DISSERTATION

USING NEURAL NETWORKS AS AN ALTERNATIVE TO STATISTICAL MODELING IN KRIGING INTERPOLATION PROCEDURES: AN INVESTIGATION

Submitted by
Anna Brandis
Department of Forest, Rangeland, and Watershed Stewardship

In partial fulfillment of the requirements
For the degree of Doctor of Philosophy
Colorado State University
Fort Collins, Colorado
Fall 2005

COLORADO STATE UNIVERSITY

July 18th, 2005

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY ANNA BRANDIS ENTITLED "**USING NEURAL NETWORKS AS AN ALTERNATIVE TO STATISTICAL MODELING IN KRIGING INTERPOLATION PROCEDURES: AN INVESTIGATION**" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Dr. Charles W. Anderson

Dr. Melinda J. Laituri

Dr. Freeman M. Smith

Dr. Denis J. Dean, Adviser

Dr. N. Thompson Hobbs, Department Head

ABSTRACT OF DISSERTATION

USING NEURAL NETWORKS AS AN ALTERNATIVE TO STATISTICAL MODELING IN KRIGING INTERPOLATION PROCEDURES: AN INVESTIGATION

Spatial interpolation techniques provide a means of predicting values of a variable of interest (e.g., an attribute such as elevation, rainfall, contaminant levels, soil type, etc.) at locations where, due to practical constraints, the variable cannot be measured. Kriging is one of the most widely used interpolation procedures, and it has been shown to produce accurate estimates in many cases. However, there are documented cases in which kriging does not produce accurate results. In these cases, it is possible that kriging procedures could be improved.

This study explored the ability of a commonly used artificial neural network (ANN) to correct the limitations of the regression-based approach to semivariogram analysis commonly employed in conventional kriging. A hybrid ANN-based kriging model (KrigANN) was developed, in which a multilayer feedforward neural network trained using the MEKA algorithm replaced the traditional kriging's regression-based semivariogram development approach, while retaining all the other properties of traditional kriging. The accuracy and precision of the predictions produced by this hybrid model were compared to those obtained from the conventional regression-based kriging

procedure by applying both models to 2250 artificially generated datasets created for this purpose.

The study demonstrated that the KrigANN model produced more accurate interpolation results than regression-based kriging at low to medium degrees of spatial autocorrelation. These findings lead to the conclusion that the ANN-based procedures proposed in this study are appropriate alternatives to the conventional regression-based approaches, when spatial data exhibit low to medium degrees of autocorrelation.

Anna Brandis
Forest, Rangeland, and Watershed Stewardship Department
Colorado State University
Fort Collins, CO 80523
Fall 2005

ACKNOWLEDGMENTS

First of all, I wish to immensely thank my advisor Dr. Denis Dean. Without his illuminating guidance, immeasurable support, and endless patience, I would have never completed this research effort. During all this time as a mentor and a friend, he fostered an environment of excellence, and kept confidence high and apprehension under control. I also express my deep gratitude to all my committee members. Their advice and encouragement was essential for the completion of this project. Dr. Charles Anderson enlightened me on the most obscure subjects of this research, and made them manageable. Dr. Melinda Laituri has been an inspiration since her very first seminar I attended years ago, remaining a precious source of advice during my academic path. Dr. Freeman Smith, being on my committee again, guided me through another academic endeavor with his insightful perspectives on personal and scientific life. I also thank Debbie Devore of the CNR Computer Labs at CSU. My deep gratitude to my managers and coworkers at ESC, Inc. in Fort Collins for being supportive, encouraging, and accommodating while being a student-worker, allowing me to maintain a professional career while achieving this goal. Last but not least, I wish to thank my family. Mom and dad allowed me to be here in many ways, and gave me the discipline needed to confront hard work. Thanks to all my beloved family and friends in Italy and the U.S., whose stimulating discussions and never-ending moral support and encouragement motivated me to never give up.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1. INTRODUCTION

1.1. Background

It is often impossible to obtain an exhaustive census of a spatial variable of interest (e.g., an attribute such as elevation, rainfall, temperature, soil type, etc.) at every location of interest. This may be due to the continuous nature of the variable, or to practical constraints that limit the number of measurements of a discrete variable that can be obtained. The lack of exhaustive sampling creates the need for methods of estimating the value of a variable of interest at places where the value of that variable has not actually been measured (Longley et al., 2002). Spatial interpolation techniques provide a means of doing this. These techniques use data measured at known sampling locations, and knowledge about the underlying spatial relationships in the data, to produce estimated values of a variable of interest at unsampled locations. Considering how often sampled data is used in spatial analysis, optimal interpolation techniques are a subject of interest for all users of spatial data (Demirhan et al., 2003).

Kriging (Cressie, 1993; Burrough and McDonnell, 1998) is one of the most widely used interpolation procedures. There are several versions of kriging. All are firmly grounded in theory in that the procedures they employ arise from implementing variations of Regionalized Variable Theory. Kriging procedures have been shown to produce accurate estimates in many cases (Gaugush, 1993; Gunnarson et al., 1998; Critto et al. 2003; Liu et al. 2004); however, there are documented cases in which they do not (Little et al., 1997; Moyeed and Papritz, 2002; Koike and Matsuda, 2003). Consequently, in at least some cases, there is room to improve the interpolations produced by conventional kriging procedures.

As are all interpolators, kriging is based on the assumption of spatial autocorrelation, which is one of the foundations of modern spatial statistics. Spatial autocorrelation is the assumption that two measurements of a single variable tend to be more similar if they are taken close together in space than two measurements of that variable taken farther apart (Tobler, 1970; Dean and Giroux-Hughes, 2004). Using this assumption, the value of a variable at an unmeasured site can be estimated based on (1) the value of the same variable measured at known sample sites, and (2) the relative locations of measured and unmeasured sites.

In order to describe the *nature* of the correlation between measurements, kriging performs a preliminary semivariogram analysis of the sample dataset (Jones, 1997). The semivariogram model produced by this analysis relates the difference between two measurements of a single variable taken at known locations (i.e., the semivariance) to the

distance between the sample locations (i.e., the lag).

Kriging uses the semivariogram model to establish a weight for each sample point located within the vicinity of an unsampled point where the value of the variable of interest is to be estimated (Jones, 1997). The weights assigned to the sample points are negatively proportional to their distance from the point being interpolated. Once the weights are determined, kriging estimates the value of the variable at the unsampled point as the weighted sum of the values of the same variable measured at the sample points. Kriging interpolation weights are chosen in order to optimize the interpolation function, i.e. to provide a Best Linear Unbiased Estimation (BLUE) of the value of that variable at the point to be interpolated (Burrough and McDonnell, 1998).

Linear regression is used in kriging's semivariogram analysis to quantify the relationship between lag and semivariance. However, the spatial data used in this analysis does not conform to all of the assumptions inherent to regression. For example, regression assumes that data points used in the analysis are independent from each other. In kriging, sample points are paired up for use in semivariogram development. This results in each original sample point belonging to multiple data pairs; consequently, the pairs observations used in semivariogram analysis cannot be considered completely independent of one another. In addition, regression assumes equal variances over the range of data, while the data pairs used in semivariogram analysis present unequal variances (heteroskedasticity). Finally, the typically noisy data seen in semivariogram analysis makes it difficult to detect the precise form of the relationship between lag and semivariance. Consequently,

subjective procedures are used to simplify the data and make detection of model forms easier. These *ad hoc* procedures are almost certainly less than optimal.

Due to these problems, regression analysis as it is used in kriging may not reliably find the best relationship between semivariance and lag. Therefore, regression may be the source of some or all of the errors in the estimated values produced by kriging. Thus, it is plausible to hypothesize that the overall accuracy of kriging could be improved by replacing kriging's regression component with an alternative modeling procedure that does not suffer from regression's limitations.

Since at least the late 1990's, *A*rtificial *N*eural *N*etworks (a.k.a., ANNs or neural networks) have emerged as an alternative to traditional statistical techniques in the development of predictive models for a variety of applications. In previous studies, ANNs often have been shown to have at least equal, and frequently superior, predictive capabilities than standard regression techniques (Bishop, 1995; Ripley, 1996; Blackard and Dean, 1999). ANNs have the advantage of making no assumptions about the form of the relationships between variables, the statistical distribution and variance properties of the data, or the independence of the observations (Hassoun, 1995). These qualities make ANNs a good alternative to regression analysis for quantifying the relationship between semivariance and lag in semivariogram analysis.

This study explored the ability of a commonly used multilayer feedforward neural network to correct the limitations of regression when employed in conventional kriging. A hybrid ANN-based kriging model was developed, in which a neural network replaced

traditional kriging's regression-based approach used to develop the semivariogram, while all other properties of traditional kriging were retained. The results from this hybrid model were compared to those obtained from the conventional kriging procedure.

1.2. Objectives

The overall objectives of this study were (1) to develop a hybrid kriging/ANN (KrigANN) model designed to improve kriging's predictive accuracy by taking advantage of ANN's unique pattern recognition capabilities, and (2) to test, evaluate, and compare the predictive accuracies of the KrigANN and traditional kriging models. The accuracy and precision of the predictions produced by both models were evaluated based on how closely they reproduced the values recorded in over 2000 artificial datasets created for this purpose.

1.3. Organization of this document

This document is organized in the following chapters. Chapter 2 provides the reader with a background of kriging interpolation techniques, artificial neural networks and regression analysis in the context of kriging and semivariogram analysis. This chapter also presents a literature review of previous studies aimed at improving kriging and semivariogram analysis. Chapter 3 presents a detailed description of the methods and procedures utilized in this study, including generation of the analyzed datasets, data

sampling, building of the conventional and ANN-based kriging models, experimental design, model comparison, and software and hardware utilized. The results of the study are presented and discussed in Chapter 4. Finally, Chapter 6 presents the conclusion of the study and recommendations for further research.

Chapter 2

2. LITERATURE REVIEW

2.1. Definition and Need for Spatial Interpolation

A basic problem that faces researchers in many disciplines is the need to convert measurements of a continuous attribute of interest taken at a finite number of sample points to maps showing the variation of the attribute over space. Spatial interpolation procedures address this issue by allowing the estimation of the value of a variable at locations where it was not measured, based on the measurements of that variable made at other sites within the same region. While only providing estimates at a single point, these procedures can be applied repeatedly, allowing for the creation of a matrix (or raster grid) of estimated values that shows the distribution of the attribute of interest over space. These raster grids can be created at any desired level of spatial resolution, and can be used in further spatial data analysis and/or display.

Spatial interpolation techniques are fundamental in geographic information science and related disciplines such as cartography, geography, and remote sensing, and they are needed in a variety of contexts where spatial data is used (Lam, 1983). Examples of

practical fields of application include:

- Natural resources management and environmental assessment studies (Webster and Oliver, 1990; US Army Corps of Engineers, 1997);
- Topography (Peuker et al. 1978; U.S. Geological Survey, 1987; Lee, 1991; Wingle, 1992);
- Agriculture and precision farming (Hosseini et al., 1994; Schloeder et al., 2001);
- Climatology and meteorology (Genton and Furrer, 1998; Matsoukas et al., 1999).

An interesting survey on some other scientific fields of application of interpolation methods is provided by Foley and Hagen (1994). In practice, interpolation finds application in all instances where a continuous surface must be generated from values measured at discrete locations. It is also mandatory in cases where maps of a discrete variable that cannot be measured at all pertinent locations are needed. Interpolation procedures are also used when transforming a raster dataset to change the resolution or orientation of the matrix in a process called re-sampling (Weibel and Heller, 1991). Considering the wide range of applications where spatial interpolation methods are needed, they are clearly a problem of interest to a great percentage of users of spatial data (Demirhan et al., 2003; Dean and Giroux-Hughes, 2004).

Spatial interpolation has been defined by Longley et al. (2001) as "intelligent guesswork." The rationale behind this guesswork is the assumption of spatial autocorrelation, one of the foundations of modern spatial statistics and geospatial analysis in general. The fundamental concept of spatial autocorrelation can be described by Tobler's First Law of geography: "everything is related to everything else, but near things

are more related than distant things'' (Tobler, 1970; Miller, 2004). In other words, spatial autocorrelation is the idea that two measurements of a single variable taken close together in space have a higher probability of being similar to one another than two measurements of that variable taken farther apart. Using the assumption of autocorrelation, most interpolation methods predict the value of an attribute Z at an unmeasured location based on (1) the value of Z measured at nearby sample points and (2) the proximity of the unmeasured location to sample sites (Isaaks and Srivastava, 1989; Cressie, 1993).

Existing spatial interpolation methods are reviewed and described in detail by a large number of authors (Ripley, 1981; Burrough, 1986; Burrough and McDonnell, 1998; Longley et al., 2001). The remainder of this section will introduce kriging, one of the most widely used interpolation techniques available today.

Interpolation methods vary from being relatively straightforward and requiring only a basic understanding of simple statistical methods, to more complex procedures requiring a deeper understanding of the concepts of geospatial statistics and spatial autocorrelation (Burrough and McDonnell, 1998). The choice of one interpolation method over another depends upon the type and complexity of the problem under investigation, the available data, the desired level of accuracy, and the available computational capabilities (Lam, 1983). Lam's (1983) review provides a simple tool for choosing the appropriate spatial interpolation technique for different applications.

Interpolation procedures can be classified into two main groups: Global and local. Global methods use all available data to determine a single function providing predictions

for the entire area of interest, and dismiss short-range local variations as random noise (Burrough and McDonnell, 1998). Global methods include regression-based approaches such as classification models and trend surface analysis. Local interpolation methods use only the data available within a restricted search area around the point where a value is to be predicted to find a mathematical function that will produce a prediction for that single point. This procedure is repeated until estimated values have been predicted for all of the unmeasured points within the area of interest. Common examples include linear and inverse distance weighting (IDW), low order polynomials, Thiessen (also called Dirichet or Voronoi) polygons, Delaunay triangulation, and spline functions such as thin plates smoothing splines (Burrough and McDonnell, 1998).

2.2. Overview of Kriging

2.2.1. Kriging as a Spatial Interpolator

Kriging (Isaaks and Srivastava, 1989; Oliver and Webster, 1990; Cressie, 1993; Burrough and McDonnell, 1998) is a hybrid interpolation procedure originated by Krige (1951) at the Paris School of Mine, and further developed by Matheron (1971) for the estimation of ore reserves in mining applications. A history of kriging is provided by Cressie (1990). Kriging is one of the most popular spatial interpolation techniques in use today, and it has found applications in numerous areas besides mining, including hydrology, agriculture, natural resource management, and environmental sciences (Oliver and Webster, 1990). A search of the scientific literature for articles describing kriging

applications and evaluations over the last decade returned more than 1,300 examples. At least since 1993, kriging and co-kriging techniques were suggested by the U.S. Fish and Wildlife Service as appropriate approaches for producing maps of estimated water quality parameters from discrete data collected at sampling points (Gaugush, 1993).

The term "kriging" encompasses a set of related methods for interpolation including variants such as simple and ordinary kriging, nonlinear kriging, block kriging, universal kriging, disjunctive kriging, and co-kriging, the latter being a multivariate extension of ordinary kriging. All versions of kriging are firmly grounded in theory in the sense that they adopt a mathematically rigorous approach based on Regionalized Variable Theory. Kriging is described in different levels of details by several authors (Journel and Huijbregts, 1978; Isaaks and Srivastava, 1989; Cressie, 1993; Burrough and McDonnell, 1998).

2.2.2. Theoretical Basis for Kriging

Kriging is a hybrid interpolation procedure in the sense that it starts by performing an initial preliminary analysis of all sampled data points to establish the nature of the relationships between the differences in the values measured at sample points and the distance between sample points (Jones, 1997; Longley et al., 2001). This is termed semivariogram analysis and since it involves all sample points, it is a form of global interpolation. However, once the semivariogram analysis is complete, its results are applied to local groups of sample points to estimate values for an unsampled point (Oliver and Webster, 1990). Thus, kriging is a hybrid interpolator because it includes both global

and local components.

Kriging is based upon the principles of Regionalized Variable Theory (RVT), which is fundamental to all geospatial statistics (Isaaks and Srivastava, 1989). In simple terms, RVT attempts to explain the variation of any property (variable) that varies as a function of its position in space (i.e., a regionalized random variable). Detailed description of the theory is provided by Journel and Huijbregts (1978), Isaaks and Srivastava (1989), Cressie (1993), Burrough and McDonnell (1998), and others.

According to RVT, the spatial variation of any regionalized random variable can be expressed as the sum of three major components (Burrough and McDonnell, 1998):

$$Z_{(s)} = m_{(s)} + \varepsilon'_{(s)} + \varepsilon'' \qquad (2.1)$$

where $Z(s)$ is the value of a variable Z at a location s; $m(s)$ is a deterministic function describing the "structural component" of the value of Z as a function of Z's location, s. The $\varepsilon'(s)$ component is a spatially correlated but otherwise random variation, known as the variation error related to space; this term represents the error in $m(s)$ estimated values that are related to the location (s) where an estimate is being generated. The ε'' component is a completely random, spatially uncorrelated noise or residual error, having a mean of zero and variance σ^2. This is the error in $m(s)$ estimated values that cannot be attributed to the location where an estimate is being generated.

Two additional assumptions are necessary to operationalize RVT: The assumptions of stationarity of difference and variance of difference (Cressie, 1993). Stationarity of

difference assumes that the spatial variation of a variable is statistically homogeneous throughout the surface, which means that the same pattern of variation can be observed at all locations on the surface (Isaaks and Srivastava, 1989; Cressie, 1993; E.S.R.I., 2003). This implies that the relationships within any set of data points remain the same regardless of which point is selected as the starting point from which the relationships are investigated (Kaluzny et al., 1998). In Equation 2.1, this implies that *m(s)* has a constant form, regardless of location (Burrough and McDonnell, 1998).

Variance of difference implies that the variability of a regionalized random variable is defined only through the magnitude of the distance between two sites at which the variable is measured, and not through the location of the sites (Isaaks and Srivastava, 1989; Cressie, 1993; Kaluzni et al., 1998). This means that once the structural components of the variation have been accounted for, the remaining variation is homogeneous and differences between sites are only a function of their distance (Burrough and McDonnell, 1998).

Together, the two assumptions just lead to Equations 2.2 and 2. 3 (Cressie, 1993):

$$E[Z_{(s)} - Z_{(s+h)}] = 0 \tag{2.2}$$

$$\text{var}[Z_{(s)} - Z_{(s+h)}] = 2\gamma(h) \tag{2.3}$$

where *(s)* and *(s+h)* are two locations separated by a distance vector *h*, *Z(s)* is the value of variable *Z* at location *(s)*, *Z(s+h)* is the value of variable *Z* at location *(s+h)*, the term *E[Z(s) – Z(s+h)]* is the expected difference between the value of *Z* at the two points, and the term *var[Z(s) – Z(s+h)]* is the variance between the values of *Z* separated by a

distance *(h)*. When Equation 2.2 is true, *m(s)* equals the mean value of the dataset, and the mean value of the term $\varepsilon'(s)$ is zero.

The function $2\gamma(h)$ from Equation 2.3 is called the variogram by most authors (Matheron, 1971; Isaaks and Srivastava, 1989; Cressie, 1993), and its magnitude is referred to as variance. The $\gamma(h)$ function and its magnitude are called, respectively, the semivariogram and semivariance. Since there are an infinite number of possible h values, the true semivariogram is unknown and it can only be estimated.

The semivariogram function characterizes and quantifies the variation in space of the regionalized variable, and the semivariance is a measure of the deviation of the values of the regionalized variable between pairs of data points at a certain distance from one another. Recalling the statistical definition of variance (often denoted by σ^2), the semivariance between two points is defined as (Cressie, 1993):

$$\hat{\gamma}(h) = \frac{1}{2}[Z_{(s)} - Z_{(s+h)}]^2 \qquad (2.4)$$

The ½ term in the semivariance indicates that each pair of data points is used only once in the analysis, while all possible data pairs are used to determine variance.

If the conditions specified by the previously described assumptions are met, the semivariance of the entire dataset can be estimated from observed data using the semivariogram estimator (Cressie, 1993), expressed by:

$$\hat{\gamma}(h) = \frac{1}{2} \frac{\sum_{i=1}^{n} [Z_{(s_i)} - Z_{(s_i+h)}]^2}{n} \qquad (2.5)$$

where n is the number of experimental pairs of sample data points separated by the distance vector h; $Z(s)$ is the value of variable Z at location (s), and $Z(s+h)$ is the value of variable Z at location $(s+h)$.

Recall that structural component of $Z(s)$'s value from Equation 2.1 denoted the portion of $Z(s)$'s value that is a systematic function of the location s. Equation 2.5 now indicates that this structural component is also a function of $Z(s+h)$ and the semivariance of the h vector. Thus, Equation 2.1 can be rewritten as Equation 2.6:

$$Z_{(s)} = f(Z_{(s+h)}, \hat{\gamma}_{(h)}) + \varepsilon'_{(s)} + \varepsilon'' \qquad (2.6)$$

This indicates that once the spatially correlated but random noise $\varepsilon'(s)$ and the random, spatially uncorrelated noise ε' are accounted for, the variation in space of the regionalized variable Z can be described by the deterministic function represented by $f(Z_{(s+h)}, \hat{\gamma}_{(h)})$, which is simply a reformatting of the definition equation of $\hat{\gamma}_{(h)}$, i.e., Equation 2.5.

2.2.3. Semivariogram Analysis

The semivariogram, essential to regionalized variable theory and to kriging, describes the nature of the spatial autocorrelation between the measured sample points

within the area under investigation (Burrough and McDonnell, 1998). The spatial autocorrelation is represented in the semivariogram by expressing some index of the difference between two measurements of a single variable as a function of the distance between the locations where the two measurements were taken. The index of difference used in most semivariogram analyses is the semivariance, which is simply half the squared difference between the two measurements. The semivariance between any two sampled data points $Z(s)$ and $Z(s+h)$ separated by a distance vector h is expressed by the Equation 2.4 above.

The distance vector h between the two data points $Z(s)$ and $Z(s+h)$ is called the lag. When not directly measured, the lag is usually expressed as the Euclidean distance in the X/Y plane between two points of coordinates (X_1, Y_1) and (X_2, Y_2), and therefore can be calculated using the Pythagorean Theorem:

$$h = \sqrt{(X_1 - X_2)^2 - (Y_1 - Y_2)^2} \tag{2.7}$$

The semivariogram in Figure 2.1 exhibits the expected form of the relationship between semivariance and lag under the assumption of spatial autocorrelation. At small distances between measurement points (i.e., small lags), the difference between measured values is slight (i.e., small semivariances), and at larger lags, larger semivariances are expected.

The semivariogram in Figure 2.1 also shows the essential elements of a typical semivariogram (Cressie, 1993; Burrough and McDonnell, 1998; E.S.R.I., 2003). The

range is the distance over which lag and semivariance are no longer correlated. If the distance between a measured data point and an unmeasured data point is larger than the range, that data point cannot make a useful contribution to an interpolation occurring at the unmeasured point.

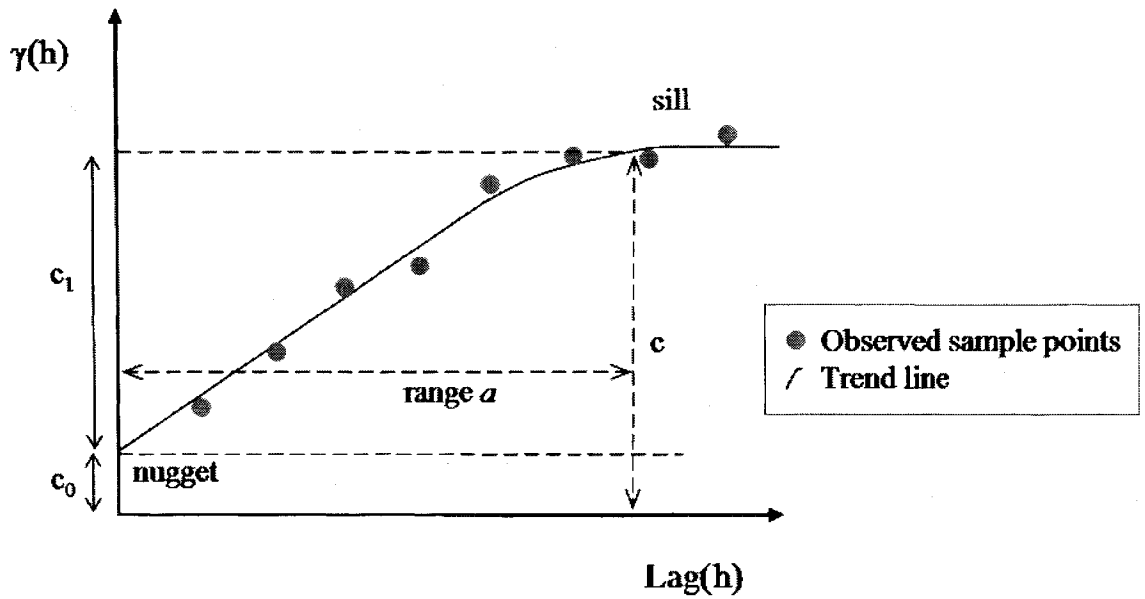

Figure 2.1 A hypothetical isotropic semivariogram (adapted from Burrough and McDonnell, 1998)

The *sill*, indicated by the sum $(c_0 + c_1) = c$, is the value reached by the semivariance when the lag reaches the range; at the sill height, the semivariogram levels off. The *nugget* variance (a.k.a. nugget effect) is a discontinuity at the origin, where the fitted semivariogram model does not pass through the origin, but intersects the *Y* axis at a positive value of $\gamma(h)$. The nugget effect is a measure of the variance among repeated

measurements taken at the same location, which means it represents the error around the true value (Cressie, 1993; Longley et al., 2001). This effect is related to the error term ε" in Equation 2.1.

Only the portion of the semivariogram between the nugget and the sill, or in other words the portion within the range, is useful for interpolation. Near the nugget, semivariance is mostly influenced by random or measurement error. Above the range, semivariance is constant and no longer depends on lag.

The form of the semivariogram curve within the range describes the nature of the spatial autocorrelation within a dataset. In a typical semivariogram analysis, the form of the curve must be specified *a priori*. The five most common forms of semivariogram models used in applied kriging are spherical, circular, exponential, Gaussian, and linear-to-sill. These models can be categorized by the presence or absence of a sill, by the behavior of the variogram near the origin (either linear or parabolic), and by the nugget variance (Journel and Huijbregts, 1978; Cressie, 1993; Burrough and McDonnell, 1998; E.S.R.I., 2003). The shape of the model's curve is characterized by either an a or r parameter (depending on the model form) in the semivariogram equations. These parameters will be called *shape parameters* in the remainder of this document. These five models are shown in Figure 2.2 and are briefly described below.

