

Mineral Resource Estimates with Machine Learning and Geostatistics

by

Matthew Samson

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science

in

Mining Engineering

Department of Civil and Environmental Engineering
University of Alberta

© Matthew Samson, 2019

ABSTRACT

Mineral resource estimation is an integral part of making informed decisions while evaluating the feasibility of a mining operation. Geostatistical tools estimate geological and spatial features with the assumptions of first and second-order stationarity. In the modelling process, Geostatisticians make subjective decisions regarding stationarity, potentially introducing bias into the estimates. Kriging is considered the best unbiased linear estimation technique for modelling geological and spatial features; however, in domains where data is non-Gaussian, and features are complex, the assumption of stationarity can cause difficulties in the modelling process. The purpose of this thesis is to present two new estimation techniques. The first estimation technique uses machine learning for geological and spatial estimations without assuming first and second-order stationarity, minimizing human interaction and potentially reducing estimation bias. The second technique is a hybrid method that consists of using geostatistical methods combined with the machine learning method. Integrating geostatistics and machine learning improves geological and spatial estimation in situations that have complex features, poorly defined domains, or non-Gaussian data.

Elliptical radial basis function networks (ERBFN) and k-means clustering are used for estimation. An ERBFN machine learning method takes advantage of a Gaussian function to generate geological estimates similar to kriging. An ERBFN does not require the assumption of stationarity and the only input features required are the spatial coordinates of the known data. The parameter required for the ERBFN is the number of nodes to model the estimations domain. Each node learns a unique anisotropy allowing for complex features to be modelled.

The hybrid estimation takes advantage of the machine learning estimation from the ERBFN and uses it as an exhaustive secondary data in ordinary intrinsic collocated cokriging. The hybrid estimation requires the assumption of stationarity and variograms must be modelled. Combining machine learning and geostatistics takes advantage of the unbiasedness of kriging while including the non-stationary features modelled in the ERBFN.

To validate the estimation techniques, examples are simulated and sampled. Machine learning, hybrid, and kriging estimates are made using the sampled dataset and compared to the exact truth. Multiple validation checks are used to compare the different estimates. The coefficient of determination and root mean squared error are used to assess model performance. Plots of the estimates and error maps are used for visual inspection to determine if modelling artifacts are present. Histograms determine if the mean and data distribution reproduction are reasonable.

The machine learning estimation method developed in this thesis is shown to produce similar results to simple kriging without requiring the assumption of first and second-order stationary. The hybrid estimation technique developed in this thesis appears to outperform simple kriging in scenarios that demonstrate non-stationary features, poorly defined domains, and non-Gaussian data. The research work of the thesis has led to a significant contribution in making spatial and geological predictions.

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Clayton Deutsch for the support, expert guidance, and inspiration essential to this thesis. I would also like thank all my colleagues at the CCG for there invaluable advice and support while writing my thesis. To my parents Cathy and Phil, my brother Christian, and girlfriend Allyson - thank you for the continuous support and motivation.

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Problem Setting and Background | 1 |
| 1.1.1 | Machine Learning | 2 |
| 1.1.2 | Kriging | 4 |
| 1.1.3 | Estimation Criteria and Validation | 4 |
| 1.2 | Thesis Statement | 5 |
| 1.3 | Thesis Outline | 5 |
| 2 | Geostatistical and Machine Learning Framework | 7 |
| 2.1 | Machine Learning Framework | 7 |
| 2.1.1 | Choosing a Machine Learning Algorithm | 7 |
| 2.1.2 | Regression | 9 |
| 2.1.3 | Training | 12 |
| 2.1.4 | Neural Networks | 14 |
| 2.1.5 | Optimization Equations | 17 |
| 2.1.6 | K-Means | 18 |
| 2.2 | Kriging Framework | 19 |
| 2.2.1 | Stationarity | 20 |
| 2.2.2 | The Variogram | 22 |
| 2.2.3 | Kriging | 22 |
| 2.3 | Estimation Criteria | 25 |
| 3 | Proposed Estimation Methods | 28 |
| 3.1 | Geostatistical Prediction With a Neural Network | 28 |
| 3.1.1 | Simple Linear Neural Network | 28 |
| 3.1.2 | Elliptical Radial Basis Function Neural Network | 35 |
| 3.2 | Combining Machine Learning and Geostatistical Estimation | 51 |
| 3.3 | Sensitivity Studies | 55 |
| 4 | JA Extension Copper Porphyry Deposit Case Study | 82 |
| 4.1 | Available Data | 82 |
| 4.2 | Modelling Parameters | 83 |
| 4.3 | Results and Analysis | 86 |
| 5 | Conclusion | 94 |

Table of Contents

| | | |
|-----|---|------------|
| 5.1 | Topics Covered and Contribution | 94 |
| 5.2 | Limitations and Future Work | 95 |
| 5.3 | Recommendations | 96 |
| | References | 97 |
| | A Appendices | 100 |

LIST OF TABLES

| | | |
|------|--|----|
| 3.1 | K-fold Simple Neural Network Results Summary | 31 |
| 3.2 | K-fold Kriging Results Summary | 32 |
| 3.3 | K-fold Simple Neural Network Using XYZ Data and Nearest 40 Data Results Summary | 33 |
| 3.4 | K-fold Simple Neural Network Using XYZ Data and Nearest 40 Data Using 5 Hidden Layers Results Summary | 35 |
| 3.5 | Summary Kernel Comparison Radial Basis Ensemble Using XYZ Data | 41 |
| 3.6 | Summary of Radial Basis Ensemble Using XYZ Data | 42 |
| 3.7 | Summary Kernel Comparison Elliptical Radial Basis Ensemble Using XYZ Data | 47 |
| 3.8 | Summary of Elliptical Radial Basis Ensemble Using XYZ Data | 47 |
| 3.9 | Summary of Elliptical Radial Basis Ensemble Using Doughnut Data Set | 50 |
| 3.10 | Summary of Simple Kriging Using Doughnut Data Set | 51 |
| 3.11 | Summary of ICCK Plus ERBFN Using Doughnut Data Set | 54 |
| 3.12 | Summary of Kriging Number of Data Sensitivity Analysis | 62 |
| 3.13 | Summary of Machine Learning Number of Data Sensitivity Analysis | 62 |
| 3.14 | Summary of ML+ICCK Number of Data Sensitivity Analysis | 63 |
| 3.15 | Summary of OICCK Plus ERBFN Using Doughnut Data Set Missing Drill Holes | 66 |
| 3.16 | Summary of Kriging Using Doughnut Data Set Missing Drill Holes | 66 |
| 3.17 | Summary of Kriging Using Lognormal Data Sets | 70 |
| 3.18 | Summary of Machine Learning Using Lognormal Data Sets | 71 |
| 3.19 | Summary of ML+ICCK Using Lognormal Data Sets | 71 |
| 3.20 | Summary of Kriging Using Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study | 75 |
| 3.21 | Summary of ML Using Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study | 75 |
| 3.22 | Summary of ML+ICCK Using Lognormal Drill Holes Spacing/Variogram Range Sensi- tivity Study | 75 |
| 3.23 | Summary of Kriging Increasing Major/Minor Variogram Range Sensitivity Study | 79 |
| 3.24 | Summary of ML Increasing Major/Minor Variogram Range Sensitivity Study | 79 |
| 3.25 | Summary of ML+ICCK Increasing Major/Minor Variogram Range Sensitivity Study | 79 |
| 4.1 | JA extension Copper Porphyry Deposit Modelling Grid Definition | 86 |
| 4.2 | JA extension Copper Porphyry Deposit SK Summary | 90 |
| 4.3 | JA extension Copper Porphyry Deposit ML Summary | 90 |
| 4.4 | JA extension Copper Porphyry Deposit ML+OICCK Summary | 90 |
| 4.5 | JA Extension Copper Porphyry Deposit Kriging Cut-off Grade K-fold 0 | 91 |

| | | |
|-----|--|----|
| 4.6 | JA Extension Copper Porphyry Deposit ML Cut-off Grade K-fold 0 | 91 |
| 4.7 | JA Extension Copper Porphyry Deposit ML+OICCK Cut-off Grade K-fold 0 | 91 |

LIST OF FIGURES

| | | |
|------|---|----|
| 1.1 | Machine Learning Steps | 3 |
| 1.2 | Machine Learning Types | 3 |
| 2.1 | Scikit Learn Algorithm Cheat-sheet | 9 |
| 2.2 | Simple Regression Network | 10 |
| 2.3 | 2D/3D Linear Regression | 10 |
| 2.4 | Sigmoid Function | 11 |
| 2.5 | Training vs Test Cost Examples | 12 |
| 2.6 | Training Cost Examples for Determining Learning Rate | 13 |
| 2.7 | Training Cost Vs R^2 | 13 |
| 2.8 | Simple Neural Network | 14 |
| 2.9 | Tanh Function | 15 |
| 2.10 | Gaussian Function | 16 |
| 2.11 | RELU Function | 17 |
| 2.12 | K-Means Example | 19 |
| 2.13 | K-Means Error Example | 20 |
| 2.14 | Example Experimental Variogram and Variogram Model | 23 |
| 2.15 | K-Fold Data Setup Example | 25 |
| 2.16 | Location Map of Error | 26 |
| 2.17 | Example Estimation Slice (Left: With artifacts, Right: No artifacts) | 26 |
| 3.1 | Simulated and Drilled Data | 29 |
| 3.2 | Simple One Layer Linear Neural Network | 30 |
| 3.3 | K-Fold Results From Simple Linear Neural Network | 31 |
| 3.4 | K-Fold Results From Simple Linear Neural Network Using XYZ data and nearest 40 Data | 33 |
| 3.5 | Simple Five Layer Linear Neural Network | 34 |
| 3.6 | K-Fold Results From Simple Linear Neural Network Using XYZ data and nearest 40 Data | 35 |
| 3.7 | Simple RBF Example | 37 |
| 3.8 | 3D RBF Example | 38 |
| 3.9 | Radial Basis Function Network In Tensorboard | 38 |
| 3.10 | Example of Using Different Number of Kernel While Train an RBFN | 40 |
| 3.11 | Example of Using Different Number of Nodes RBFN k-fold 3 Results | 41 |
| 3.12 | K-Fold Results Variogram Results From Radial Basis Neural Network Using XYZ Data | 42 |
| 3.13 | K-Fold Results From Radial Basis Neural Network Using XYZ Data | 43 |
| 3.14 | K-Fold Results From Radial Basis Neural Network Using XYZ Data | 44 |

| | | |
|------|--|----|
| 3.15 | Simple Simulated and Sampled Data Example for RBFN to ERBFN | 45 |
| 3.16 | Results From Simple Simulated and Sampled Data Example for RBFN to ERBFN | 46 |
| 3.17 | Example of Using Different Number of Nodes ERBFN k-fold 3 Results | 47 |
| 3.18 | K-Fold Results From Elliptical Radial Basis Neural Network Using XYZ Data | 48 |
| 3.19 | Complex Structure Example, Doughnut Data | 49 |
| 3.20 | K-Fold Results From Elliptical Radial Basis Neural Network Using Doughnut Data Set . | 50 |
| 3.21 | Drill Hole Data Reproduction for ERBFN and Kriging | 51 |
| 3.22 | Machine Learning And Geostatistical Estimation Work Flow | 52 |
| 3.23 | K-Fold Results From ICCK Plus ERBFN Using Doughnut Data Set | 53 |
| 3.24 | Drill Hole Data Reproduction for ERBFN and Kriging | 54 |
| 3.25 | Small Example Demonstrating the Difference Between ICCK and OICCK | 55 |
| 3.26 | Simulated Data Set for Number of Data Sensitivity Study | 56 |
| 3.27 | Sensitivity Study 100 Drill Holes Main | 57 |
| 3.28 | Sensitivity Study 75 Drill Holes Main | 58 |
| 3.29 | Sensitivity Study 50 Drill HolesvMain | 59 |
| 3.30 | Sensitivity Study 25 Drill Holes Main | 60 |
| 3.31 | Summary Graphs of RMSE and R^2 Values for Number of Data Sensitivity Study | 61 |
| 3.32 | Doughnut Data Set With Missing Drill Holes | 64 |
| 3.33 | K-Fold Results From OICCK Plus ERBFN Using Doughnut Data Set Missing Drill Holes | 65 |
| 3.34 | Cumulative Distribution Functions of Lognormal Data Sets | 67 |
| 3.35 | Example of Lognormal Data Set | 68 |
| 3.36 | Example of Lognormal Estimation Standard Deviation = 2.0 | 69 |
| 3.37 | Lognormal Sensitivity Summary Graphs | 70 |
| 3.38 | Example of Lognormal Data Set DH/Varg Range = 0.46 | 72 |
| 3.39 | Example of Lognormal Data Set DH/Varg Range = 1.46 | 72 |
| 3.40 | Example of Lognormal Estimation Drill Holes Spacing/Variogram Range = 0.46 | 73 |
| 3.41 | Summary Graph for Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study | 74 |
| 3.42 | Example of Simulated and Drilled Data For Major/Minor Vargiogram Range Sensitivity Study = 2.75 | 76 |
| 3.43 | Example of Simulated and Drilled Data For Major/Minor Vargiogram Range Sensitivity Study = 5.25 | 76 |
| 3.44 | Example of Major/Minor Vargiogram Range Sensitivity Study = 5.25 Main | 77 |
| 3.45 | Summary Graph for Increasing Major/Minor Vargiogram Range Sensitivity Study | 78 |
| 3.46 | Run time vs Number of Nodes for 100 Training Steps | 80 |
| 3.47 | Run Time vs Number of Nodes for Prediction | 81 |
| 4.1 | JA extension Copper Porphyry Deposit Location Map of Cu Data | 83 |

| | | |
|-----|--|----|
| 4.2 | JA extension Copper Porphyry Deposit Data Spacing Study | 84 |
| 4.3 | JA extension Copper Porphyry Deposit De-clustered Histogram of Cu Data | 85 |
| 4.4 | JA extension Copper Porphyry Deposit Cu Variograms | 86 |
| 4.5 | JA extension Copper Porphyry Deposit K-Fold 2 Results | 88 |
| 4.6 | JA extension Copper Porphyry Deposit Final Estimate Results | 89 |
| 4.7 | JA extension Copper Porphyry Deposit Cu Swath Plot X | 92 |
| 4.8 | JA extension Copper Porphyry Deposit Cu Swath Plot Y | 92 |
| 4.9 | JA extension Copper Porphyry Deposit Cu Swath Plot Z | 93 |

LIST OF ABBREVIATIONS

| | |
|-------|--|
| 2d | Two dimensional |
| 3d | Three dimensional |
| Adam | Adaptive Moment Estimation Optimization |
| CV | Coefficient of Variation |
| CDF | Cumulative Density Function |
| ERBF | Elliptical Radial Basis Function |
| ERBFN | Elliptical Radial Basis Function Neural Network |
| FLOP | Floating-point Operations Carried Out Per Second |
| ICCK | Intrinsic Collocated Cokriging |
| LMC | Linear Model of Coregionalization |
| ML | Machine Learning |
| OICCC | Ordinary Intrinsic Collocated Cokriging |
| OK | Ordinary Kriging |
| RBF | Radial Basis Function |
| RBFN | Radial Basis Function Neural Network |
| RV | Random Variables |
| RELU | Rectified Linear |
| ReV | Regionalized Variables |
| RMSE | Root Mean Squared Error |
| RMSp | Root Mean Squared prop |
| SK | Simple Kriging |
| SRF | Stationary Random Functions |
| UCS | Ultimate Compression Strength |

CHAPTER 1

INTRODUCTION

1.1 Problem Setting and Background

With the recent increase in available computation power and increases in available data, machine learning (ML) has become an area of significant research interest. It is estimated that from 1993 until 2017, the total number of floating-point operations carried out per second (FLOPS) on a supercomputer has increased from 124 billion to 93,000 trillion: almost one million times greater (Roser & Ritchie, 2019). Along with an increase in computational power, there has been an increase in available data storage. Hard drive capacity has increased exponentially since 1980 with massive hard drives having megabytes of memory to significantly smaller hard drives with terabytes of memory (Roser & Ritchie, 2019). Not only has memory and computation power increased, but the cost per computation has decreased significantly. In 1990 a computer that cost \$1000 was capable of making 10^{-5} calculations per second, in 2010 a computer that cost \$1000 was capable of making 10^{10} calculations per second (Roser & Ritchie, 2019). Similar computational power increase statistics can be found in William D. Nordhaus "The Progress of Computing" that discusses the computer power increase over the twentieth-century (Nordhaus, 2001).

