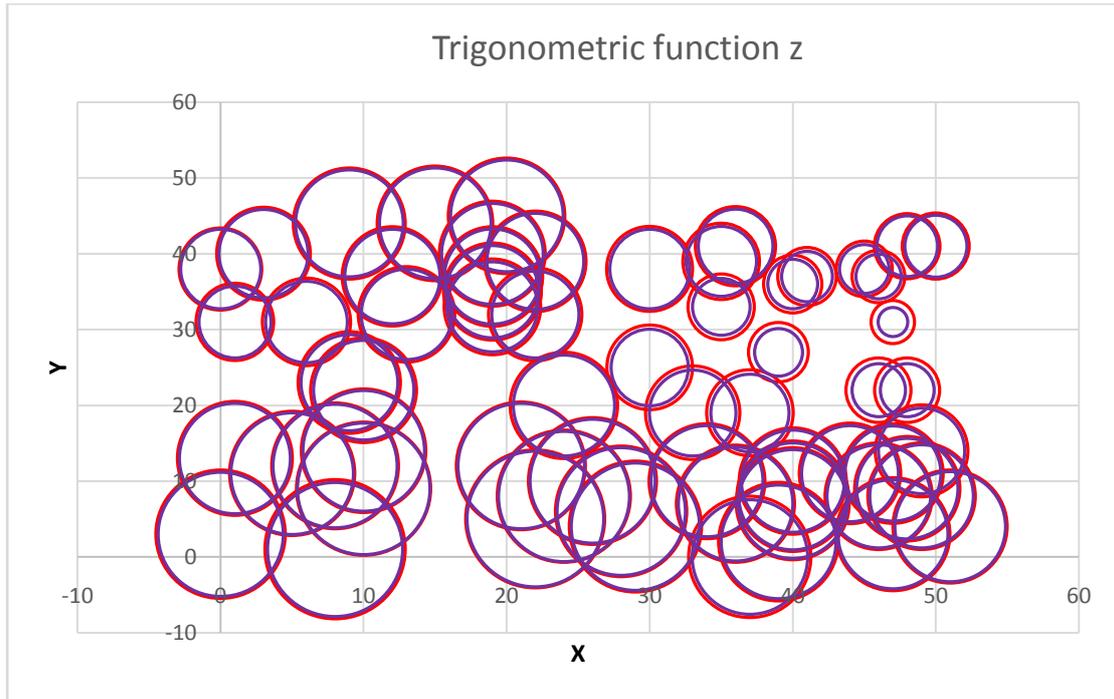
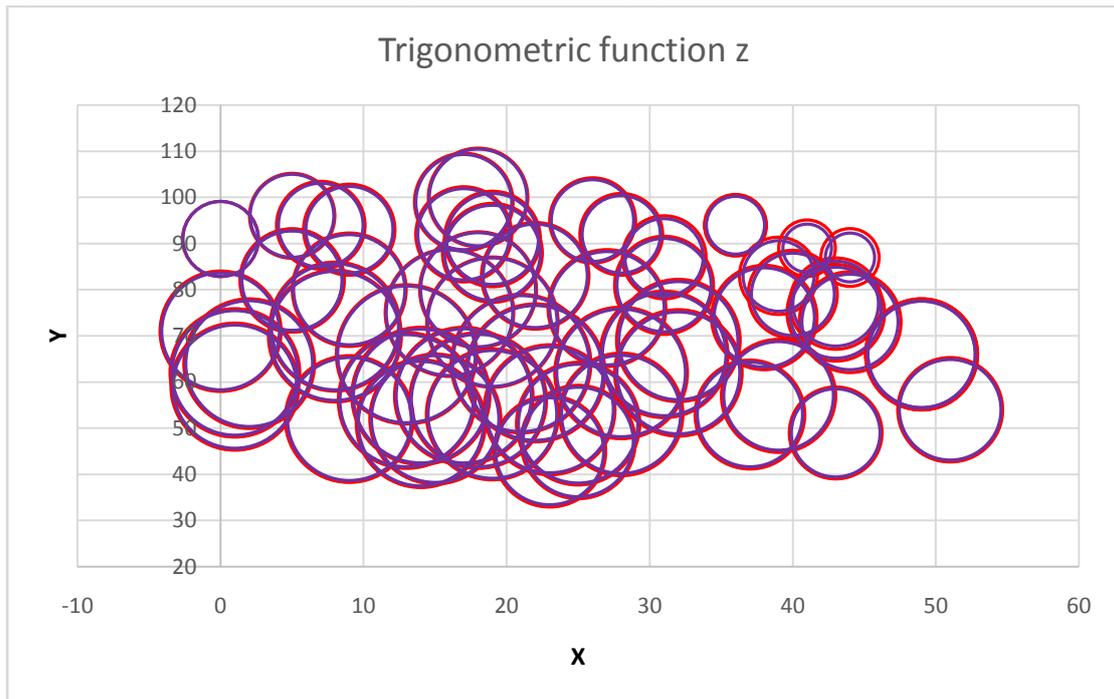