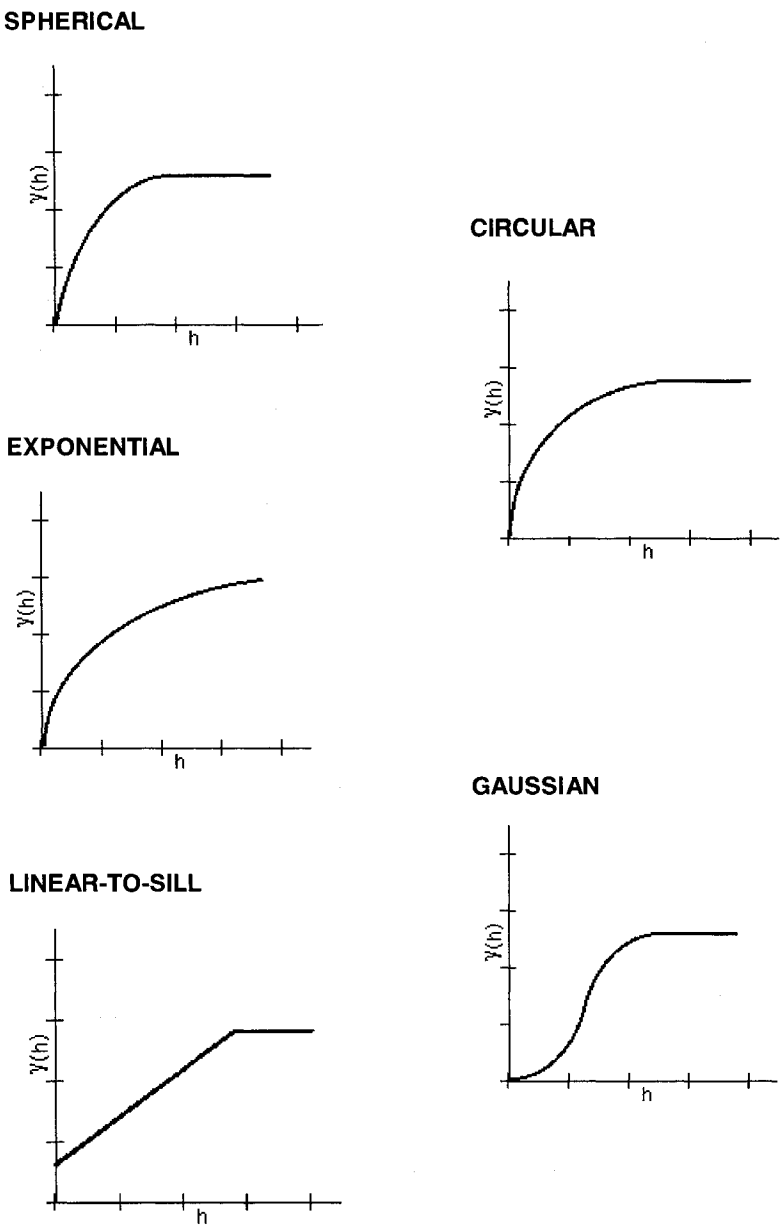

Figure 2.2 Five typical standard semivariogram models: spherical, circular, exponential, Gaussian and linear to sill; *h* is lag and $\gamma(h)$ is semivariance (from E.S.R.I. 2003. ARC/INFO Online Help, Version 8.3).

a) Spherical model

The spherical model is the most commonly used semivariogram form (Isaaks and Srivastava, 1989). This model has a sill and a linear behavior at the origin, and it reaches a sill at a finite distance equal to a (i.e. the range of spatial correlation is a). The standard equation of the spherical model is (Burrough and McDonnell, 1998):

$$\gamma(h) = c_0 + c_1\left\{\frac{3h}{2a} - \frac{1}{2}\left(\frac{h}{a}\right)^3\right\} \quad \text{when } 0 < h \leq a \tag{2.8}$$

$$= c_0 + c_1 \qquad \text{when } h > a$$

$$= c_0 \qquad \text{when } h = 0$$

In Equation 2.8, $\gamma(h)$ is the semivariance, c_0 is the nugget variance, $c = c_0+c_1$ is the sill variance, h is the lag distance, and a is the range.

b) Circular model

The circular model reaches a sill at the range a, and the behavior at the origin is linear. The standard equation of the circular model is (E.S.R.I, 2003):

$$\gamma(h) = c_0 + c_1\left\{1 - \frac{2}{\pi}\cos^{-1}\left(\frac{h}{a}\right) \times \sqrt{1 - \left(\frac{h^2}{a^2}\right)}\right\} \quad \text{when } 0 < h \leq a \tag{2.9}$$

$= c_0 + c_1$ when $h > a$

$= c_0$ when $h = 0$

where all terms are defined as previously.

c) Exponential model

The exponential model is similar to the spherical model, with the major difference that it rises more steeply than the spherical model and then flattens out more gradually; therefore it reaches the sill slower than does the spherical model (Isaaks and Srivastava, 1989; Journel and Hujbreght, 1978). The exponential model has a sill and a linear behavior close to the origin, but it reaches the sill asymptotically. Its standard equation is (Burrough and McDonnell, 1998):

$$\gamma(h) = c_0 + c_1\left\{1 - \exp\left(-\frac{h}{r}\right)\right\} \text{ when } h \neq 0 \qquad (2.10)$$

$= c_0$ when $h = 0$

The term r in Equation 2.10 is the radius and it is a parameter related to the form of the semivariogram model curve. All other terms are as described previously.

d) Gaussian model

The Gaussian model is often used to model extremely continuous phenomena (Isaaks and Srivastava, 1989). This model also reaches a sill, but the behavior at the origin is parabolic. The sill is reached asymptotically (Journel and Huijbreghts, 1978). The standard equation of the Gaussian model is:

$$\gamma(h) = c_0 + c_1\left\{1 - \exp\left(-\frac{h^2}{r^2}\right)\right\} \quad \text{when } h > 0 \tag{2.11}$$

$$= c_0 \qquad \text{when } h = 0$$

All terms are as described previously.

e) Linear-to-sill model

The linear-to-sill model reaches a sill at the range a, with a linear behavior from the origin to the sill. The standardized form of the linear-to-sill model is:

$$\gamma(h) = c_0 + c_1\left(\frac{h}{a}\right) \quad \text{when } 0 < h < a \tag{2.12}$$

$$= c_0 + c_1 \qquad \text{when } h \geq a$$

$$= c_0 \qquad \text{when } h = 0$$

where all the terms are as described previously.

2.2.4. Implementing Kriging

Based on the assumption of spatial autocorrelation and RVT, kriging estimates the value of an attribute Z at an unmeasured location as a function of the spatial relationship of this unmeasured location to measured sites and the values of the Z attribute at the measured sites. Mathematically, the general kriging model can be expressed by the following equation (Dean and Giroux-Hughes, 2004):

$$Z_{(s)} = f(Z_{s+h_1}, h_1, Z_{s+h_2}, h_2, ..., Z_{s+h_n}, h_n) \qquad (2.13)$$

where $Z(s)$ is the value of some variable Z to be interpolated at the unmeasured location (s), $Z(s+h_x)$ represents the value of the variable Z measured at sample location x, and h_x is the lag (distance) between sample point x and unmeasured point (s).

When the variation of Z is a function of both the magnitude of the distance vector h_x and the direction θ_x of the vector connecting points (s) and $(s+h_x)$, the process is called anisotropic, as is the semivariogram representing it. Anisotropic processes imply that the variation of Z is not the same in all directions for a given distance between points (Cressie, 1993) and involve adding a θ_x term to Equation 2.13. When the variation of Z is purely a function of the magnitude of the distance vector h between two spatial locations, and it is not a function of direction, the variogram is called isotropic. The present study will consider only isotropic processes.

Kriging is essentially a weighted averaging technique. The Z value of a variable at unmeasured locations is estimated as the weighted linear combination of measured Z values at sampled locations, where the weights assigned to each sampled point are negatively proportional to their distance from the point being interpolated. This is expressed by the following equation (Burrough and McDonnell, 1998):

$$\hat{Z}(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i) \qquad (2.14)$$

where $\hat{Z}(s_0)$ is the estimated value of the variable Z at an unmeasured location (s_0), $Z(s_i)$ is the measured value of Z at the sampled locations s_1, s_2, ... s_n, and λ_i is the weight associated to the location (s_i). The sum of the weights $\sum \lambda_i$ must be equal to 1 to ensure that the estimate is unbiased (i.e. mean residual error equal to 0) and to guarantee that the weights are chosen to minimize the estimation variance. For these reasons, ordinary kriging is considered a Best Linear Unbiased Estimator (BLUE) (Isaaks and Srivastava, 1989; Lucifredi et al., 2000). The term "best" refers to the minimization of the variance σ^2 of the prediction errors; the term "linear" implies that the predictions being made are constructed as linear combinations of the observed data; and the term "unbiased" refers to the aim of ensuring that the mean residual error is equal to zero (Isaaks and Srivastava, 1989). In addition, kriging is considered an "exact interpolator", because the predicted and observed value of the variable being interpolated are equal at the sample points.

Kriging is also regarded as an optimal interpolator because in addition to unbiased overall estimates, individual estimation variances (errors) can be determined for each point

at which a Z value is estimated and can be mapped like the estimates themselves. The user can therefore determine what confidence can be assigned to each estimate. The ability to calculate estimation variances for each estimate differentiates kriging from many other interpolations techniques (Oliver and Webster, 1990; Schultz at al., 1998).

Kriging uses semivariogram analysis to determine the optimal set of weights (i.e., the λ_i values from Equation 2.14) for interpolation. In typical kriging, semivariogram analysis is conducted by pairing up observed measurements of a variable Z from a sampled dataset to build what we will call a pairs dataset, recording observed lags and semivariances between measured sample pairs. A trend model is then fit to this pairs dataset using one of the previously described standard model forms and some type of regression analysis. This trend model is termed the experimental semivariogram to differentiate it from the pairs data from which it is derived, which is termed the observed semivariogram.

Semivariogram model fitting is a fundamental step of kriging, because the optimal semivariogram model is needed to find the best set of weights for interpolation. Lack of attention to this important step would lead to inaccurate kriging estimates. Semivariogram estimation has been subject of many studies spanning several decades (Cressie, 1985; McBratney and Webster, 1986; Genton, 1998a; Genton 1998b; Muller and Zimmerman, 1999; Maglione and Di Blasi, 2004).

In standard kriging, the mathematical form of the theoretical semivariogram model providing the best possible fit to the observed data points must be specified *a priori* by the analyst based on experience, or must be found after trial-and-error experimentation

through an Exploratory Data Analysis (EDA) process (Hartwig and Dearing, 1979; Kaluzny et al., 1998). The semivariogram model is usually chosen from among five common models mentioned previously, but the shape parameters of these models (e.g. a in the spherical, circular, and linear-to-sill models, and r in the exponential and Gaussian models) must be found using trial-and-error and/or heuristic techniques.

Several methods are used for evaluating the goodness-of-fit of experimental semivariogram models, including the Maximum Likelihood (ML) method, the Restricted Maximum Likelihood or Residual Maximum Likelihood (REML) method, the Minimum Norm Quadratic (MINQ) estimation method, and the Least Squares methods. Extensive literature exists regarding these methods (Journel and Huijbregts, 1978; Isaaks and Srivastava, 1989; Cressie, 1993). The least-squares-based goodness-of-fit criteria are recommended by Cressie (1993) because they require the fewest assumptions about the distribution of the Z data. This group of methods measures the closeness of the observed semivariogram to the plot of the experimental semivariogram by mean of the sum of the squared errors (SSE), or residuals (David, 1977; Journel and Huijbregt, 1978; Clark 1979; Cressie, 1993). The SSE is defined as the sum of the squared differences between experimental (observed) semivariance and theoretical model's (predicted) semivariance for all data pairs (Cressie, 1993):

$$SSE = \sum_{i=1}^{n} \left(\gamma(h) - \hat{\gamma}(h) \right)^2 \qquad (2.15)$$

where $\gamma(h)$ is the observed and $\hat{\gamma}(h)$ is the predicted semivariance.

A comparison of different techniques for semivariogram fitting by Zimmermann and Zimmermann (1991) suggested that the application of more convoluted and more computationally intensive techniques does not significantly improve results compared to ordinary least squares or weighted least squares regression methods (Kaluzny et al., 1998). Thus, the ordinary least squares method was used in this study for semivariogram estimation of standard kriging. This is in keeping with common practice, where the overwhelming majority of kriging applications use ordinary least squares techniques.

Once the form and the parameters of the best experimental semivariogram model are determined, the model can be used to calculate weights to be used for interpolation. The weights are chosen in order to produce estimates of $Z(s)$ with mean residual error equal to zero (unbiased estimates) and to minimize the prediction error (error variance) for the selected semivariogram model. The kriging prediction error at any interpolated location s_0, also called kriging variance, is given by (Burrough and McDonnell, 1998):

$$\hat{\sigma}_e^2 = \sum_{i=1}^{n} \lambda_i \gamma(s_i, s_0) + \phi \qquad (2.16)$$

Where $\gamma(s_i,s_0)$ is the estimated semivariance of Z between the observed location s_i and the unmeasured location s_0, and λ_i is the interpolation weight associated with location s_i. The term φ is a Lagrange multiplier and is required to assure the minimization of the function (Taylor, 1955). It can be demonstrated that the minimum prediction error for the

semivariogram is reached when the semivariance of Z between the sampled point s_j and the unsampled point s_0 is equal to the sum of the semivariances of Z between each observed data pair (s_i, s_j) multiplied by the associated weight λ (Cressie, 1993; Isaaks and Srivastava, 1989; Burrough and McDonnell, 1998). That is, when:

$$\gamma(s_j, s_0) = \sum_{i=1}^{n} \lambda_i \gamma(s_i, s_j) + \phi \quad \text{for all } j \tag{2.17}$$

The set of weights that satisfy Equation 2.17 is calculated using matrix algebra operations. The derivation of these operations is explained in detail by Journel and Huijbregts (1978), Isaaks and Srivastava (1989), and Cressie (1993), among others. In practice, the set of weights can be calculated using Equation 2.18, which is the matrix algebra equivalent of Equation 2.17 (Isaaks and Srivastava, 1989; Burrough & McDonnell, 1998):

$$\left[\frac{\lambda}{\phi}\right] = A^{-1}b \tag{2.18}$$

where λ is the vector of weights, A^{-1} is the inverse matrix of semivariances between pairs of sampled data points, and b is the vector of semivariances between the point to be estimated and each sampled data point. The semivariances between all point pairs represented by A and b are calculated using the fitted theoretical semivariogram equation.

Finally, the value of Z at unmeasured point s_0 is estimated as the sum of the values at the sampled points s_i, each multiplied by its corresponding weight (Burrough and

McDonnell, 1998):

$$Z(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i) \qquad (2.19)$$

The kriging procedure just described can be repeated for as many unmeasured points as desired. When the unmeasured points are arranged to represent the cell center of a raster grid, an estimation surface describing the distribution of the Z attribute is produced.

2.2.5. Previous Literature Regarding Kriging

Kriging is regarded as an optimal unbiased spatial interpolator and it has been widely used in a variety of fields. Estimation of the spatial distribution of contaminants in the environment is a common application. For example, Little et al. (1997) evaluated and compared eight methods of ordinary and universal kriging for predicting contaminants and water quality parameters in an urbanized estuary in South Carolina. The differentiating factors for the eight methods were (1) Euclidean distance ("as the crow flies") versus in-water distance ("as the fish swim"), (2) form of the experimental semivariogram, and (3) use of a model trend component identified as the distance from the unknown site to the inlet mouth of the estuary. The use of in-water distance versus Euclidean distance, as well as the choice of the other factors, did not show consistent or dramatic improvement of kriging's prediction accuracy for any given setting and any given interpolation variable. However, the study pointed out the ability of the kriging interpolation method to adapt to the structure of the data. The results suggested further research for the development of improved GIS-based kriging analysis models allowing the use of more appropriate

parameters for the aquatic environment, such as in-water distances, and the development of more valid significance tests for the comparison of various methods.

In a study by Critto et al. (2003), both kriging and Principal Component Analysis (PCA) were used for the characterization of soil and groundwater properties surrounding a contaminated illegal landfill site near the Venice lagoon (Italy). Both kriging and PCA proved to be effective for risk assessment and analysis, allowing the development of a conceptual model and the identification of exposure scenarios from a relatively small dataset. In particular, kriging was applied to a few measured parameters considered to be good indicators of the migration of contaminants from soil to groundwater. This model increased the researchers' understanding of the system and allowed them to draw conclusions regarding the soil as a potential source of contaminants for the underlying aquifer, and regarding the aquifer as a potential pathway for contamination of the lagoon ecosystem.

Liu at al. (2004) used indicator kriging to evaluate the arsenic contamination potential in three coastal aquifers of Yun-Lin County in Taiwan. Indicator kriging estimates the probability of exceeding a specific threshold value at a given location. Taking advantage of a high correlation between measured arsenic concentration at measured depths within the aquifer, and considering the high association of arsenic concentration and source with sedimentary formations, indicator kriging was successfully used to predict the probability of arsenic concentration exceeding the water-quality standards within the aquifers. The indicator kriging method was chosen to overcome the scarceness of available groundwater

quality data from monitoring wells. The study allowed identification of areas posing different levels of risk to human health, and was proposed as a useful guide for water service companies to identify suitable aquifers for use as drinking water supply.

Other interesting examples of applications of kriging are the estimation of forest characteristics in forest management planning (Gunnarsson et al., 1998), the analysis of the spatial variability of soil properties for precision farming in agronomy and soil science (Bocchi et al., 2000), the estimation of weather parameters such as rainfall in climatological studies (Yuan and Duchon, 2001), and the development of geophysical models for seismic monitoring in geology (Schultz et al., 1998).

2.3. Overview of Regression Analysis

Regression is a well established and widely used statistical procedure that quantifies the relationship between variables. Extensive discussions and critiques of regression analysis can be found in Anderson et al. (1991), Johnston (1991), Kleinbaum et al. (1998), Berk (2004), and Mickey et al. (2004), among others.

The simplest regression model is the simple linear model, where the relationship between two variables is represented by a straight line and expressed by the following:

$$\hat{y} = a + bx \tag{2.20}$$

The term $\hat{y}$ in the equation is the estimated value of an unknown (dependent) variable

whose value we are trying to estimate, x is a known (independent) variable related to y, b is a coefficient indicating the slope of the regression line, and a is a value indicating the Y-intercept of the regression line. The regression model is used to predict the unknown value of the dependent variable based on the measured values of the predictive variable.

In simple regression, the method of the least squares is the most common approach used to find optimal values for the a and b parameters. This method finds a and b values that minimize the sum of the squared errors, or residuals, between the predicted values of the dependent variable produced by the model and the actual (measured) values. Mathematically, this involves minimizing Equation 2.21:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad (2.21)$$

where SSE is the sum of squared errors of the model, y_i is the observed value of the dependent variable for the i^{th} observation, $\hat{y}_i$ is the estimated value of the dependent variable for the i^{th} observation, and n is the number of samples used in the analysis.

Kriging is dependent upon some form of regression analysis to quantify the relationship between semivariance and lag within the semivariogram analysis procedures described in the previous section. The overall accuracy of kriging has been demonstrated for decades, and therefore regression has proven to be reasonably successful in this context. Nevertheless, the spatial data used in kriging analysis to develop the semivariogram model may not conform to all of the assumptions inherent to regression.

One of the assumptions of regression analysis is the independence of each observation used in the regression analysis from all of the other observations used in the analysis. Violating this assumption can result in biased coefficient estimates (Johnston, 1991). The observations used in kriging's semivariogram regression analysis are constructed from pairs of observed measurements of the *Z* variable, and a single observed measurement of the Z variable is part of several pairs. Therefore, since the same observed data points are used to construct multiple point pairs, it is not correct to assume that each paired observation is in fact independent of all other paired observations.

In addition, regression assumes equal variance of the dependent variable over the entire range of the independent variable. This condition is referred to as homoskedasticity. The variance of a set of data is the average squared deviation of each observed data value from the mean for the dataset, expressed by (Johnston, 1980; Anderson et al., 1991):

$$s^2 = \frac{1}{n}\sum_{i=1}^{N}(y_1 - \overline{y})^2 \qquad (2.22)$$

where y_i is the value of the variable y at the i^{th} observation, $\overline{y}$ is the mean value of y for the dataset, and n is the number of observations.

Kriging interpolations are based on the assumption of spatial autocorrelation, which implies larger variance of the dependent variable (i.e. semivariance of the *Z* attribute between paired points) as the value of the independent variable (lag distance) increases. This means that the pairs data used in kriging's semivariogram development process

typically exhibit unequal variances over the range of data. This condition is referred to as heteroskedasticity (shown in Figure 2.3) (Isaaks and Srivastava, 1989). Heteroskedasticity has a negative impact on the consistency of the regression's parameter estimates, so that the estimates for any particular value of the dependent variable would not be as accurate as would the estimates for other values (Isaaks and Srivastava, 1989). Although there are methods for attempting to remove heteroskedasticity before performing regression analysis, such as the logarithmic transformation of the data, they may remove relevant information from the dataset as well, and are not commonly used in kriging analysis (Johnston, 1991).

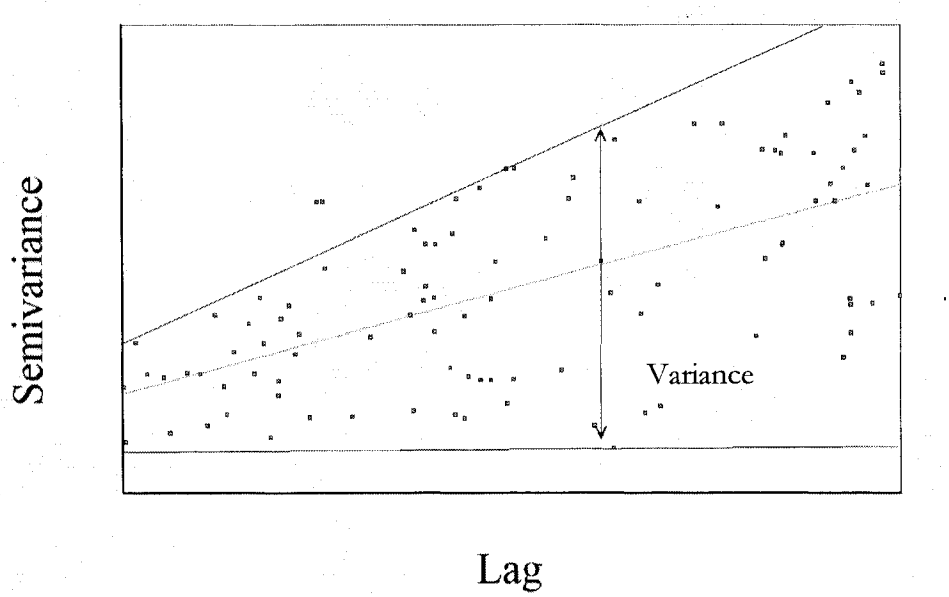

Figure 2.3 Heteroskedasticity of data in an experimental semivariogram.

Regression also depends upon the assumption that all of the variables involved in the analysis have Gaussian (i.e. normal) distributions. The verification of this assumption for

the sample data used to build the semivariogram is often infeasible, because of the lack of very large samples (Johnston, 1980).

Simple regression analysis attempts to fit a straight line to the plot of the observed data. For the regression model to be accurate, it is necessary to ensure that the relationship between the independent and dependent variables is in fact linear (Johnston, 1980). If this is not the case, the raw data can be transformed to achieve linearity. In conventional kriging, semivariogram analysis involves transforming the observed sample data to achieve linearity by fitting one of the five semivariogram models described in the previous section. Transformation of the raw variable values to achieve linearity may introduce problems of interpretation and subjectivity (Johnston, 1980; Dean and Giroux-Hughes, 2004).

Regression analysis is also very sensitive to noise in the data. The observed data used in semivariogram analysis is typically extremely noisy, making it very difficult for the analyst to discern the form of the relationship between semivariance and lag. The noisy data problem in kriging has been addressed by several authors (Oliver and Webster, 1990; Demirhan, 2003). A common technique to reduce noise aggregates the pairs data into classes or groups. Using this technique, the semivariogram plot is not built by using the raw pairs data, but instead by using averaged lag and semivariance values obtained through a *grouping* process. As shown in Figure 2.4, the groups are created by dividing the raw pairs data into m groups having an equal range of lag values. For example, if the lags in the raw pairs data ranges from 0 to 10 distance units, and the number m of groups is

set to 5, the first group would include lags ranging from 0 to 2, the second group would include lags ranging from 2 to 4, and so forth. Once the number of groups and the exact range of lag values within each group are set, the average lag and semivariance values of the pairs data points falling in each group are determined. These average lag and semivariance values are used in the regression portion of semivariogram analysis, instead of the raw pairs data.

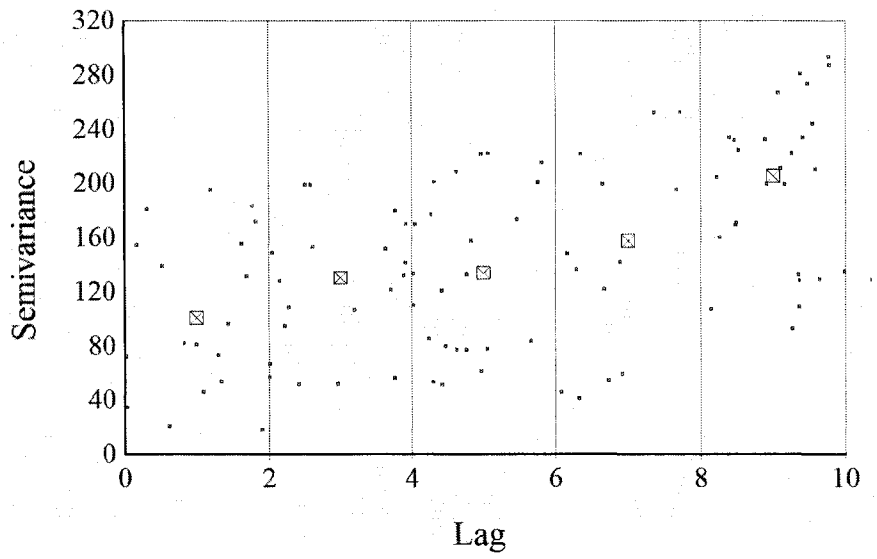

Figure 2.4 Noisy dataset and grouping technique. Raw lag/semivariance observations shown as points; group averages as boxes. (Reprinted by permission from Denis Dean).

Grouping reduces the noise in the data, therefore making the form of the semivariogram model more apparent to the analyst. However, the technique introduces additional subjectivity in the analysis, because there is no objective method to find the optimal grouping characteristics for a set of data, and different grouping characteristics may lead to different results for the same dataset. Furthermore, the removal of noise from

the data by averaging raw semivariances and lags may remove not only spurious noise, but also meaningful data variation.

As a result of all of the above mentioned problems, regression may be a "weak link" preventing kriging from being an even more powerful interpolator. The present study addresses this issue by evaluating the effects of replacing the regression portion of kriging analysis with an alternative method.

2.4. Overview of Artificial Neural Networks (ANN)

Artificial Neural Networks (a.k.a. ANNs or neural networks) are a computer-based form of artificial intelligence (AI) inspired by the design and functioning of the brain. ANNs have the capability to perform intelligent tasks such as learning by example, generalizing learned knowledge, and recognizing patterns (Nigrin, 1993; Haykin, 1994; Bishop, 1995; Hassoun, 1995; Ripley, 1996). Haykin (1994, p.2) provides a definition of neural networks adapted from Aleksander and Morton (1990):

> "A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:
>
> 1) Knowledge is acquired by a network through a learning process
>
> 2) Internode connection strengths known as the synaptic weights are used to store the knowledge."

Early work on neural networks dates back to the 1950's and 1960's, leading to the publication of Minsky's book on "Perceptrons" (Minsky and Papert, 1969; Wasserman,

1989). After a long period of disinterest, neural networks re-emerged in the late 1980's and have since been applied to solve prediction, classification, and control problems in applications as diverse as earth science, atmospheric science, engineering, finance, and medicine (Fisher, 1994; Haykin, 1994; Antonić et al., 2001; Sung et al., 2001; Liu et al. 2002). Neural networks excel at modeling very complex functions, including solving nonlinear problems. In addition, the level of user knowledge needed in treating complex nonlinear problems via ANN-based analysis is lower than needed when applying more traditional nonlinear statistical methods (StatSoft, 2003). In many existing studies, ANNs have been shown to have at least equal, and frequently superior, predictive capabilities than standard regression techniques (Bishop, 1995; Ripley, 1996; Blackard and Dean, 1999).

Over the past decade, researchers in the geospatial sciences have frequently used ANNs together with Geographic Information Systems (GIS) in order to create descriptive and predictive models that benefit from the strengths of both technologies: The spatial capabilities of GIS and the pattern recognition capabilities of ANNs. Linked GIS/ANN models have been most frequently applied to remote sensing problems (e.g., Coulter et al., 1999; Liu et al., 2002; Skidmore et al., 1997; Sunar and Ozkan, 2001; Tapiador and Casanova, 2003). However, they have also being used in a wide range of other applications including ecology (Blackard and Dean, 1996, 1998,1999; Antonić et al., 2001; Hilbert and Ostendorf, 2001), snow cover modeling (Tappeiner et al., 2001), spatial decision making issues (Gimblett and Ball, 1995; Zhou and Civco, 1996), scenic

evaluation of urban landscapes (Sung et al., 2001), and several other fields.

An ANN consists of a number of simple, interconnected processing units called nodes, units, or neurons. In the most common network architecture, the nodes are arranged into layers, typically consisting of one input layer, one output layer, and one or more hidden layers (Figure 2.5). Referring to the layer they belong to, nodes are also referred to as input, output, and hidden. The input layer contains one node for each predictive variable included in the model and the output layer contains a node for each variable to be predicted by the model. One or more hidden layers are located between the input and the output layers. The number of nodes per hidden layer is dictated by the complexity of the relationship between the variables under study.

Nodes are connected by so-called synaptic connections, or synapses, which transport communication signals. Each node in the network processes input signals received from either the external environment (in the case of input nodes) or from connected nodes in the previous layers (for hidden and output nodes). Each node aggregates the signals it receives into a weighted sum, and based on this sum, it generates an output signal.

Once the information has been processed by all layers in the network, the output signals produced by the output nodes are taken as the predicted (output) values produced by the model for the response variables represented by the various output nodes.

The neural network just described presents a type of "multilayer feedforward" architecture. The "multi-layer" designation refers to the number of computational layers.

Only the hidden and output layers are considered computational, because no computation is performed in the input layer nodes (Haykin, 1994). The "feedforward" designation refers to a multilayer network where the signal travels exclusively in the forward direction from the input layer to the output layer. This is in contrast with more complex recurrent networks, where the signal can be transmitted backward from the output or hidden layers toward the input layer, creating one or more feedback loops.