ML can be used in many different applications from simple regression to handwritten number classification and, recently, autonomous driving. ML algorithms provide an exciting opportunity to simplify and automate a process that usually would be tasked to humans (Musumeci et al., 2019). Having a machine perform these tasks will help reduce human error and allow time to be allocated to more desirable tasks (Boutaba et al., 2018). For this thesis, ML will be used to generate geological/spatial estimates at unsampled locations while reducing the amount of human interaction required in the modeling process. Traditionally geological/spatial modeling is done using a set of techniques known as Geostatistics (C. V. Deutsch & Journel, 1992).

Geostatistics is primarily concerned with estimating/predicting geological and spatial features such as grade, thickness, porosity, or rock type that are then used by engineers and geologists to make an informed decision while evaluating mineral deposits (Rossi & Deutsch, 2016). It is worth noting that Geostatistics does not follow a classical statistical framework; it was developed to suit the purposes of spatial modeling (Matheron, 1963). Matheron used this example to demonstrate that Geostatistics does not follow a classical statistical framework:

'A given deposit is explored by drilling, it would suffice to cut the cores into 5mm pieces instead of 50cm pieces to obtain 100 times more samples, and therefore 100 times high accuracy. This, of course, is wrong.'

Matheron describes how the extra data would be considered redundant due to spatial continuity.

In Geostatistics there are many different ways to perform predictions and classifications each with their own unique advantages and disadvantages; however, for this thesis, the primary techniques that will be used for making Geostatistical predictions will be Simple Kriging (SK) and Ordinary Kriging (OK) (Rossi & Deutsch, 2016).

1.1.1 Machine Learning

ML research can be dated back to 1949 when Donald Hebb introduced the Hebbian Learning Rule, which states:

'When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B is increased.'

The Hebbian Learning Rule can be viewed as the building blocks of today's neural networks, however; the first perception network was not developed until 1958 by Frank Rosenblatt which was a supervised neural network used for simple binary classification (Wang, Raj, & Xing, 2017).

ML is typically divided into supervised learning and unsupervised learning (Dey, 2016). Both supervised, and unsupervised learning methods follow a similar three-step process of training, testing, and application, as seen in Figure 1.1 (*Machine Learning: a brief breakdown*, 2018). In supervised learning, the ML algorithm has input features x which are then used to predict the solution y , during training the solution to the hypothesis is known and used to generate/test the ML algorithm which is why the process is known as supervised learning. After the ML algorithm is trained and the hypothesis is validated using k-fold validation or cross-validation the ML algorithm can be used to predict/classify similar problems with the same input features x where the solution y is not known. Different types of supervised learning techniques include but are not limited to regression, decision trees, random forests, support vector machines, and neural networks (Dey, 2016). For the purposes of this thesis, the key supervised ML method that will be explored will be neural networks.

In unsupervised learning the ML algorithm has input features x which are then used to predict the solution y , however; while training the ML algorithm the answer to the hypothesis y is not known. The solution to the hypothesis will need to be verified by using k-fold validation, cross-validation, or by minimizing error in dimension reduction techniques. Unsupervised learning techniques include but are not limited to k-mean clustering, principal component analysis, and reinforcement learning (Usama et al., 2017). For the purposes of this thesis, the primary unsupervised ML method that will be explored will be k-means clustering. A brief example of supervised vs. unsupervised machine learning techniques can be seen in Figure 1.2.

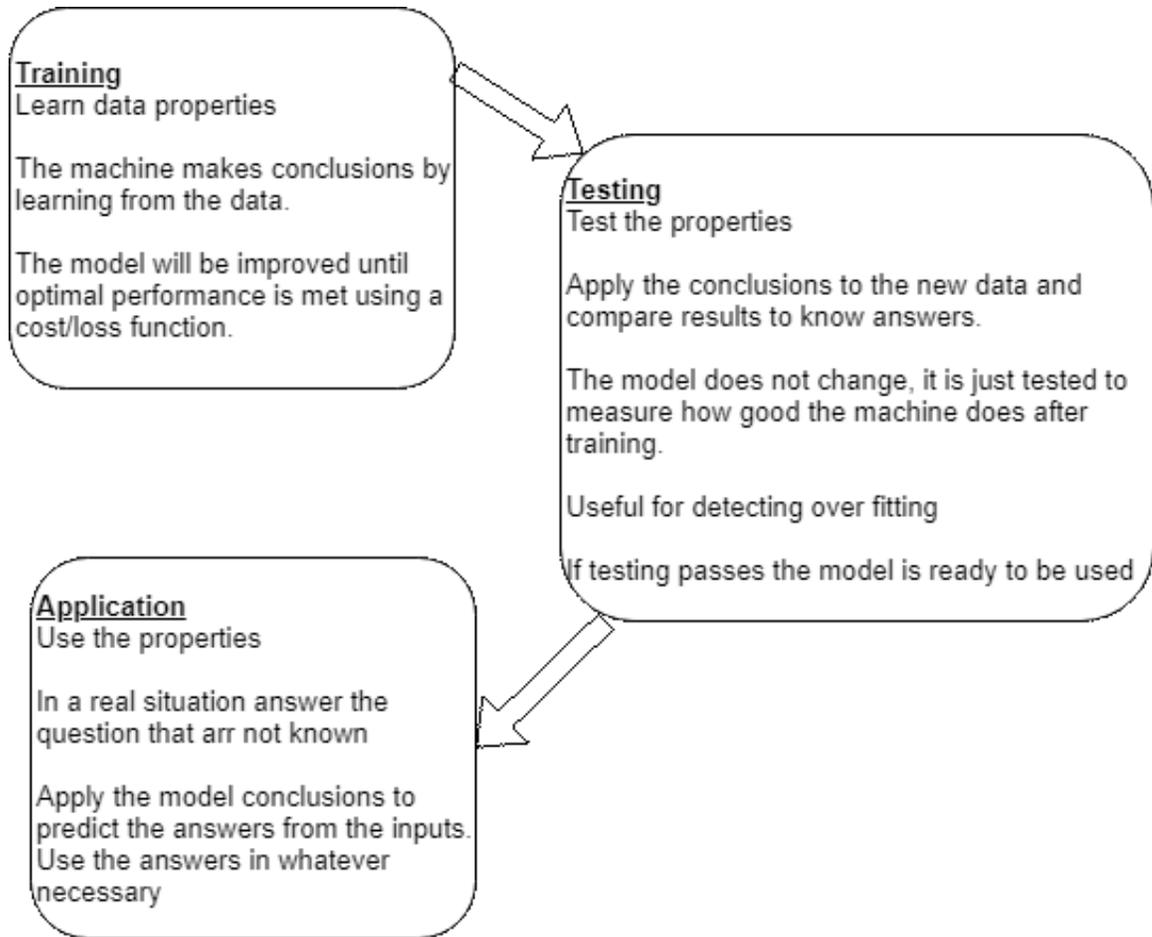


Figure 1.1: Stages in Machine Learning (Similar to *Machine Learning: a brief breakdown* (2018))

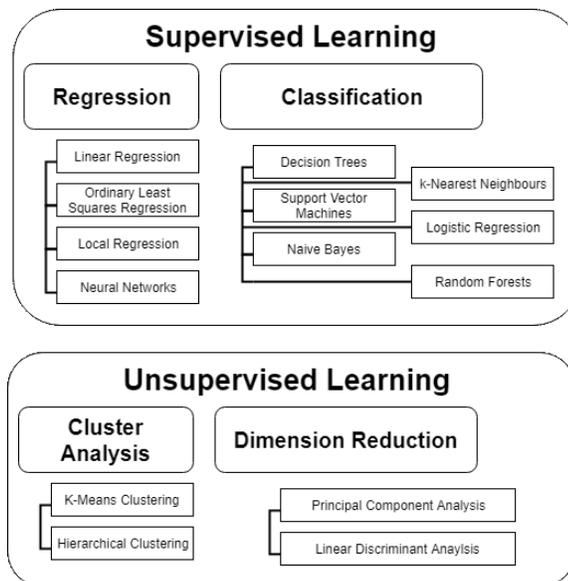


Figure 1.2: Types of Machine Learning (Similar to *Machine Learning: a brief breakdown* (2018))

1.1.2 Kriging

Kriging is a method of estimating/interpolating spatial data, and is described by Matheron as:

'It consists in estimating the grade of a panel by computing the weighted average of available samples, some being located inside others outside the panel. The grades of these samples being x_1, x_2, \dots, x_n we attempt to evaluate the unknown grade (z) of the panel with a linear estimator z^ .'*

Where the weights are often sum to one and minimize the estimation variance. Further discussion into estimation variance, unbiasedness, and the framework behind kriging will be discussed in the kriging section of Chapter 2; however, how the weights for simple kriging are calculated with an assumption of second-order stationarity is discussed below.

Second-order stationarity assumes that the variance of the random variable is constant throughout the estimation domain. The weights for kriging are calculated based on distance and variance at that distance; hence, an anisotropic model must be calculated (Boisvert, 2010). This anisotropic model is known as a variogram. Experimental variogram points are calculated, and then a variogram model is fit to those points (Ortiz & Deutsch, 2002). Variograms are typically calculated in three directions: the principal direction of the deposit, a secondary direction defining the plane of greatest continuity, and a minor perpendicular direction of anisotropy. From these three variograms, the variance in any direction and distance can be calculated and used to determine the kriging weights (Gringarten & Deutsch, 1999). Criteria must be established to determine if the estimate is acceptable and which method of estimation is best. In the next section, an overview of estimation criteria is presented.

1.1.3 Estimation Criteria and Validation

Generally, the truth is not known, which can make it difficult to perform a true comparison/validation of results. Example deposits could be simulated and then sampled so the estimations can be compared to the truth. A data spacing and data configuration study could be performed.

When working with real-world problems, the truth is unknown, and other estimation validation techniques are implemented. Validation methods include K-fold validation, appropriate statistic reproduction, and visual inspection for errors and artifacts. K-fold validation is a validation method divides the data into a training/estimation dataset and a validation dataset (Reich & Barai, 1999). For example, in 5-fold validation, the data would be divided 5 folds consisting of 20% of the data each. The full estimation process would take place five different times with the 80% training/estimation datasets, and then the results would be compared to the 20% left out. The R-Squared (R^2) statistic measures the explained variance from the dependent variable to the total variance in the predicted variable (FuquaSchoolofBusiness, 2019). The root mean squared error (RMSE) could also be used as a comparison of the difference between the estimated values and the validation values. The RMSE is in the units of the data. The mean of the estimation should be compared to the de-clustered mean

of the sample data and be similar. Histogram reproduction can also be assessed. Ideally, the known data histogram should be similar to the estimation histogram.

1.2 Thesis Statement

In order to address issues connected with assuming first and second-order stationarity while making an estimation with spatial data and human error introduced in the kriging workflow a machine learning workflow will be proposed in this thesis that does not require the assumption stationarity. The algorithm will minimize human interaction, perhaps minimizing error and bias. The thesis statement:

An integrated machine learning and geostatistical framework for modelling spatial data that more effectively takes into account complex anisotropy while minimizing human interaction required in the modelling process which will improve mining resource estimates.

An integrated ML and Geostatistical approach consists of two ML techniques with a geostatistical finisher, the first ML method is k-mean clustering, the second an elliptical radial basis neural network (ERBFN), and the geostatistical finisher is intrinsic collocated cokriging (OICCK). The ERBFN will optimize multiple parameters using a form of gradient descent in order to minimize the error between the known data locations and the estimation while keeping the estimation geologically sound and free of artifacts. The ML algorithm will not require variograms to be modeled or kriging parameters to be determined and instead will simply require the X,Y,Z data location of the known data as input features x and the known data sample y to compare the estimate with in order to optimize the machine learning parameters. The output of the ML algorithm will result in an estimation that is very similar to kriging in terms of RMSE, R^2 , histogram reproduction, stational reproduction, and will be free of estimation artifacts. The ML estimate more effectively reproduces complex anisotropy and is used as a super secondary variable in OICCK, enforcing the complex anisotropy in the geostatistical prediction. The hybrid algorithm outperforms kriging in complex scenarios. The estimation results from the ML algorithm and hybrid ML+OICCK will be compared to the kriging estimations results using K-fold validation.

1.3 Thesis Outline

Chapter Two establishes the framework of estimation for both ML and Geostatistics that are prerequisites for the modeling process proposed in this thesis. Chapter Two discusses relevant equations and concepts. Chapter Three compares Geostatistical estimates to the ML estimates and hybrid estimates with simulated models. Chapter Four shows a case study comparing the Geostatistical estimation to ML estimation with k-fold validation. Finally, Chapter Five concludes and summarizes the findings and discusses future work and limitations of the ML algorithm. The code used

to develop the machine learning estimation is developed in python using the TensorFlow machine learning library (*TensorFlow*, n.d.).

CHAPTER 2

GEOSTATISTICAL AND MACHINE LEARNING FRAMEWORK

In Chapter 1, Machine Learning (ML) and Kriging were broadly defined for estimation based on input features and weights. In kriging the estimates are calculated based on the input features denoted as x_i , which are the grades at the sampled locations then multiplied by the weights w_i . In ML the input features denoted as x_i are used to generate the estimate, which can be any relevant data at the sampled location. The estimation techniques explored in this thesis use X,Y,Z coordinates as the primary input features. The ML weights w_i are determined by a form of optimization using the known values y_s . Kriging weights are determined based on a variogram model and on the X,Y,Z data locations relative to the estimation location. The data are multiplied by the weights to calculate an estimate. In ML, the weights are determined by known values at X,Y,Z locations, and then combined for an estimate; which is very different from kriging.