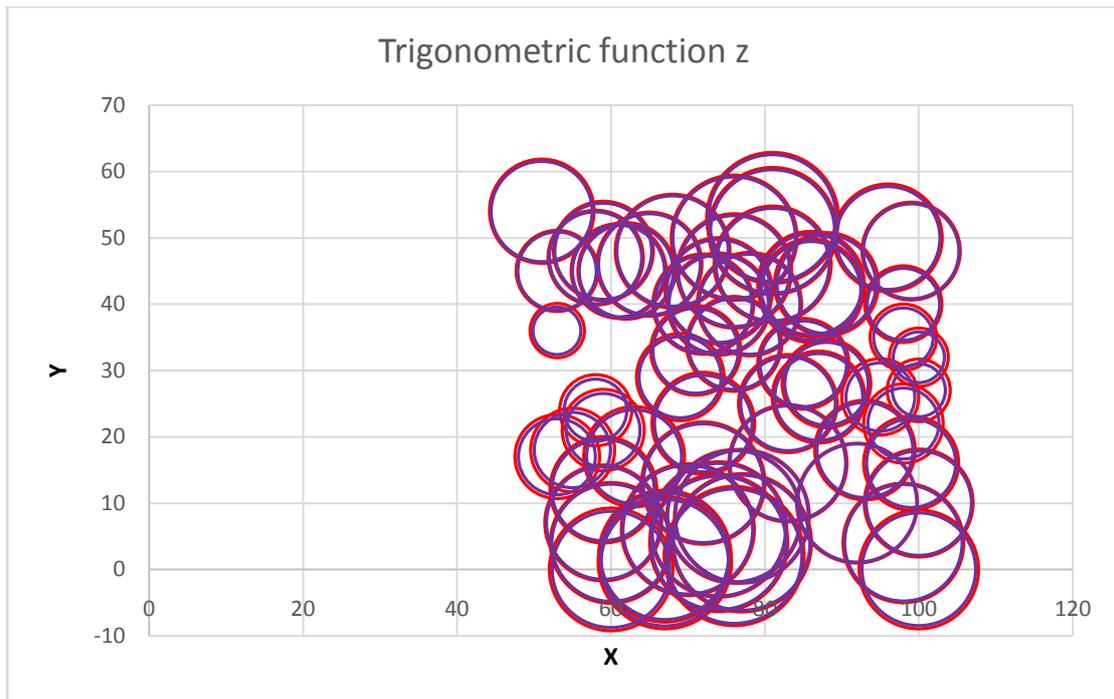
Figure 3.31: Random dataset training (entire set).



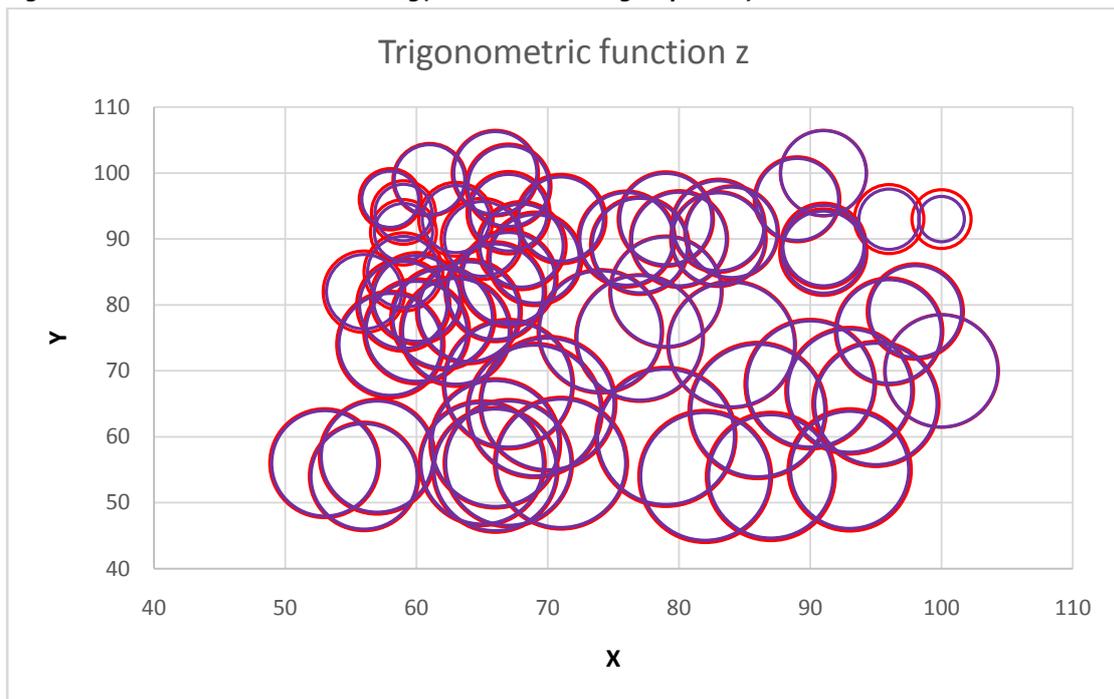
**Figure 3.32: Random dataset training (detail - bottom left quarter).**