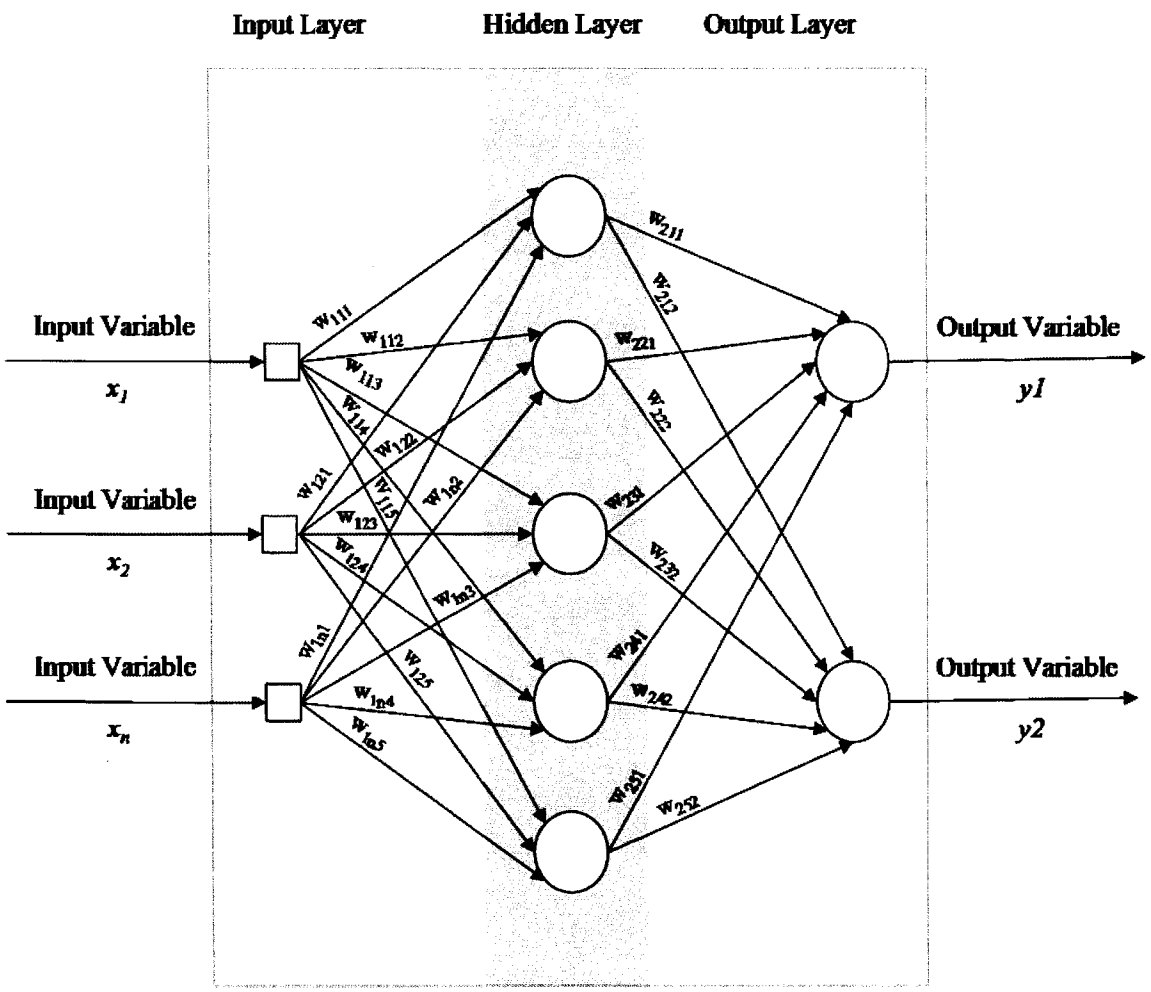

Figure 2.5 Schematic diagram of a simple multilayer feedforward neural network.

Several types of neural networks exist for different types of applications, but the multilayer feedforward network is one of the most common and it has been successfully applied in many applications (Ripley, 1981; Wasserman, 1989; Haykin, 1994; Bishop, 1995). Most feedforward networks are fully connected, meaning that every node in each layer of the network receives a signal from each node in the preceding layer, as depicted in Figure 2.5. It is possible to create partially connected networks, where some of the communication links (synaptic connections) between nodes are absent. However, such partially connected networks are not common (Haykin, 1994).

A fully connected, multilayer, feedforward neural network such as that described, with one input layer, one or more hidden layers, and a single output layer is commonly referred to as multilayer perceptron or MLP network (Haykin, 1994; Bishop, 1995). Multilayer perceptrons are among the most popular neural networks used today, and they have been successfully applied to complex and difficult problems in very diverse fields. The remainder of this section will describe the process of constructing MLP models in greater detail.

The synapses that connect the nodes of an MLP are fundamental to the operations of the neural network model. Each synapse has an associated strength, or weight, called a synaptic weight. These weights influence the strength of the signal received by each hidden or output node in the network. The sum of the weighted inputs received by a node determines the activity level, or activation potential, of the node (Wassermann, 1989). Internally, each node contains a simple mathematical function (known as an activation or

transfer function) which first evaluates the activity level and determines if it is great enough to warrant sending an output signal, and second, if an output signal is warranted, how strong the output should be. In this sense, a node is essentially an information-processing unit similar to the neurons in the mammalian brain (Wassermann, 1989; Haykin, 1994). The standard model of a node with hypothetical synaptic weights and activation function can be represented as shown in Figure 2.6.

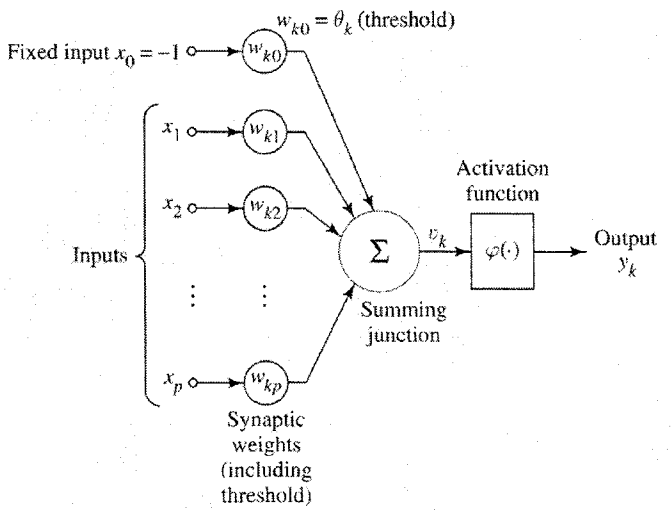

Figure 2.6 The standard model of a hypothetical artificial node (Haykin, 1994, Fig 1.6a).

The model represented in Figure 2.6 shows a set of inputs x_1, x_2,..., x_p applied to the artificial node k. Each input signal is multiplied by an associated synaptic weight w_{k1}, w_{k2}, ..., w_{kp}. All the weighted inputs are aggregated together, producing the activity level signal v_k of the node. The threshold θ_k shown in Figure 2.6 is an external threshold parameter of the node k, which has the effect of lowering the activity level of the node. The threshold

can be considered as an additional weight w_{k0} equal to the desired threshold value θ_k and assigned to a new input signal $x_0 = -1$. The net signal is processed by an activation function $\varphi(\cdot)$, which determines the magnitude of the node's output signal.

This model of a neural network node can be mathematically described by the following (Haykin, 1994):

$$y_k = \varphi_k(v_k) = \varphi_k(-\theta_k + \sum_{j=1}^{p} w_{kj} x_j) \tag{2.23}$$

where y_k is the output signal of the node k; $\varphi_k(\cdot)$ is the node's activation function; v_k is node k's activity level; θ_k is node k's threshold parameter; w_{kj} is one of the $w_{k1}, w_{k2}, ..., w_{kp}$ synaptic weights adjusting the strength of the signal from node j to node k, and $x_1, x_2, ..., x_p$ are the inputs to node k from nodes 1, 2, ..., p. In Equation 2.23, the activation function $\varphi_k(\cdot)$ produces an output signal of zero if the term $-\theta_k + \sum_{j=1}^{p} w_{kj} x_j$ is less than or equal to zero (this is only true for this particular activation function). The equation indicates that the output signal of a node is a function of its net input (or activation potential), given by the sum of the threshold (or otherwise a bias) value and the weighted sum of the inputs (Wasserman, 1989; Haykin, 1994).

Different types of activation functions are available, and it is possible to use different forms of activation function in different nodes within the same network (Gurney, 1997; Dean and Giroux-Hughes, 2004). The threshold function (McCulloch and Pitts, 1943) is

the simplest type of activation function and it is expressed by the following (Haykin, 1994):

$$\varphi(v_k)=\begin{cases}1 & if\ v_k \geq 0\\ 0 & if\ v_k < 0\end{cases} \tag{2.24}$$

According to this function, the amplitude of the output signal y_k of a node k is equal to one if its net input v_k is greater than or equal to zero; the output signal is equal to zero otherwise.

Sigmoid functions are the most common class of activation function used in ANNs. An example of a sigmoid function is the logistic function, commonly used in multilayer perceptrons, and expressed mathematically as (Haykin, 1994):

$$\varphi(v_k)=\frac{1}{1+exp(-av_k)}, \text{ with } -\infty < v_k < \infty \tag{2.25}$$

where a is the slope parameter of the sigmoid function, and the amplitude of the net input lies in the range $-\infty < v_k < \infty$. When the net input v_k signal is processed by a sigmoidal function, the output signal y_k assumes a continuous range of values between zero and one. The ability to assume a continuous range of values distinguishes the sigmoid activation function from the threshold function, where the output signal assumes a value of either zero or one. Continuous differentiability is an important characteristic of the sigmoid activation function, because the derivative of the activation function is used to compute weight updates in backpropagation training (discussed later). The sigmoidal function is

also called a "squashing function," because it compresses (squashes) the possible range of values for the node's output signal, regardless of the value of the net input signal (Wassermann 1989; Haykin, 1994). An additional feature of the sigmoidal function is that it provides a form of control of the nonlinear gain of the node (Wassermann, 1989). Gain is defined as the ratio of the change in magnitude of the output signal to a small change in magnitude of the net input. For small net inputs, the slope of the sigmoidal curve is steep, producing a high gain in output signal. This allows small signals to pass through the function without excessive attenuation, i.e. the output signal cannot become excessively small. As the magnitude of the net input increases, the slope decreases, producing reduced gain in output signal (Wassermann, 1989). In this way, the sigmoid function provides a form of automatic gain control.

Another example of a sigmoidal activation function commonly used in multilayer perceptrons is the hyperbolic tangent function, expressed as (Haykin, 1994):

$$\varphi(v_k) = tanh\left(\frac{v_k}{2}\right) = \frac{1-exp(-v_k)}{1+exp(-v_k)} \tag{2.26}$$

where the magnitude of the output signal $\varphi(v_k)$ lies inside the continuous range from -1 to +1. The hyperbolic tangent function differs from the logistic function in that it has an antisymmetric form with respect to the origin, and the output signal $\varphi(v_k)$ may assume negative values (Haykin, 1994).

As any other model, an ANN must be parameterized before it can be used to infer

unknown information from known information (StatSoft, 2003; Patterson, 1996). Parameterization involves finding values for the synaptic weights, threshold parameters, and coefficients of the activation functions used throughout the network. This is accomplished through a training process based on (1) a training dataset consisting of observed pairs of input and output data, and (2) a set of systematic rules referred to as a training or learning algorithm. During the training process, the neural network is "trained" to learn the relationship between predictive (input) variables and response (output) variables. The objective of training is to develop a network that produces the observed (or at least close to the observed) outputs in response to the application of the observed inputs (Wassermann, 1989).

Numerous training algorithms are available for different types of networks, each having its own strengths and limitations. The four basic types of training processes are error-correction learning, Hebbian learning, competitive learning, and Boltzman learning (Haykin, 1994).

Error-correction learning is based on the iterative correction of the prediction error produced by the network. This class of training algorithms includes the standard backpropagation learning algorithm, which is the most popular and widely used training algorithm available today (Haykin, 1994). This algorithm was independently invented in three separate research efforts (Werbos, 1974; Parker, 1982; Rumelhart, Hinton and Williams, 1986a, 1986b), and it is described in detail in several texts (Bishop, 1995; Hassoun, 1995; Haykin, 1994; Ripley, 1996; Rumelhart et al., 1995).

Backpropagation treats training as an optimization problem where the objective is to reduce the errors between the desired and the predicted responses of the network (Haykin, 1994). Training is accomplished by progressively adjusting the synaptic weights, threshold values, and activation function parameters of the network from some initial set of values (usually determined randomly) until they converge to optimal values such that the application of a determined input produces a desired output (Wassermann, 1989). In neural network training, the term "convergence speed" of the training process is used in reference to the time it takes for the network to find these optimal values (Wassermann, 1989).

The three main tasks involved in training a neural network are: (1) assigning values to the weights associated with the synapses between nodes (i.e., the w_{kj} values in Equation 2.23), (2) assigning values to the threshold parameters of each node, and (3) quantifying the parameters of the activation functions of each node in the network (i.e., the a parameter in Equation 2.25). Collectively, these three items constitute the free parameters of the network model. Training usually starts by specifying *a priori* the form of the activation function (e.g. threshold, logistic, hyperbolic tangent, etc.) to be assigned to each node in the network. It is possible to assign either the same or different activation function forms to different nodes throughout the neural network (Gurney, 1997). Once the appropriate form(s) of activation functions are selected, each free parameter needs to be quantified. In the backpropagation algorithm, a common approach is to set these initial values to small random numbers that are uniformly distributed inside a small range (Haykin, 1994). Typically, all the weights are initialized to random values between -0.1 and 0.1 (Charles Anderson, personal communication, May 30th 2005). Each node's threshold value is also

treated as an additional weight input to the node, where the input signal is fixed as -1 and whose initial random weight is allowed to vary during the subsequent training process (see Figure 2.6) (Haykin, 1994). The activation function's parameters are usually set to an initial value of 1 (Charles Anderson, personal communication, May 30th 2005).

Once initialized, the neural network is presented with a training dataset containing examples of input and output variables, and the errors between observed and predicted response produced by the network are determined. The pairs of predicted and observed response variables are called training pairs (Haykin, 1994; Wassermann, 1989). Weights and threshold values (which, as shown in Figure 2.6, are treated as weights) are iteratively adjusted to reduce the error between observed and predicted response. The process of iteratively computing predicted values and refining network weights (and by extension, threshold values) continues until some predefined stopping criteria is met.

The training process just described leaves the third of the network's free parameters - the activation function parameter - unchanged. This omission is not as significant as it first appears. This situation is analogous to a statistical problem where there are *n* "observations" (free parameters, in this case), but only *n-1* degrees of freedom. When viewed in this light, modification of weights and threshold values effectively achieves the same results as would the adjustment of the activation function's parameters (Charles Anderson, personal communication, May 30th 2005).

Because of the way threshold values are treated during training, throughout the remainder of this discussion, when used in the context of network training, the term

"weights" will refer to both standard weights and threshold values.

The backpropagation learning algorithm relies on differences between observed and predicted response variable values (i.e., the *error signal*) to compute adjustments for the network's synaptic weights. The amount of error for an individual node in the output layer is simply the difference between the observed response for the output variable represented by the node and the predicted response produced by the network. Thus, the error signal $e_k(n)$ of output node k for training pair n is quantified as (Haykin, 1994):

$$e_k(n) = d_k(n) - y_k(n) \tag{2.27}$$

where $d_k(n)$ is the observed response and $y_k(n)$ is the predicted response produced by node k in the neural network for training pair n.

The amount of error for all nodes in the output layer of the network is quantified by using a suitable summation function across all output nodes. This error (or cost) function can take a variety of forms. One of the most commonly used is the sum of the squared errors over all the output nodes (Haykin, 1994; Bishop, 1995):

$$\varepsilon(n) = \frac{1}{2}\sum_{i=1}^{K} e_k{}^2(n) \tag{2.28}$$

where $\varepsilon(n)$ is the error for the network model for training pair n, and the summation is over all the k output nodes of the network (k =1, 2, ... K). Note that the ½ factor in the equation is used to simplify the differentiation of the error function, which will be required

later in the training process (Haykin 1994).

The average squared error E_{av} for the total number of training pairs in the training dataset is given by averaging the sum of squared errors $\varepsilon(n)$ from Equation 2.28 over the training set (Haykin, 1994; Charles Anderson, personal communication, May 30th 2005):

$$E_{av} = \frac{1}{N}\sum_{n=1}^{N}\varepsilon(n) \tag{2.29}$$

where N is the total number of training pairs in the training set, and n is an index identifying one of the *1, 2, ... N* training pairs.

The instantaneous sum of squared errors (Equation 2.28), as well as the average squared error (Equation 2.19), are dependent upon all the free parameters in the network model. The objective of training the neural network is to find the optimal set of free parameters values that minimize the selected error function (Haykin, 1994). To achieve this objective, the error function can be expanded to link the weights to the measure of the average squared error produced by the forward pass of the backpropagation algorithm (Haykin, 1994; Charles Anderson, personal communication, May 30th 2005):

$$E_{av}(W) = \frac{1}{N}\sum_{n=1}^{N}\varepsilon(n, W) \tag{2.30}$$

where $E_{av}(W)$ is the average squared error term over all training pairs in the training set, W is the set of weights to be estimated, and N is the total number of training pairs presented to the neural network. The error function in Equation 2.30 quantifies the

network's predictive error as a function of a specific choice of the synaptic weights.

In backpropagation, which is an intrinsically nonlinear problem, minimization of the error function with respect to the synaptic weights of the network is accomplished using the method of gradient descent (Bishop, 1995). The method of gradient descent, sometimes known as steepest descent, can be visualized by referring to the *error surface* (Haykin, 1994; Bishop, 1995). Figure 2.7 shows the error surface represented as a multidimensional surface created by taking each of the N weights and thresholds of the network as a dimension in space, and the error as the $N+1^{th}$ dimension. The error surface can be plotted for every configuration of the network synaptic weights and thresholds considered during training. Minimization of the error function is reached at the lowest point (along the error dimension) on this surface, which identifies the best combination of values for the free parameters of the network. This lowest point on the error surface is called *global minimum*. Unfortunately, for general networks with multiple layers of synaptic weights, the error function is typically nonlinear, and the error surface is very convoluted, with valleys, saddles, plateaus, and channels (Wassermann, 1989). These may create *local minima*, which are lower than the surrounding surface, but above the global minimum. Local minima represent non-optimal solutions.

As a consequence of the convoluted nature of the error surface, it is not possible to analytically determine where the global minimum is located. Therefore, the optimal combination of weights corresponding to a global minimum is sought through an exploration of the error surface. Starting from arbitrary initial values of the weights, i.e. an

arbitrary point on the error surface, the training algorithm looks for a global minimum by gradually following the slope of the error surface in the direction of steepest descent. But this search may find a local and not the global minimum (Wassermann, 1989; Haykin, 1994).

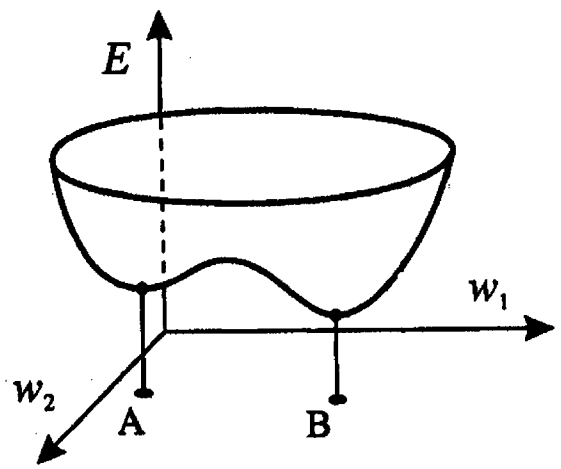

Figure 2.7 A geometrical picture of the error function as a surface sitting above weight space. Points A and B represent minima of the error function (Adapted from Fig. 7.1 in Bishop, 1995).

A useful strategy for avoiding the problem of becoming "trapped" in local minima relies on the way weights are adjusted during training. The neural network is first initialized by setting all weights to random initial values. Training starts with random large weight adjustments, retaining only those adjustments that reduce the error function. The average weight adjustment is then progressively reduced until, hopefully, a global minimum is reached (Wassermann, 1989). This strategy effectively "jumps around" the

error surface, exploring large regions of the surface before zeroing in on an optimal (or nearly optimal) solution. With the gradual reduction of the weight adjustment, the set of weights corresponding to the global minimum should ultimately be reached (Wassermann, 1989).

When the weights are adjusted after each training pair is presented to the network, the training process is said to be in pattern mode. In this mode, a training iteration corresponds to the presentation of one training pair to the network. In practical applications, pattern mode training is impractically slow, so weights are instead adjusted after all training pairs contained in the training dataset are presented to the network. This is termed batch mode (Haykin, 1994). In batch mode, each complete pass through the training dataset is called an epoch, so that a training iteration corresponds to an epoch. The batch mode for training was adopted in this study.

Training occurs through a number of iterations. Ideally, every iteration should make the network more knowledgeable about the training data presented to it, and therefore the prediction error should be reduced. The training progression in the backpropagation algorithm for multilayer perceptrons can be summarized by the following steps (Wassermann, 1989):

1) Select a network configuration with an initial number of hidden layers for the network of an initial number of nodes in each hidden layer. Also, specify the form of activation function (e.g. threshold, logistic, hyperbolic tangent, etc.) that each node should utilize.
2) Assign an initial value to all the free parameters in the network and to the learning rate parameter in Equation 2.31 (explained later).

3) Present the network with a training dataset containing pairs of inputs and corresponding target output values (these are the actual measured values for the output variables) and let the network produce predicted output values.

4) Calculate the prediction error as a measure of the difference between the outputs predicted by the network and the target outputs. If this is the first training iteration, jump to Step 7. Otherwise, go to Step 5.

5) Compare prediction errors from the current and previous training iterations. If the prediction error from the current iteration is lower than that for the previous iteration, go to Step 7. Otherwise, return the network's weight values to those from the previous iteration and then move on to Step 6.

6) Modify the learning rate parameter so that the average size of the weight changes in the current iteration will be different from that in the previous iteration. The learning rate is the term η in the delta rule Equation 2.31 (explained later) and it is used to allow control of the average size of weight changes.

7) Adjust all the weight values by a calculated amount. This amount is computed following an error-correction rule, called the delta rule, defined in Equation 2.31 (explained later).

8) Evaluate the current network using a stopping rule (described shortly). If the stopping rule applies, stop. Otherwise, return to step 3.

Each complete iteration of the backpropagation training algorithm consists of two passes through the network: a forward pass (step 3 in the above procedure) and a backward pass (Step 5) (Haykin, 1994). In the forward pass, signals propagate forward through the network. An input vector containing pairs of predictive variable values is fed to the network, and signals travel forward through the network from the input toward the output layers until the network produces an output vector of predicted values for the response variable. The predicted output vector is compared to the desired response vector and a

measure of the error is calculated.

In the backward pass, the weights (both conventional and the weight associated with the threshold parameter) associated with each node are adjusted according to an error-correction rule designed to minimize the amount of error between predicted and desired response. This correction process starts with the nodes in the output layer and proceeds backward toward the input layer. The backward pass is what gives the backpropagation learning algorithm its name, and it is through this step that training is achieved (Haykin, 1994).

The delta rule (Haykin, 1994), also known as Widrow-Hoff learning (after Widrow and Hoff, 1960), establishes the amount of change to be applied to each weight at each iteration in the training process. Based on the delta rule, the adjustment applied to a given weight is proportional to the product of the error signal produced by the node the synapse leads to and the input signal entering the synapse. This means that the correction applied to a synapse with a large input signal and a large error signal will be greater than the correction applied to a synapse with the same error signal but lower input signal (Haykin, 1994). Using the delta rule, the adjustment $\Delta w_{kj}(n)$ applied to synaptic weight w_{kj} running from node j to node k at training iteration n is expressed by (Widrow and Hoff, 1960; Haykin, 1994):

$$\Delta w_{kj}(n) = \eta\, \delta_k(n)\, y_j(n) \tag{2.31}$$

where η is a learning rate parameter, $\delta_k(n)$ is the local error gradient, and $y_j(n)$ is the

magnitude of the signal from node j to node k. Note that the output signal of node j is the input signal for node k before weight adjustment.

The term η in the delta rule equation is a positive constant determining the learning rate. The learning rate coefficient allows control over the average magnitude of weight changes at each iteration (Wassermann, 1989; Haykin, 1994). This parameter is typically specified *a priori* by the analyst and it is crucial to reaching a good compromise between the speed of training and the obtainment of an optimal solution. If η is too small, it will take a long time for the algorithm to converge to an optimal solution (i.e., find the global minimum). Conversely, if η is too large, the algorithm will move too fast over the error surface, diverging away from the minimum of the error function.

The local error gradient $\delta_k(n)$ is a measure of how a unit change to the synaptic weight w_{kj} for node k at time n will effect the measure of the associated overall error. This factor represents the direction of change in weight space over the error surface toward an optimal solution (global minimum) for the synaptic weight w_{kj} at time n. The local error gradient for a node k at time n is equal to the product of the error signal for the node in question times the derivative of its associated activation function (Haykin, 1994). Calculation of the error signal is simple if the node in question belongs to the output layer, because its desired and predicted outputs are known. However, if the node belongs to a hidden layer there are no specified response values and calculation of the error signal is more complex. Therefore, the computation of the value of the local error gradient depends upon the location of node k in the network. If the node belongs to the output layer, the error signal

can be calculated based on Equation 2.27 as the difference between desired response and predicted response. The local error gradient $\delta_k(n)$ for an output node k is then equal to (Haykin, 1994):

$$\delta_k(n)=e_k(n)\varphi_k(v_k(n)) \quad \text{(node } k \text{ is an output node)} \tag{2.32}$$

where $e_k(n)$ is the error signal for node k at iteration n, and $\varphi_k(v_k(n))$ is the derivative of the node's activation function.

However, if the node in question belongs to a hidden layer, a specified desired response for the node is not available, so the error signal must be determined iteratively from the error signals of all the nodes in the subsequent hidden or output layer to which the node is connected. Thus, the error of each hidden node is quantified as a "shared responsibility" for the error signal produced by the output layer (Haykin, 1994) [♦]. Using the delta rule, the local error gradient $\delta_k(n)$ for a hidden node k at iteration n is defined by the following (Haykin, 1994):

$$\delta_k(n)=\sum_{l=1}^{m}\delta_l(n)w_{lk}(n)\varphi_k(v_k(n)) \quad \text{(node } k \text{ is a hidden node)} \tag{2.33}$$

Where the error signal of hidden node k is calculated as the weighted sum of all the local error gradients δ's computed for all the m nodes in the subsequent hidden or output layer that are connected to node k. The term $w_{lk}(n)$ represents the weight associated with the synaptic connection between node k and the connected node l in the subsequent layer,

[♦] Note that the need for this derivative is the reason why the ½ term was introduced in Equation 2.28.

while the term δ_l is the local error gradient for node l. In the equation, m is the number of nodes connected to node k in the subsequent layer, and l is one of these connected nodes.

Once all the terms in the delta rule are evaluated, the new updated value $w_{kj}(n+1)$ of the synaptic weight between presynaptic node j and postsynaptic node k at iteration $(n+1)$ can be calculated as the sum of the value of the weight at time n plus the weight adjustment calculated using the delta rule (Haykin, 1994):

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{2.34}$$

The iterative training of the network continues through each training epoch, with netwrok weights being refined after each iteration. Theoretically, the ability of the network to predict response variables from the training dataset should improve with each iteration. However, continuing the training process indefinitely would cause the network to become "over-trained" or "over-fitted." An over-trained network focuses its training on nuances of the training data and thus loses its generalizability, which means that while the model may be able to predict values from the training dataset with errors close to zero, it loses the ability to make accurate predictions for other datasets. The problem of over-fitting is not specific to neural network training; it is also encountered in statistical analysis where a probabilistic model is fit to a set of experimental data.

One of the simplest approaches to prevent over-training is called early stopping. Using this approach, the training process continues iteratively until a predetermined stopping criterion is met. This is usually a predetermined value of the root mean square

error (RMSE) of the neural network model's predictions or a set number of training epochs (in the case of batch training). These parameters are typically set *a priori* by the user.

A more sophisticated method for preventing over-training involves the use of a separate dataset independent of the training dataset (Fine, 1999). This second subset, here called testing subset, is used to periodically test the performance of the ANN during the training process. This is accomplished by periodically feeding the testing dataset's predictor variables to the ANN in-training, and computing an RMSE value based on the ANN's performance with the testing dataset. Training of the ANN stops when this testing RMSE is minimized, thereby preventing over-training (Haykin, 1994; Gurney, 1997). Calculation of the RMSE occurs at every X iterations through the training data (epochs in batch training), where X is defined *a priori* by the user.

Once the ANN is fully trained, a third subset of the data, called in this study validation subset, is used to cross-validate and confirm the performance of the model. Since the validation dataset was not used in building the ANN model, it provides an unbiased estimate of the final network's predictive ability. Once cross-validated, the ANN is ready to be used as a predictive model (Bishop, 1995; Ripley, 1996).

Despite its success, backpropagation has its limitations, including the risk of becoming trapped in local minima, very high computational resources requirements for the training process, and extremely low speed of convergence (Wasserman 1989; Haykin 1994; Bishop, 1995). Several variants to the classic backpropagation training algorithm have been proposed to overcome some or all these limitations. For example, the Quickprop

scheme (Fahlman, 1989), the weight decay procedure (Hinton, 1986), and versions of the Extended Kalman Filter (EKF) (Singhal and Wu, 1989) are designed to address some or all of these issues.

The EKF methods provide excellent convergence and quality of solutions and are among the most promising enhanced training methods (Haykin, 2001). However, such improved performance incurs higher computational cost at each iteration and much higher computer storage requirements (Shah and Palmieri, 1990; Lary and Mussa, 2004). The goal of EKF training is the same as that of backpropagation, i.e., reducing the mean square error (Palmieri, 1991a, 1991b). The difference lies in the manner in which the synaptic weights are updated.