This Chapter focuses on the framework behind estimation in both geostatistics and ML. Simple regression, neural networks, activation functions, training/optimization, and k-means are the primary discussion points for machine learning. Stationary/domain selection, variography, simple kriging, ordinary kriging, intrinsic collocated cokriging, and ordinary intrinsic collocated cokriging are the primary discussion points for geostatistics. Different methods for model-checking are presented at the end.

2.1 Machine Learning Framework

Machine learning was theorized in 1949 with the first perception network being built in 1958 for binary classification; Wang et al. (2017). Since 1958 there have been many advancements in machine learning; algorithms have been developed to do everything from binary classification to text/image recognition. There are many methods to perform these tasks. The primary focus is on regression and classification in supervised learning and cluster analysis in unsupervised learning. The first step in determining the ML algorithm is to determine the hypothesis or objective of the problem and then to determine the available data.

2.1.1 Choosing a Machine Learning Algorithm

A minimum of 50 data are required for a straightforward classification or regression problem. The performance of the algorithm generally scales with the amount of useful data (Pedregosa et al., 2011). The category the ML problem falls under should be determined, see Chapter 1 Figure 1.2 where

four main types of machine learning are introduced: regression, classification, cluster analysis, and dimension reductions. One must consider the available data, that is, the available features x_s and the response variable being predicted y_s . If there is no known response data, an unsupervised technique is required. Prediction with supervised neural networks requires a subset of the data to have a known solution to train the network before making a prediction. Hyperparameters are tuned based on the data available and the ML algorithm chosen. Scikit-learn, a machine learning package for python, has come out with a useful algorithm cheat-sheet that can be seen below in Figure 2.1 (Pedregosa et al., 2011). Microsoft Azure's machine learning studio has a very similar workflow, methodology, and cheat-sheet to the Scikit-learn method (Xiaoharper, n.d.). For a neural network, the hyperparameters would include the number of hidden layers, nodes in a hidden layer, learning rate, and activation functions for the hidden layers. K-means hyperparameters would include the initial cluster center, number of iterations, number of clusters to inspect, and method of error calculation. Determining these hyperparameters is explored below.

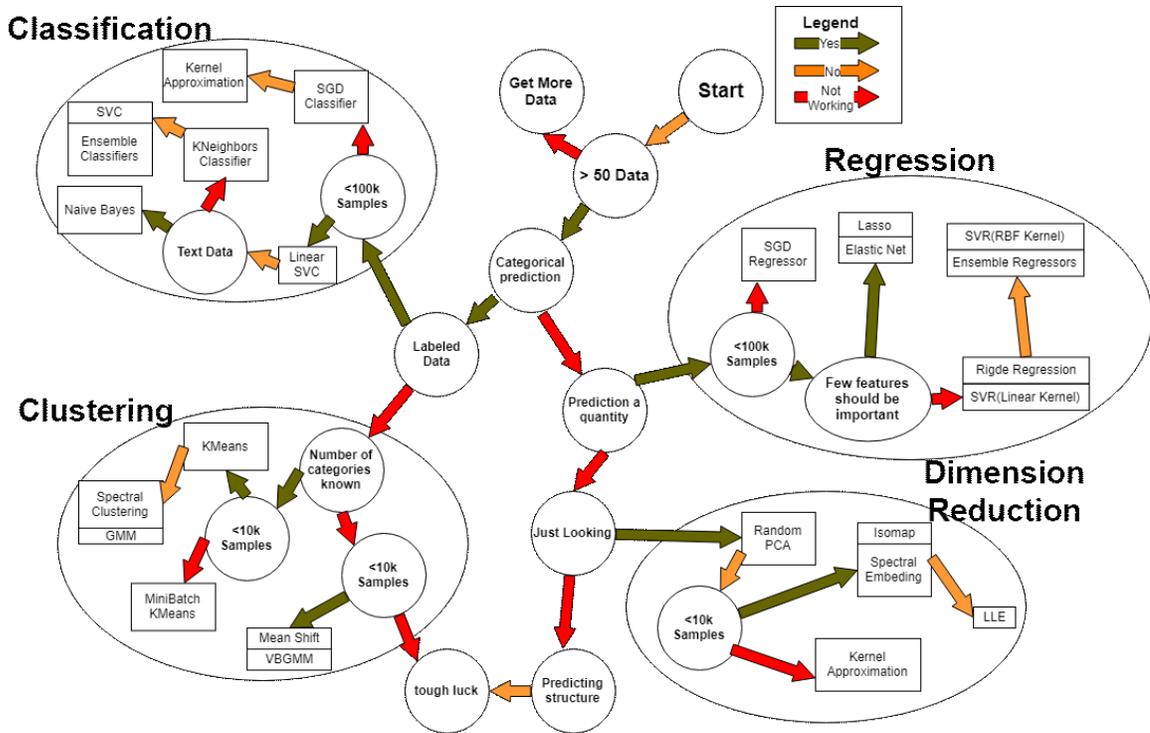


Figure 2.1: Scikit Learn Algorithm Cheat-sheet Similar to Pedregosa et al. (2011)

2.1.2 Regression

There are many types of regression including linear and logistic. Regression is a form of supervised learning that aims to predict a dependent variable based on the available data. A function is fit to the data that minimizes the distance between the function and all the points (Stulp & Sigaud, 2015). Figure 2.3 is an example of linear regression in 2D and 3D using ultimate compression strength (UCS) and quartz content in the 2D example and including hornblende into the 3D example. Equation 2.1 is the regression algorithm with y^* being the estimate. When k is equal to one, it is considered linear regression when k is greater than one; it would be considered polynomial regression. Figure 2.2 is a simple visualization of a regression network.

$$y^*(x) = w_0 + w_1x_1^i + w_2x_1^i + \dots + w_nx_n^i \quad (2.1)$$

$x_1 \dots x_n$ are the features and $w_1 \dots w_n$ are the weights. The weights and features are trained by the

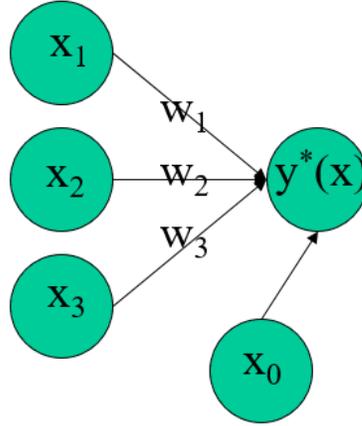


Figure 2.2: Simple Regression Network

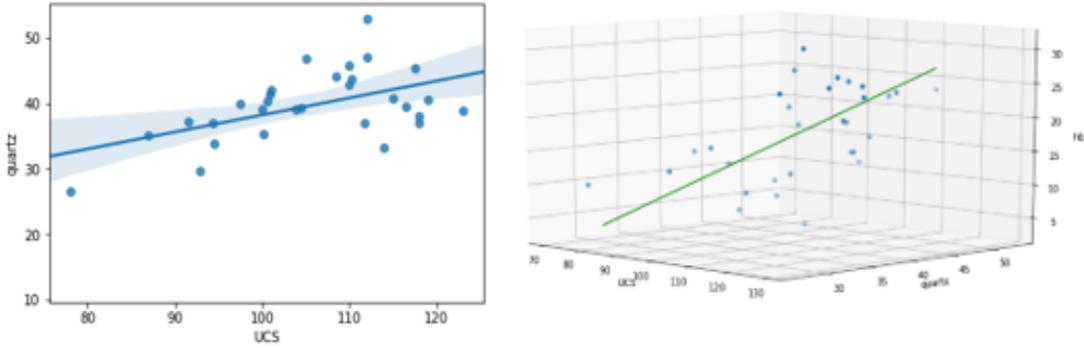


Figure 2.3: 2D/3D Linear Regression

learning algorithm. For unbiased estimation, the w_0 term is added (Stulp & Sigaud, 2015).

To determine the weights $w_1 \dots w_n$ the error for learning step must be calculated, and then the weights must be adjusted. The known truths are compared to the predicted outputs of the ML algorithm to calculate, J , the error of the algorithm (Equation 2.2). In the error equation, m denotes the number of training examples, and i denotes the training example.

$$J(w) = \frac{1}{2m} \sum_{i=1}^m (y^*(x^i) - y^i)^2 \quad (2.2)$$

Once the error has been calculated for the current learning step the weights must be modified for each feature this done with Equation 2.3.

$$w_{j+1} = w_j - \alpha \left(\frac{1}{2m} \sum_{i=1}^m (y^*(x^i) - y^i)^2 \right) x_j^i \quad (2.3)$$

In the weight modification equation w_j is the j^{th} parameter for $j = 1, \dots, n$ from the regression equation, α is the learning rate and controls how fast the weights are able to change, typically a learning rate much less than one is used; however, recently adaptive learning rates have been introduced in to machine learning algorithms (Dorffner, 2001). Using Equations 2.2 and 2.3 the regression algo-

rithm is trained until the desired fit has been reached. Note that the weights for simple linear or polynomial regression can be directly calculated and need not be estimated iteratively.

Similar to polynomial/linear regression, logistic regression is a classification tool; however, the error and regression equations are slightly different. Logistic regression takes advantage of the sigmoid function $g(x)$, Equation 2.4, with e being Euler's number, and a visual representation of the sigmoid function in Figure 2.4

$$g(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

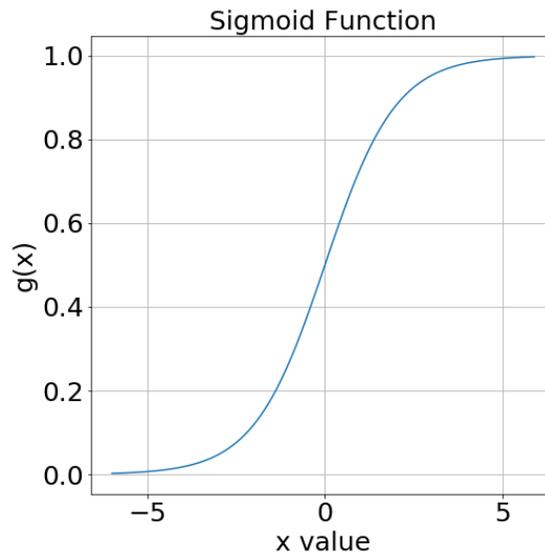


Figure 2.4: Sigmoid Function

The logistic regression formula is:

$$y^* = \frac{1}{1 + e^{-w^T x}} \quad (2.5)$$

With w_j^T denoting the transpose of the weights in vectorized form. In this form of logistic regression all values will fall between zero and one, hence a threshold will have to be set. This threshold T will vary depending on the problem.

$$y^*(x) = \begin{cases} 1, & x < T \\ 0, & x \geq T \end{cases} \quad (2.6)$$

Error is calculated as follows:

$$J(w) = \frac{1}{-m} \sum_{i=1}^m y^i \log(y^*(x^i)) + (1 - y^i) \log(1 - y^*(x^i)) \quad (2.7)$$

The weight modification formula is similar to Equation 2.3. A more general weight modification equation is shown in the optimization equation subsection to follow. Logistic regression is typically

used for classifying things like spam email, credit card transactions, and medical diagnosis (Ng & Katanforoosh, 2018). A method of evaluating the error in a regression model and how to use that error in determining the weights has been described, the next step is to look at the training.

2.1.3 Training

After the architecture of the machine learning algorithm has been determined, the next step is to train the weights of the algorithm. Typically the data are split into training, testing, and a development set. The training and testing set have known solutions, and the development set will not. A common test/train split would be a test set consisting of 20% of the known data, and the training set consisting of 80% of the known data (Raschka, 2018). As suggested by the naming convention, the training data set trains the ML algorithm. The test data set is used to ensure that the ML algorithm is not over-fitting the training data.

While training the ML algorithm, the error for both the training data set and testing data set are continuously calculated using Equation 2.2. Ideally, the error of the training dataset decreases at the same rate as the testing data set. If the error of the training dataset and the error of the testing data set are significantly different, this would indicate over-training. When the training data and testing data error look similar, and then the test data error begins to deviate from the training data, this would indicate an early stop at the point of separation is required. In Figure 2.5, an example of good training, bad training, and early stop are available.

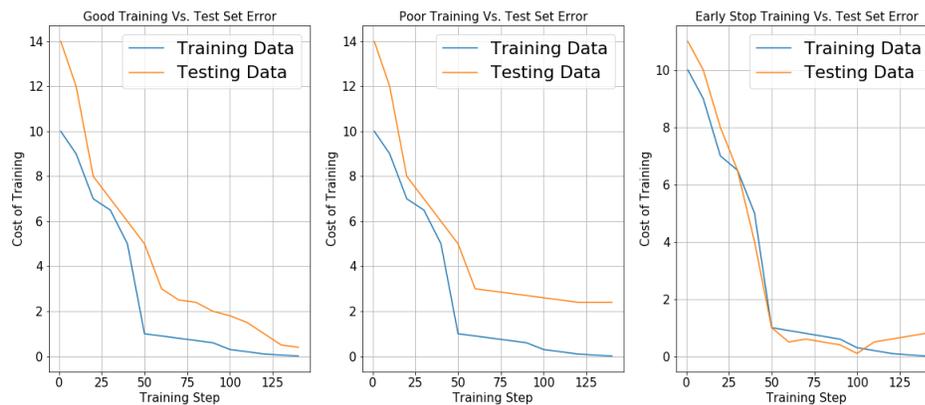


Figure 2.5: Training vs Test Cost Examples

The training cost can also be used to help tune the learning rate for the ML algorithm. Below in Figure 2.6, three examples can be seen. On the left would be an example of the correct learning rate with good training; this can be seen by the initial training cost-reducing quickly then slowing down as it approaches the minimum error. In the middle would be an example of a learning rate that is too large. Finally, on the right is an example of a learning rate that is too small as the initial cost decreases relatively slowly compared to the other example.