**Figure 3.33: Random dataset training (detail - top left quarter).**



**Figure 3.34: Random dataset training(detail - bottom right quarter).**



**Figure 3.35: Random dataset training(detail - top right quarter).**

### 3.2.5 Observations

It is clear that splitting the training data into two concentrated groups has improved the ANNs accuracy. On the first group of data the ANNs errors have dropped by more than one factor, while on the second they have slightly increased. However, as it can be observed in the produced contour maps of the aquifer’s height, the split

data set provided predictions with better details, compared to those produced with the entire data set.

This is to be expected because the neural network this way is trained to adapt to each specific localized dataset in a better way. As the positions of the measurements are scattered in a non-canonical way, the splitting of the dataset in two sub-sets which are more canonically distributed allows for a more uniform distribution of the RBF centers and a better predictive capability of the network.

## CHAPTER 4: CONCLUSIONS

This study presented the use of a specific type of artificial neural network, namely a radial basis function network (RBFN) for the spatial prediction of an aquifer's height. This method has proved to be a success, as the error measurements are very low and the predicted spatial distribution of the aquifer's height is close to the corresponding measured distribution.

Our main conclusion regarding the function of the RBFN is that datasets that have a more uniform distribution increase the accuracy of the network, as this allows for a more uniform distribution of the RBFN centers. This could be determined from the increase in accuracy when the dataset was split from an unevenly distributed dataset, to two more canonically distributed sub datasets. It could also be seen when the RBFN was trained and tested on an artificial dataset with a random but almost uniform spatial distribution, providing very satisfactory results. This characteristic is related to the use of the Gaussian function for the RBFN, which provides a strong response to the area close to its center.

Finally, the use of RBF artificial neural networks for this application shows their ease of use and practicality, as there was no use of modelling of the physical and chemical processes of our study area. The artificial neural network provided the solutions without determining the relationship between the input data and the results. Therefore they provide a viable alternative technique to conventional hydrological modelling techniques.

## REFERENCES

- F. Akram, M.G. Rasul, M.M.K. Khan, M.S.I.I Amir (2012), "A Comparative View of Groundwater Flow Simulation Using Two Modelling Software - MODFLOW and MIKE SHE", *Proceedings of the 18th Australasian Fluid Mechanics Conference*, Launceston, Australia.
- J. Arnold, R. Srinivasan, R.S. Muttiah, J.R. Williams (1998), "Large Area Hydrologic Modeling and Assessment Part I: Model Development", *JAWRA Journal of the American Water Resources Association*, vol. 34(1), pp. 73-89.
- A.C. Bagtzoglou, F. Hossain (2008), "Radial basis function neural network for hydrologic inversion: an appraisal with classical and spatio-temporal geostatistical techniques in the context of site characterization", *Stochastic Environmental Research and Risk Assessment*, vol. 23(7), pp. 933–945.
- E. Binaghi, V. Pedoia, A. Guidali, M. Guglielmin (2013), "Snow cover thickness estimation using radial basis function networks", *The Cryosphere*, vol. 7, pp. 841–854.
- M.T. Brown, L.R. Wicker (2000), "Discriminant Analysis", *Handbook of Applied Multivariate Statistics and Mathematical Modeling*, pp. 209-235.
- R. Callan (1999), *The essence of Neural Networks*, Prentice Hall Europe.
- A. Gemitzi, C. Petalas, V. Pisinaras, V.A. Tsihrintzis(2008), "Spatial prediction of nitrate pollution in ground waters using neural networks and GIS: an application to South Rhodope aquifer (Thrace, Greece)", *Hydrological Processes*, vol. 23, pp. 372–383.
- V. Gholami, M. Sebgati, Z. Yousefi (2016), "Integration of artificial neural network and geographic information system applications in simulating groundwater quality", *Environmental Health Engineering and Management Journal*, vol. 3(4), pp. 173-182.
- S. Gopal (2017), "Artificial neural networks in geospatial analysis", *The International Encyclopedia of Geography*, John Wiley & Sons Ltd.
- S. Grimaldi, S.-C. Kao, A. Castellarin, S.-M.Papalexiou, A. Viglione, F. Laio, H. Aksoy and A. Gedikli (2011), "Statistical Hydrology". In: P. Wilderer (Ed.) *Treatise on Water Science*, vol. 2, pp. 479–517, Oxford Academic Press.
- T. Harter (2003), *Basic Concepts of Groundwater Hydrology*, Agriculture and Natural Resources (ANR), Publication 8083, University of California, Davis, USA.
- S. Haykin (1999), *Neural Networks a Comprehensive Foundation*, 2<sup>nd</sup> Edition, Prentice-Hall.