Using the EKF approach, a multilayer network can be seen as a non-linear dynamic system whose state is a vector containing all the network weights (Singhal and Wu, 1989). Consequently, training can be treated as a state estimation problem for a non-linear dynamic system (Singhal and Wu, 1989; Lary and Mussa, 2004). In other words, the problem of finding optimal weights can be treated as a non-linear parametric system identification problem (Stan and Kamen, 2000). In the state estimation form, the behavior of the neural network at solution is expressed by the following mathematical equations (Datum et al., 1996; Stan and Kamen, 2000; Haykin, 2001):

$$w_{t+1} = w_t + \omega_t \ , \tag{2.35}$$

$$d_t = g_t(w_t, u_t) + \varepsilon_t \tag{2.36}$$

Where the subscript t denotes discrete time, w_t and w_{t+1} are the vectors containing all the network's weight parameter values at times t and $t+1$, ω_t is the process noise vector (i.e. the model's error), d_t is the desired output vector of the network, expressed as a nonlinear function $g_t(w_t, u_t)$ of the vector w_t of estimated weights and the input vector u_t at time t, and ε_t is the measurement noise vector. Equation 2.35 is called the process equation, and it indicates that the state of the ideal neural network is given by the weight vector w_t, but the actual state of the network is corrupted by the process noise ω_t. Equation 2.36 is called the observation equation, and it describes the target response vector d_t of the network as a non-linear function of the input vector u_t and the weight vector w_t, summed to the measurement noise vector ε_t. Both the process noise ω_t and the measurement noise ε_t are assumed to be independent, zero-mean white noise processes of covariance matrices Q_t and R_t given by the following (Haykin, 2001):

$$E[\omega_t \omega_l^T] = \delta_{t,l} Q_t \tag{2.37}$$

$$E[\varepsilon_t \varepsilon_l^T] = \delta_{t,l} R_t \tag{2.38}$$

where the superscript T denotes matrix transposition. Applying a global EKF-based strategy to the solution of the state estimation problem, the goal of the training process is finding the state vector (i.e. the weight vector w) that minimizes the mean squared error of the predicted state w using all the observed data (Haykin, 2001). The cost function to minimize at time n is (Datum et al., 1996):

$$E(\hat{w}_n) = \frac{1}{2}\sum_{t=1}^{n}\left\| d_t - g(\hat{w}_n, u_t) \right\|^2 \lambda^{n-t} \tag{2.39}$$

where $\hat{w}_n$ is the current estimate of the weight vector, and λ^{n-t} $(0 < \lambda \leq 1)$ is called the "forgetting factor" and it is used to assign exponentially lower weight to the training examples presented in the past. The cost function is minimized at each time step during training (Datum et al., 1996).

For given d_t, g_t, Q_t, and R_t values, the updated estimate of the weight vector w_{t+1} for the network can be obtained using the following extended Kalman recursions (Singhal and Wu, 1989; Haykin, 2001):

$$A_t = \left[R_t + H_t^T P_t H_t\right]^{-1}, \tag{2.40}$$

$$K_t = P_t H_t A_t, \tag{2.41}$$

$$\hat{w}_{t+1} = \hat{w}_t + K_t \xi_t, \tag{2.42}$$

$$P_{t+1} = P_t - K_t H_t^T P_t + Q_t. \tag{2.43}$$

The recursions above indicate that the updated estimate of the weight vector w_{t+1} is a function of the current values of the vector $\hat{w}_t$, the Kalman gain matrix K_t, and the error

vector ξ_t. The vector $\hat{w}_t$ is the estimate of the weight vector (i.e. the state) of the network system at update time step t. The error vector $\xi_t = d_t - \hat{y}_t$ is the error between the desired network's output vector d_t (as seen in Equation 2.36 above) and the predicted network output vector $\hat{y}_t$ for the training sample presented at time step t. Hence, the predicted output vector is $\hat{y}_t = g_t(w_t, u_t)$. The global scaling matrix A_t is used to compute the Kalman gain matrix K_t and it is a function of the measurement noise covariance matrix R_t, the matrix H_t of derivatives of the network's outputs with respects to all weight parameters, and the approximate error covariance matrix P_t. The approximate error covariance matrix P_t includes second-order derivative information about the training problem, and it is updated recursively (at every P_{t+1}) together with the weight vector estimate. The updated matrix P_{t+1} is obtained using the previous values of P_t, the Kalman gain matrix K_t, the derivative matrix H_t, with an augmentation by the process noise covariance matrix Q_t. The Kalman gain matrix K_t is used to update the values of the weight vector (from w_t to w_{t+1}); this matrix is a function of the matrices P_t, H_t, and A_t (Haykin, 2001). It should be noted that the approximate error covariance matrix P_t must be initialized and the process noise covariance matrices R_t and Q_t must be measured at the beginning of training (Haykin, 2001). Also, note that the network weights are updated based on second-order derivative information, as opposed to the first-order derivatives used by backpropagation (Singhal and Wu, 1989; Haykin, 2001). The EKF algorithm computations described above are computationally expensive, and may require

prohibitively high amounts of storage for each iteration (Shah and Palmieri, 1990).

The Multiple Extended Kalman Algorithm (MEKA) is an alternative EKF-based training algorithm which has been proposed by Shah and Palmieri (1990). MEKA is designed to retain the advantages of the EKF approach over the backpropagation algorithm, while limiting the associated costs (Shah and Palmieri, 1990).

The MEKA procedure simplifies the EKF approach by partitioning the global problem into a set of sub-problems, therefore reducing dimensionality (Stan and Kamen, 2000). The partition is scaled down to the level of the single node (Palmieri et al., 1991a). While still computationally intensive, the MEKA algorithm has been shown to reduce convergence times and improve the quality of the solution compared to standard backpropagation, with reasonable complexity and storage requirements during training (Shah and Palmieri, 1990).

The MEKA algorithm solves the global problem of minimizing the cost function for the neural network by independently minimizing the squared error at the output of each node, and implementing multiple applications of the EKF algorithm locally at the level of each node (Palmieri et al., 1990). The local state model equations for each node are the following (Shah and Palmieri, 1990; Datum et al., 1996):

$$w_{i(t+1)} = w_{i(t)} = w_{Oi} \tag{2.44}$$

$$d_{i(t)} = \varphi_{i(t)}(w_{Oi}^{T}, u_{i(t)}) + \varepsilon_{i(t)} \tag{2.45}$$

where $w_{i(t)}$ is the weight vector at the output of the i^{th} node at time step t (i.e. the current state), $d_{i(t)}$ is the desired node's output, $u_{i(t)}$ is the node's input vector, and $\varepsilon_{i(t)}$ is the error at the output of the i_{th} node.

The method used to calculate the error $\varepsilon_{i(t)}$ is similar to that of standard backpropagation. The optimal weight vector for the i^{th} node is the vector providing minimization of the error function. The error function to be minimized at time n becomes (Datum et al., 1996):

$$E(\hat{w}_{i(n)}) = \sum_{t=1}^{n} \varepsilon_{i\,(t)}^{2} \lambda^{n-t} \tag{2.46}$$

Like the EKF algorithm, the weight updates are obtained using multiple extended Kalman recursions (see Equations 2.40 to 2.43 above) implemented at the level of each node (Datum et al., 1996).

Using the MEKA approach, a copy of the P matrix must be maintained for each node, because each node receives its effective input vector, even if such input is shared with other nodes. This creates the need of higher computational resources and storage requirements for MEKA as compared to the backpropagation. However, the algorithm uses only information that is available at each node, and therefore it is still local and less computationally intensive than global EKF approaches (Shah and Palmieri, 1990).

In summary, the steps of MEKA training are the following (Datum et al., 1996):

1) Present the network with an input training dataset until it produces a predicted output;
2) Calculate the error vector ξ;
3) Propagate the error backward to the output of each node and calculate the H matrix;
4) Compute the Kalman gain vector K for each node;
5) Update the weight vector w and the approximate error covariance matrices P for each node;
6) Repeat until the network error is minimized.

The faster convergence of the MEKA and the EKF family of algorithms compared to the backpropagation's gradient descent method is due to the fewer number of training iterations needed to reach convergence (Stan and Kamen, 2000). Backpropagation's gradient descent uses single training samples and derivatives of error to find the direction and magnitude in which weights should change. Considering a training process in pattern mode, the direction of steepest gradient descent toward optimum weights is found at each iteration using the gradient of the network function for the latest training sample (Datum et al., 1996). This means that the error and the gradient are only evaluated for one training sample at each iteration, not for a complete training set, and the gradient descent search is made using only the information about that single training sample. Therefore, it takes several repetitions through the training data for backpropagation to find an optimal solution, and it takes a longer time to train the network (Datum et al., 1996).

This undesirable slow training characteristics of backpropagation does have the advantage that the memory requirements are not excessive, because only weights need to

be stored during training. Conversely, the MEKA algorithm uses more training samples to find the direction of gradient descent and the magnitude of weight adjustments. The direction of steepest gradient descent is found at each iteration using the gradient of the network function for all the training samples presented so far (Datum et al., 1996). This means that the error for one training pair is representative of the error of the complete training set, and the use of the error information about the training set allows a complete search of the gradient descent (Datum et al., 1996). Therefore, the MEKA algorithm requires less training repetitions and it reaches a minimum error faster compared to backpropagation, but every iteration is computationally more complex and requires more memory. Considering these factors, the MEKA algorithm is a viable alternative to backpropagation if an adequate amount of computer storage is available.

Both the backpropagation and the MEKA algorithms were tested for this study. The MEKA algorithm was chosen for the analysis, due to its faster convergence at comparable performance.

2.5. Comparing ANNs and Regression Analysis in the Context of Kriging

Artificial neural networks offer several advantages compared to regression approaches for quantifying the relationship between semivariance and lag in the development of semivariogram models. For example, ANNs have the advantage of making no assumptions about independence of the observations, about the variance properties (homoskedasticity) and statistical distribution of the data, or about the form of the relationship between

variables (Hassoun, 1995). In addition, ANNs are very tolerant to noisy data.

Because neural networks make no assumption about the independence of the observations (Hassoun, 1995), the problems caused by the non-independence of the lag/semivariance paired observations used to develop the semivariogram are eliminated.

Neural networks have the ability to quantify and replicate the relationships between variables regardless of the variance properties of the data; therefore, their performance is not affected by the heteroskedasticity commonly found in semivariogram analysis. In addition, neural networks do not assume a Gaussian distribution of the data, and therefore the dataset does not need to be normalized.

The fact that ANNs make no assumptions about the form of the relationship between variables removes the subjectivity introduced by the *a priori* or EDA selection of the regression form and shape parameters value for the semivariogram model. ANNs do require *a priori* specification of the number of hidden nodes and the forms of the activation functions. Fortunately, in specifying the number of hidden nodes, the analyst needs only to be concerned with the consequence of specifying too few nodes. An insufficient number of hidden nodes would create a model that is too simplistic to capture the nature of the relationship between variables, negatively affecting the predictive capability of the final neural network model. Thus, under-specifying the number of hidden nodes has real negative consequences. Conversely, the only negative consequence of specifying an excessive number of hidden nodes is an increased amount of time and computer resources needed to run the model, while the predictive ability of the network is not impacted

(Gurney, 1997). Therefore, the potential issue posed by *a priori* selection of the number of nodes per hidden layer can be resolved simply by specifying a generous number of hidden nodes.

The issues related to the *a priori* selection of the form of the activation function for each node in the network is more complicated. When using regression analysis, the specific mathematical form of the semivariogram model (e.g. spherical, exponential, etc.) is rigidly specified *a priori* by the analyst, thus limiting the regression analysis to the evaluation of a single mathematical form. However, in neural network analysis the final mathematical form of the model is determined by the combination at least four factors: (1) the mathematical form of each activation function for each node, (2) the parameters assigned to each activation function, (3) the weights assigned to each synaptic connection in the network, and (4) the number of nodes in the network. Therefore, while *a priori* selection of the form of the activation function necessarily limits the range of potential mathematical forms evaluated by the network model, the number of mathematical forms evaluated will still be very large even for the most simplistic neural network (Dean and Giroux-Hughes, 2004).

In addition, ANNs are highly tolerant to noisy data. This should make ANNs capable of analyzing raw semivariance and lag observations. This eliminates the subjectivity associated with the *ad hoc* grouping procedures used in conventional kriging.

It must be noted that although neural networks offer numerous benefits for solving complex problems, they are not a panacea. Neural networks have their own disadvantages

compared to traditional statistical analysis techniques. The most noticeable disadvantage is the increased amount of computational time and computer resources required to construct and run a neural network model. In addition, as discussed earlier, construction of the neural network model does require *a priori* or EDA selection of some values for the initial configuration of the network, such as learning rate, number of epochs, etc. (Dean and Giroux-Hughes, 2004). Furthermore, a neural network model usually incorporates a certain amount of stochasticity, which may result in suboptimal solutions.

ANNs represent a suitable alternative to regression analysis for quantifying the relationship between lag and semivariance in the kriging environment. The replacement of a regression-based approach with an ANN-based approach for developing the semivariogram is expected to improve the predictive ability of kriging, or at least achieve comparable results.

2.6. Previous Efforts to Improve Kriging Procedures

The conventional kriging interpolation technique has been used successfully to solve a variety of problems. Nevertheless, there are instances in which conventional kriging does not perform as well as could be expected. In addition, several examples are available of accepted kriging variants that replace the intrinsically linear regression-based approach of kriging with alternative methods.

For example, Schultz et al. (1998) developed a robust non-stationary Bayesian kriging

algorithm for interpolating spatial correction values in geophysical models used for seismic monitoring. The datasets encountered in this type of application typically suffer from highly non-uniform and sparse spatial distribution, and an appropriate interpolation algorithm must account for these characteristics. Schultz et al. (1998)'s study showed that their method provided reliable predictions and an appropriate estimate of prediction uncertainty.

Moyeed and Papritz (2002) compared the performance of linear and several nonlinear kriging methods, with respect to both the precision of their predicted values and their performance in modeling prediction uncertainty. These methods were applied to measurements of the topsoil concentration of copper and cobalt in the Border Region of Scotland. The authors used two positively skewed datasets with weak spatial dependence, with one dataset showing stronger degree of autocorrelation than the other. They modeled spatial dependence for all their kriging models using semivariograms developed via traditional regression-based approaches. Their study found no clear differences in the precision of ordinary kriging versus a number of standard kriging variants, including disjunctive, indicator, lognormal, and model-based kriging. However, ordinary kriging did not perform as well as the nonlinear methods in modeling the prediction uncertainty, especially for the dataset showing a weaker degree of autocorrelation and more substantial skeweness.

Other authors have proposed alternative kriging methodologies that involve replacing the fundamentally linear regression-based approach to semivariogram analysis found in

conventional kriging with alternative techniques. For example Carle and Fogg (1996, 1997) suggested the application of continuous-lag Markov chain models to geostatistical estimation and simulation techniques, such as cokriging and kriging, in geology. Two- and three-dimensional continuous Markov chains can address patterns that traditional geostatistical semivariogram models cannot.

Walker and Loftis (1997) discussed an alternative to ordinary kriging (which they termed the Least Absolute Deviation method, or LAD) for treating groundwater monitoring data. This sort of data often contains outliers and may not conform to kriging's assumption of normally distributed observations. For this type of data, the proposed LAD method showed improved performance over ordinary kriging.

Emery (2002) described a "sequential indicator simulation method" based on indicator kriging that overcomes the limitations of Gaussian models and allows for simulating regionalized variables with highly skewed distributions, which cannot be appropriately handled in a Gaussian frame.

Saito and Goovaerts (2001) presented a variant of universal kriging (kriging with a trend) for the spatial prediction of contaminants concentration. Their method accounts for the presence of local trends in the data (not conforming to the assumption of local stationarity of ordinary kriging) by integrating information about pollutant source location and transport direction into the spatial mapping of contaminants. Such techniques were shown to outperform ordinary kriging.

We are not aware of any previously published studies that examined the specific hybrid ANN-based kriging modeling technique presented in this study. However, there are several examples of studies where kriging and/or regression based interpolation approaches were compared to alternative systems that involved neural networks.

For example, Mukhopadhyay (1999) compared the performance of traditional kriging to an artificial neural network model for the estimation of transmissivity values in the Dammam Formation Aquifer in Kuwait. The ANN model acted as a global interpolator and produced an output of interpolated transmissivity at unknown well locations, using inputs of geographic coordinates of wells and hydrogeological data measured at well sites. The study suggested that the ANN produced better estimates at well locations as compared to the kriging model, provided that enough training data were available to cover the complete range of variation within the study area, and that appropriate hydrogeological input variables were used.

In a study done by Koike and Matsuda (2003) for a limestone mine in southwestern Japan, a neural network was used to estimate the content distributions of calcium oxide (CaO) and impurities in limestone ores. These factors determine the suitability of limestone for industrial use and its commercial value. The input data for the network consisted of geographic coordinates and elevation of the unknown points, together with data on rock and fossil types. The neural network-based method showed its superiority compared to kriging, producing more reliable spatial models with lower estimation errors and smoothing effects for the type of data under investigation. It should be noted that the

neural network approach was chosen because the spatial correlation of the impurities content data was not clearly shown by variogram analysis. This is in accordance with the expectations of this study, which hypothesizes that an ANN approach would outperform regression techniques for semivariogram development in situations where only moderate amounts of spatial autocorrelation are present.

Rigol et al. (2001) used a neural network to generate temperature surfaces for an area in Yorkshire, United Kingdom. Air temperatures were interpolated as a function of distances to coast and rivers as well as temperature, time, elevation, and other terrain variables measured at sampled points. The authors reported that the predictions of their ANN model reached a level of accuracy comparable to those reported in the literature for ordinary kriging, and concluded that neural networks represent a viable method for temperature estimation.

Merwin et al. (2002) examined the performance of ANNs as a spatial interpolation technique for the construction of DEMs depicting the area of Tijeras, New Mexico. The overall accuracy of the DEMs derived from the neural network was determined as compared to the elevation surfaces generated using kriging. The ANN estimated elevations at unknown points as a function of geographic location (either absolute or relative) and other elevation related attributes (slope, aspect, and average distance, sine, and cosine between an interpolated point and its sample neighbor points) at the sampled points. The results showed a better accuracy of the DEM estimates derived by kriging. However, the authors noticed that the pattern of under- and over-estimations of the ANN occurred in

regions of highest and lowest elevation values within the range of values in the sample. Therefore, they concluded that the systematic error contained in the ANN-derived predictions was to be attributed to the inability of neural networks to accurately predict the entire range of values contained in the sample data (this property is also shared by other interpolation techniques).

Yama and Lineberry (1999) developed a neural network model for predicting sulfur content values from spatial coordinates of sample locations in a mineral field in northern West Virginia. The study showed that the neural network performed as accurately, and in some cases better than, a kriging model. The authors suggested the suitability of neural networks as an alternative to geostatistical approaches for predicting mineral values from spatial coordinates.

Other examples of comparative studies evaluating kriging or other regression-based techniques compared to ANN-based models can be found in Cortez et al. (1997), Skidmore (1997), Blackard and Dean (1999), Johnston (1999), Matsoukas et al. (1999), and Lucifredi et al. (2000).

Lucifredi et al. (2000) compared three models for predictive monitoring and maintenance of hydroelectric power systems: linear multiple regression, kriging, and neural network. While the authors concluded that the kriging model overrides the difficulties presented by the application of both the multiple regression and the neural network models, and they recommended a mixed approach using combined kriging and

neural networks that could optimize results.

There are handful examples of alternative interpolation systems combining neural networks with kriging. However, none of them involved a hybrid ANN-based kriging approach identical to that proposed in this study. For example, a group of researchers from the Institute of Nuclear Safety in Moscow, Russia proposed an interpolation approach combining neural networks and kriging and designed to improve spatial data analysis (Kanevski et al., 1995; 1997a; 1997b; 1999; Demyanov et al., 1998). These authors recognized the difficulties of developing a valid semivariogram model for environmental data presenting complex spatial patterns at different scales, containing a great deal of noise, and suffering from significant measurement errors. For example, the fallouts of the explosion in 1986 at the Chernobyl nuclear power plant resulted in serious environmental consequences effecting large areas of Europe. The radioactive materials were transported over large distances, and their deposition produced extremely complex spatial patterns, which have proven to be difficult to analyze (Kanevski et al., 1997a). In order to deal with this issue, Kanevski et al. (1995; 1997a) introduced a hybrid Neural Network Residual Kriging (NNRK) model for use in case studies that evaluated soil contamination by Chernobyl radionuclides. They later presented variations of the original model, such as the Neural Network Residual Simulated Kriging (NNRSA) approach (Kanevski et al., 1997a).

In the proposed NNRK model (and its variations), a feedforward neural network was used to estimate global nonlinear trends; after semivariogram analysis of the residual errors from the ANN predictions, kriging was used to estimate such residuals; the final

hybrid model's predictions were obtained summing the original ANN estimations to the residuals predicted by kriging (Kanevski et al., 1995; 1997a). This approach was considered after the observation of a stationarity in the variogram of the residuals that was not present in the original data. The proposed approach "enables more freedom versus the hypothesis of stationarity, and versus the linearity of BLUE interpolation" (Kanevski et al., 1997a, page 533). In addition, the ANN offers the advantage of modeling highly nonlinear global trends for the area of study regardless of noise in the data, which traditional kriging does not handle well. On the other side, kriging captured the correlation missed by the ANN, and therefore using kriging (precise interpolator) for residuals estimation improved the ANN predictions (approximate interpolator) (Kanevski et al., 1997a). The approach proposed by these authors can be considered a trend surface approach, where the ANN is used to set the trend. The hybrid NNRK model and/or its variations were also successfully applied to case studies involving climatic data (Demyanov et al., 1998) and forest contamination data (Kanevski et al., 1999).

In two similar studies, Wang et al. (1999a; 1999b) used a similar method based on the trend surface approach proposed by Kanevski et al. (1995), but using a radial basis function ANN instead of a feedforward ANN. The authors combined neural networks and kriging for stochastic modeling of reservoir properties in the A'nan Oilfield in North China. In the proposed integrated technique, the authors used a neural network to estimate porosity distribution, and then used kriging to estimate the residuals from the neural network predictions after variogram analysis of the residuals at well locations. The resulting residual maps were combined with the porosity distribution obtained by the

neural network to produce an optimized final porosity distribution map. The final network predictions were therefore the sum of the original network predictions and two different realizations of their errors. The authors found the results of the integrated technique to be realistic and true to the known properties of the oilfield, and the results from the integrated technique to be superior to kriging alone. In addition, they concluded that this technique was fast and straightforward and eliminated any need for cross-correlation modeling.

The findings of studies exploring alternative kriging variants, such as those mentioned above, suggest that the performance of the conventional kriging's intrinsically linear approach to semivariogram analysis may be a limitation. The identification of an appropriate alternative methodology may further improve the interpolation capabilities of kriging, at least in those instances where the relationship between the variables under study is more complex than what can be modeled under the assumptions of linear regression-based semivariogram analysis.

None of the studies just described involved anything directly like the hybrid ANN-based kriging model proposed in this study. In all the reviewed studies, the ANN used in the interpolation system directly predicted the value of the variable(s) being interpolated at unmeasured sites, as a function of (1) the value of the variable itself or related predictive variable(s) at other measured sites, and (2) the absolute (geographic position) or relative (distance) location of the unmeasured site with respect to the measured sites. None of these studies adopted the approach of replacing the regression-based semivariogram analysis portion of kriging with a neural network, as presented in this study. In such approach, the

neural network is used to predict semivariance values for the variable(s) being interpolated (e.g., one of the attributes considered in the reviewed studies, such as elevation, impurities concentration, reservoir porosity, aquifer transmissivity, radioactive soil contamination, etc.) as a function of the lag distance to the measured sites. The semivariances predicted by the ANN are then used in the hybrid ANN-based kriging model to produce a predicted value of the variable.

Giroux-Hughes (2002) started the investigation of the ANN/kriging hybrid approach that is continued in this study. Giroux and Dean (2000) and Giroux-Hughes (2002) found the hybrid to be more effective than traditional regression-based kriging methods for databases with moderate degree of spatial autocorrelation. However, these previous studies did not account for all forms of semivariogram models typically used in traditional regression-based kriging, and did not evaluate various shape parameter values and grouping techniques. Therefore, the performance of the hybrid ANN/kriging model may have been compared to non-optimal traditional kriging models. Furthermore, only rudimentary adjustments to basic ANN parameters were investigated. As a result, those authors recommended further investigations of the KrigANN approach. This study is an attempt to follow up upon this recommendation.

Chapter 3

3. METHODS AND PROCEDURES

3.1. Overview

This study compared two spatial interpolation models:

1) A conventional kriging system that used experimental semivariograms generated via regression techniques;

2) A hybrid ANN-based kriging system (KrigANN) that used an artificial neural network (ANN) to create experimental semivariograms, but otherwise retained all of the characteristics of conventional kriging.

Both the models were evaluated using an experimental database containing 2250 artificially generated and independent GIS raster datasets. For each dataset, data values were randomly divided into model building and model validation subsets. The model building subset was used to construct both a traditional kriging and a hybrid kriging/ANN interpolation model for each dataset. The accuracy of the predictions from these interpolation models was evaluated by comparing their predicted values to known values from the validation subset. How these models and datasets were created and analyzed will

be the subject of the remainder of this chapter.

The major steps in the experimentation process were: data generation, sampling, model building, and model evaluation. All of these operations were performed using a custom program written in the Microsoft Visual Basic programming language. The program was named the *Raster Data Generator and Interpolator* (RDGI), and its operations are summarized in the flowchart shown in Figure 3.1.

The RDGI program started its operations by generating raster datasets, based on a number of controlling factors specified by the user (Figure 3.2). Controlling factors considered in this study were (1) the number of raster cells used in the model building data subsets for both the conventional kriging and the KrigANN models, (2) the number of sample cells used to estimate each individual cell value while validating the models, and (3) the value of an H-parameter that specifies the target degree of autocorrelation in each raster dataset. The program used this H-parameter in the Midpoint Displacement Method (described in Section 3.2), which was used to build all of the artificial datasets used in this study.

Once a dataset was constructed, the program randomly divided the cells in the dataset in two groups. A user-specified number of randomly selected cells were placed into a model building subset, which was used to build both the conventional kriging and the KrigANN models. All remaining cells were placed into the validation subset, which was used to cross-validate each model by comparing predicted and actual cell values.

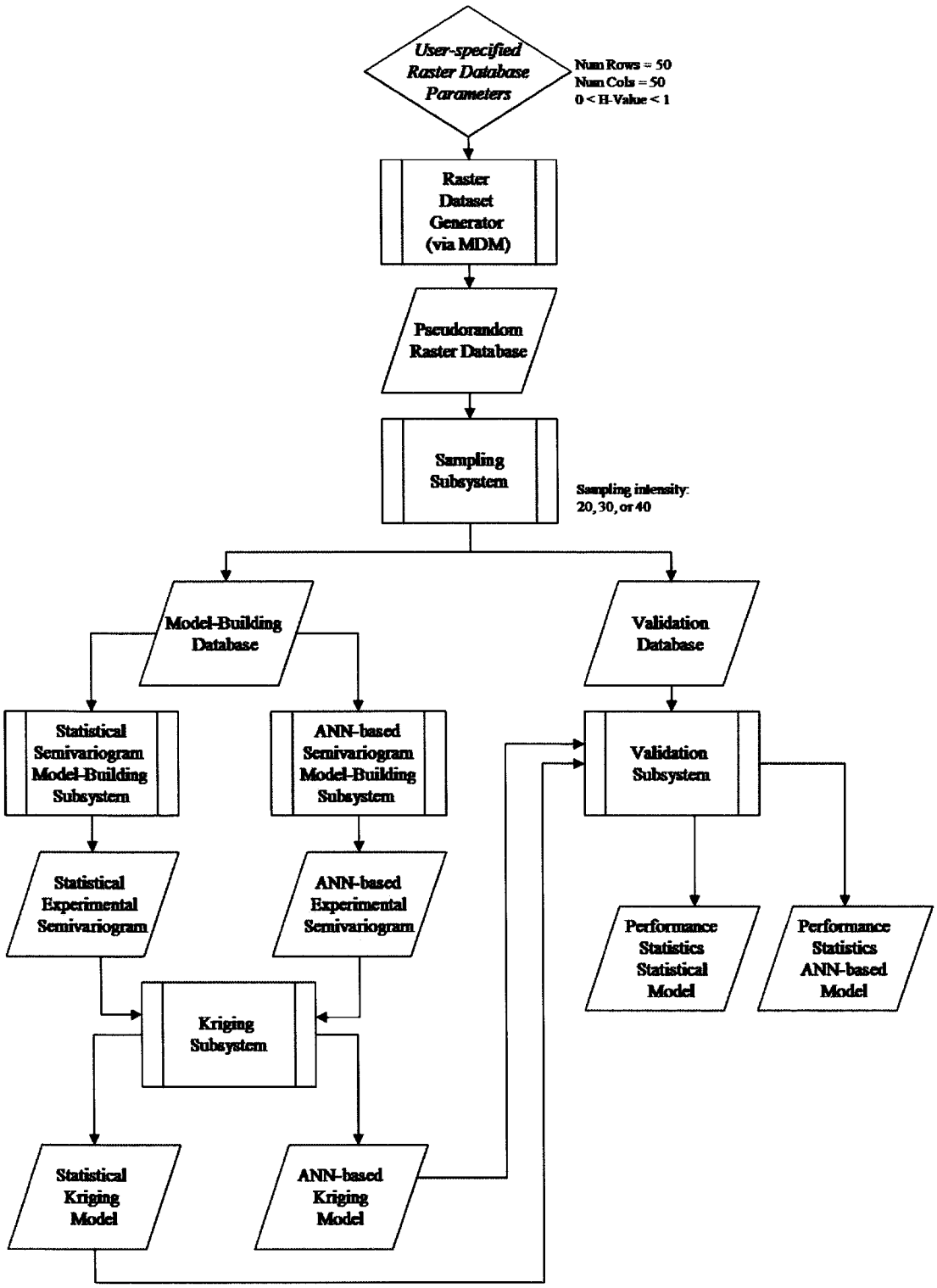

Figure 3.1 Flowchart of Raster Data Generator and Interpolator (RDGI) program operations.