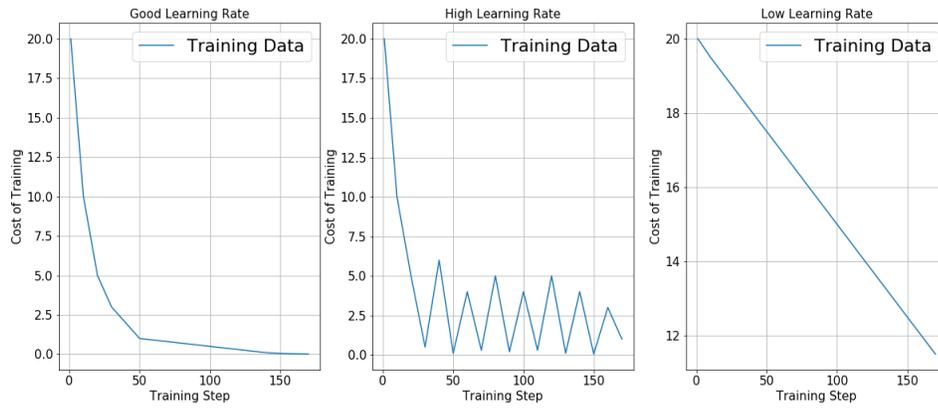


Figure 2.6: Training Cost Examples for Determining Learning Rate

Along with looking at the error/cost of the train/testing data set for a polynomial regression problem, at each training iteration, the coefficient of determination or R^2 value is calculated and used as an indicator for training performance. Equation 2.8 shows the R^2 metric and an example of how the desired training cost vs. R^2 value is shown in Figure 2.7. Ensuring that both the R^2 value is high and the cost of training is low is essential to a good estimation. The cost of training scales with the data values, whereas R^2 does not.

$$R^2 = 1 - \frac{\sum (y_i - y_i^*)^2}{\sum (y_i - \bar{y})^2} \quad (2.8)$$

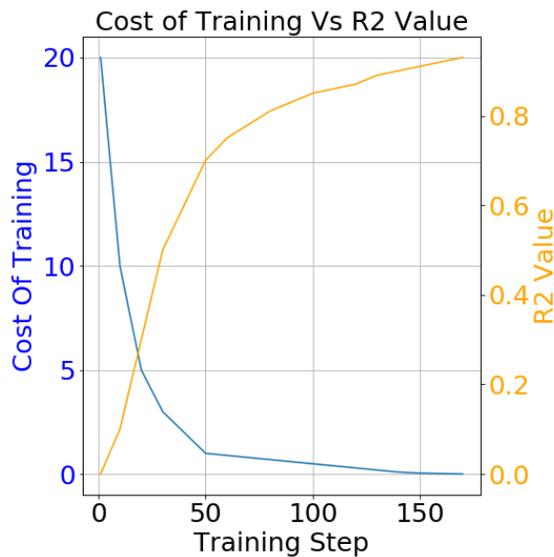


Figure 2.7: Training Cost Vs R^2

Neural networks are similar to regression. The critical difference is that a neural network has

at least one hidden layer. Regression only has an output layer and an input layer. The second significant difference is the use of activation functions on the hidden layers.

2.1.4 Neural Networks

Figure 2.8 shows an example of a simple neural network, the architecture can be compared to the regression architecture in Figure 2.2. The input layer is the first layer in the simple neural network, the middle layers are the hidden layers, and the final layer is the output layer. A set of weights connects each of the layers, and typically hidden layers have a form of activation function.

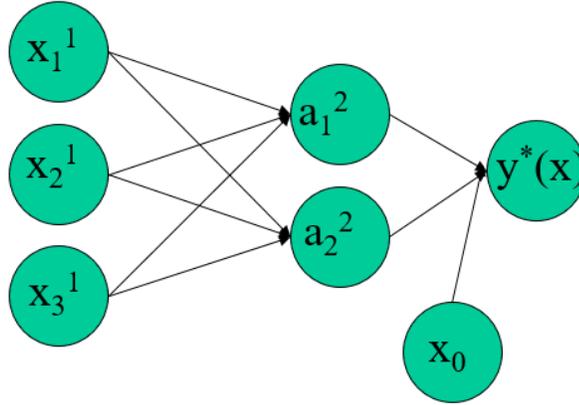


Figure 2.8: Simple Neural Network

The equations for solving for the optimal solution for the network in Figure 2.8 are below in 2.9 and 2.10. The equation for the hidden layer nodes is:

$$\begin{aligned} a_1^2 &= g(x_1^1 w_1^{11}) + g(x_1^1 w_2^{11}) + g(x_3^1 w_3^{11}) \\ a_2^2 &= g(x_1^1 w_1^{12}) + g(x_2^1 w_2^{12}) + g(x_3^1 w_3^{12}) \end{aligned} \quad (2.9)$$

with $g()$ being the activation function. The following equation is for the output of the neural network:

$$y^*(x) = g(a_1^2) + g(a_2^2) + x_0 \quad (2.10)$$

From the equations for a simple neural network with only one hidden layer, it is easy to imagine how complex a network could get with multiple layers, more nodes, or different activation functions on different layers.

Activation functions are used to introduce non-linearities into a neural network (Manavazhahan, 2017). There are many different types of activation functions, each useful in different scenarios. A few examples of activation functions include to Sigmoid, Soft-max, Tanh, Gaussian, and Rectified Linear. Soft-max and sigmoid are similar and used for classification problems. The sigmoid equation is defined in Equation 2.4 (Farnoush, 2017). When attempting to classify multiple classes, using the soft-max activation function on the output layer results in an output with the probability of each

class. The sum of a soft-max activation function is one (Equation 2.11).

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n_{classes}} e^{x_j}} \quad (2.11)$$

The Tanh is a shift version of the sigmoid function and output values from a Tanh function can range from negative one to positive one. Typically Tanh is used in hidden layers (Farnoush, 2017). The Tanh activation function is illustrated below in Equation 2.12 and Figure 2.9.

$$\tanh(x_i) = 2 * \frac{1}{1 + e^{-2x_i}} - 1 \quad (2.12)$$

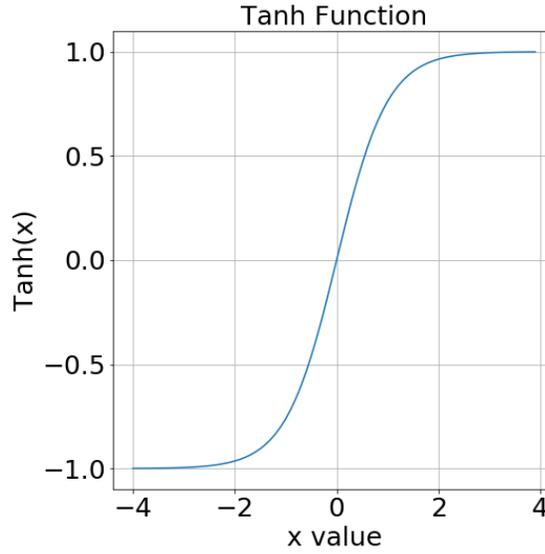


Figure 2.9: Tanh Function

A Gaussian activation function is used in the hidden layer when attempting to keep estimations smooth. The value of the Gaussian is also based on the input data range and can be either positive or negative (Manavazhahan, 2017). Equation 2.13 is the equation, where cc is the center of the Gaussian function, and r is the radius of the Gaussian function. Figure 2.10 is a graph of Gaussian functions with different radii and a center of zero.

$$Gaussian(x_i) = e^{-(r*(cc-x_i))^2} \quad (2.13)$$

In higher dimensional problems typical Euclidean distance is used; however, in more complex situation when the covariance should be considered using a Mahalanobis distance is beneficial. Equation 2.14 is the Euclidean distance in 3D.

$$d_e(x, y, z) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (2.14)$$

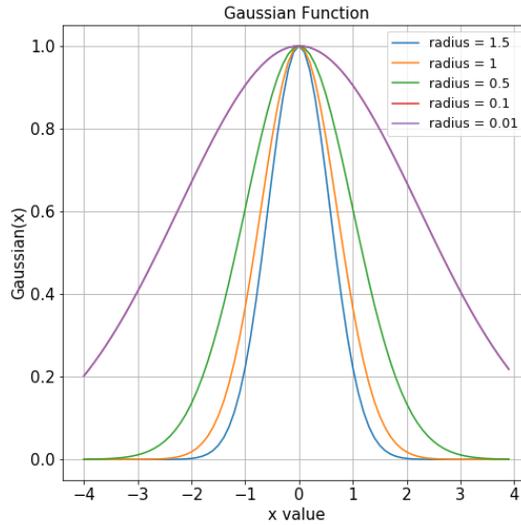


Figure 2.10: Gaussian Function

Equation 2.15 is the Mahalanobis distance between two random vectors.

$$d_m(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T S^{-1} (\mathbf{x} - \mathbf{y})} \quad (2.15)$$

Where $(\mathbf{x} - \mathbf{y})$ is the difference between the estimation location and the cluster center, $(\mathbf{x} - \mathbf{y})^T$ would be the transpose of the difference, and S^{-1} is the inverse of the covariance matrix between the cluster center and the estimation location (Todeschini, Ballabio, Consonni, Sahigara, & Filzmoser, 2013). For n dimensions space $(\mathbf{x} - \mathbf{y})$ is represented with Δ .

The rectified linear (RELU) activation function is a piecewise linear function that stops at zero and goes to infinity (Manavazhahan, 2017). Equation 2.16 is the RELU activation function.

$$RELU(x) = \begin{cases} 0, & x < 0 \\ mx, & x \geq 0 \end{cases} \quad (2.16)$$

RELU is typically used as a hidden layer activation function. Figure 2.11 is a graphical representation of the RELU activation function.

Different activation functions can be used in different layers. It is generally best to start simple when designing a neural network and gradually get more complex by increasing the number of hidden layers and the number of nodes in the hidden layers. Activation functions are chosen based on the problem. Starting simple and becoming more complex also helps keep the machine learning algorithm more computationally efficient. The degrees of freedom can be viewed as the number of nodes and layers, having too many nodes and layers can often lead to over-fitting the training data or make optimization very slow.

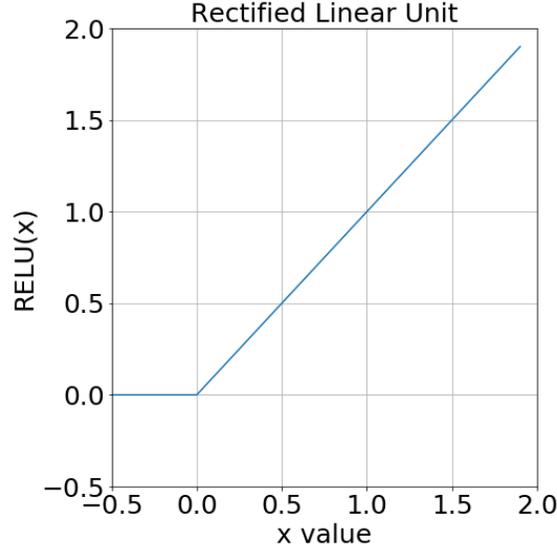


Figure 2.11: RELU Function

2.1.5 Optimization Equations

Equation 2.3 gradient descent iterations:

$$w_{j+1} = w_j - \alpha \frac{\delta}{\delta w_j} J(w) \quad (2.17)$$

Where $\frac{\delta}{\delta w_j} J(w)$ is the gradient of the cost with respect to the current weight.

Other optimization methods include: momentum-based gradient descent, adaptive gradient optimization, root mean squared prop (RMSprop), and the adaptive moment estimation optimization (Adam) (Wang et al., 2017).

Momentum based on gradient descent proposed by Boris T. Polyak is a method of speeding up the optimization process by not only depending on the current gradient but by introducing a velocity term (Polyak, 1964). The velocity term is the uncentred variance of the gradient. The equation for momentum-based gradient descent is:

$$w_{j+1} = w_j - \alpha V_j \quad (2.18)$$

where V_j is the velocity term:

$$V_j = \beta V_{j-1} + (1 - \beta) \frac{\delta}{\delta w_j} J(w) \quad (2.19)$$

V is initially set to zero and β which controls the rate of momentum change is generally set to 0.9; however, this value should be adjusted based on the problem being solved.

RMSprop uses an exponential moving average to help determine how the weights are modified, which speeds up the learning process (Ruder, 2017). Equation 2.20 is RMSprop:

$$w_{j+1} = w_j - \frac{\alpha}{\sqrt{S_j + \epsilon}} \frac{\delta}{\delta w_j} J(w) \quad (2.20)$$

With S_j being the exponential moving average:

$$S_j = \beta S_{t-1} + (1 - \beta) \left(\frac{\delta}{\delta w_j} J(w) \right)^2 \quad (2.21)$$

In this method S_j is initialized to zero with β typically being set to 0.9 and ϵ being set to a very small number. The Adam optimizer is a combination of the previous two optimization techniques using both a velocity term and an exponential moving average value. Equation 2.22 is the Adam optimizer:

$$w_{j+1} = w_j - \frac{\alpha}{\sqrt{\hat{S}_j + \epsilon}} \hat{V}_j \quad (2.22)$$

The \hat{S}_j and \hat{V}_j are introduced to help counteract biases introduced when b_1 and b_2 are close to one (Ruder, 2017):

$$\hat{S}_j = \frac{S_j}{1 - \beta_1^j} \quad (2.23)$$

$$\hat{V}_j = \frac{V_j}{1 - \beta_1^j} \quad (2.24)$$

S_j and V_j are then similar to as seen above:

$$S_j = \beta_1 S_{t-1} + (1 - \beta_1) \left(\frac{\delta}{\delta w_j} J(w) \right)^2 \quad (2.25)$$

$$V_j = \beta_2 V_{j-1} + (1 - \beta_2) \frac{\delta}{\delta w_j} J(w) \quad (2.26)$$

The Adam optimization equation is a combination of the RMSprop and the momentum-based gradient descent equation. S and V are initialized at zero, β_2 is generally set to 0.9, and ϵ is set to a very small number, the only difference being β_1 is generally set to 0.99. Again, these hyperparameters should be set to values that best suit the given problem. The optimization equations presented are for supervised learning, K-means is an unsupervised learning technique and is discussed in the following section.

2.1.6 K-Means

K-means is an unsupervised ML technique that separates samples into n clusters C with equal variance while minimizing the inertia (Pedregosa et al., 2011). K-means divides the samples into n centers μ , which are also the mean of the data. The centers are not necessarily from the data samples; however, the data samples and the centroid must exist within the same space. The inertia (I) of a cluster is:

$$I = \sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (2.27)$$

where $\|x_i - \mu_j\|$ is the Euclidean distance between the point and the centre the point is assigned to. The next step after calculating the inertia of each cluster is to adjust the cluster centres and potentially reassign the data to a new cluster centre location. Generally cluster centres are randomly

initialized, the cluster centres are then adjusted using:

$$\mu_j = \frac{1}{C_j} \sum_{i=1}^{C_i} x_i \quad (2.28)$$

where C_i is the i^{th} cluster center. These two steps are repeated until the minimum potential inertia between all clusters is reached. Figure 2.12 is an example of k-means clustering (Maklin, 2019). Note that as the number of centroids approaches the number of data, the total inertia of the system

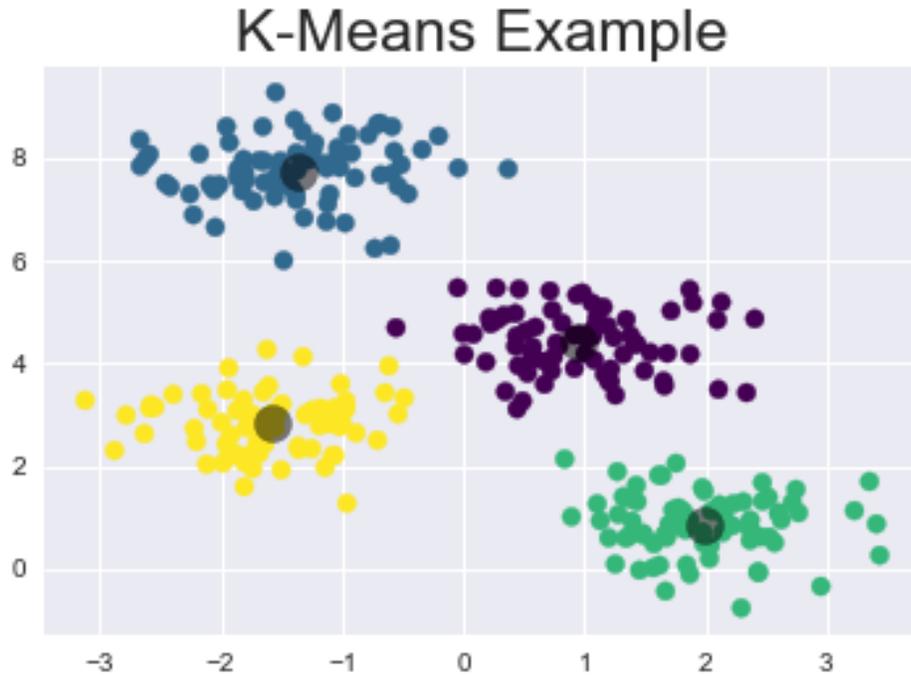


Figure 2.12: K-means Example (Similar to Maklin (2019))

decreases (Figure 2.13). When the number of nodes is equal to the number of data, the inertia is zero; however, choosing a large number of nodes is computationally inefficient. The elbow method is a common method used in determining the optimal number of nodes. A simple example is shown in Figure 2.13 where the decreasing error ($y - axis$), of a K-means system as the number of centroids increases ($x - axis$), for this example the elbow would be around 20 centroids. After this point, information gained from adding centroids is not substantial, and the information loss is not significant (A Syakur, K Khotimah, M S Rochman, & Dwi Satoto, 2018).