D.O. Hebb (1949), *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York.

J.J. Hopfield (1982), "Neural Networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences, USA*, vol. 79, pp. 2554-2558.

J.W. Kalat (2001), *Biological Psychology*, 7<sup>th</sup> edition, Wadsworth Thomson Learning.

R, Kalra, M.C. Deo, R. Kumar, V. K. Agarwal (2005), "RBF network for spatial mapping of wave heights", *Marine Structures*, vol. 18(3), pp. 289-300.

B. Krose, P. Smagt (1996), *An Introduction to Neural Networks*, 8<sup>th</sup> Edition, The University of Amsterdam.

J.F. Mas, J.J. Flores (2008), "The application of artificial neural networks to the analysis of remotely sensed data", *International Journal of Remote Sensing*, vol. 29(3), pp. 617-663.

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

M.L. Minsky (1961), "Steps toward artificial intelligence", *Proceedings of the Institute of Radio Engineers*, vol. 49, pp. 8-30, (Reprinted in: E.A. Feigenbaum, and J. Feldman, (Eds.), *Computers and Thought*, pp. 406-450, McGraw-Hill, New York).

M.L. Minsky (1967), *Computation: Finite and Infinite Machines*, Englewood Cliffs, Prentice-Hall.

M.L. Minsky, S.A Papert (1969), *Perceptrons*, MIT Press.

S.A. Nelson (2015), *Ground water*, Lecture Notes, Tulane University.

I.K. Nikolos, E.S. Zografos, A.N. Brintaki (2007), "UAV Path Planning Using Evolutionary Algorithms", in J.S. Chahl et al. (Eds.), *Innovations in Intelligent Machines - 1*, (Series: Studies in Computational Intelligence – SCI, vol. 70), pp. 77-111, Springer.

S. Noack, A. Knobloch, S.H. Etzold, A. Barth, E. Kallmeier (2014), “Spatial predictive mapping using Artificial Neural Networks”, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-2, ISPRS Technical Commission II Symposium, 6 – 8 October, Toronto, Canada.

M.J.L. Orr (1996), *Introduction to Radial Basis Function Networks*, Centre for Cognitive Science, University of Edinburgh, Scotland, U.K.

M. Richarme (2002), “Eleven Multivariate Analysis Techniques: Key Tools In Your Marketing Research Survival Kit”.<https://www.decisionanalyst.com/whitepapers/multivariate/>

B.D. Ripley (1996), *Patern Recognition and Neural Networks*, 1<sup>st</sup> Edition, Cambridge University Press.

N. Rochester, J.H Holland, L.H Haibt, W.L Duda (1956), “Tests on a cell assembly theory of the action of the brain using a large digital computer”, *IRE Transactions on Information Theory*, vol. IT-2, pp. 80-93.

F. Rosenblatt (1958), “The perceptron: A probabilistic model for information storage and organization in the brain”, *Psychological Review*, vol. 65, pp. 386-408.

D.E. Rumelhart, G.E. Hinton, R.J. Williams (1986), “Learning representations of back-propagation errors”, *Nature*, vol. 323, pp. 533-536.

C. Rusu, V. Rusu(2006),“Radial Basis Functions Versus Geostatistics in Spatial Interpolations”, *International Federation for Information Processing (IFIP)*, vol. 217, pp. 119–128.

S.M. Seth(2009), “Physically based hydrological modelling”.  
<https://www.geospatialworld.net/article/physically-based-hydrological-modelling/>

P.J. Werbos (1974), *Beyond Regression: New tools for prediction and analysis in the behavioral sciences*, Ph. D Thesis, Harvard University, Cambridge, MA, USA.

B. Widrow, M.E. Hoff (1960), “Adaptive switching circuits”, *IRE WESCON Convention Record*, pp. 96-104.

N. Wiener (1948), *Cybernetics: Or Control and Communication in the Animal and the Machine*, Wiley, New York.

C. Woodford (2018), “Neural Networks”.  
<http://www.explainthatstuff.com/introduction-to-neural-networks.html>

Chong Yu. Xu(2002), *Hydrologic Models*,Textbooks of Uppsala University, Department of Earth Sciences Hydrology.

The USGS Water Science School (2016),“Infiltration-The water cycle”.<https://water.usgs.gov/edu/watercycleinfiltration.html>