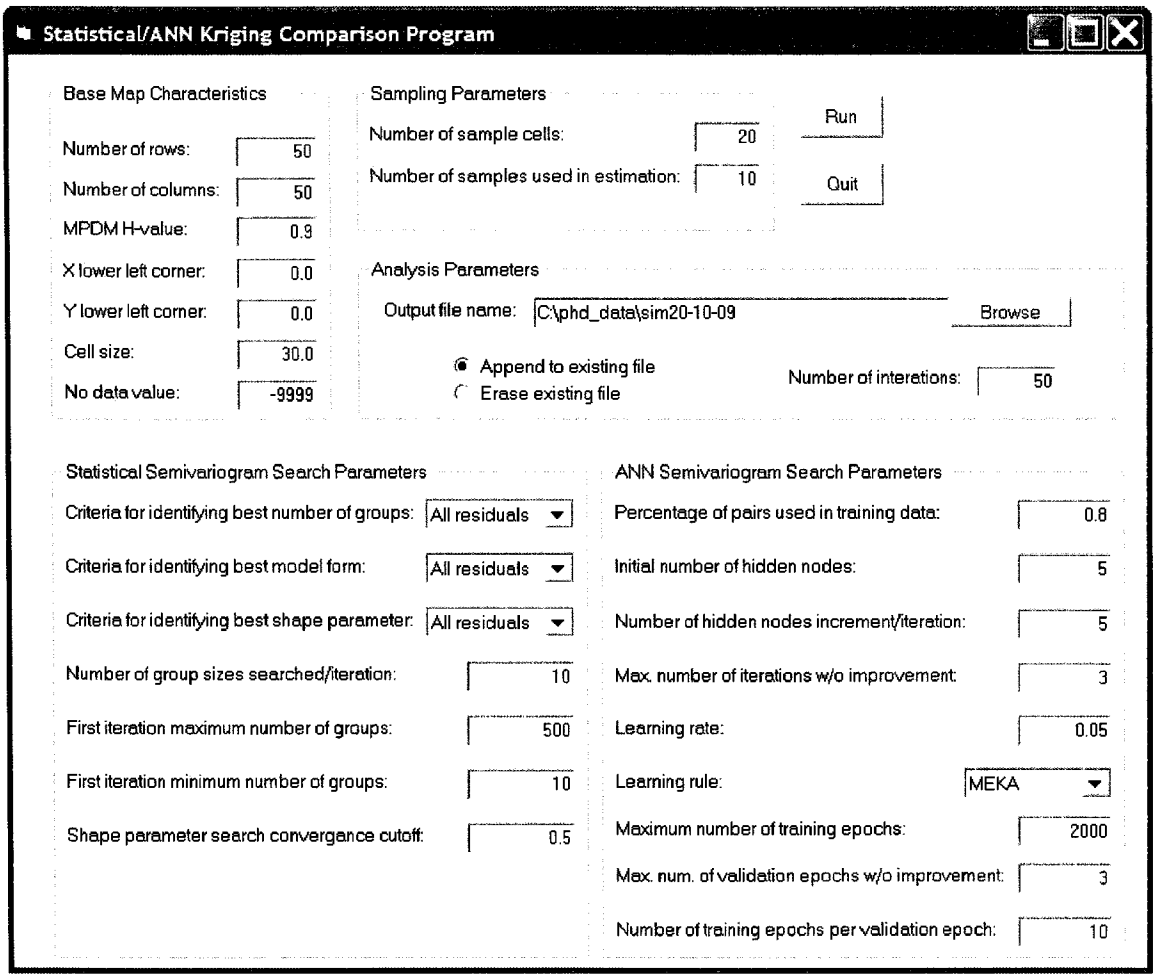

Figure 3.2 Main user interface of the RDGI program.

The program constructed both conventional kriging and KrigANN models from the model building data subset. Heuristic techniques were used to find optimal parameters for the conventional kriging's regression model and for the ANN.

Finally, the RDGI program used both the conventional kriging and KrigANN models to generate estimates of the values contained in each raster cell in the validation dataset. These estimates were compared to the known 'true' values from the validation dataset in

order to evaluate the predictive accuracy of each model.

All the operations in this experimentation process were repeated for each of the 2250 raster maps used in this study's GIS database. The following sections explain this process in detail.

3.2. Generating Artificial Raster Datasets

Artificially generated GIS databases were used to test and compare the two predictive models under investigation in this study. The use of artificial datasets, usually generated using an automated algorithm in a computerized process, is a common procedure in statistical analysis (Laurini and Thompson, 1994; Jones, 1997; Kaluzny et al., 1998; Burrough and McDonnell, 1998; E.S.R.I., 2003). Artificial datasets offer a number of advantages over real-world data. Artificial datasets allow for the creation of databases where the factors relevant to the study can be controlled. The algorithms used to generate the datasets can be modified in different ways, thereby creating different combinations of parameters. By alternatively varying various variables or factors (while keeping other factors constant) artificial data sets make it possible to thoroughly investigate the effects of each varying variable or factor on the model.

Other advantages of using artificial datasets include the ability to manipulate the level of randomness inherent in the data generation process, and to generate a large number of different datasets showing the same relationships between variables. Together, these

characteristics make it possible to generate multiple datasets that share the same underlying theoretical model, but are not identical. These multiple datasets can be used as multiple observations in statistical analyses, thereby gaining the repetitions necessary to draw meaningful conclusions from analyses of the datasets.

A number of methods are available for generating artificial spatial datasets. In this study, the Midpoint Displacement Method (MDM) described by Peitgen and Saupe (1988) and introduced by Fournier et al. (1982) was used to generate raster databases. This method produces pseudo-random databases with targeted amounts of spatial autocorrelation. The amount of spatial autocorrelation in a dataset created via the MDM procedure can be controlled by means of a parameter termed the Hurst exponent, also called *H*-exponent, or *H*-parameter (Mandelbrot, 1983; Peitgen and Saupe, 1988; Peitgen et al., 1992). The *H*-parameter ranges between zero and one, and is proportional to the amount of autocorrelation existing in a dataset produced by the MDM algorithm. Values of *H*-parameter close to 0 produce a dataset with a great deal of fragmentation, while *H*-parameter values close to 1 produce datasets with a great deal of a spatial autocorrelation (Mandelbrot, 1983; Peitgen and Saupe, 1988).

Unfortunately, due to the stochastic nature of the MDM procedure, the *H*-parameter is not a perfect measure of the degree of autocorrelation in any given dataset produced via the MDM procedure. Thus, two datasets created with the same *H*-parameter will present slightly different degrees of actual spatial autocorrelation. This condition is shown in Figure 3.3 for six different realizations of raster datasets created using the MDM method.

In order to work around this problem, in this study the *H*-parameter was used to specify a target degree of spatial autocorrelation, but actual spatial autocorrelation was measured using the Moran's I autocorrelation index (Moran, 1950). Moran's index was calculated for each dataset after its creation by the MDM method.

The Moran's I index is a standard measure of spatial autocorrelation, and has been in widespread use since its first publication (Moran, 1950). The index can be calculated using Equation 3.1 (Goodchild, 1986; Lee and Wong, 2001):

$$I = n \frac{\sum_{i=1}^{n} \sum_{j=1}^{n_i} w_{ij}(y_i - \mu)(y_j - \mu)}{\sum_{i=1}^{n} (y_i - \mu)^2 (\sum_{i=1}^{n} \sum_{j=1}^{n_i} w_{ij})} \qquad (3.1)$$

Where I is the Moran's I index, n is the number of values taken into account (in our case the number of raster cells); n_i is the number of neighbors of cell i used in the calculation; the terms y_i and y_j are the values of variable Z at locations i and j (in our case, the values in raster cells i and j); μ is the mean of all cell values, and w_{ij} are the spatial weights for location i in respect to j, representing the relative proximity of i and j. In our case, we chose to express the distances in terms of cell widths, so the weight values are 1 for directly adjacent cells, and the square root of two for diagonally adjacent cells.

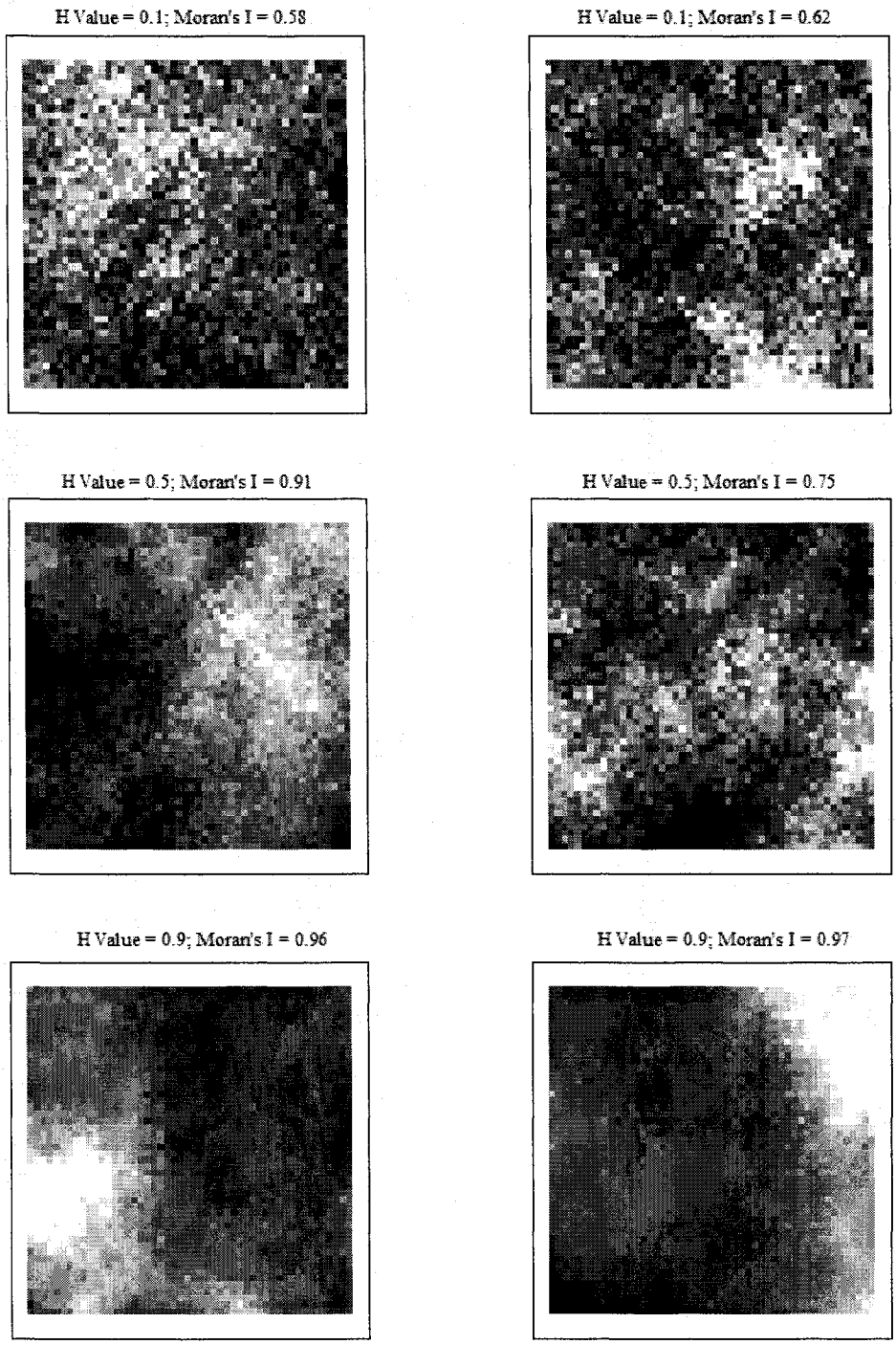

Figure 3.3 Examples of realizations of raster datasets created via the MDM method, showing target *H*-value and actual Moran's I autocorrelation index. Highest values are shown in white.

The values of Moran's index express the degree of clustering or dispersion within a dataset. Similar to the Pearson's correlation coefficient, the value of this index ranges from –1.0 to +1.0, where a value near 0 indicates a random pattern, i.e. no autocorrelation. A value close to +1.0 indicates a high degree of clustering of adjacent points with similar characteristics, i.e. extreme autocorrelation. Finally, values close to -1.0 indicate dispersion, or low clustering, and could indicate a uniform pattern (Isaaks and Srivastava, 1989; Lee and Wong, 2001). It should be noted that the ranges of the Moran's I index (from -1 to +1) and of the *H*-parameter (from 0 to +1) are different, as are their respective meanings. Therefore, although these parameters are directly related to one other, there is not a precise correspondence between the values of the two parameters for the same dataset.

3.3. Data Sampling and Model Building

3.3.1. Sampling

In order to build and evaluate the traditional and ANN-based kriging models, the cells in each raster dataset were randomly divided into two independent subsets. One subset of data points (the validation subset) was set aside for later use in evaluating the models (see Section 3.5). The other subset of data points (the model building subset) was used to construct both the conventional and KrigANN models. The data sampling and management scheme used in this study is shown in flowchart form in Figure 3.4.

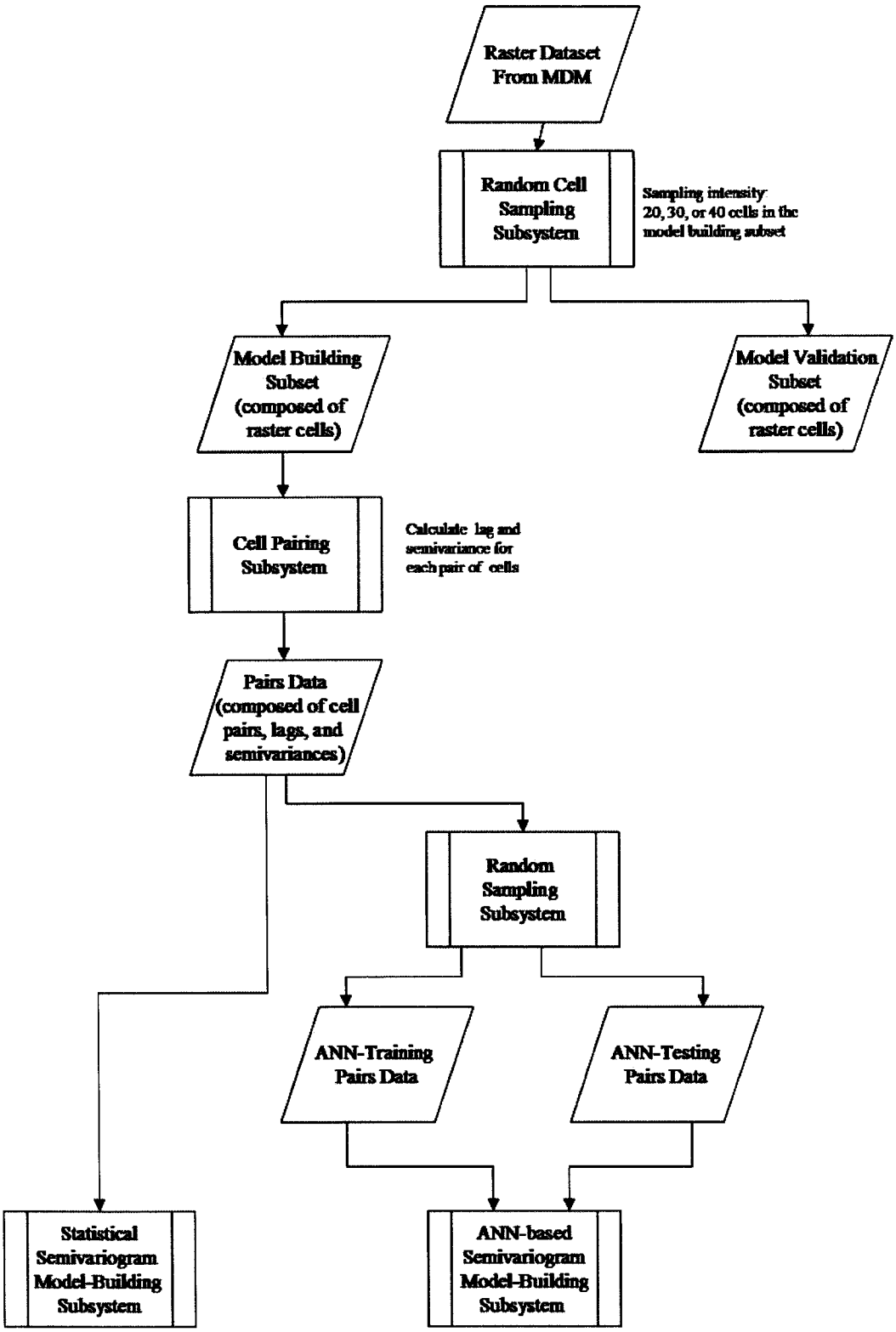

Figure 3.4 Data sampling scheme in the RDGI program.

The first step in this process was to randomly divide the cells from the raster database into model building and validation subsets. The exact number of cells in the model building subset was set by the user; all remaining cells were placed into the validation subset.

The individual raster cell values in the model building subset required further processing before they could be used in the model building process. Each raster cell was paired with all other cells in the subset, thereby producing $[n \times (n-1)]/2$ pairs from a model building dataset containing n raster cells. For each pair, semivariance (1/2 the squared difference in cell values), and lag (Euclidean distance between cell centers) were computed. The calculated lag and semivariance values were used as the predictive (lag) and response (semivariance) variables in the experimental semivariogram construction process.

For the KrigANN model (but not the conventional kriging model), the pairs data in the model building subset was further subdivided as shown in Figure 3.4. This was required in order to produce ANN-training and ANN-testing datasets (as described in the previous chapter) for use in building the ANN-based experimental semivariogram model. The ANN-training subset was used to train the neural network, and the ANN-testing subset was used to measure the ability of the neural network to predict semivariance values as a function of lag. Note the exchange in terminology for these datasets relative to much of the traditional ANN literature (Bishop, 1995; Ripley, 1996). This change is intended to avoid confusion with the terminology used in the statistics literature. The change ensures

that the term "validation dataset" described in this study is in fact used to validate both the conventional and KrigANN models.

3.3.2. Building Conventional Experimental Semivariogram Models

Experimental semivariograms for the conventional kriging model were found using an iterative heuristic approach designed to find the optimal or near optimal semivariogram for any given dataset. The heuristic techniques used in this study included three intertwined sub-routines searching for (1) the best model form, (2) the best grouping characteristics, and (3) the best shape parameter values for the semivariogram equation (either a or r in Equations 2.8 through 2.12).

The search routine started by evaluating each of the five mathematical forms typically used in semivariogram analysis (linear-to-sill, exponential, Gaussian, spherical, and circular). For each model form, multiple grouping characteristics (number and range of lag groups) were evaluated. For each model form and grouping characteristic, multiple shape parameter values were evaluated. The combination of model form, grouping characteristics, and shape parameter values that optimized a specific measure of model performance were identified as producing the optimal semivariogram.

Model performance was evaluated using the root mean square error (RMSE) value of all residuals. Note that since many forms of the semivariogram model consist of a portion of the model that is defined via a regression equation (i.e., the portion of the model within the semivariogram's range) and a second portion that is defined as a constant (i.e., the

portion of the model on the semivariogram sill), this criteria is not the same as simply finding the regression model with the lowest RMSE.

The search for the optimal semivariogram model started by selecting a model form. Each of the five model forms were evaluated sequentially, and the combination of grouping characteristics and shape parameter values that produced the best model for each form were identified. Once this was accomplished, selecting the optimal form was simply a matter of selecting the form that produced the best model.

For any given model form, the process of finding optimal grouping and shape parameter values started by selecting an integer number of groups into which the model building data would be divided. The shape parameter value that produced the best model for that number of groups was identified using techniques to be described shortly. A heuristic technique was then used to select an alternative number of groups to be evaluated, and the process of finding the optimal shape parameter value for this new number of groups was repeated. Ultimately, when this heuristic concluded that the best combination of number of groups and shape parameter values had been found, the search process stopped.

The heuristic that searched for the best number of groups started by determining the range of group numbers to be evaluated. The minimum number of groups to be considered was arbitrarily set to 10; it was felt that fewer groups would not result in enough observations to produce a valid analysis. The maximum number of groups was set to half the number of observations in the model building pairs data; this would ensure that on

average at least two observations fell into each group.

Once the range of numbers of groups to be evaluated was determined, the process of finding the optimal number of groups proceeded iteratively. Each iteration started by dividing the range of group numbers by a predefined number (10 in this study). This resulted in 10 specific numbers of groups to be evaluated. For example, consider the situation shown in Figure 3.5. Assume the range of number of categories to be considered is 10 to 1810. Dividing this range by 10, an evaluation spacing of 200 is produced. Thus, numbers of groups 10, 210, 410, ..., 1810 are identified for further evaluation.

Optimal shape parameters are found for each of these numbers of groups, and the resulting models are compared (using the previously mentioned RMSE criterion). Suppose the model produced using 810 groups is identified as the best of the models evaluated. This number of groups is bracketed by the 610 and 1010 number of groups. Thus, 610 to 1010 becomes the range of group numbers to be evaluated in the second iteration of the heuristic algorithm (second tier of Figure 3.5).

Figure 3.5 shows that in the second iteration, the number of groups providing the best semivariogram model is found to be 921. This number is bracketed by 877 and 965, which becomes the range of values to be evaluated in the third iteration. The iterative search technique continues evaluating smaller and smaller ranges of numbers of groups, until the range is reduced to a point where all integers within the range can be evaluated. The best model from this range is returned as the best overall model. In Figure 3.5, the number of

groups providing the best semivariogram model overall is found to be 907 groups.

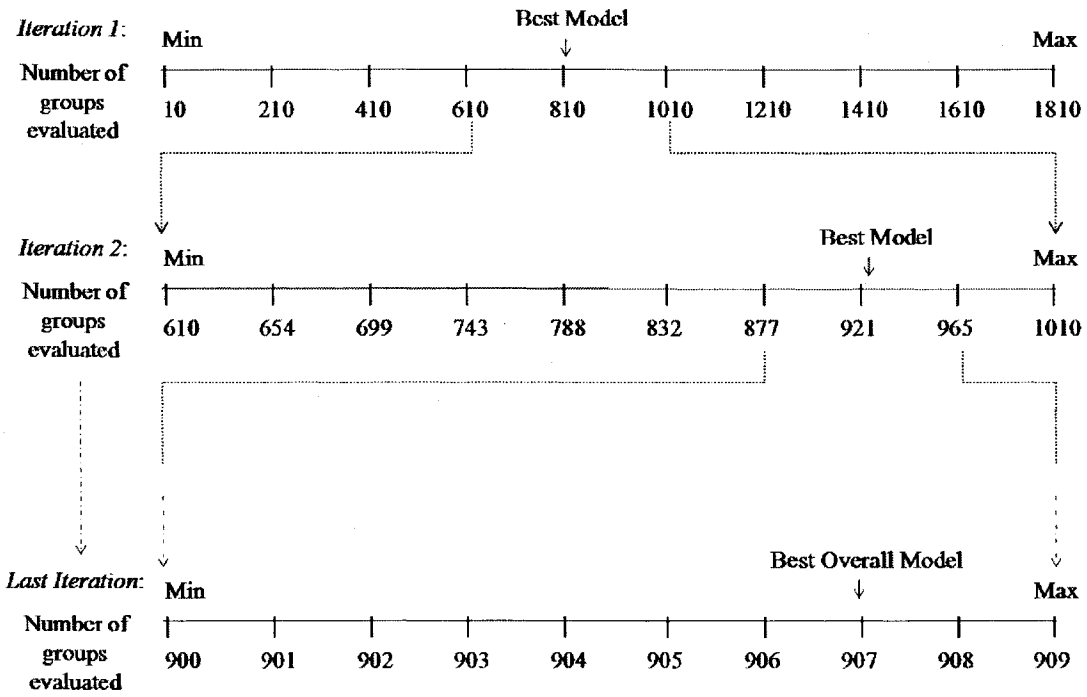

Figure 3.5 Iterative search routine to find optimal grouping characteristics.

Within the number of groups search routine just described, an inner search was used to find the best value for the shape parameter. This search sub-routine started by constructing two models using initial values for the shape parameter. The initial minimum shape parameter value was set to 0.001 (i.e., just slightly greater than zero), and the initial maximum shape parameter value was set to 1/2 the maximum lag value in the model building dataset. These models were evaluated using the minimum RMSE criteria described previously, and the better model was identified.

The search process proceeded by identifying the model that was produced using the

current value of the shape parameter (in the first iteration of the search, this was the initial shape parameter value that produced the better model), the best model produced thus far (initially, this was the same as the current model), and the previous model (initially, this was the model produced by the initial shape parameter value that did not produce the better model).

With these models (and corresponding shape parameter values) in hand, the heuristic search for the optimal shape parameter value proceeded using the following rules:

1) $RMSE_{Best} < RMSE_{Previous} < RMSE_{Current}$, or $RMSE_{Previous} < RMSE_{Best} < RMSE_{Current}$: New shape parameter value equals average of current value and the shape parameter value associated with the minimum of $RMSE_{Best}$ and $RMSE_{Previous}$.
2) $RMSE_{Current} < RMSE_{Previous} < RMSE_{Best}$, or $RMSE_{Current} < RMSE_{Best} < RMSE_{Previous}$: New shape parameter value equals average of current value and the shape parameter value associated with the maximum of $RMSE_{Best}$ and $RMSE_{Previous}$.
3) $RMSE_{Best} < RMSE_{Current} < RMSE_{previous}$: New shape parameter value equals average of best and current shape parameter values.
4) $RMSE_{Previous} < RMSE_{Current} < RMSE_{Best}$: New shape parameter value equals average of best and current shape parameter values.
5) $RMSE_{Previous} < RMSE_{Best} = RMSE_{Current}$: New shape parameter value equals current value plus the difference between the current and previous shape parameter values.
6) $RMSE_{Best} = RMSE_{Current} < RMSE_{Previous}$: New shape parameter value equals current value minus the difference between the current and previous shape parameter values.

7) $RMSE_{Current} < RMSE_{Previous} = RMSE_{Best}$: New shape parameter value equals current value plus 1.5 times the difference between current and best shape parameter values.

8) $RMSE_{Previous} = RMSE_{Best} < RMSE_{Current}$: New shape parameter value equals current value minus 1.5 times the difference between current and best shape parameter values.

Using these rules, the value of the shape parameter was iteratively refined. This process continued until the difference between the new shape parameter value and the previous value decreased to some convergence criterion defined *a priori* by the analyst. In this study, the convergence criterion was set to 0.05.

Once the heuristic just described found an optimal shape parameter value, the previously described heuristic could find an optimal number of groups. This in turn allowed the system to identify an optimal model form. In this fashion, a near optimal statistical semivariogram function was identified.

3.3.3. Building ANN-based Experimental Semivariogram Models

Several architectural and training parameters must be identified in order to implement any type of neural network. The KrigANN system used the most common type of neural network; a multi-layer, feed-forward, fully connected system. The architecture of all KrigANN neural networks consisted of one node in the input layer (corresponding to the single input into the model, i.e. the lag distance), one node in the output layer (corresponding to the single output of the model, i.e. semivariance), and a variable number of nodes in its single hidden layer. Each node in every neural network used a sigmoid

activation function.

Preliminary evaluation of the basic backpropagation and MEKA learning algorithms showed that backpropagation was unmanageably slow. As a result, the MEKA algorithm was used to train all the neural networks in the KrigANN system.

The MEKA training algorithm requires *a priori* specification of both the learning rate parameter η (see delta rule Equation 2.31) and the initial value of the synaptic weights. We chose to use a learning rate of 0.05, which is a small value relative to learning rates used in many other studies we found in the literature. We were able to use this small learning rate (which typically produces excellent final results, but takes a great deal of time) due to the excellent performance of the MEKA algorithm.

All nodes in KrigANN neural networks were assigned random initial synaptic weights. This is a common procedure and simply reflects a lack of *a priori* knowledge regarding the optimal final values of these weights.

The use of one hidden layer is usually sufficient for a feedforward network (Wong et al. 1995; Fowler and Clarke, 1996; Blackard, 1998). However, the optimal number of nodes in this single hidden layer depends on the complexity of the problem under study, and no single method of determining the number of hidden nodes is entirely appropriate for every situation (Wong et al. 1995; Fowler and Clarke, 1996; Blackard, 1998). In this study, the optimal number of nodes in the hidden layer was determined using a simple heuristic approach. First, an initial neural network was created and trained using 5 hidden

nodes (this number was defined *a priori* by the analyst). The RMSE of this model against the ANN-testing dataset was computed, and the iterative search started. The next iteration created and trained a network with 10 hidden nodes (again, the increase of 5 additional nodes was chosen *a priori* by the analyst), and this new model's RMSE against the ANN-testing dataset was computed. This process continued, creating networks with 15, 20, and more hidden nodes, until there was no improvement in the RMSE for 3 consecutive increments of the number of hidden nodes. At this point, the neural network with the lowest RMSE was selected as the final model.