2.2 Kriging Framework

Simple kriging in geostatistics is the best linear unbiased estimation (BLUE). In this section, the framework behind ordinary kriging and simple kriging are explored and explained. The first step is to determine the estimation domain where the assumptions of first and second-order stationarity

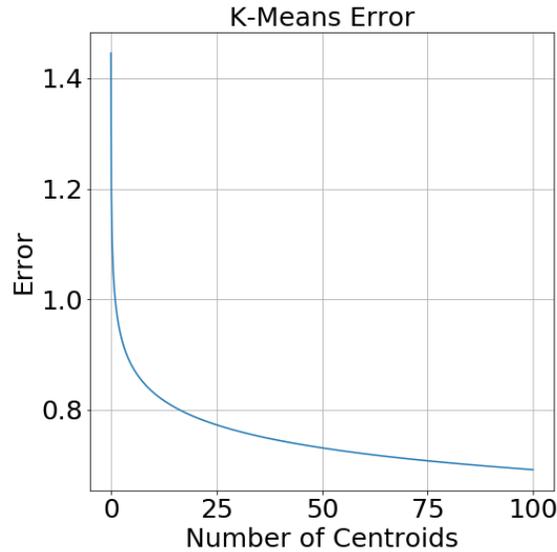


Figure 2.13: K-Means Error Example

are valid. Then, an experimental variogram is calculated and fit to provide the variance/covariance values for the kriging estimation.

2.2.1 Stationarity

Stationarity and domain decisions are required for all statistical analyses and are one of the most crucial steps in any statistical analysis. A bad decision of stationarity can result in poor results (McLennan, 2007). There is no one correct answer when it comes to deciding stationarity. The decision is subjective, which can make it difficult in some cases to determine the correct domain. In Jason McLennan Ph.D. thesis, five steps for making a reasonable decision of stationarity for geostatistical models are presented:

1. Choosing the number and types of domains for numerical petrophysical property modelling
2. Modeling the domain boundaries
3. Boundary type detection and model mixing
4. Trend modelling
5. Predicting with a trend model.

Often the number and type of domains are based on the deposit. Typically when defining the domain, it is best to start off looking at the entire deposit and then to split the deposit based on multiple smaller, more geologically homogeneous domains. Decreasing the size of the domain can help increase geological precision; however, decreasing the size of the domain results in a decrease

in the number of data samples available for making estimation and inferring the stationary random functions (SRF) more difficult (Wilde, 2011). In Jason McLennan Ph.D. it states that:

'A stationary random function (SRF) is a probabilistic representation of a petrophysical property with the constant expected value and covariance moments independent of location.'

SRF encompasses first and second-order stationarity. As mentioned above, domain selection is subjective; hence is typically done manually and deterministically. Manually digitizing domains deterministically can be time-consuming and is unable to capture uncertainty in size, shape, and orientation of the domain (Wilde, 2011). Within the defined domain, we have petrophysical properties distributed. These variables referred to as regionalized variables (ReV). Geostatistics aims to predict ReV at an unknown location u , given the location-dependent random variables (RV) (McLennan, 2007).

Within the domain, a continuous RV has an infinite number of possibilities. The cumulative density function (CDF) fully characterizes continuous random variables. A stationary random functions assumes that:

$$m(\mathbf{u}) = E\{X(\mathbf{u})\} \quad \forall \mathbf{u} \in D \quad (2.29)$$

$$C_x(\mathbf{u}, \mathbf{u}') = E\{X(\mathbf{u}) \cdot X(\mathbf{u}')\} - E\{X(\mathbf{u})\} \cdot E\{X(\mathbf{u}')\} \quad \forall \mathbf{u} \in D \quad (2.30)$$

Where $m(\mathbf{u})$ is first order stationarity assuming that the expected value for each RV pair $X(\mathbf{u})$ and $X(\mathbf{u}')$ that are distance of \mathbf{h} apart, and has the same second order covariance C_x at distance \mathbf{h} for all \mathbf{u} within the domain D , where D may be a subset of N domains $\{D = D_1, \dots, D_N\}$ and the assumptions for valid SRF must be reconsidered within each subset domain.

Boundary surface models are required when splitting geological domains into multiple domains consisting of valid SRFs. When modelling boundary surfaces, both deterministic and probabilistic techniques can be employed; however, this thesis is not focused on probabilistic methods. The deterministic method used for estimating boundary conditions is, in fact, kriging, generally with a Gaussian variogram to ensure smooth distance functions (Wilde, 2011). If possible, using global kriging for surface boundary estimation is recommended to help prevent artifacts from the modelling processes.

The final three steps in decision of stationarity are available in Jason McLennan Ph.D. thesis *"The Decision of Stationarity"*; however, it is important to discuss the first steps to make it clear that the assumption of stationarity is necessary and in complex deposits multiple kriging and boundary modelling steps might be required. The next section explores variogram modelling. It is important to note that a variogram is required to be modelled separately for each variable in each domain.

2.2.2 The Variogram

The variogram $\gamma(\mathbf{h})$ is a distance function that indicated the dissimilarity of two points separated by a distance of $\mathbf{h} = |\mathbf{u} - \mathbf{v}|$ where \mathbf{u} and \mathbf{v} are spatial locations; however, before modelling a variogram the experimental variogram must be modelled from pairs of observed data $\{x(\mathbf{u}), x(\mathbf{u} + \mathbf{h})\}$. The equation of the experimental variogram is as follows:

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2 \cdot n(\mathbf{h})} \sum_{i=1}^{n(\mathbf{h})} [x(\mathbf{u}_i) - x(\mathbf{u}_i + \mathbf{h})]^2 \quad \forall \mathbf{u}_i, \mathbf{h}, \mathbf{u}_i + \mathbf{h} \in D \quad (2.31)$$

where $n(h)$ is the total number of pairs sampled at distance h . The variogram is then modelled by fitting the experimental point with a variogram function. There are three widely used variogram functions; however, there are many more potential options (Manchuk, 2017). The three variogram types mainly used are, exponential:

$$\text{exp}(h) = 1 - \exp^{-3h/a} \quad (2.32)$$

spherical:

$$\text{Sph}(h) = 1.5(h/a) - \exp^{-3(h/a)^2} \quad (2.33)$$

and gaussian:

$$\text{Gaus}(h) = 1 - \exp^{-3(h/a)^2} \quad (2.34)$$

where a is the isotropic range of that portion of the variogram function. Often multiple variogram structures with different isotropic ranges and types are used to ensure a good variogram fit. The equation for a variogram model is:

$$\gamma(\mathbf{h}) = \sum_{i=0}^{nst} C_i \Gamma_i(\mathbf{h}) \quad (2.35)$$

where C_i is the contribution of each structure, $\Gamma_i(\mathbf{h})$ is the i^{th} variogram structure and nst is the number of structures in the variogram model. Typically the 0^{th} variogram structure is a nugget effect which accounts for out of average discontinuity at the short scale (Matheron, 1963). Figure 2.14 is an example of an experimental variogram. The variogram is then related to covariance $C(\mathbf{h})$ via the variance σ^2 :

$$C(\mathbf{h}) = \sigma^2 - \gamma(\mathbf{h}) \quad \forall \mathbf{h} \in D \quad (2.36)$$

the covariance is then used in the kriging equation for estimation as it represents the spatial variability of the RV (Qu, 2018).

2.2.3 Kriging

Every domain defined with the assumption of a valid SRF has an experimental variogram calculated. Then a variogram is modelled based on the experimental variogram points. The next step is to

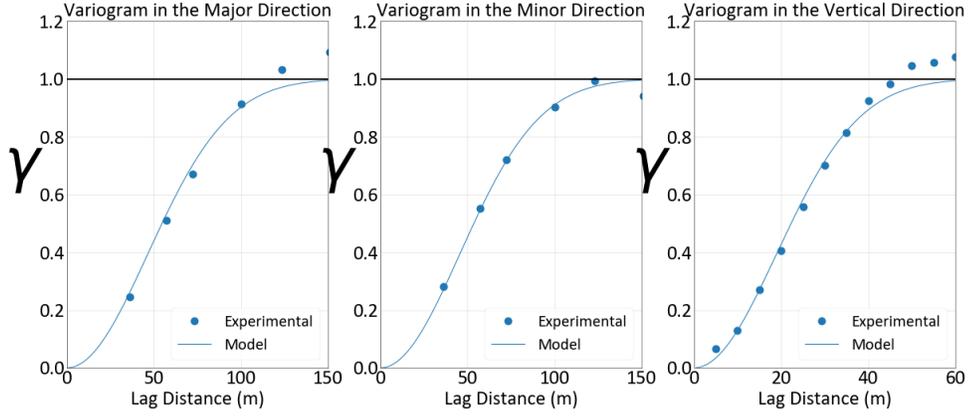


Figure 2.14: Example Experimental Variogram and Variogram Model

generate an estimate by kriging using the variogram model. As mentioned above in Chapter 1 kriging is essentially a linear estimation based on a summation of weights multiplied by data points, in simple kriging the mean of the domain is used in the prediction hence SK equation is:

$$y^*(\mathbf{u}) - m = \sum_{i=1}^n \lambda_i \cdot [x_i - m] \quad (2.37)$$

where λ_i is the weight assigned to the i^{th} known data sample x_i . The weights are calculated using the covariance between known sample locations and the covariance unknown and known sample locations. Generally, kriging with all data is not practical, so using a large number (100) is recommended (J. Deutsch & Deutsch, 2012). A more general form of the equation is:

$$\sum_{j=1}^n \lambda_j C_{i,j} = C_{i,*} \quad i = 1, \dots, n \quad (2.38)$$

with the SK estimation variance give as:

$$\sigma_{sk}^2 = \sigma^2 - \sum_{i=1}^n \lambda_i C_{i,*} \quad (2.39)$$

To move to ordinary kriging where knowing the mean is not required, the sum of the kriging weights is constrained to one and a Lagrange multiplier is added (λ_L):

$$y^*(\mathbf{u}) = \sum_{i=1}^n \lambda_i \cdot x_i \quad (2.40)$$

$$\begin{cases} \sum_{j=1}^n \lambda_j C_{i,j} + \lambda_L = C_{i,*} & i = 1, \dots, n \\ \sum_{j=1}^n \lambda_j = 1 \end{cases} \quad (2.41)$$

Intrinsic collocated cokriging (ICCK) is a multivariate estimation technique, develop by Olena Babak, that takes advantage of exhaustive collocated data (Babak & Deutsch, 2009). ICCK aims to reduce artificial variance inflation while using collocated cokriging. Traditionally the multivariate geostatistical full cokriging is used. The workflow is complex and requires variograms for each vari-

able being estimated, across variograms between the variables, and finally, a linear model of coregionalization (LMC), which represents the spatial correlations between all estimation location and all known variables (Babak & Deutsch, 2007). Building an LMC can be difficult and time-consuming, hence leading to the development of ICCK, which only requires the variogram of the primary variable and then scales the secondary variable based on the correlation between the primary and secondary data. The ICCK equation is as follows:

$$y^*(\mathbf{u}) - m_y = \lambda_{sc} \cdot [z_o - m_z] + \sum_{i=1}^n \lambda_i \cdot [x_i - m_y] + \sum_{i=1}^{sn} \lambda_{si} \cdot [z_{si} - m_z] \quad (2.42)$$

where λ_{sc} represent the weight given to the secondary value, z_o , at estimation location and z_{si}, m_z are the secondary data and mean of the secondary data. A more in-depth description of cokriging and intrinsic collocated cokriging is available in Olena Babak and Clayton V. Deutsch "An Intrinsic Model of Coregionalization that Solves Variance Inflation in Collocated Cokriging"

Ordinary intrinsic collocated cokriging (OICCK) has the same constraint of the weights summing to one similarly to OK. Generally speaking, this is not particularly useful, by using the Lagrange formalism to constrain the sum of the weights to one will result in most of the weight being assigned to the collocated data when there is no primary data within variogram range. Generally this is not desired when making a multivariate estimation. Using OICCK assigns a majority of the weight to the secondary data when outside of variogram range with a small amount of the weight being assigned to the unbiased estimation of the mean at that location. This is desired and these reason are discussed in Chapter 3, Proposed Estimation Methods. The OICCK equations are as follow:

$$y^*(\mathbf{u}) = \lambda_{sc} \cdot [z_o] + \sum_{i=1}^n \lambda_i \cdot [x_i] + \sum_{i=1}^{n_{sec}} \lambda_{si} \cdot [z_{si}] \quad (2.43)$$

$$\begin{bmatrix} C_{1,1} & C_{2,1} & \dots & C_{n,1} & C_{s1,1} & C_{s2,1} & \dots & C_{sn,1} & C_{s0,1} & 1 \\ C_{1,2} & C_{2,2} & \dots & C_{n,2} & C_{s1,2} & C_{s2,2} & \dots & C_{sn,2} & C_{s*,2} & 1 \\ \dots & & & \dots & \dots & & & \dots & \dots & \dots \\ \dots & & & \dots & \dots & & & \dots & \dots & \dots \\ \dots & & & \dots & \dots & & & \dots & \dots & \dots \\ C_{1,n} & C_{2,n} & \dots & C_{n,n} & C_{s1,n} & C_{s2,n} & \dots & C_{sn,n} & C_{s*,n} & 1 \\ C_{1,s1} & C_{2,s1} & \dots & C_{n,s1} & C_{s1,s1} & C_{s2,s1} & \dots & C_{sn,s1} & C_{s*,s1} & 1 \\ C_{1,s2} & C_{2,s2} & \dots & C_{n,s2} & C_{s1,s2} & C_{s2,s2} & \dots & C_{sn,s2} & C_{s*,s2} & 1 \\ \dots & & & \dots & \dots & & & \dots & \dots & \dots \\ \dots & & & \dots & \dots & & & \dots & \dots & \dots \\ \dots & & & \dots & \dots & & & \dots & \dots & \dots \\ C_{1,s2} & C_{2,sn} & \dots & C_{n,sn} & C_{s1,n} & C_{s1,sn} & \dots & C_{s2,sn} & C_{sn,sn} & \\ C_{1,s*} & C_{2,s*} & \dots & C_{n,s*} & C_{s1,*} & C_{s2,s*} & \dots & C_{sn,s*} & C_{s*,s*} & 1 \\ 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \dots \\ \dots \\ \lambda_n \\ \lambda_{s1} \\ \lambda_{s2} \\ \dots \\ \dots \\ \dots \\ \lambda_{sn} \\ \lambda_{sc} \\ \lambda_L \end{bmatrix} = \begin{bmatrix} C_{1,*} \\ C_{2,*} \\ \dots \\ \dots \\ \dots \\ C_{n,*} \\ C_{s1,*} \\ C_{s2,*} \\ \dots \\ \dots \\ \dots \\ C_{sn,*} \\ C_{sc,*} \\ 1 \end{bmatrix} \quad (2.44)$$

Where $C_{n,sn}$ would represent the correlation between the secondary and primary variables (s standing for secondary), $C_{n,n}$ is the correlations between primary variable, $C_{sn,sn}$ would be the correlation between the secondary variable, and finally the kriging weights λ and λ_L is the Lagrange multiplier weight.