The optimal number of epochs needed to fully train any neural network was found using a well established heuristic approach. This technique involved batch training, where the network is updated after every complete pass through the training data, i.e., after every epoch. After each 10 epochs (10 being another parameter chosen *a priori* by the analyst), the network was presented with the data from the ANN-testing dataset, and the RMSE of the network's predictions were noted. Ten more training epochs were then conducted, and the process of evaluating the network against the ANN-testing data was repeated. This cyclic process of training and evaluation against the ANN-testing subset continued until either (1) at least three consecutive evaluations against the ANN-testing data produced no decrease of the RMSE in the best model found so far, or (2) a maximum number of training epochs (2000, in this study) were completed. Both the cutoff of three non-improving evaluations and the maximum of 2000 training epochs were parameters set *a priori*. The best neural network found at the end of this process was saved as the final

neural network.

Any semivariogram must conform to the principles of Regionalized Variable Theory. To ensure such conformity, some necessary constrains were placed on the semivariograms produced by neural network modeling. The semivariograms were constrained so that the lag/semivariance relationship was only defined for non-negative lags, and any predicted semivariance generated by the ANN as less than zero were reset to zero. Typically, this was only necessary for very small lags in analyses where the model training data did not contain any observations involving small lags. In these cases, the ANN would extrapolate beyond the training data to produce its predictions, and negative extrapolations occasionally resulted.

Some authors maintain that valid semivariograms must be nondecreasing. However, there are real world examples of variables showing decreasing semivariograms, as discussed in Section 4.3. Therefore, the ANNs used in this study were not constrained to produce nondecreasing semivariograms. The impact of this will be evaluated in the Results chapter.

Examples of different semivariogram models produced using the RDGI program by both regression-based and ANN-based techniques are shown in Figure 3.6.

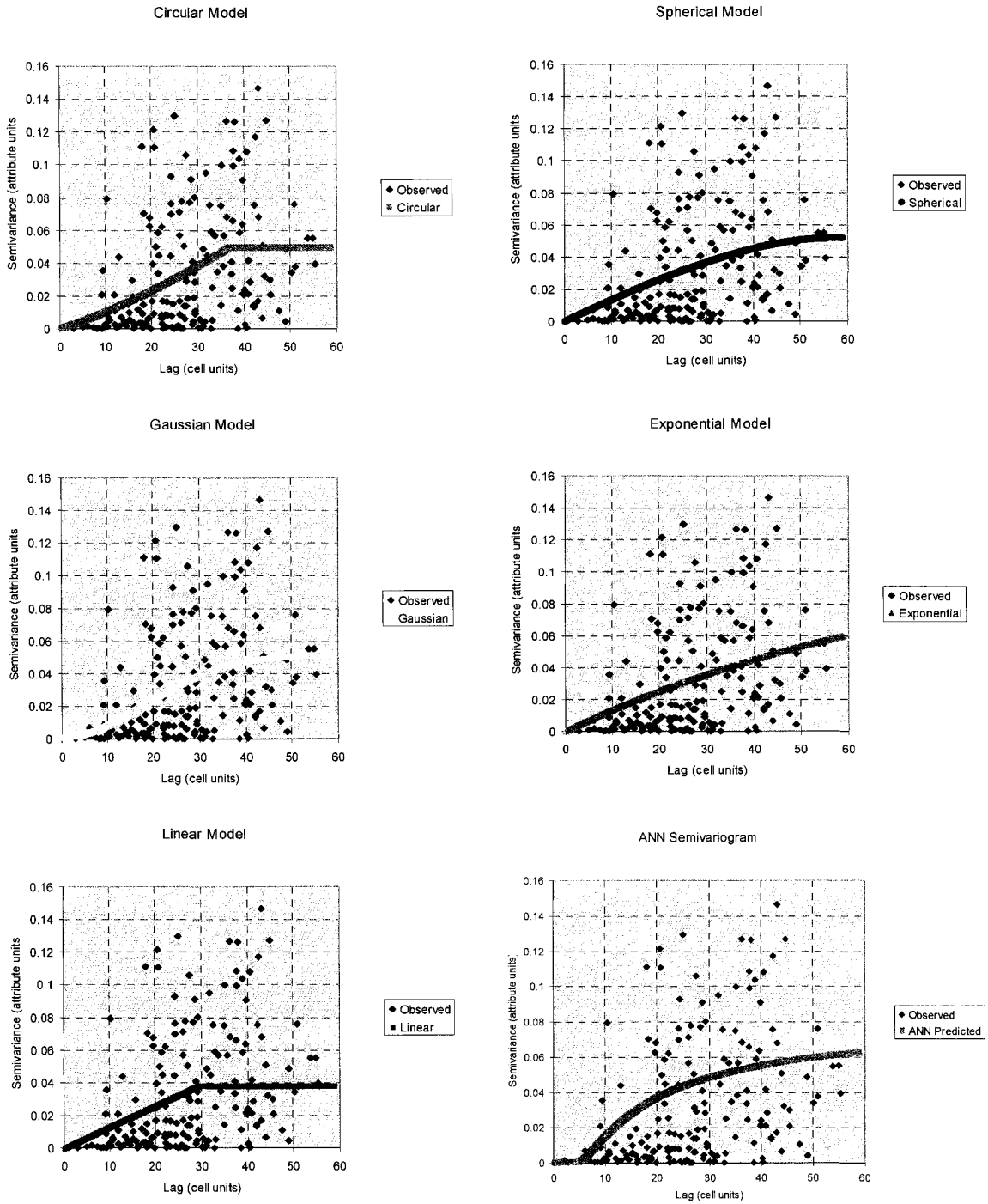

Figure 3.6 Examples of typical kriging semivariogram models and an ANN-based semivariogram model generated using the RDGI program.

3.3.4. Producing Kriging Estimates

The conventional and KrigANN interpolation systems shared the same overall structure. The only difference between the two models was that in the KrigANN system the regression-based approach for constructing the semivariogram used by conventional kriging was replaced by a neural network. Once the semivariograms were constructed for each model, standard procedures typically used in kriging were employed to produce predicted values for each cell in the validation subset of data.

For both the conventional and ANN-based kriging models, the optimal semivariogram identified by the search routines just described was used to predict semivariance values. The predicted values were then used to populate the A matrix and b vector from Equation 2.18. This equation was used to construct kriging weights. Finally, these weights were applied to the Z values of selected sampled cells to produce conventional or ANN-based kriging estimates of the Z values in the validation dataset (Equations 2.16). The same weights, applied to predicted semivariances between each validation point to be interpolated and each selected sample point, produced a value of kriging variance for each interpolated value of Z (Equation 2.19).

The interpolated values of Z and the related kriging variances were calculated using only a selected number of sample cells. This is a common procedure in kriging. The number of neighboring sample cells used to develop estimates for each validation cell was set by the user. This parameter was referred to as sampling utilization size.

Examples of an original raster datasets, with the corresponding predictions and

estimation variances produced by both conventional kriging and ANN-based kriging are shown in Figure 3.7. This figure also shows the sample points extracted from the original dataset for the model-building subset of data.

3.4. Experimental Design

The RDGI program developed for this study was used to generate a spatial database containing a total of 2250 raster datasets suitable for use in kriging. The controlling factors investigated for each dataset were: *H*-parameter value (target spatial autocorrelation), sampling intensity, and sampling utilization. Five different *H*-parameters values, three sampling intensities, and three sampling utilization values were investigated, for a total of 5 x 3 x 3 = 45 combinations of controlling factors. For each combination of controlling factors, 50 dataset realizations were created, for a total of 45 x 50 = 2250 realizations. A summary of the realizations created for the experimental database is shown in Table 3.1.

Each dataset used in this study consisted of a uniform grid of 50 x 50 points (for a total of 2500 points per dataset). Each point had *X*, *Y*, and *Z* coordinates, the latter being created via the previously described MDM process.

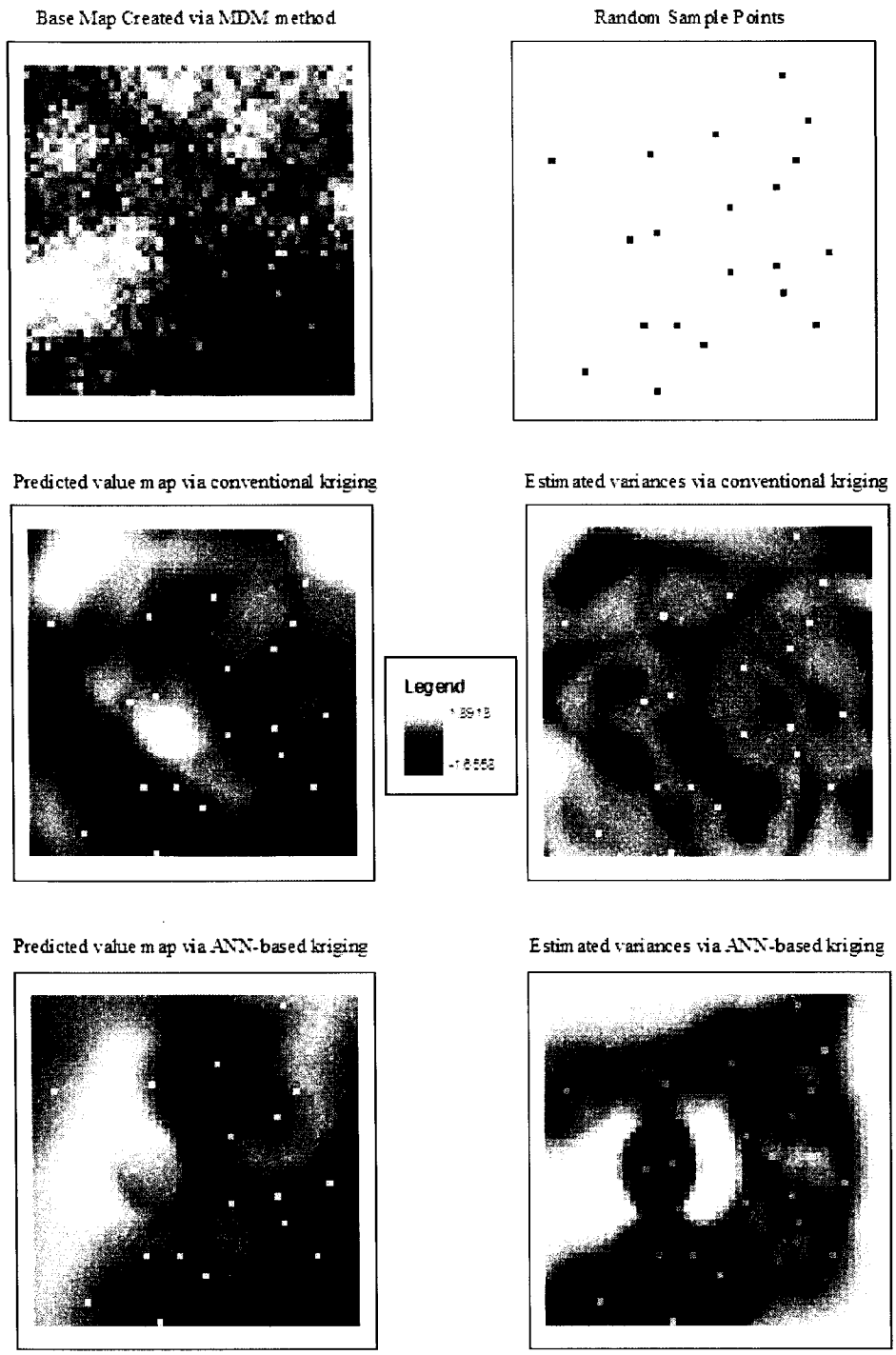

Figure 3.7 Raster datasets showing actual values, predicted values, and estimation variances for conventional kriging and ANN-based kriging. *H*-value = 0.05.

Table 3.1 Summary of raster dataset realizations produced for the experimental GIS database used in the study.

H Parameter Value	*Sampling Intensity*			*Sampling Utilization*		*Number of Raster Dataset Realizations (Total: 2250)*
	Num Sample Cells	*Percent Cells in Sample*	*Num Paired Cells*	*Num Samples Used for Interpolation*	*Percent Sample Used*	
0.1	20	0.8%	190	5	25%	50
				10	50%	50
				15	75%	50
	30	1.2%	435	5	17%	50
				10	33%	50
				15	50%	50
	40	1.6%	780	5	13%	50
				10	25%	50
				15	38%	50
0.3	20	0.8%	190	5	25%	50
				10	50%	50
				15	75%	50
	30	1.2%	435	5	17%	50
				10	33%	50
				15	50%	50
	40	1.6%	780	5	13%	50
				10	25%	50
				15	38%	50
0.5	20	0.8%	190	5	25%	50
				10	50%	50
				15	75%	50
	30	1.2%	435	5	17%	50
				10	33%	50
				15	50%	50
	40	1.6%	780	5	13%	50
				10	25%	50
				15	38%	50
0.7	20	0.8%	190	5	25%	50
				10	50%	50
				15	75%	50
	30	1.2%	435	5	17%	50
				10	33%	50
				15	50%	50
	40	1.6%	780	5	13%	50
				10	25%	50
				15	38%	50
0.9	20	0.8%	190	5	25%	50
				10	50%	50
				15	75%	50
	30	1.2%	435	5	17%	50
				10	33%	50
				15	50%	50
	40	1.6%	780	5	13%	50
				10	25%	50
				15	38%	50

The *H*-parameter value was used to investigate the impact of varying degrees of spatial autocorrelation of the kriging models. Any form of kriging assumes the existence of spatial autocorrelation, so it is reasonable to expect that varying the strength of autocorrelation will impact kriging results. The *H*-parameter values provided an indirect control over the degree of spatial autocorrelation within each generated dataset. The *H*-parameter values considered in this study were 0.1, 0.3, 0.5, 0.7, and 0.9.

The sampling intensity parameter indicates the number of sample cells randomly selected from each dataset for building both the statistical kriging and the ANN-based kriging models. It is plausible to assume that the number of samples used in any form of kriging will impact the results of the analysis. Sampling intensities of 20, 30, and 40 samples were tested, respectively corresponding to 0.8%, 1.2%, and 1.6% of the total points in each 50 x 50 dataset.

The sampling utilization parameter is the number of neighboring sample cells used for estimation of *Z* values at the interpolation points. Previous studies and texts indicate that this number can dramatically impact kriging results. Sampling utilization sizes of 5, 10, and 15 sample cells were examined.

3.5. Model Comparison

Once the conventional kriging and KrigANN models were applied to any given dataset, each was used to generate predicted *Z* values and estimation variances for each

point in the validation subset (i.e., each point not included in the model building subset). The quality of the predictions and variances were assessed using a variety of metrics.

Overall quality of the predictions generated for any given dataset were summarized by calculating the Root Mean Square Error (RMSE) of the predictions relative to the known values of the validation subset. The effects on these RMSE values of each controlling factor considered in the experiment was also evaluated: Degree of spatial autocorrelation, sampling intensity, and sampling utilization size.

In addition, the effect of semivariogram modeling parameters on the RMSE of the predictions of each model was evaluated. For conventional kriging, the semivariogram modeling parameters considered were model form, number of lag groups, and shape parameter value. For ANN-based kriging, the semivariogram modeling parameters considered were number of hidden nodes and number of training epochs.

Conventional wisdom holds that a desirable feature of kriging is the fact that estimation variances are typically low near sample points and increase at locations farther from sample points. We investigated this property by examining the correlation between estimation variances and distance to the nearest sample point. The effects of sampling intensity and sampling utilization size on the estimation variance and its spatial distribution were also evaluated.

Finally, computational performances were evaluated in order to compare the practicality of the two interpolating systems.

3.6. Hardware and Software

Most of the operations in the data building, data sampling, model building, and model comparison processes were computationally intensive and were repeated numerous times. These intensive and repetitive operations were executed using a custom program specifically developed for this study and written using the Microsoft Visual Basic 6 programming language. This custom program was named the Raster Data Generator and Interpolator (RDGI).

The feedforward neural network in the hybrid KrigANN model was developed using the MLP/X Neural Network ActiveX Control and COM Object by Windale Technologies, which was integrated in the custom RDGI program. The MLP/X ActiveX DLL allows for relatively easy implementation of multilayer perceptrons in a wide range of Windows applications (Windale Technology, 2001). This object requires no user interface and can be accessed by any ActiveX compatible development environment, including Visual Basic 6. The MLP/X implements the backpropagation and the MEKA training algorithms.

The GS+ package by Gamma Design Software and the ArcGIS 8.x GIS software by Environmental Science Research Institute (E.S.R.I.) were used to test the accuracy of the geostatistical routines and calculations used in the kriging and KrigANN models developed for this study. The ArcGIS 8.x software was also used for GIS data management, manipulation, spatial analysis, and map display. The SAS statistical package was used for statistical analysis of the model performance variables. A trial copy of the CoPlot and CoStat software package version 6.303 by CoHort Software was used for

producing 3-D graphs.

Analyses were conducted on numerous personal computers, each with slightly different configurations. However, all systems were based on Intel Pentium CPUs and ran the Windows XP Professional operating system. Note that for any given dataset, both conventional and KrigANN models were developed on a single machine, thus allowing valid performance comparisons to be made.

For printed reports and maps, a color Gestetner DSc38 laser printer was used.

Chapter 4

4. RESULTS AND DISCUSSION

The RDGI program developed for this study allowed the generation and analysis of 2250 realizations of artificial raster datasets showing different degrees of autocorrelation. Both the regression-based and the ANN-based (KrigANN) kriging models were applied to each dataset, evaluating for each model the effect of (1) varying the level of autocorrelation, (2) varying the overall sampling intensity used to develop the models, and (3) varying the sampling utilization, i.e. the number of samples utilized in calculating individual interpolated values. For each of the 2250 realizations, the RDGI program produced a number of metrics describing either specific characteristics of the dataset or the model, or the performance of each of the two models. These variables are presented in Appendix A together with basic descriptive statistics, and they will be described in further detail in the following sections. For cross referencing purposes, abbreviated names for the variables will be given in parenthesis within the text.

4.1. Autocorrelation Effects

The effect of the Moran's index on the performance of both the regression-based and the ANN-based kriging was evaluated in terms of the average RMSE of the predicted values produced by the models, as compared to the actual values in the dataset constructed by the RDGI program. The RMSE values were calculated using two slightly different techniques. In the first method, the calculation included the sample cells used to build the model, while in the second method, the RMSE values were calculated without the sample cells. Because kriging is an exact interpolator, including the sample cells used to develop the model in the computation of the RMSE would be expected to reduce the estimation error (Burrough and McDonnell, 1998). The RMSE values calculated in the two different manners returned the expected results, with the first method consistently producing slightly lower RMSE values than the second method. Other variables introduced later in this chapter were also calculated using the two techniques, and in all cases produced the expected results. Thus, to avoid repetition, only results computed without using the sample points will be presented throughout the remainder of this chapter.

For our first analysis, the 2250 RMSE values in the dataset were divided into 10 groups, each representing an equal range of Moran's index. Average RMSE and Moran's index values were computed for each group, and these averages are plotted in Figure 4.1.

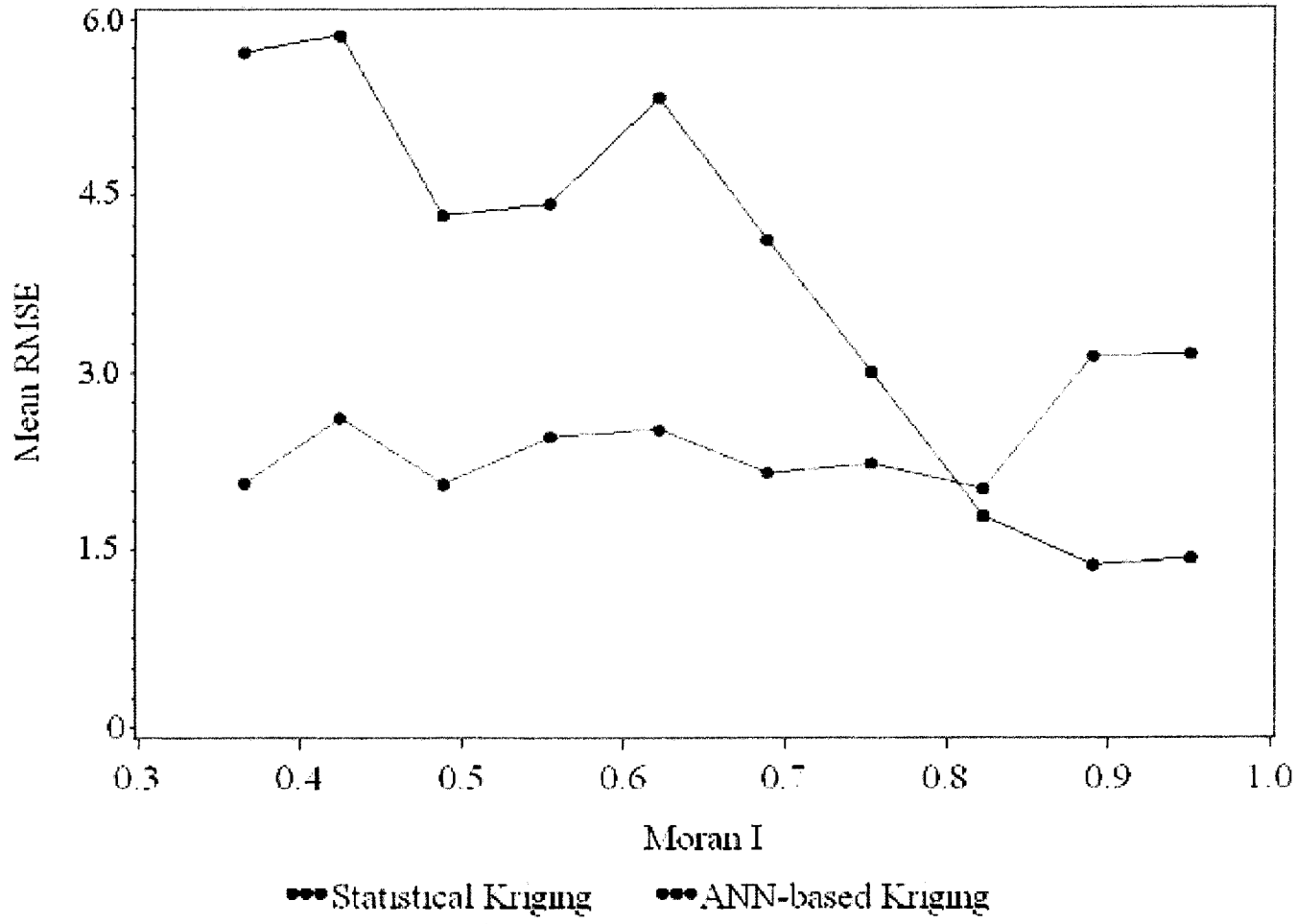

Figure 4.1 Effect of Moran's index on RMSE produced by statistical and ANN-based kriging.

The results in Figure 4.1 indicate that the ANN-based model performed better than statistical kriging at low and medium degrees of autocorrelation, while statistical kriging performed better at higher degrees of autocorrelation. The RMSE curves in this plot also indicate that ANN-based kriging was more stable, producing RMSE values that did not fluctuate as much as those produced by statistical kriging.

The results in Table 4.1 show the values used to produce the plot in Figure 4.1. At all Moran's index values below 0.72, the differences between the RMSE values are significantly different (at $\alpha = 0.05$), with the ANN-based model producing lower RMSE (i.e. more accurate predictions). At Moran's values greater than 0.86, RMSE differences were again significant, but this time the statistical model produced lower RMSEs. For Moran's values between 0.72 and 0.86, the differences between the two predictions were insignificant.

Thus, for 72% of the range of possible Moran's index values, the ANN-based model provided better results, and in 86% of the range it performed at least as well as regression-based kriging. The better performance of regression-based kriging at high degrees of spatial autocorrelation was expected, because high degrees of autocorrelation imply a semivariogram with relatively little noise and a strong trend in the lag/semivariance relationship, both of which favor regression analysis. However, when these conditions are lacking (i.e. at low to medium degrees of spatial autocorrelation), the ANN-based kriging approach produces more accurate results.

Table 4.1 Effect of Moran's Index on RMSE produced by statistical and ANN-based kriging models. Shaded rows represent groups where the difference between statistical and ANN-based kriging results were insignificant at the $\alpha = 0.05$ level.

Range of Moran's Index	Sample Size	Statistical Model Mean RMSE	ANN-based Model Mean RMSE	RMSE Difference	H0: RMSE Difference = 0 T Value	p-value
0.30 - 0.37	13	285.549	103.077	182.472	2.70447	0.01238
0.37 - 0.44	101	292.854	130.689	162.165	3.69661	0.00034
0.44 - 0.51	164	216.385	102.651	113.734	4.34221	0.00003
0.51 - 0.58	192	221.085	122.537	98.547	3.85977	0.00016
0.58 - 0.65	175	265.716	125.54	140.176	3.12889	0.00209
0.65 - 0.72	162	205.623	107.198	98.425	2.90922	0.00415
0.72 - 0.79	223	150.051	111.243	38.808	1.53241	0.12693
0.79 - 0.86	252	89.246	100.495	-11.249	-0.64841	0.51739
0.86 - 0.93	454	68.379	156.417	-88.038	-4.97038	<0.0001
0.93 - 1.00	514	71.383	157.268	-85.885	-6.25452	<0.0001

4.2. Sampling Effects

The performance of the two predictive models was further evaluated by conducting an analysis of variance (ANOVA) to examine the impact of a number of variables on the prediction RMSEs. Preliminary analysis of the data examined both the effects of individual variables, and the effects of sets of multiple variables. These analyses led to identical conclusions, so to avoid repetition, only the ANOVAs performed with the multiple variables will be presented here.

The ANOVA results in Table 4.2 show the effect of the Moran's index (*MI*), the total number of samples used to build the pairs dataset and therefore the semivariogram model (*NumSamp*), and the number of samples used for interpolation of individual *Z* values

(*NumSampUsed*) on the RMSE of regression-based kriging (*StatRMSE*). The ANOVA results indicate that the performance of the regression-based kriging model is significantly impacted by the value of the Moran's index (p-value < 0.0001). RMSE increases as Moran's I decreases, as expected. Conventional statistical kriging is designed for autocorrelated datasets, and therefore its performance should be negatively correlated with the degree of autocorrelation, which is measured by Moran's index.

Table 4.2 ANOVA Results for statistical kriging model RMSE versus Moran's index, number of samples, and number of samples used for interpolation.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	21735825.1	7245275.0	97.13	<.0001
Error	2246	167541494.5	74595.5		
Corrected Total	2249	189277319.6			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
MI	32.38872962	-12.83	<.0001
NumSamp	0.70519783	1.18	0.2371
NumSampUsed	1.41039552	11.18	<.0001

The ANOVA results in Table 4.2 also show that the number of sample cells used to develop the model did not significantly impact the RMSE of the prediction (p-value = 0.23). This somewhat surprising result indicates that increasing the sample size from 20 cells to 30 and 40 did not improve the ability of the model to produce accurate predictions of the values in the experimental dataset. Obviously, this finding should be considered valid only for the range of sample sizes considered in this study (20, 30, and 40), and it

certainly does not imply that the sample size could be reduced indefinitely (e.g., it is not valid to use this result to conclude that a sample of one cell would be acceptable for developing a statistical kriging model). This effect may be explained by the fact that the pairing process considerably increases the number of samples effectively used to build the semivariogram model. For example, if 20 sample cells are extracted from the original raster dataset, the actual number of observations used to build the model is 190 (i.e., $n(n-1)/2$ pairs), where n is the number of sample cells. Under the central limit theorem, statistical models like regression relatively quickly reach a sample size containing all of the information in a dataset, thereby rendering additional sampling unnecessary (Anderson et al., 1991). It is entirely plausible that the smallest sample size used in this study (20 cells, or 190 pairs) captured all of the meaningful information in the dataset, and therefore increasing sample sizes provided no additional useful data.

On the other hand, the results from Table 4.2 show that the RMSE of statistical kriging's predictions was significantly affected by the number of neighboring sample cells used to make each individual cell prediction (p-value < 0.0001), with higher RMSE values (i.e., less accurate predictions) at higher sample utilization sizes. One possible reason for this is that the neighboring cells used to make the estimations were chosen without restrictions on their distance from the point being interpolated. Therefore, it is possible to assume that a higher number of neighboring cells is more likely to include interpolation points that are beyond the range of autocorrelation in the semivariogram. These far away points would not make a relevant contribution to the accuracy of the estimation, and could

contribute to meaningless noise to the system and therefore produce greater RMSEs.

The results of the ANOVA conducted on the RMSEs of the ANN-based kriging model (*AnnRMSE*) are shown in Table 4.3. Similar to the results obtained for regression-based kriging, the RMSE of the ANN-based model is significantly impacted by Moran's index. However, this effect is less significant (p-value = 0.005). This outcome is likely due to the ANN's ability to find trends in the noisy pairs datasets encountered when performing semivariogram analyses on datasets with little autocorrelation. Thus, the level of autocorrelation in the data should not impact the performance of an ANN-based model as much as it does regression-based kriging.