2.3 Estimation Criteria

A geomodeller must set up a method to validate estimations. K-fold validation is a common method used to validate estimation techniques where all relevant statistics are calculated based on each fold. K-fold validation splits the data set into k . Each set will have $\frac{100}{k}\%$ of the data. The estimation is run k times, withholding one fold of the data for testing and using the rest of the folds for training. After all folds have run, relevant statistics are calculated based on the training and test folds (Raschka, 2018). Figure 2.15 is a sketch of how the data would be split up in k-fold validation.

| % Data | 20% | 20% | 20% | 20% | 20% |
|--------|-------|-------|-------|-------|-------|
| Fold 1 | Test | Train | Train | Train | Train |
| Fold 2 | Train | Test | Train | Train | Train |
| Fold 3 | Train | Train | Test | Train | Train |
| Fold 4 | Train | Train | Train | Test | Train |
| Fold 5 | Train | Train | Train | Train | Test |

Figure 2.15: K-Fold Data Setup Example

Each estimation method is compared based on the R^2 values, and the RMSE is calculated on the

test set. Equation 2.8 is the R^2 equation and the RMSE equation is as follows:

$$RMSE = \sqrt{\frac{\sum (y_i - y_i^*)^2}{n_{data}}} \quad (2.45)$$

Where y_i would be the test validation data, and y_i^* would be the estimated data. RMSE scales with the data units. The R^2 value and RMSE are considered. It is also important to check for histogram reproduction and de-clustered mean reproduction.

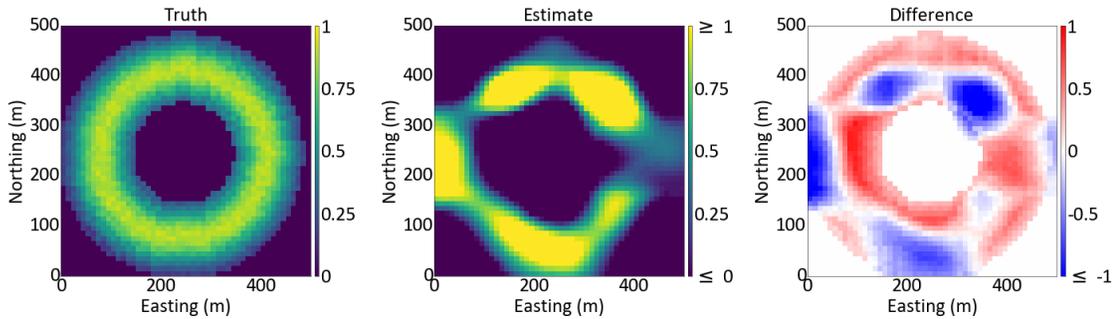


Figure 2.16: Location Map of Error

The second visual check involves inspecting slices of the estimation to ensure that the estimation is free of artifacts; such as jagged shadow-looking features that are not geological. An example of a prediction with artifacts can be seen below in Figure 2.17 on the left.

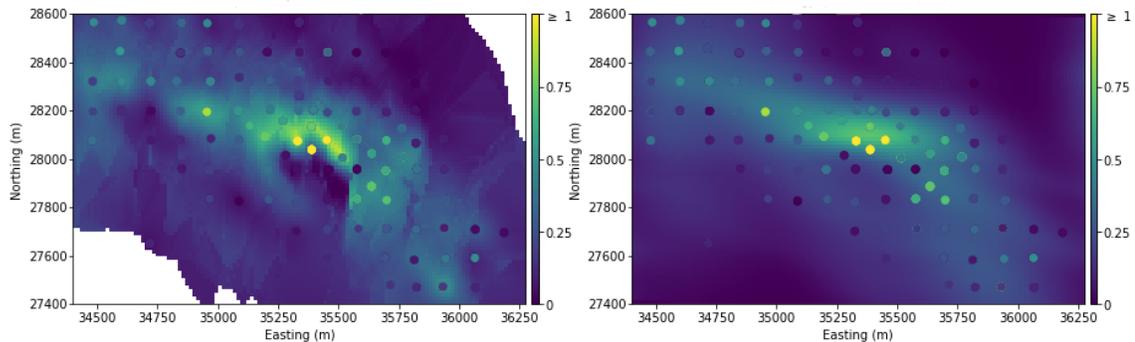


Figure 2.17: Example Estimation Slice (Left: With artifacts, Right: No artifacts)

Scenarios where the full solution to the problem is known, are best suited to compare different estimation methods; this is because the exact solution is known, and we can have a direct comparison between the different estimation methods and the truth. The K-fold estimations are used to validate the method, but also, by comparing it to the exact truth, it is possible to make a directed comparison of errors between estimation methods. Theoretically, K-fold and full model comparison results should agree.

The next chapter explores the proposed machine learning estimation technique for spatial models and a hybrid method that includes ML and geostatistics.

CHAPTER 3

PROPOSED ESTIMATION METHODS

In this chapter, two estimation methods that involve machine learning (ML) are explored. ML does not require the assumption of stationarity in estimation; however, ML is a data driven method and a form of regression; hence, data are not reproduced. Kriging, requires the assumption of stationarity, which, is not always valid; however, kriging is a model-driven estimation, and, reproduces data at their locations. The first method explored is an elliptical radial basis function neural network (ERBFN), a ML technique. The second method explored is a hybrid of ML and kriging. The second method involves making an estimation with ERBFN and then using simple or ordinary intrinsic collocated cokriging using the ERBFN results as the secondary data in the intrinsic co-kriging workflow. A hybrid estimation brings the best of both estimation methods into a single estimation. The ML approach brings the non-stationary features into the estimation, while the kriging method enforces data reproduction and unbiasedness. A neural network with a more traditional activation functions, or a deep network is shown to be less effective than an elliptical radial basis function network.

3.1 Geostatistical Prediction With a Neural Network

This section explores, estimations with neural networks and multiple types of activation functions. Estimations are made varying the size, depth, and activation functions used. The proposed network that is the primary focus of this thesis is a simple one-layer network using RBF to activate the hidden layer; however, there is still an issue with anisotropy using a simple RBF as an activation function. Hence the RBF is converted to an elliptical radial basis function (ERBF) to reproduce anisotropy more effectively. In this section, a 3D data set is simulated and drilled so that the exact solution to the problem is known. The different types of networks are tested and compared to a simple kriging model.

3.1.1 Simple Linear Neural Network

In this section, a spatial prediction is made using a simple linear neural network and compared to a kriged model. Figure 3.1 illustrates the simulated data for this study. The data set was simulated using a drill hole spacing to variogram range of 40% in the major and minor direction and a range of 3% in the vertical direction resulting in 600 data points. A fivefold validation method is used on the synthetic data set (Figure 3.1), resulting in 480 data points in each training data set and 120 data

points in each testing data set. Figure 3.2 shows the results of a simple linear neural with 25 nodes in the hidden layer, using X, Y, Z values as features.

Figure 3.3 illustrates the results from one fold. Table 3.1 is a statistical summary of the results for the simple neural network and can compare to kriging results in Table 3.2.

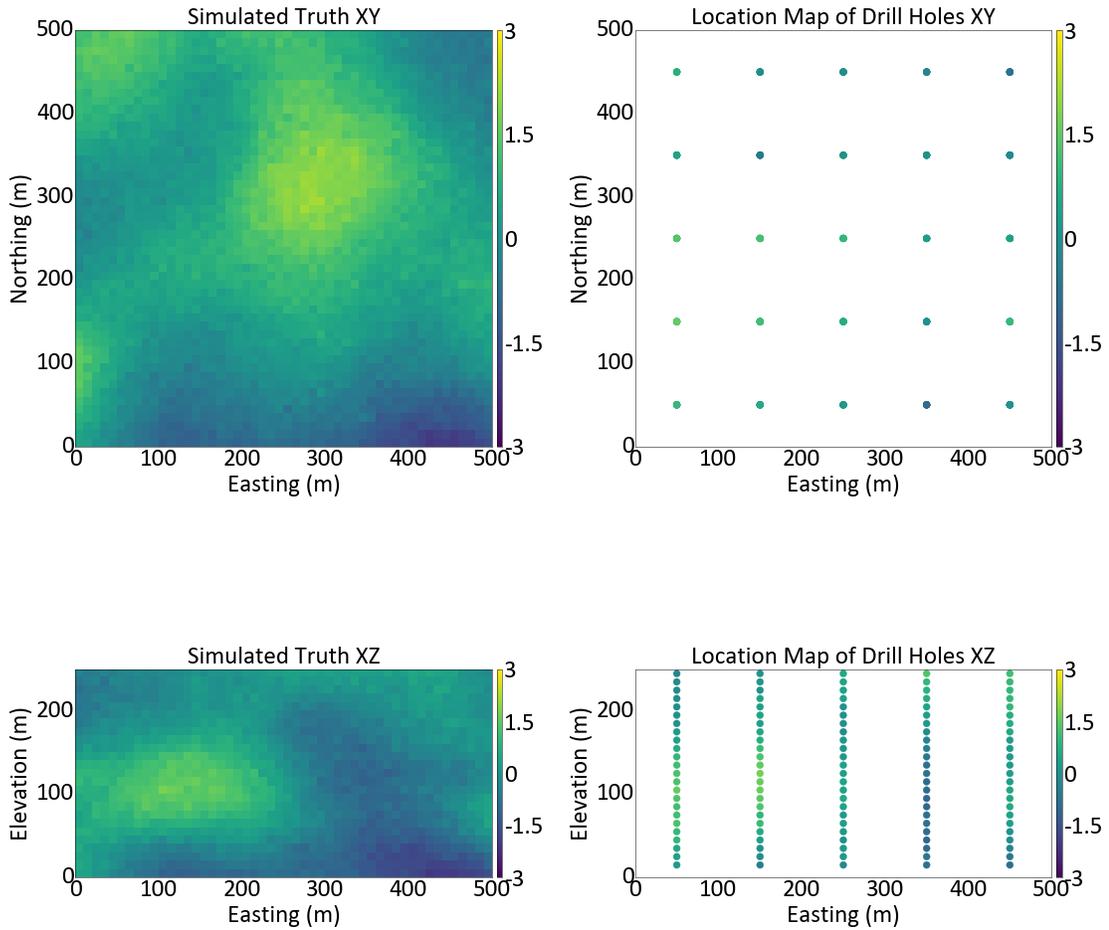


Figure 3.1: Simulated and Drilled Data

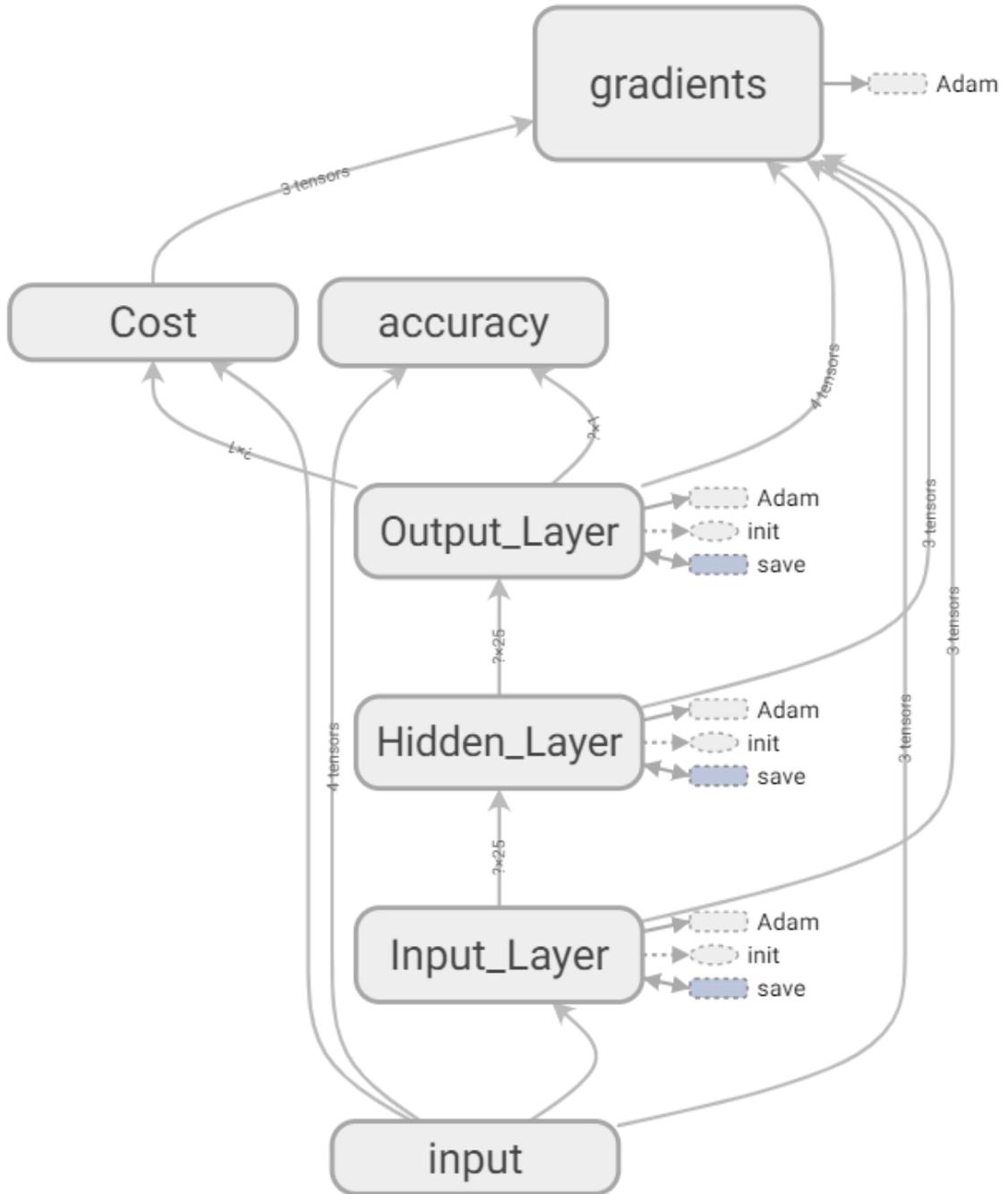


Figure 3.2: Simple One Layer Linear Neural Network (*TensorBoard: Graph Visualization | TensorFlow Core | TensorFlow, n.d.*)