Table 4.3 ANOVA Results for ANN-based kriging model RMSE versus Moran's index, number of samples, and number of samples used for interpolation.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3074396.6	1024798.9	15.53	<.0001
Error	2246	148226637.1	65995.8		
Corrected Total	2249	151301033.7			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
MI	30.46462589	2.81	0.005
NumSamp	0.66330444	2.43	0.0152
NumSampUsed	1.32660874	5.73	<.0001

In addition, the form of the RMSE - Moran's I relationship is reversed, with higher Moran's index values associated with higher RMSE values. This result is not easily

explained in terms of theory; nonetheless, observation of the plot in Figure 4.1 suggests an explanation. The curve of the RMSE values produced by ANN-based kriging as a function of Moran's index is relatively flat, with small spikes likely related to random noise in the datasets. The negative effect of higher Moran's index values on the accuracy of the model must be a consequence of this random noise.

The ANOVA results also show that the sampling intensity has a significantly stronger effect on the RMSE results of ANN-based kriging than it does on regression-based kriging. Typically, a large number of data points are needed by a neural network to identify the unknown form of the best model reproducing the relationships between variables (Haykin, 1994; Gurney, 1997). Conversely, as explained previously, under the central limit theorem statistical models like regression relatively quickly reach an adequate sample size (Anderson et al., 1991). In statistical kriging the sample points are forced to fit a model type with a predefined form, so relatively few samples are needed to fit the predefined model to the data. ANN-based models have no predefined form, and therefore need additional samples to properly create the model.

The ANOVA results in Table 4.3 indicate that the number of neighboring sample cells used to calculate the prediction is a statistically significant predictor of the variation of RMSEs (p-value = <0.0001), with higher RMSE values for larger sample utilization sizes. This can be explained using the same logic as was presented for the previous ANOVA analysis, that is, larger sample utilizations result in interpolations that incorporate sample points located quite far from the interpolation point.

4.3. Effects of Semivariogram Modeling Parameters

The ANOVA results presented in Table 4.4 show that the variation in RMSE produced by the statistical kriging model is not significantly affected (p-value = 0.6225) by the number of lag groups that produced the semivariogram model that best fits the experimental data (*BestNumGroups*). This is reassuring, because it implies that the heuristic techniques used to identify the best statistical model were not biased and consistently found the ideal or near-ideal number of groups, regardless of whether that ideal was a large or a small number.

Table 4.4 ANOVA Results for statistical kriging model RMSE versus best number of groups.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	20404.7	20404.7	0.24	0.6225
Error	2248	189256914.9	84189.0		
Corrected Total	2249	189277319.6			

Conversely, the model form (*BestStatModel*) used in regression-based kriging significantly impacted the RMSE results of the model (p-value < 0.0001), as shown in Table 4.5. The independent variable used in this analysis was an integer identifying the model form (spherical, exponential, Gaussian, circular or linear-to-sill) chosen by the heuristic procedures (described in the Methods chapter) as best fitting the experimental data.

Table 4.5 ANOVA Results for statistical kriging model RMSE versus best statistical model.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	9412235.7	9412235.7	117.64	<.0001
Error	2248	179865083.8	80011.2		
Corrected Total	2249	189277319.6			

The mean, standard deviation, minimum and maximum RMSE values produced by the models using each standard form are shown in Table 4.6. The Gaussian model performed substantially more poorly than the other model forms, while the differences between the other forms were not especially dramatic. This outcome is likely due to the nature of the MDM method used in the data generation process. The nature of the MDM method will result in a specific form of semivariogram relationship, and these findings indicate that this form is not well represented by the Gaussian model. Before a conclusion could be made upon the universally most applicable form of semivariogram, several different procedures for generating artificial datasets should be tested.

Table 4.6 Effect of semivariogram model form on RMSE for the statistical kriging model.

Best Model Form	N	Mean	Standard Deviation	Minimum	Maximum
Linear	109	67.92734	33.62965	7.44127	123.11615
Gaussian	666	279.86755	394.77403	8.88620	2393.65000
Circular	1030	98.51284	252.48436	6.93351	2605.25000
Spherical	321	46.69212	32.61949	7.18209	114.93411
Exponential	124	37.43763	30.26410	7.98939	122.96867

The number of training epochs (*NumAnnTrainEpochs*) at which the neural network in the ANN-based kriging model reached convergence was recorded for each simulation, as was the number of hidden nodes (*NumHidden*) in the final trained neural network. Table 4.7 shows the results of the ANOVA for the RMSE produced by ANN-based kriging versus these variables. The results indicate that there is a mild effect of the number of training epochs on the RMSE of the model (p-value = 0.0186). The optimal number of training epochs varied from 10 to 2000, the latter being the maximum number allowed to take place training any single ANN.

Table 4.7 ANOVA Results for ANN-based kriging model RMSE versus number of epochs and number of hidden nodes.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	5293188.9	2646594.5	40.73	<.0001
Error	2247	146007844.7	64979.0		
Corrected Total	2249	151301033.7			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
NumAnnTrainEpoch	0.00792336	-2.36	0.0186
NumHidden	0.46557782	8.29	<.0001

The optimal number of hidden nodes in the final trained network ranged from 5 to 75 (see Appendix A). The effect of the number of hidden nodes in the final network was very significant in explaining the variation of the model's RMSE (p-value < 0.0001). This implies that more complex ANNs that could reflect relatively convoluted relationships

between semivariance and lag generally outperformed simpler ANNs that reflected less convoluted relationships.

4.4. Effects of Non-decreasing versus Unconstrained Semivariograms

Some authors maintain that the form of the relationship between lag and semivariance must be non-decreasing, i.e., there can be no range of lags over which an increase in lag results in a decrease in semivariance (Cressie, 1993). By design, the semivariogram forms used in conventional statistical semivariogram analysis are non-decreasing. Conversely, in a neural network the form of the relationship between lag and semivariance is not constrained in any way.

However, it is relatively easy to find real world examples of datasets that would produce decreasing semivariograms. For example, consider a semivariogram relating elevation to lag in a terrain characterized by parallel ridges and valleys, as shown in Figure 4.2. Using a base point located at the bottom of one of these valleys and following a transect perpendicular to the ridges, lag and semivariance would initially show an increasing relationship as we moved up a ridge. However, after the crest of the ridge is reached and we start to move down the other side, lag and semivariance would show a decreasing relationship (Dean and Giroux-Hughes, 2004).

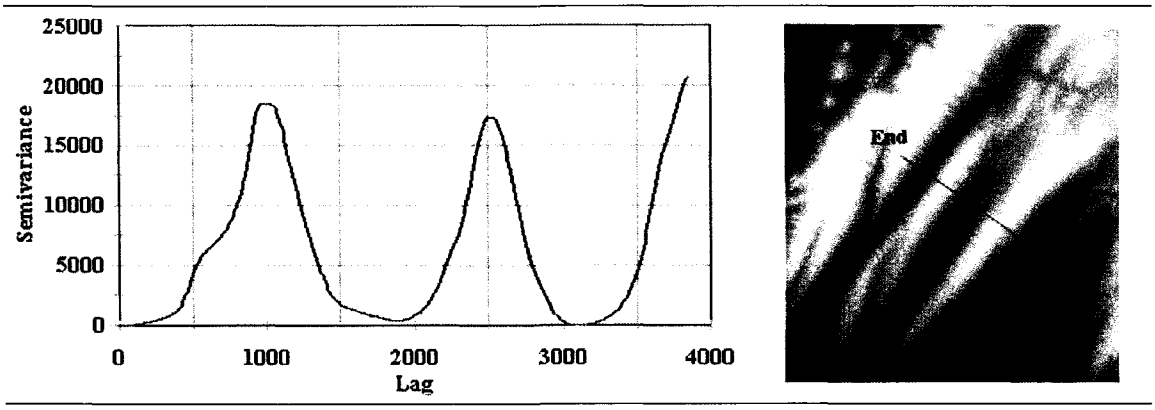

Figure 4.2 Example of a real world dataset producing a decreasing semivariogram. The data shown comes from a U.S.G.S. Digital Elevation Model for Western Maryland. White shows high elevation; black shows low elevation. (Reprinted by permission of Denis Dean, 2005).

In order to eliminate these effects, it is a common procedure to de-trend the dataset prior to kriging analysis. For example, a statistical approach (such as regression) may be used to construct a model that explains the overall pattern in the data (in the case of Figure 4.2, the overall pattern in the data is the pattern of parallel ridges and valleys). This overall trend is then subtracted from the observed sample values to obtain residuals, and the residual values are used in kriging analysis. The trend and kriging estimates are subsequently added together to form final predictions. These procedures allow the data used in semivariogram development to conform to the assumption inherent in kriging that there should be no noticeable trend in the data, i.e. local stationarity (Isaaks and Srivastava, 1989). However, de-trending is not always trivial; it is often a very laborious process, and it requires a good understanding of the phenomenon described by the dataset.

ANN-based kriging eliminates the need to de-trend datasets. Since ANN-based semivariogram analysis can fit models to any dataset without the need for *a priori*

definition of the form of the model, the need for de-trending is eliminated. This should be an advantage for the ANN-based kriging models investigated in this study.

The ANN-based kriging model produced 1025 decreasing and 1225 non-decreasing semivariograms (roughly a 45/55% split), as shown in Table 4.8. The results indicate that in cases where ANN-based kriging produced a non-decreasing semivariogram, the statistical approach outperformed ANN-based kriging by a small amount. However, in cases where ANN-based kriging found decreasing semivariograms to best fit the data, the ANN-based model substantially outperformed conventional statistical kriging.

Table 4.8 RMSE results produced by ANN-based kriging showing decreasing semivariograms, compared to RMSE results for non-decreasing semivariograms. Statistical kriging's RMSE results are also shown.

	N	Mean RMSE	Stdandard Deviation	Minimum	Maximum
Non-decreasing Semivariograms					
ANN-based kriging	1225	158.462	303.778	8.840	2482.000
Regression-based kriging	1225	123.922	267.398	6.934	2605.250
Decreasing Semivariograms					
ANN-based kriging	1025	101.768	188.940	9.634	1894.970
Regression-based kriging	1025	159.113	314.164	7.138	2408.420

When analyzing unfamiliar datasets using conventional kriging techniques, the analyst may not know *a priori* if a decreasing semivariogram component is present in the data. Preliminary EDA analyses may be needed to resolve this issue. However, the findings presented here suggest that using ANN-based kriging can produce accurate

predictions regardless of the presence of a trend in the experimental data, such as decreasing semivariograms. This eliminates the need of laborious de-trending techniques and may render them obsolete.

4.5. Kriging Variance Results

The standard deviation (or variance) of each interpolated value produced via a kriging process is known as the kriging standard deviation or variance. Kriging variance, also known in conventional kriging as "kriging error" or "estimation variance", gives information about the reliability of each interpolation over the area of interest (Burrough and McDonnell, 1998). In this study, kriging standard deviations were computed for both statistical (*MeanStdStat*) and ANN-based kriging (*MeanStdAnn*). The effect of sampling intensity and sampling utilization on the mean kriging standard deviation was examined for both models, and the results are presented in Tables 4.9 and 4.10. The results presented in these tables indicate that neither model's standard deviation estimates were impacted by sampling intensity or utilization.

The correlation between the kriging standard deviation at each interpolated point and the distance from that point to the nearest sample point gives information about the spatial distribution of the kriging estimation variance. Distances from each interpolation point to the nearest sample point were calculated. The correlation between kriging variance and distance to the nearest sample point was calculated using Pearson's correlation coefficient for both statistical kriging (*StdDistCorrStat)* and ANN-based kriging (*StdDistCorrAnn*).

The impact of sampling intensity and sampling utilization sizes on the distance-variance correlation was examined for both models. The results of these analyses are presented in Tables 4.11 and 4.12.

Table 4.9 ANOVA Results for statistical kriging estimation variance versus number of samples and number of samples used for interpolation.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	94.226	47.113	0.06	0.9380
Error	2247	1654403.831	736.272		
Corrected Total	2249	1654498.057			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
NumSamp	0.07006056	-0.32	0.7527
NumSampUsed	0.14012112	-0.17	0.8655

Table 4.10 ANOVA Results for ANN-based kriging estimation variance versus number of samples and number of samples used for interpolation.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	154.985	77.5	0.1	0.9093
Error	2247	1830890.318	814.815		
Corrected Total	2249	1831045.303			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
NumSamp	0.07370280	0.08	0.9376
NumSampUsed	0.14740560	0.43	0.6679

Table 4.11 ANOVA Results for statistical kriging model: index of correlation between kriging estimation variance and distance versus number of samples and number of samples used for interpolation.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	15.5498517	7.7749258	127.14	<.0001
Error	2247	137.414046	0.0611544		
Corrected Total	2249	152.9638977			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
NumSamp	0.00063851	-1.21	0.2273
NumSampUsed	0.00127702	-15.9	<.0001

Table 4.12 ANOVA Results for ANN-based kriging model: index of correlation between kriging estimation variance and distance versus number of samples and number of samples used for interpolation.

a) Overall ANOVA Results

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	1.921114	0.960557	10.09	<.0001
Error	2247	213.9191532	0.0952021		
Corrected Total	2249	215.8402672			

b) Breakdown of Variance Components

Parameter	Standard Error	t Value	Pr > |t|
NumSamp	0.00079667	-3.91	<.0001
NumSampUsed	0.00159334	-2.2	0.0276

The ANOVA results presented in Tables 4.11 show that the sampling intensity did not significantly impact the degree of correlation between kriging variance and distance for the

statistical kriging model (p-value = 0.2273).

The effect of sampling intensity on the distance-variance correlation is significant on the ANN-based model (*p*-value < 0.0001), as shown in Table 4.12. In particular, the distance-variance correlation decreases as the sampling intensity increases. This is in keeping with the material presented by Burrough and McDonnell (1998), who state that kriging variance is linked to the density of sample data points, and becomes higher as the density decreases.

The ANOVA results in Table 4.11 and Table 4.12 also show that the sampling utilization significantly impacted the degree of correlation between estimation variance and distance of both models. This can be explained by the effect of the sampling utilization on the prediction error itself. Recalling the ANOVA results from Tables 4.3 and 4.4, the RMSE produced by both kriging models increased with increasing sample utilization sizes. Since the RMSE value increases and the distance decreases with increasing sample utilization size, the value of the Pearson's correlation coefficient is also expected to increase.

4.6. Computational Considerations

The amount of time needed to find the optimal regression-based (*TimeStat*) and ANN-based (*TimeAnn*) kriging model for each raster dataset was recorded in seconds. Given that the simulations were conducted on multiple computers with different performance

characteristics, comparison of absolute times would not be valid. However, since both the statistical and ANN-based models for any given dataset were developed on the same computer, the use of time differences and time ratios is valid and does not involve any bias produced by different computers.

Recall that the experimental design used in this study investigated five different levels of the MDM's *H*-parameter. Preliminary analysis of time differences showed a pattern of similar results for all five groups of simulations produced with different *H*-parameter; therefore, results will be presented here only for one group of simulations.

The results of time comparisons for the group of 450 simulations developed using *H*-parameter = 0.1 are presented in Figure 4.3. This chart shows the ratio between the time needed to find the statistical kriging model and the time needed to find the ANN-based model for each simulation. The results indicate that for most simulations, the ANN-based kriging method found the optimal interpolation model for the experimental dataset faster than regression-based kriging. For most simulations (344 of 450), the time needed for statistical kriging to find the best interpolation model was from 50% to 100% higher than the time needed to find the best ANN-based kriging model. However, for the remaining 106 simulations, the ANN-based model required as much as 450% longer than did the statistical model.

This result was unexpected, considering that the neural network literature indicates that training a neural network may be very time consuming and resource intensive (Haykin, 1994). However, much of this literature involves only the common

backpropagation training algorithm, while the MEKA algorithm was used in this study. The MEKA algorithm is known for delivering faster convergence times and better solutions than backpropagation (Shah and Palmieri, 1990; Palmieri et al., 1991a, 1991b; Stan and Kamen, 2000), and the results of this study support these findings.

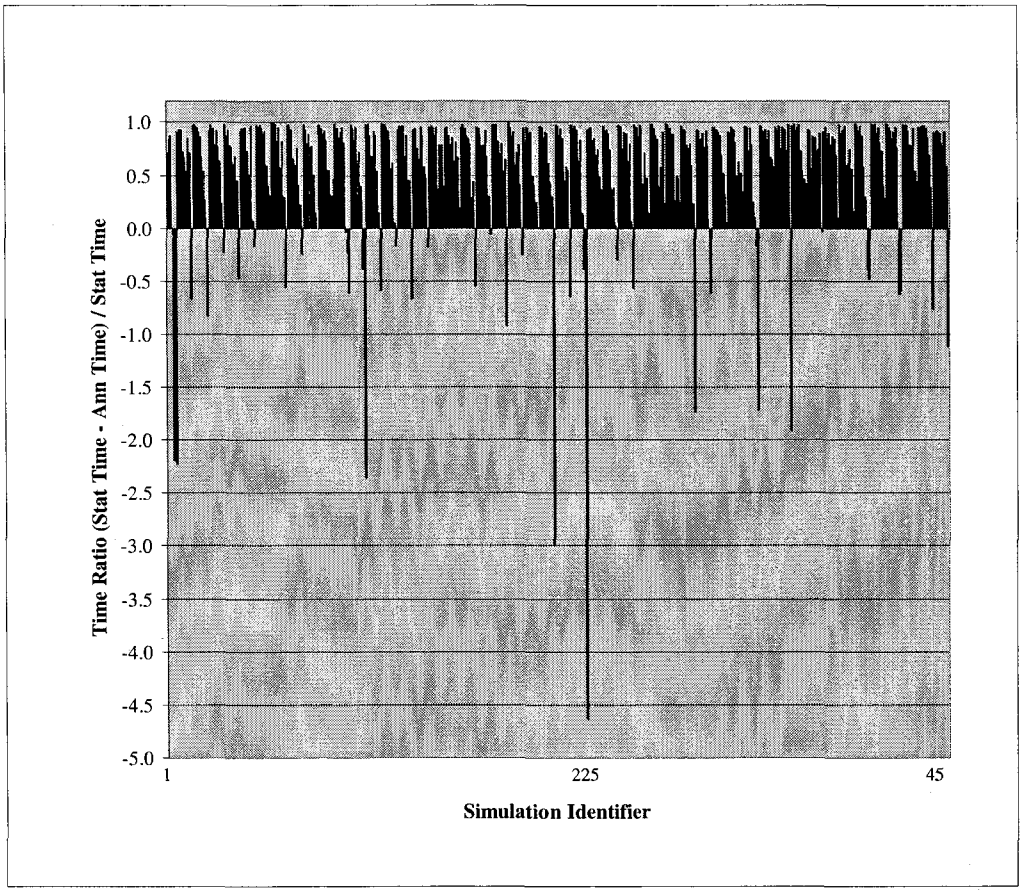

Figure 4.3 Comparison of time needed to build statistical and ANN-based kriging models.

The time performance of the two models was also analyzed by looking at the effect of the number of samples used to develop the models and the number of samples used for individual interpolations. The three-dimensional graphs in Figures 4.4 and 4.5 show that

both models slow down as both sampling parameters increase. For the ANN-based model, the relationship between execution time and these sampling parameters is far more regular and uniform than it is for statistical kriging. This can be explained by the heuristics used to identify the best semivariogram model. These heuristics evaluate multiple semivariogram models, and the time required to find the optimal combination of model's form, lag grouping characteristics, and shape parameter values may vary in different measure, and increase considerably, depending on how many combinations need to be evaluated before the optimal model is found.

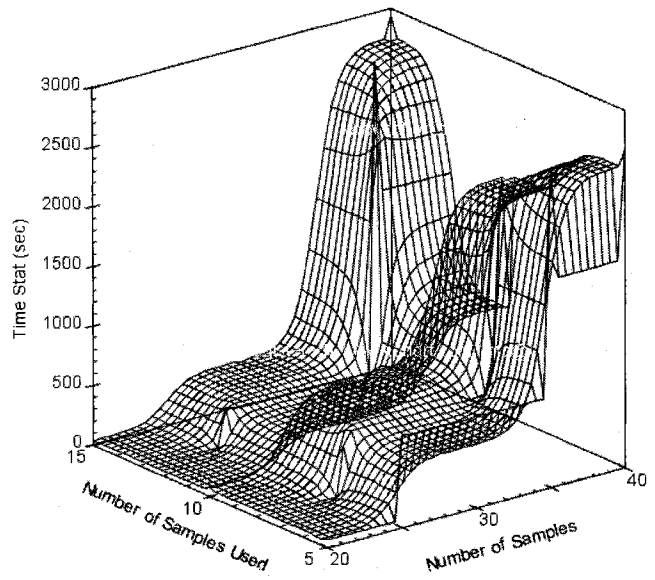

Figure 4.4 Time needed to find optimal statistical kriging model as a function of sampling intensity and sampling utilization.

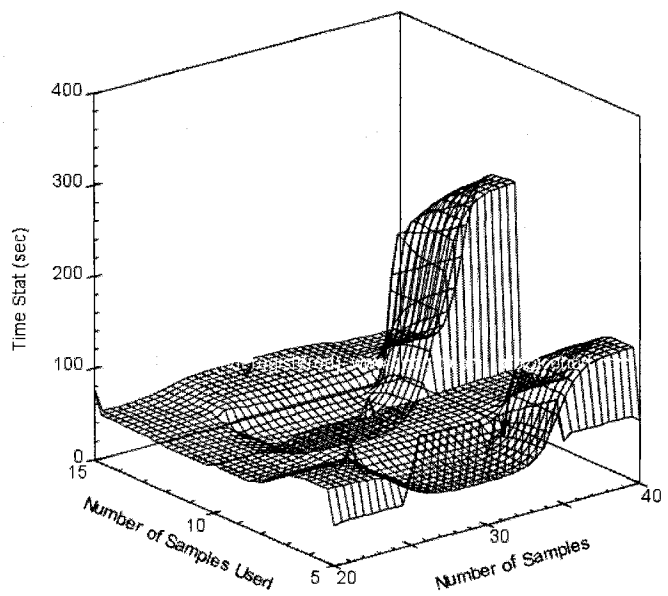

Figure 4.5 Time needed to find optimal ANN-based kriging model as a function of sampling intensity and sampling utilization.

Chapter 5

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

The overall objectives if this study were (1) to develop a hybrid ANN-based kriging interpolation model (KrigANN) designed to improve kriging's accuracy by taking advantage of ANN's unique pattern recognition capabilities, and (2) to test, evaluate and compare the predictive accuracies of the KrigANN model and conventional kriging by applying them to the same set of artificially generated datasets. The RDGI software demonstrated that developing an ANN-based kriging model was possible, and that neural networks offer a viable alternative to the regression analysis approach used in conventional kriging. The study also demonstrated that the KrigANN model produced more accurate interpolation results than regression-based kriging at low to medium degrees of spatial autocorrelation. This conclusion supports the findings of previous studies where the hybrid ANN-based kriging model was first proposed (Giroux and Dean, 2000; Giroux-Hughes, 2002). These findings also lead to the conclusion that the ANN-based procedures proposed in this study are appropriate alternatives to statistical approaches in quantifying the lag/semivariance relationships, when spatial data exhibit low to medium degrees of

autocorrelation.

Conversely, traditional kriging performed better than ANN-based kriging at high levels of spatial autocorrelation. However, in this case both the KrigANN and the conventional kriging models produced accurate interpolations. When strong spatial autocorrelation is present in the data, the use of a traditional approach to semivariogram development (designed for highly spatially autocorrelated data) is ideal. However, the use of ANN-based kriging in these situations does not produce dramatic decreases in prediction accuracy.

The use of search techniques to find the best possible regression-based semivariogram model and the best grouping characteristics for the semivariogram, as recommended by Giroux and Dean (2000) and Giroux-Hughes (2002), did not improve the relative accuracy of conventional kriging compared to ANN-based kriging.

One of the improvements of the ANN-based kriging model developed and evaluated in this study, compared to the model proposed by Giroux and Dean (2000) and Giroux-Hughes (2002) was the ability of the model used here to construct kriging estimation variances. This ability is an important characteristic of kriging interpolations, because it provides information about the confidence of the interpolation. In many cases, this information is essential for decision making, where a certain action can be taken only if the estimation is within a specified confidence interval (Journel and Huijbreghts, 1978). Despite the fact that the ANN-based model produced slightly higher average estimation variances overall (average for statistical kriging = 33.46; average for ANN-based kriging =

45.83), this increase was minimal compared to the overall improvement in prediction accuracy at moderate degrees of spatial autocorrelation.

The ANN-based kriging model used in this study was not limited to produce only non-decreasing semivariograms. The results presented here imply that this may eliminate the need to de-trend datasets where the semivariance/lag relationship involves decreasing components.

Because the overall accuracy of ANN-based kriging is impacted by sampling intensity to a larger extent than is conventional kriging, ANN-based systems may require larger sample sizes than regression-based kriging. However, the ANN-based kriging model presented here still performed well with relatively small sample sizes.

Previous studies evaluating ANN-based kriging used the backpropagation training algorithm, which presented sometimes impractical processing times (Personal communication, Denis Dean, 2004). With this limitation in mind, the MEKA algorithm was used in this study due to its reported better speed and quality of solutions compared to standard backpropagation (Shah and Palmieri, 2000). The use of the MEKA training algorithm provided significantly improved speed to the ANN-based kriging model, while producing highly accurate predictions. This indicates that the MEKA algorithm is more appropriate than backpropagation for the proposed ANN-based interpolation method.

5.2. Recommendations for Further Research

The findings of this study indicate that the hybrid ANN-based kriging model has a potential to become a standard alternative to traditional kriging as an interpolation technique. However, the performance of ANN-based kriging systems could be further evaluated and improved as follows:

1) Consider higher variation and number of sampling intensity sizes

The experimental design used here allowed for the evaluation of the effect of the number of samples used to develop the model on its overall prediction accuracy, but the range of sample sizes investigated (20 to 40) was rather small. Consideration of more sample sizes and higher variability of the sample sizes may allow inference of more specific information about the effect of the sampling intensity on the accuracy of the KrigANN model.

2) Consider distance in the selection of sample utilization sizes

In this study, it was shown that the performance of the ANN-based kriging was negatively impacted by higher sampling utilization sizes. This is counterintuitive, to say the least. This unexpected result may be explained by the fact that the sample points used for calculating an interpolation value were chosen without regards to their distance from the point being interpolated. Therefore, depending on the location of the original sample cells in relation to the point being interpolated, the samples used in interpolation could be

located at a distance beyond the range of spatial autocorrelation present in the dataset. Such sample cells would be too far away from the unknown site to make a relevant contribution to the accuracy of the interpolation. If sample utilization sizes were calculated by selecting only sample cells within an appropriate distance from the point being interpolated (i.e., as many samples as available within a maximum lag distance), the effect of sample utilization on the alternative ANN-based kriging model may be further explained.

3) Consider evaluating different neural network architecture and training parameters

In the present study, the controlled neural network parameters were the number of training epochs and the number of hidden nodes. The effect of different combinations of these parameters and of different training algorithms, feedback architectures, and so forth should be further evaluated.

4) Examine reasons for ANN's lower performance at higher autocorrelations

In this study, the ANN lost its advantage at high degrees of spatial autocorrelation. Further research should be directed at examining the reasons for this effect. A possible explanation is that the ANN models used in this study may have been too complex and may have overfitted the data. If this was the case, further research may find simpler ANNs to perform better than conventional kriging at all degrees of spatial autocorrelation.

5) *Direct further research at the application of the ANN-based kriging model to real world datasets.*

The current study strongly supports the validity of an ANN-based kriging model. This conclusion was made using artificially generated datasets, which provided valuable information on the effect of different model parameters on the overall prediction accuracy of the ANN-based kriging model. However, further validation using real-world data is an obvious next step in evaluating the KrigANN approach.

BIBLIOGRAPHY

Anderson, D.R., D.J. Sweeney, and T.A. Williams. 1991. Introduction to Statistics: Concepts and Applications. St. Paul, Minnesota: West Publishing Company. 714 pages.

Antonić, O.J., Križan, A. Marki, and D. Bukovec. 2001. *Spatio-temporal Interpolation of Climatic Variables over Large Region of Complex Terrain using Neural Networks.* **Ecological Modelling**. 138: 255–263.

Berk, R.A. 2004. Regression Analysis: A Constructive Critique. Thousand Oaks: California Sage Publications. 259 pages.

Bishop, C.M. 1995. Neural Networks for Pattern Recognition. New York: Oxford University Press. 482 pages.

Blackard, J.A., and D.J. Dean. 1996. *Evaluating Integrated Artificial Neural Networks and GIS Techniques for Predicting Likely Vegetative Cover Types.* In: **Proceedings of the 1st Southern Forestry GIS Conference** (Greg J. Arthaud and W. C. Hubbard, editors). The University of Georgia. Athens, Georgia. 416 pages.

Blackard, J. A., and D. J. Dean. 1998. *Comparative Predictive Accuracies of Artificial Neural Networks and Discriminant Analysis.* In: **Proceedings of the 2nd Southern Forestry GIS Conference** (H. J-H. Whiffen and W. C. Hubbard, editors). The University of Georgia. Athens, Georgia. 339 pages.

Blackard, J.A., and D.J. Dean. 1999. *Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables.* **Computers and Electronics in Agriculture.** 24:131-151.

Bocchi, S., A. Castrignano, F. Fornaro, and T. Maggiore. 2000. *Application of Factorial Kriging for Mapping Soil Variation at Field Scale.* **European Journal of Agronomy.** 13(4): 295-308.

Burrough, P. A., and R.A. McDonnell. 1998. Principles of Geographical Information Sytems (second edition). Oxford, England: Oxford Press. 333 pages.

Burrough, P.A. 1986. Principles of Geographical Information Systems for land Resources Assessment. Oxford, England: Clarendon Press. 193 pages.

Carle, S.F., and G.E. Fogg. 1996. *Transition Probability-based Indicator Geostatistics.* **Mathematical Geology.** 28(4): 453-477.

Carle, S.F., and G.E. Fogg. 1997. *Modeling Spatial Variability with One and Multidimensional Continuous-lag Markov Chains.* **Mathematical Geology**. 29(7): 891-918.

Clark, I. 1979. Practical Geostatistics. London, England: Elsevier Applied Science. 129 pages.

Cortez, L.P., A.J. Sousa, F.O. Durao, J.Q. Rogado and J.A. Simoes. 1997. *A Neural Network Approach for Natural Resources Estimation.* E.Y. Baafi and N.A. Schofield (eds.). In: **Geostatistics Wollongong '96**. 2: 1149-1162. Netherlands: Kluwer Academic Publishers.

Coulter, L., D. Stow, B. Kiracofe, C. Langevin, D. M. Chen, S. Daeschner, D. Service and J. Kaiser. 1999. *Deriving Current Land-Use Information for Metropolitan Transportation Planning Through Integration of Remotely Sensed Data and GIS.* **Photogrammetric Engineering and Remote Sensing**. 65(11):1293-1300.

Cressie, N. 1985. *Fitting Variogram Models by Weighted Least Squares.* **Mathematical Geology**. 17(5): 563-586.

Cressie, N. 1993. Statistics for Spatial Data. New York, New York: John Wiley and Sons. 900 pages.

Cressie, N. 1990. *The Origins of Kriging.* **Mathematical Geology**. 22: 239-252.

Critto, A., C. Carlon, and A. Marcomini. 2003. *A Characterization of Contaminated Soil and Groundwater Surrounding an Illegal Landfill (S. Giuliano, Venice, Italy) by Principal Component Analysis and Kriging.* **Environmental Pollution.** 122(2): 235-244.

Datum, M.S., F. Palmieri, and A. Moiseff. 1996. *An Artificial Neural Network for Sound Localization using Binaural Cues*, **Journal of the Acoustical Society of America**. 100(1): 372-383.

David, M. 1977. Geostatistical Ore Reserve Estimation. Amsterdam, Netherlands: Elsevier. 364 pages.

Dean, D. J., and E. M. Giroux-Hughes. 2004. *Neural Networks as an Alternative to Statistical Modeling in Kriging Procedures.* Presentation at the **Third International Conference on Geographic Information Science**, October 20-23 2004. University of Maryland.

Demirhan, M., A. Ozpinar, and L. Ozdamar. 2003. *Performance Evaluation of Spatial Interpolation Methods in the Presence of Noise.* **International Journal of Remote Sensing**. 24(6): 1237-1258.

Demyanov, V., Kanevsky M., Chernov S., Savelieva E., and Timonin V. 1998. *Neural Network Residual Kriging Application for Climatic Data.* **Journal of Geographic Information and Decision Analysis**. 2(2): 215-232.

E.S.R.I. 2003. *ARC/INFO Online Help*, Version 8.3. Redlands, California: Environmental Sciences Research Institute.

E.S.R.I. 2004. *ArcGIS Desktop Online Help*, Version 9. Redlands, California: Environmental Sciences Research Institute.

Emery, X. 2002. *Conditional Simulation of Nongaussian Random Functions.* **Mathematical Geology.** 34(1): 79-100.

Fahlman, S.E. 1989. *Faster Learning variations on backpropagation: an Empirical Study.* In: **Proceedings of the 1988 Connectionist Models Summer School**, Pittsburg. Eds D. Touretzky, G. Hinton, & T. Sejnowski, p. 38-51. San Mateo, California: Morgan-Kaufmann.

Fine, T.L. 1999. Feedforward Neural Network Methodology. New York, New York: Springer Publishing. 340 pages.

Fischer, M.M. 1994. *From Conventional to Knowledge-Based Geographic Information Systems.* **Computers, Environment and Urban Systems**. 18(4):233-242.

Foley, T.A., and Hagen H. 1994. *Advances in Scattered Data Interpolation.* **Survey on Mathematics for Industry.** 4:71-84.

Fournier, A., D. Fussel, and L. Carpenter. 1982. *Computer Rendering of Stochastic Models.* **Communications of the ACM.** 25(6): 371–384.

Fowler, C.J., and B.J. Clarke. 1996. *Corporate Distress Prediction: A Comparison of the Classification Power of a Neural Network and a Multiple Discriminant Analysis Model.* **Accounting Forum.** 20(3-4): 251-269.

Gaugush, R. F. 1993. *Kriging and Cokriging Applied to Water Quality Studies* U.S. Fish and Wildlife Service, Environmental Management Technical Center. (SuDoc I 49.109/2:93-R 027), Onalaska, Wisconsin.

Genton, M.G., and Furrer R. 1998. *Analysis of Rainfall Data by Simple Good Sense: is Spatial Statistics Worth the Trouble?* **Journal of Geographic Information and Decision Analysis.** 2(2): 11-17

Genton, M. G., 1998a. *Highly Robust Variogram Estimation.* **Mathematical Geology.** 30(2): 213-221.

Genton, M.G., 1998b. *Variogram Fitting by Generalized Least Squares using an Explicit Formula for the Covariance Structure.* **Mathematical Geology.** 30(4): 323-345.

Gimblett, R. H., and G. L. Ball. 1995. *Neural Network Architectures for Monitoring and Simulating Changes in Forest Resource Management.* **AI Applications.** 9(2):103- 123.

Giroux, E.M., and D.J. Dean. 2000. *Neural Networks as an Alternative to Statistical Modeling in the Semivariogram Analysis Portion of Co-Kriging Procedures.* In: **Proceedings of the Third Southern Forestry GIS Conference** (William G. Hubbard and J. B. Jordin, editors). Oct 10-12 2000; Athens. Athens, Georgia: The University of Georgia. Published as a CD.

Giroux-Hughes, E. M. 2002. *Neural Networks as an Alternative to Statistical Modeling in the Semivariogram Analysis Portion of Kriging Procedures.* **Master Thesis.** Fort Collins, Colorado: Colorado State University. 23 pages.

Goodchild, M.F. 1986. Spatial Autocorrelation. Catmog 47. Norwich, England: Geo Books.

Gunnarsson, F., S. Holm, P. Holmgren, and T. Thuresson. 1998. *On the Potential of Kriging for Forest Management Planning.* **Scandinavian Journal of Forest Research.** 13(2): 237-245.

Gurney, K.N. 1997. An Introduction to Neural Networks. London: UCL Press Limited. 234 pages.

Hartwig, F., and Dearing, B.E. 1979. Exploratory Data Analysis. Newberry Park, California: Sage Publications. 83 pages.

Hassoun, M.H. 1995. Fundamentals of Artificial Neural Networks. Cambridge, Massachusetts: MIT Press. 511 pages.

Haykin, S. 1994. Neural Networks. A Comprehensive Foundation. New York, New York: Macmillan College Publishing. 696 pages.

Haykin, S. 2001. Kalman Filtering and Neural Networks. New York, New York: Wiley Inter-Science, John Wiley and Sons. 284 pages.

Hilbert, D. W., and B. Ostendorf. 2001. *The Utility of Artificial Neural Networks for Modelling the Distribution of Vegetation in Past, Present and Future Climates.* **Ecological Modelling**. 146(1-3): 311-327.

Hinton, G.E. 1986. *Learning Distributed Representations of Concepts*. In: **Proceedings of the Eighth Annual Conference of the Cognitive Science Society.** Amherst, 1986. Hillsdale: Erlbaum. pp. 1-12.

Hosseini, E., J. Gallichand, and D. Marcotte. 1994. *Theoretical and Experimental Performance of Spatial Interpolation Methods for Soil Salinity Analysis.* **Trans. ASAE** 37: 1799–1807.

Isaaks, E. H., and R. M. Srivastava. 1989. Introduction to Applied Geostatistics. New York, New York: Oxford University Press. 561 pages.

Johnston, D.O. 1999. *Predicting Cyclosporine Area under the Concentration-time Curve is Better Achieved using Artificial Neural Networks than Linear Regression.* **British Journal of Clinical Pharmacology**. 47(5): 589-590.

Johnston, R.J. 1991. Multivariate Statistical Analysis in Geography: a Primer on the General Linear Model. New York, New York: John Wiley & Sons. 280 pages.

Jones, C. 1997. Geographic Information Systems and Computer Cartography. Edinburgh Gate, England: Addison Wesley Longman. 319 pages.

Journel, A. G. and C.J. Huijbregts. 1978. Mining Geostatistics. London: New York Academic Press. 1981 Edition. Repr. with corrections. 600 pages.

Kaluzny, S.P., S.C. Vega, T.P. Cardoso, and A.A. Shelly. 1998. S+ Spatial Stats. New York, New York: Springer Publishing. 327 pages.

Kanevski, M., M. F. Maignan, V. Demyanov, and M. F. Maignan. 1997b. *How neural network 2-D interpolations can improve spatial data analysis; Neural Network Residual Kriging (NNRK).* In: **Proceedings of IAMG '97; the Third annual conference of the International Association for Mathematical Geology,** 3: 549-554. Pawlowsky-Glahn: Vera Editor (Universitat Politecnica de Catalunya, Barcelona, Spain).

Kanevski, M., R. Arutyunyan, L. Bolshov, V. Demyanov, and M. F. Maignan. 1995. *Artificial Neural Networks and Spatial Estimations of Chernobyl Fallout.* **Geoinformatics**. 7:5-11.

Kanevski, M., R. Arutyunyan, L. Bolshov, V. Demyanov; S. Chernov, E. Savelieva, V. Timonin, M. Maignan, and M. F. Maignan. 1999. *Contaminated Forests; Recent Developments in Risk Identification and Future Perspectives.* **NATO Science Series.** Partnership Sub-series 2, Environmental Security, 58:249-256. Dordrecht, Netherlands: Kluwer Academic Publishers.

Kanevski, M., V. Demyanov, and M. Maignan. 1997a. *Spatial estimations and simulations of environmental data by using geostatistics and artificial neural networks.* In: **Proceedings of IAMG'97; the Third Annual Conference of the International Association of Mathematical Geology**, 3:533-538, Barcelona, Spain: V. Pawlowsky Glan (Ed.), CIMNE.

Kleinbaum, D.G., L.L. Kupper, and K.E. Muller. 1998. Applied Regression Analysis and Other Multivariate Methods. Pacific Grove, California: Duxbury Press. 798 pages.

Koike, K., and S. Matsuda. 2003. *Characterizing Content Distributions of Impurities in a Limestone Mine using a Feedforward Neural Network.* Natural Resources Research International Association for **Mathematical Geology**. 12(3): 209-222.

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

Lam, N., 1983. *Spatial Interpolation Methods: A Review.* **The American Cartographer.** 10(2): 129-149.

Lary, D. J., and H. Y. Mussa. 2004. *Using an Extended Kalman Filter Learning Algorithm for Feed-Forward Neural Networks to Describe Tracer Correlations.* **Atmospheric Chemistry and Physics Discussions**, European Geosciences Union, 4: 3653–3667.

Laurini, R., and D. Thompson. 1994. Fundamentals of Spatial Information Systems. San Diego, California: Academic Press. 680 pages.

Lee, J. 1991. *Comparisons of Existing Methods for Building Triangular Irregular Network Models of Terrain from Grid Digital Elevation Models.* **International Journal of Geographical Information Systems.** 5: 267-285.

Lee, J., and D.W.S. Wong. 2001. Statistical analysis with ArcView GIS. New York, New York: John Wiley. 192 pages.

Little, L.S., D. Edwards, D.E. Porter. 1997. *Kriging in estuaries: As the crow flies, or as the fish swim?* **Journal of Experimental Marine Biology and Ecology**, 213(1): 1-11.

Liu, C.W., C.S. Jang, and C.M. Liao. 2004. *Evaluation of Arsenic Contamination Potential using Indicator Kriging in the Yun-Lin Aquifer (Taiwan).* **Science of the Total Environment**. 321(1-3): 173-188.

Liu, X., A. K. Skidmore and H. Van Costen. 2002. *Integration of Classification Methods for Improvement of Land-Cover Map Accuracy.* **Journal of Photogrammetry and Remote Sensing**. 56:257-268.

Longley, P.A., M.F. Goodchild, D.J. Maguire, and D.W. Rhind. 2001. (Reprinted March 2002) Geographic Information Systems and Science. London, England: John Wiley and Sons. 454 pages.

Lucifredi, A., C. Mazzieri, and M. Rossi. 2000. *Application of Multiregressive Linear Models, Dynamic Kriging Models and Neural Network Models to Predictive Maintenance of Hydroelectric Power Systems.* **Mechanical Systems and Signal Processing.** 14(3): 471-494.

Maglione, D. S., and A. M. Diblasi. 2004. *Exploring a Valid Model for the Variogram of an Isotropic Spatial Process.* **Stochastic Environmental Research and Risk Assessment.** 18(6): 366–376.

Mandelbrot, B. 1983. The fractal geometry of nature. New York, New York: W.H. Freeman. 468 pages.

Matheron, G. 1971. *The Theory of Regionalised Variables and its Applications.* **Le Cahier du Centre de Morphologie Mathématique de Fontainebleau.** 5: 211-222.

Matsoukas, C., and S. Islam. 1999. *Fusion of Radar and Rain Gage Measurements for an Accurate Estimation of Rainfall.* **Journal of Geophysical Research.** 104(D24): 437-431, 450.

McBratney, A. B., and Webster R. 1986. *Choosing Functions for Semivariograms of Soil Properties and Fitting Them to Sampling Estimates.* **Journal of Soil Science.** 37: 617-639

McCulloch, W. S., and W. H. Pitts. 1943. *A Logical Calculus of the Ideas Imminent in Nervous Activity.* **Bulletin of Mathematical Biophysics.** 5: 115-133.

Merwin, D.A., R.G. Cromley, and D.L. Civco. 2002. *Artificial Neural Networks as a Method of Spatial Interpolation for Digital Elevation Models.* **Cartography and Geographic Information Science**. 29(2): 99-110.

Mickey, R. M., O.J. Dunn, and V. Clark Hoboken. 2004. Applied Statistics: Analysis of Variance and Regression. Hoboken, New Jersey: Wiley-Interscience. 448 pages.

Miller, H. J. 2004. *Tobler's First Law and Spatial Analysis.* **Annals of the Association of American Geographers**. 94(2): 284–289

Minsky, M., and Papert S. 1969. Perceptrons: An Introduction to Computational Geometry. Cambridge, MA: MIT Press. 258 pages.

Moran, P.A.P. 1950. *Notes on Continuous Stochastic Phenomena,* **Biometrika.** 37:17-23.

Moyeed, R.A., and Papritz A. 2002. *An Empirical Comparison of Kriging Methods for Nonlinear Spatial Point Prediction.* **Mathematical Geology**. 34(4): 365-386.

Mukhopadhyay, A. 1999. *Spatial Estimation of Transmissivity using Artificial Neural Network.* **Ground Water**. 37(3): 458-464.

Muller, W.G., and Zimmerman D.L. 1999. *Optimal designs for variogram estimation.* **Environmetrics**. 10(1): 23-37.

Nigrin, A. 1993. Neural Networks for Pattern Recognition. Cambridge, Massachusetts: MIT Press. 511 pages.

Oliver, M.A., and R. Webster. 1990. *Kriging: a Method of Interpolation for Geographical Information Systems.* **International Journal of Geographical Information Science**. 4(4): 313-332

Palmieri, F., M., A. Datum, and A. Moiseff. 1991a. *Sound Localization with a Neural Network Trained with the Multiple Extended Kalman Algorithm.* **International Joint Conference on Neural Networks**. Seattle. 1: 125-131.

Palmieri, F., M. Datum, A. Shah, and A. Moiseff. 1991b. *Learning Binaural Sound Localization through a Neural Network.* In**: Proceedings of the 1991 Institute of Electrical and Electronic Engineers Bioengineering Conference**, Seventeenth Annual Northeast, 13-14.

Parker, DB. 1982. *Learning-logic.* Invention report, S81-64, File 1. Stanford University Office of Technology Licensing.

Patterson, D. 1996. Artificial Neural Networks: Theory and Applications. Singapore, Prentice Hall.

Peitgen, H. O., J. Hartmut, and D. Saupe, 1992. Chaos and Fractals: New Frontiers of Science, New York, New York: Springer-Verlag. 984 pages.

Peitgen, H.O., and D. Saupe. 1988. The Science of Fractal Images. New York, New York: Springer-Verlag. 312 pages.

Peuker, T.K., R. J. Fowler, J.J. Little, and D.M Mark. 1978. *The Triangulated Irregular Network.* In: **Proceedings of the DTM Symposium.** American Society of Photogrammetry – American Congress on Surveying and Mapping, St. Louis, Missouri: 24-31.

Rigol, J.P., H.J. Claire, and N. Stuart. 2001. *Artificial Neural Networks as a Tool for Spatial Interpolation.* 15(4): 323-343

Ripley, B. 1981. Spatial Statistics. New York: John Wiley. 252 pages.

Ripley, B.D. 1996. Pattern Recognition and Neural Networks. New York, New York: Cambridge University Press. 403 pages.

Rumelhart, D.E, R. Urbin, R. Golden, and Y. Chauvin. 1995. *Backpropagation: The basic Theory.* In: **Backpropagation: Theory, Architectures, and Applications.** (Y. Chauvin and D.E. Rumelhart, eds.) Hillsdale, New Jersey: Laurence Erlbaum Associates. Pages 1-33.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986a. *Learning Internal Representations by Error Propagation.* In Rumelhart, D., McClelland, J., and the PDP Research Group editors, Parallel Distributed Processing. 1: 318-362. Cambridge, MA: MIT Press.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986b. *Learning Representations of Back–Propagation.* **Nature.** 323: 533–536.

Saito, H., and P. Goovaerts. 2001. *Accounting for Source Location and Transport Direction into Geostatistical Prediction of Contaminants.* **Environmental Science & Technology.** 35(24): 4823-4829.

SAS Institute, Inc. 1999. *Statistical analysis system (SAS)*, version 8. Cary, North Carolina.

Schloeder, C.A., N.E. Zimmerman, and M.J. Jacobs. 2001. *Comparison of Methods for Interpolating Soil Properties using Limited Data.* **Soil Science Society of America Journal.** 65:470-479.

Schultz, C.A., S.C. Myers, J. Hipp, and C.J. Young. 1998. *Nonstationary Bayesian Kriging: A Predictive Technique to Generate Spatial Corrections for Seismic Detection, Location, and Identification.* **Bulletin of the Seismological Society of America**. 88 (5): 1275-1288.

Shah, S., and F. Palmieri. 1990. *MEKA-a fast, local algorithm for training feedforward neural networks.* **IJCNN International Joint Conference on Neural Networks**, Jun 17-21 1990. 3:41-46.

Singhal, S., and L. Wu. 1989. *Training Multilayer Perceptrons with the Extended Kalman Filter.* In **Proceedings of the International Conference on ASSP**. 1187–1190.

Skidmore, A. K., B. J. Turner, W. Brinkhof, and E. Knowles. 1997. *Performance of a Neural Network: Mapping Forests using GIS and Remotely Sensed Data.* **Photogrammetric Engineering and Remote Sensing**. 65(3):501-514.

Stan, O., and E. Kamen. 2000. *A Local Linearized Least Squares Algorithm for Training Feedforward Neural Networks.* **Institute of Electrical and Electronic Engineers Transactions on Neural Networks**. 11(2): 487-495.

StatSoft, Inc. 2003. "Neural Networks", http://www.statsoft.com/textbook/stneunet.html. Accessed: March 13, 2005

Sunar, F., and C. Ozkan. 2001. *Forest Fire Analysis with Remote Sensing Data.* **International Journal of Remote Sensing**. 22(12): 2265-2277.

Sung, D. G., S. H. Lim, J. W. Ko, and G. S. Chao 2001. *Scenic Evaluation of Landscape for Urban Design Purposes using GIS and ANN.* **Landscape and Urban Planning**. 56(1-2): 75-85.

Tapiador, F. J., and J. L. Casanova. 2003. *Land Use Mapping Methodology using Remote Sensing for the Regional Planning Directives in Segovia, Spain.* **Landscape and Urban Planning**. 62: 103-115.

Tappeiner, U., G. Tappeiner, J. Aschenwald, E. Tasser, and B. Ostendorf. 2001. *GIS-Based Modeling of Spatial Pattern of Snow Cover Duration in an Alpine Area.* **Ecological Modelling**. 138(1-3): 265-275.

Taylor, A.E. 1955. Advanced Calculus. Waltham, Massachusetts: Blaisdel Publishing Company. 786 pages.

Tobler, W. R. 1970. *A Computer Movie Simulating Urban Growth in the Detroit Region.* **Economic Geography**. 46: 234–40.

U.S. Army Corps of Engineers. HTRW Report. 1997. *Practical Aspects of Applying Geostatistics at Hazardous, Toxic, and Radioactive Waste Sites.* CEMP-RT, Technical report ETL: 1110-1-175.

U.S. Geological Survey. 1987. *Digital Elevation Model – Data Users' Guide.* US Department of the Interior, USGS, Reston, Virginia.

Walker, D.D., and J.C. Loftis. 1997. *Alternative Spatial Estimators for Ground-Water and Soil Measurements.* **Ground Water**. 35(4):593-601

Wang, L., P.M. Wong, and S.A.R. Shibli. 1999b. *Modeling Porosity Distribution in the A'nan Oilfield: Use of Geological Quantification, Neural Networks, and Geostatistics.* **SPE Reservoir Evaluation and Engineering.** 2(6): 527-532.

Wang, L., P.M. Wong, M. Kanevski, and T.D. Gedeon. 1999a. *Combining Neural Networks with Kriging for Stochastic Reservoir Modeling.* **In Situ**. 23(2): 151-169.

Wasserman, P.D. 1989. Neural Computing: Theory and Practice. New York: Van Nostrand Reinhold. 230 pages.

Webster, R. and Oliver M.A. 1990. Statistical Methods in Soil and Land Resource Survey. Oxford, England: Oxford University Press. 316 pages.

Weibel, R. and Heller, M. (1991). *Digital Terrain Modelling.* In: Maguire, D. J., Goodchild, M. F., and Rhind, D. W. (eds.) Geographical Information Systems: Principles and Applications, pp.269-297, Longman, London.

Werbos, P. J. 1974. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* **PhD thesis**. Harvard University.

Widrow, B, and Hoff M. 1960. *Adaptive switching circuits.* 1960 IRE WESCON convention record. New York, New York: Institute of Radio Engineers.

Windale Technology. 2001. *MLP/X Neural Network ActiveX Control and COM Object.* http://windale.com/mlpx.php3. Accessed: February 6, 2005.

Wingle, W.L. 1992. *Examining Common Problems Associated with Various Contouring Methods, Particularly Inverse-Distance Methods, using Shaded Relief Surfaces.* In**: Geotech '92 Conference Proceedings**, Lakewood, Colorado, pp. 362-376.

Wong, P.M., I.J. Taggart, and T.D. Gedeon. 1995. *Use of Neural Network Methods to Predict Porosity and Permeability of a Petroleum Reservoir.* **AI Applications.** 9(2): 27-37.

Yama, B.R., and G.T. Lineberry. 1999. *Artificial Neural Network Application for a Predictive Task in Mining.* **Mining Engineering**. 51(2): 59-64.

Yuan, M., and C.E. Duchon. 2001. *Estimation of Daily Area-Average Rainfall in Central Florida using Arithmetic Averaging and Kriging.* **Physical Geography**. 22(1): 42-58

Zhou, J., and D. L. Civco. 1996. *Using Genetic Learning Neural Networks for Spatial Decision Making in GIS.* **Photogrammetric Engineering and Remote Sensing**. 62(11): 1287-1295.

Zimmerman, D. L., and M.B. Zimmerman. 1991. *A Comparison of Spatial Semivariogram Estimators and Corresponding Ordinary Kriging Predictors.* **Technometrics**. 33: 77-92

APPENDIX A

Appendix A. List of variables used for evaluating model's performance, including descriptive statistics and descriptions (n = 2250).

	VARIABLE NAME	MEAN	STANDARD DEVIATION	MINIMUM	MAXIMUM	VARIABLE DESCRIPTION
1	IntNum	n/a	n/a	n/a	n/a	Iteration number
2	NumRows	n/a	n/a	50	50	Number of rows in the raster map
3	NumCols	n/a	n/a	50	50	Number of columns in the raster map
4	NumSamp	30	10	20	40	Number of samples used to develop the model
5	NumSampUsed	10	5	5	15	Number of samples used to calculate the estimate
6	H	0.5	0.3	0.1	0.9	H Value
7	MI	0.8	0.2	0.3	1.0	Moran's Index
8	BestStatModel	n/a	n/a	n/a	n/a	Best statistical model (1 = linear, 2 = Gaussian, 3 = circular, 4 = spherical, 5 = exponential)
9	BestShape	1647.03	1439.35	160.11	6048.78	Best statistical model's shape parameter
10	BestNumGroups	107	102	10	441	Best statistical model number of groups
11	NumAnnTrainEpoch	348	685	10	2000	ANN model number of training epochs
12	NumHidden	16	12	5	75	ANN model number of hidden nodes
13	DecreasingSV	n/a	n/a	n/a	n/a	TRUE if ANN semivariogram never decreases; FALSE otherwise.

14	StatRMSEWith	139.100	288.251	6.892	2584.330	RMSE between actual and predicted (via statistical model) value including sample cells
15	StatRMSEWithout	139.953	290.105	6.934	2605.250	RMSE between actual and predicted (via statistical model) value without sample cells
16	DistCorrStatWith	0.268	0.166	-0.082	0.831	Correlation between ABS (predicted –actual) values and distance to sample points, statistical model and including sample cells
17	DistCorrStatWithout	0.252	0.173	-0.094	0.833	Correlation between ABS (predicted –actual) values and distance to sample points, statistical model and without sample cells
18	MinStdStatWith	26.136	27.377	0.000	120.286	Minimum Kriging standard deviation, statistical model/including sample points
19	MeanStdStatWith	33.401	27.117	0.003	128.422	Mean Kriging standard deviation, statistical model/including sample points
20	MaxStdStatWith	50.603	63.828	0.063	1418.020	Maximum Kriging standard deviation, statistical model/including sample points
21	StdStdStatWith	2.940	4.326	0.000	108.939	Std of Kriging standard deviations, statistical model/including sample points
22	MinStdStatWithout	27.451	26.966	0.000	120.926	Minimum Kriging standard deviation, statistical model/without sample points
23	MeanStdStatWithout	33.468	27.123	0.003	128.708	Mean Kriging standard deviation, statistical model/without sample points
24	MaxStdStatWithout	50.603	63.828	0.063	1418.020	Maximum Kriging standard deviation, statistical model/without sample points
25	StdStdStatWithout	2.871	4.292	0.000	109.572	Std of Kriging standard deviations, statistical model/without sample points

26	StdDistCorrStatWith	0.756	0.260	-0.193	0.984	Correlation between kriging standard deviation and distance to sample points, statistical model and including sample points
27	StdDistCorrStatWithout	0.756	0.261	-0.194	0.984	Correlation between kriging standard deviation and distance to sample points, statistical model and without sample points
28	TimeStat	923	1128	23	11290	Number of seconds needed to find best statistical model
29	AnnRMSEWith	131.815	257.731	8.769	2467.060	RMSE between actual and predicted (via ANN model) value including sample cells
30	AnnRMSEWithout	132.635	259.374	8.840	2482.000	RMSE between actual and predicted (via ANN model) value without sample cells
31	DistCorrAnnWith	0.102	0.123	-0.257	0.620	Correlation between ABS (predicted –actual) values and distance to sample points, ANN model and including sample cells
32	DistCorrAnnWithout	0.083	0.124	-0.268	0.612	Correlation between ABS (predicted –actual) values and distance to sample points, ANN model and without sample cells
33	MinStdAnnWith	0.000	0.000	0.000	0.000	Minimum Kriging standard deviation, ANN model/including sample points
34	MeanStdAnnWith	45.279	28.189	6.645	150.364	Mean Kriging standard deviation, ANN model/including sample points
35	MaxStdAnnWith	144.120	200.810	7.489	2474.540	Max Kriging Std, ANN model/including sample points
36	StdStdAnnWith	13.877	15.780	0.670	167.710	Std of Kriging standard deviations, ANN model/including sample points
37	MinStdAnnWithout	27.566	29.969	0.000	134.878	Minimum Kriging standard deviation, ANN model/without sample points

38	MeanStdAnnWithout	45.829	28.533	6.698	151.577	Mean Kriging Std, ANN model/without sample points
39	MaxStdAnnWithout	144.120	200.810	7.489	2474.540	Maximum Kriging standard deviation, ANN model/without sample points
40	StdStdAnnWithout	10.999	17.052	0.000	168.913	Std of Kriging standard deviations, ANN model/without sample points
41	StdDistCorrAnnWith	0.266	0.169	-0.329	0.805	Correlation between kriging standard deviation and distance to sample points, ANN model and including sample points
42	StdDistCorrAnnWithout	0.270	0.310	-3.136	1.579	Correlation between kriging standard deviation and distance to sample points, ANN model and without sample points
43	TimeAnn	61	52	1	801	Number of seconds needed to find ANN model