3. Proposed Estimation Methods

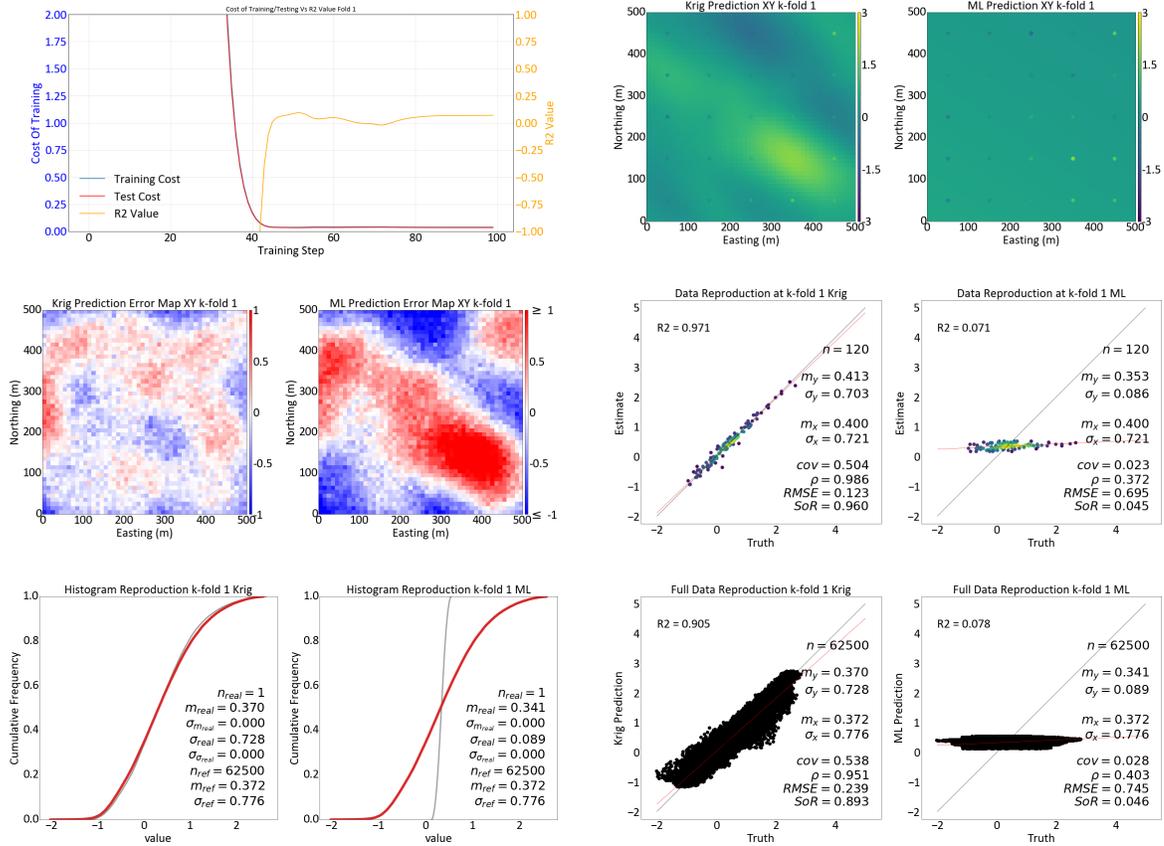


Figure 3.3: K-Fold Results From Simple Linear Neural Network

| Simple Neural Network | | | | | | |
|-----------------------|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 1 | 0.695 | 0.071 | 0.745 | 0.078 | 0.089 | 0.341 |
| 2 | 0.747 | 0.108 | 0.739 | 0.094 | 0.111 | 0.361 |
| 3 | 0.710 | 0.164 | 0.713 | 0.155 | 0.318 | 0.269 |
| 4 | 0.754 | 0.075 | 0.748 | 0.069 | 0.079 | 0.334 |
| 5 | 0.700 | 0.062 | 0.749 | 0.067 | 0.085 | 0.449 |
| Average | 0.114 | 0.096 | 0.739 | 0.093 | 0.136 | 0.351 |

Table 3.1: K-fold Simple Neural Network Results Summary

| Fold | Kriging | | | | | |
|---------|---------------|-------------|-------------------|-----------------|-------|-------|
| | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 1 | 0.123 | 0.971 | 0.239 | 0.905 | 0.728 | 0.370 |
| 2 | 0.109 | 0.980 | 0.219 | 0.921 | 0.739 | 0.369 |
| 3 | 0.107 | 0.981 | 0.239 | 0.905 | 0.720 | 0.373 |
| 4 | 0.110 | 0.980 | 0.238 | 0.906 | 0.727 | 0.369 |
| 5 | 0.119 | 0.973 | 0.229 | 0.913 | 0.733 | 0.372 |
| Average | 0.114 | 0.977 | 0.233 | 0.910 | 0.729 | 0.371 |

Table 3.2: K-fold Kriging Results Summary

From the k-fold results it is evident that the simple neural network using X,Y,Z locations is unsuccessful. The simple neural network estimate values are close to the mean, resulting in an R^2 value of only 0.096. The kriged estimate has an R^2 of 0.910.

Figure 3.6 shows the results from the machine learning using X,Y,Z coordinates and the nearest 40 data. Adding the extra features appears to improve the estimation. The R^2 value increase to 0.742. The new ML model is considerably better than the previous model. When looking at the statistics in Table 3.3, the kriging results are still significantly better. The final check is a visual inspection to ensure that the ML estimation is artifact-free. From the prediction maps in Figure 3.6, the estimate appears to be similar to a nearest neighbour estimate as the width and height of the block are precisely the drill hole spacing; these are estimation artifacts. Although this ML estimation was better than the previous ML estimation, it is still not nearly as good as the kriged estimate in Table 3.2. In the next network, the depth is increased to five hidden layers in an attempt to improve the results.

3. Proposed Estimation Methods

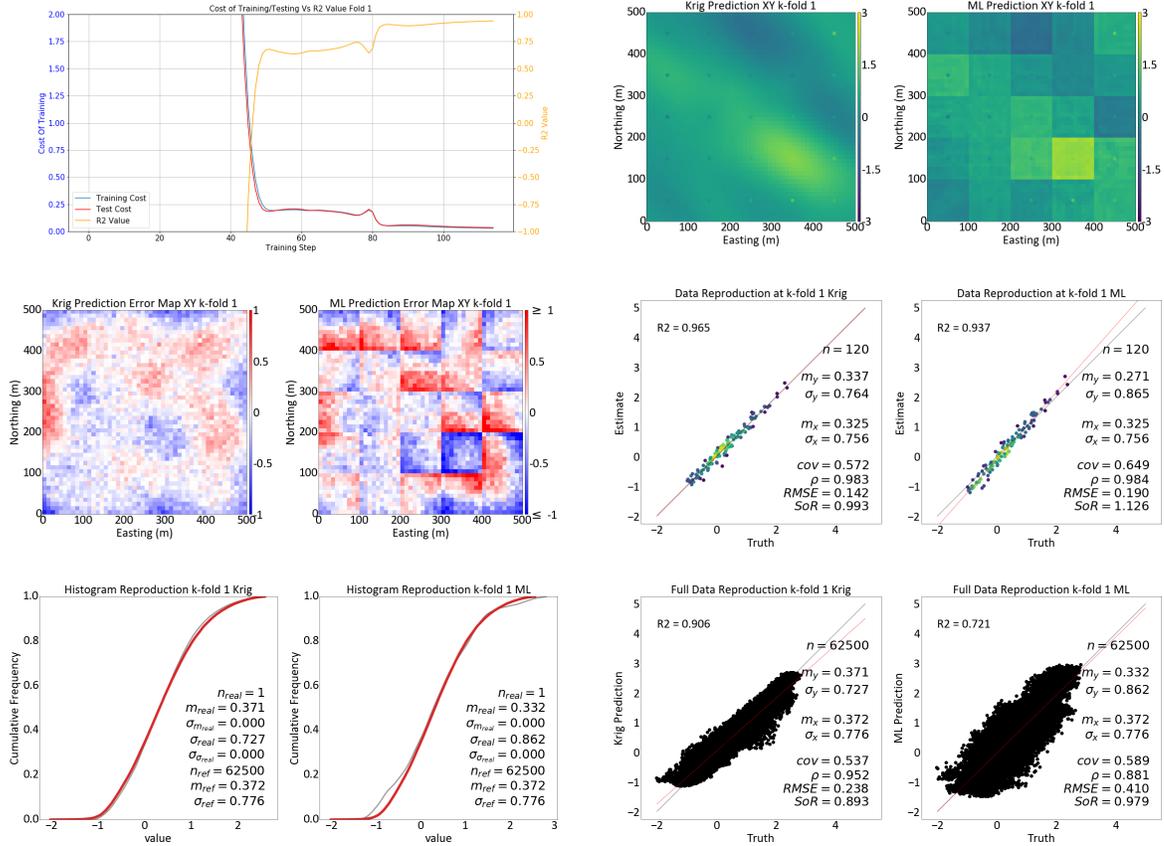


Figure 3.4: K-Fold Results From Simple Linear Neural Network Using XYZ data and nearest 40 Data

| Simple Neural Network Using XYZ Data and Nearest 40 Data | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 1 | 0.695 | 0.071 | 0.745 | 0.078 | 0.089 | 0.341 |
| 2 | 0.747 | 0.108 | 0.739 | 0.094 | 0.111 | 0.361 |
| 3 | 0.710 | 0.164 | 0.713 | 0.155 | 0.318 | 0.269 |
| 4 | 0.754 | 0.075 | 0.748 | 0.069 | 0.079 | 0.334 |
| 5 | 0.700 | 0.062 | 0.749 | 0.067 | 0.085 | 0.449 |
| Average | 0.114 | 0.096 | 0.739 | 0.093 | 0.136 | 0.351 |

Table 3.3: K-fold Simple Neural Network Using XYZ Data and Nearest 40 Data Results Summary

Figure 3.5 shows the structure of the five-layer hidden layer neural network and the results of one fold are shown in Figure 3.6. Table 3.4 is a summary of the five-layer hidden layer neural network. Adding layers to the network did not have much impact on the results. Adding layers does not appear to be the solution to fixing the artifact issues. In the next section, using a Gaussian activation function to activate the hidden layers, such as in Figure 2.10, is explored.

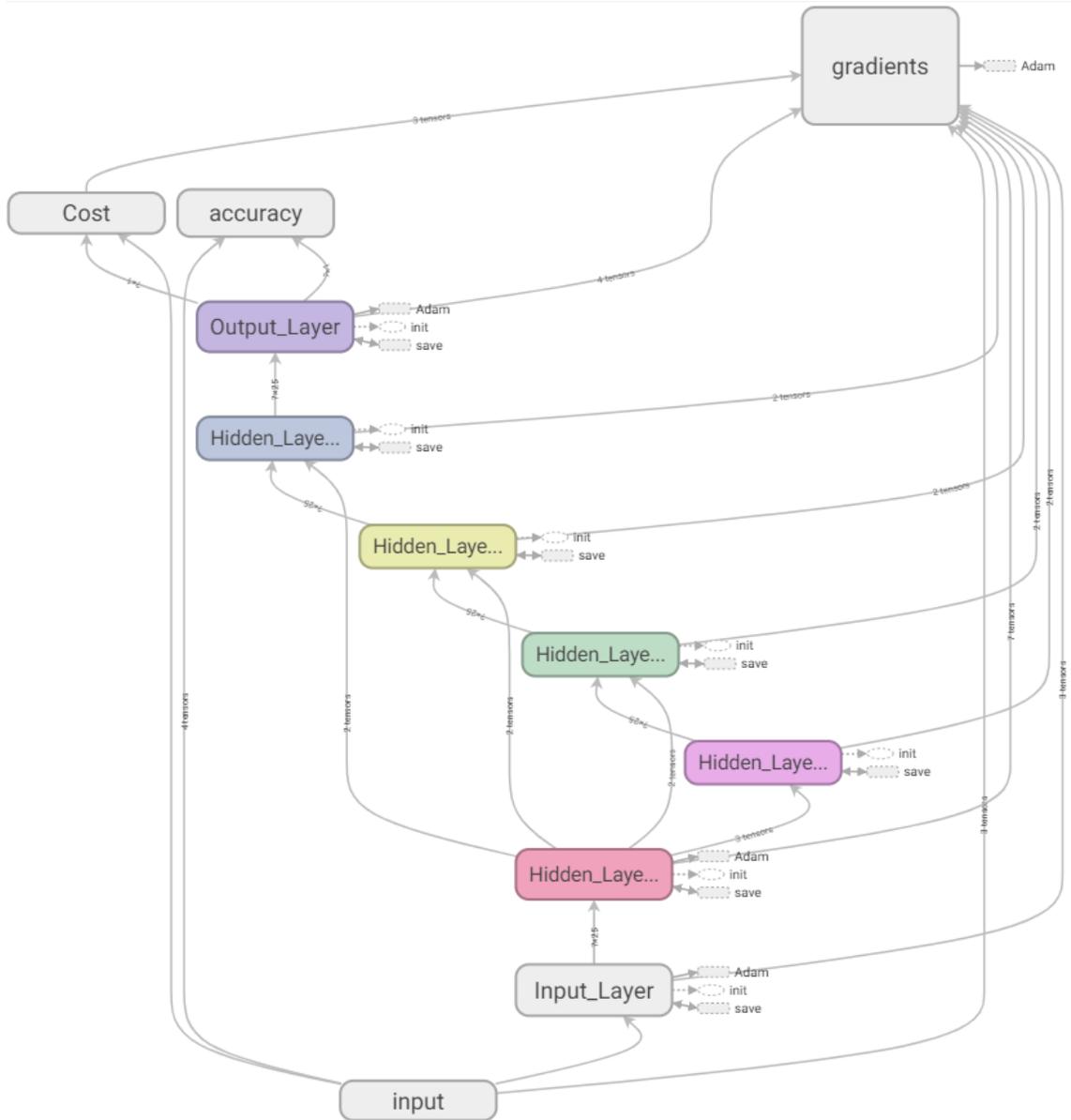


Figure 3.5: Simple Five Layer Linear Neural Network (*TensorBoard: Graph Visualization | TensorFlow Core | TensorFlow, n.d.*)

3. Proposed Cost Estimation Methods

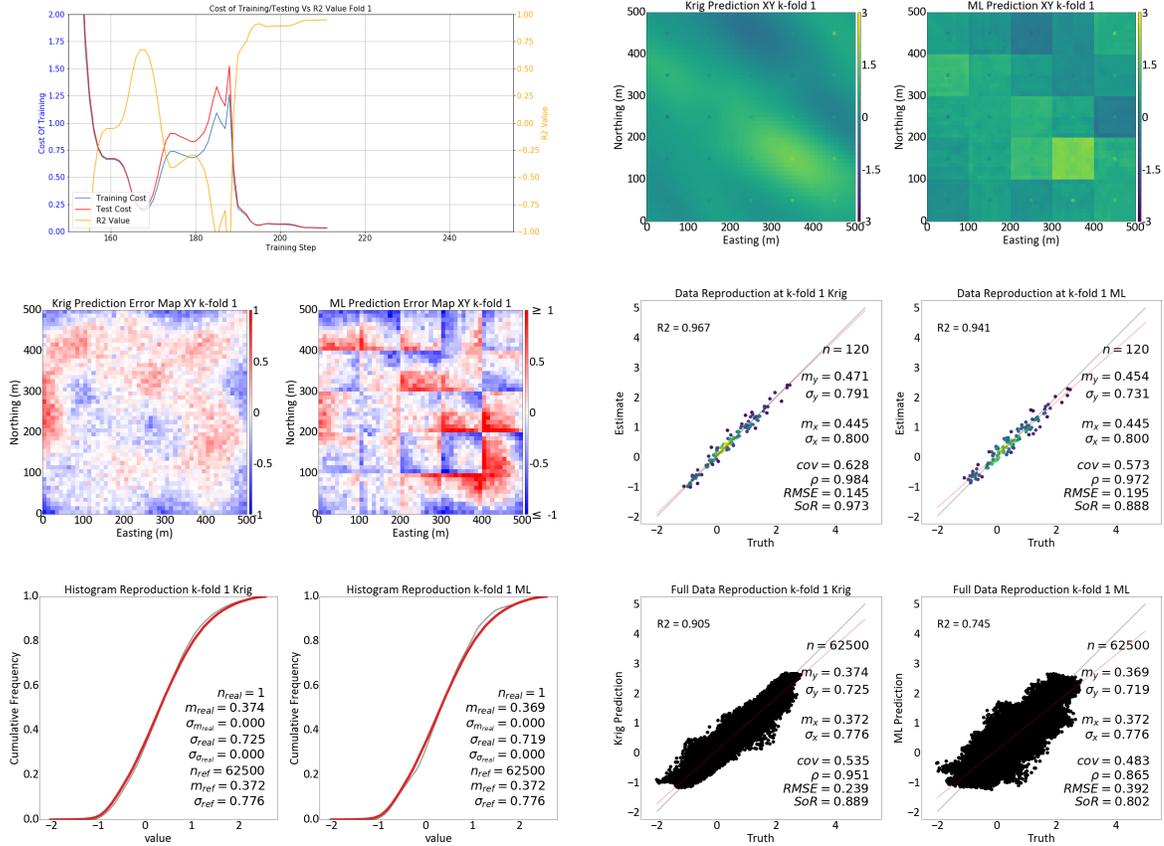


Figure 3.6: K-Fold Results From Simple Linear Neural Network Using XYZ data and nearest 40 Data

| Simple Neural Network XYZ Data 40 Closest Data With 5 Hidden Layers | | | | | | |
|---|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 1 | 0.195 | 0.941 | 0.392 | 0.745 | 0.719 | 0.369 |
| 2 | 0.185 | 0.935 | 0.409 | 0.722 | 0.825 | 0.358 |
| 3 | 0.185 | 0.944 | 0.390 | 0.747 | 0.766 | 0.337 |
| 4 | 0.216 | 0.904 | 0.398 | 0.737 | 0.739 | 0.328 |
| 5 | 0.182 | 0.947 | 0.382 | 0.758 | 0.694 | 0.378 |
| Average | 0.193 | 0.934 | 0.394 | 0.742 | 0.749 | 0.354 |

Table 3.4: K-fold Simple Neural Network Using XYZ Data and Nearest 40 Data Using 5 Hidden Layers Results Summary

3.1.2 Elliptical Radial Basis Function Neural Network

This section explores the elliptical radial basis function neural network (ERBFN). The ERBFN is a modified version of a radial basis function neural network (RBFN) using Mahalanobis distance instead of Euclidean distance. ERBFN have been used in text-independent speak verification, for example, in M. W. Mak and C. K. Li "Elliptical Basis Function Networks and Radial Basis Function Net-

works for Speaker Verification: Comparative Study." Examples using the RBFN and ERBFN are explored to demonstrate why using Mahalanobis distance is preferable to using Euclidean distance.

Cristian Rusu and Virginia Rusu use the RBFN in "*Radial Basis Functions Versus Geostatistics in Spatial Interpolations*" to predict the 2004 Spatial Interpolation Comparison dataset. They found that using an RBFN generates similar results to kriging and was the best option for geostatistical prediction with machine learning. Due to the similarity and ease of using an RBFN to make a prediction, they recommended using an RBFN over kriging for spatial interpolation (Rusu & Rusu, n.d.).

Three parameters are required when using a Gaussian activation function Equation 2.13: the number of Gaussian kernels, the radius r of each kernel, and the location of each kernel center cc . To determine the kernel centers of each RBF k-means is first used to get an initial location of the centers, the centers' locations are then optimized through training the neural network. Each kernel has a radius of one when training is initiated. The ADAM optimizer is used throughout training to optimize the radius of the kernel (Kingma & Ba, 2014). Finally, the number of kernels must be selected. Choosing the number of kernels is challenging; if the number of kernels is too small, the estimation is smooth and not be able to fit the data adequately. If the number of kernels is too large, the data is overfit, resulting in artifacts in the estimate. When estimations are overfit a small Gaussian filter can be applied to reduce the noise of the estimation (Elboher & Werman, 2012). In Figure 3.7 and Figure 3.8 show a visual representation of how the RBFN makes a prediction. Essentially the RBFN is the sum of multiple kernels multiplied by a weight at the prediction location to generate an estimate.

3. Proposed Estimation Methods

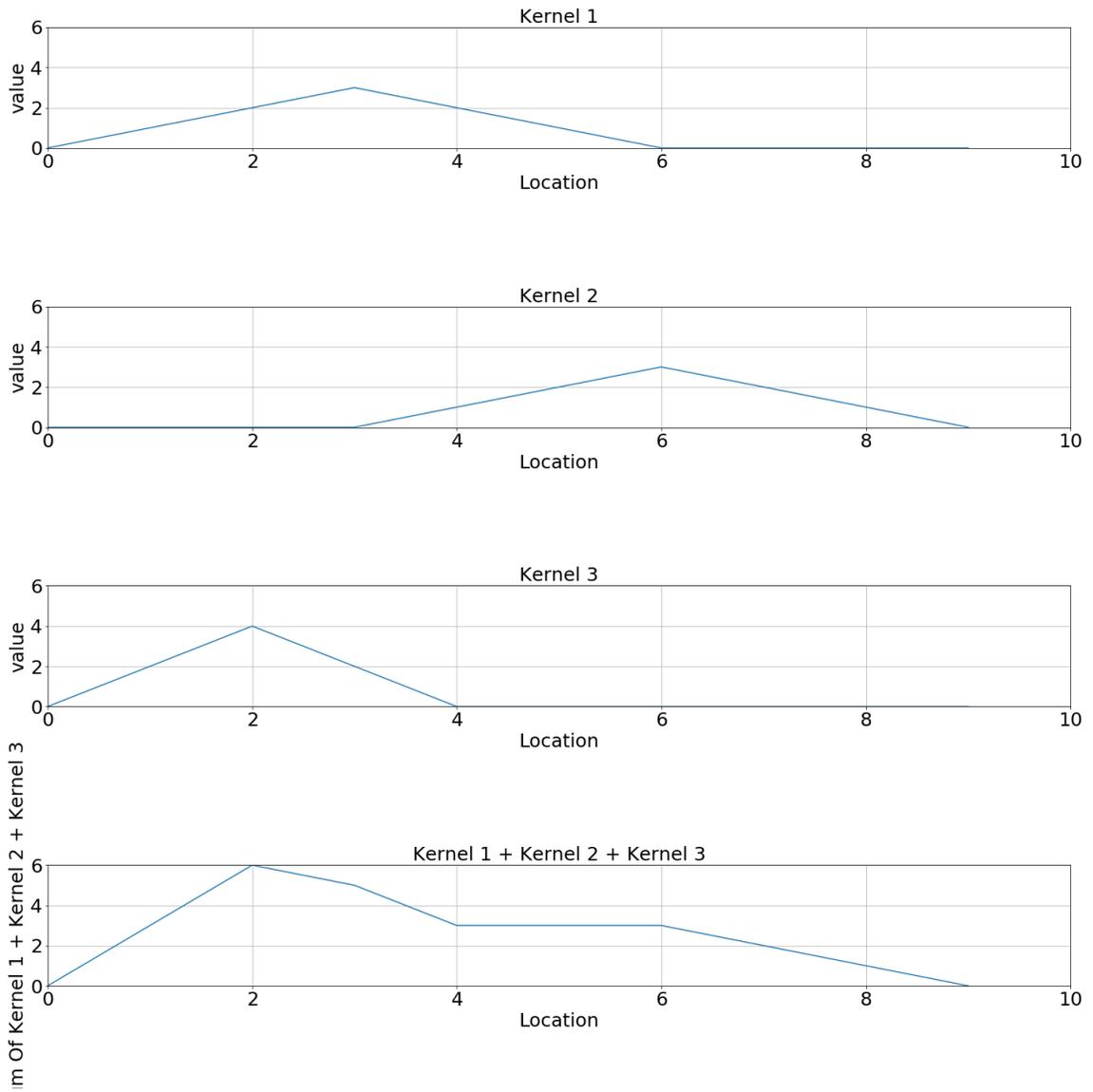


Figure 3.7: Simple RBF Example

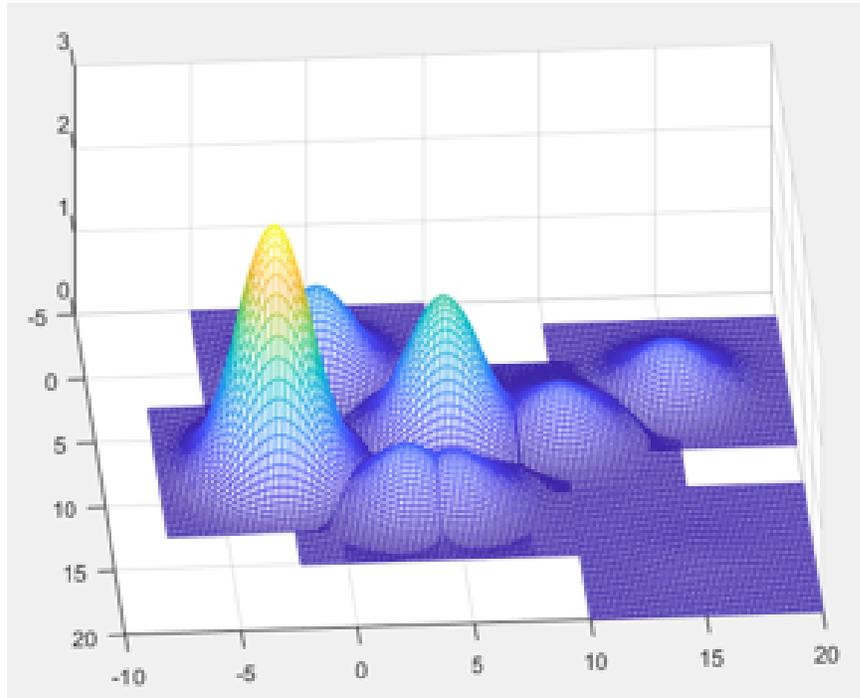


Figure 3.8: 3D RBF Example

The matrix form of the equation can be seen below in Equation 3.1.

$$\begin{bmatrix} e^{-(r\|cc_1-x_1\|)^2} & \dots & e^{-(r\|cc_1-x_i\|)^2} \\ \vdots & \ddots & \vdots \\ e^{-(r\|cc_n-x_i\|)^2} & \dots & e^{-(r\|cc_n-x_i\|)^2} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (3.1)$$

Where $\|cc_n - x_i\|$ is the Euclidean distance between the cluster center and the estimation locations. An example of RBFN graph can be seen in Figure 3.9.

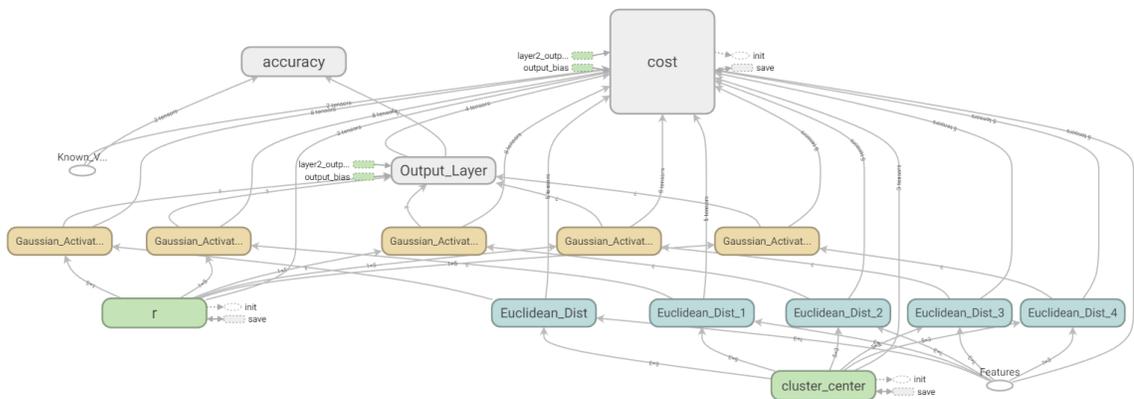


Figure 3.9: Radial Basis Function Network In Tensorboard (Note each Gaussian Kernel has its own radius value r) (TensorBoard: Graph Visualization | TensorFlow Core | TensorFlow, n.d.)

Everything is being optimized by the ML algorithm except the number of kernels. The more complex the deposit, the more kernels that are required. A few scenarios could be tested to ensure that the network is training to an acceptable error. Overfitting can occur if there are too many kernels.

Figure 3.10 shows the error and R^2 of the ERBFN as it trains. From Table 3.5, 15 kernels train the slowest and have the second-worst statistical results; there are not enough kernels to fit the data. 30 kernels train the fastest and has the worst results; there are too many kernels and the data is overfit.

The overfitting can be confirmed by looking at Figure 3.10 in the 30 kernels example. It is evident that there are too many nodes, and it can be seen clearly in the error map. Looking at the Summary Table 3.5 it can be seen that in the k-fold, the best results come from a balance between a low R^2 and a high $RMSE$ as they are both equally important.

3. Proposed Estimation Methods

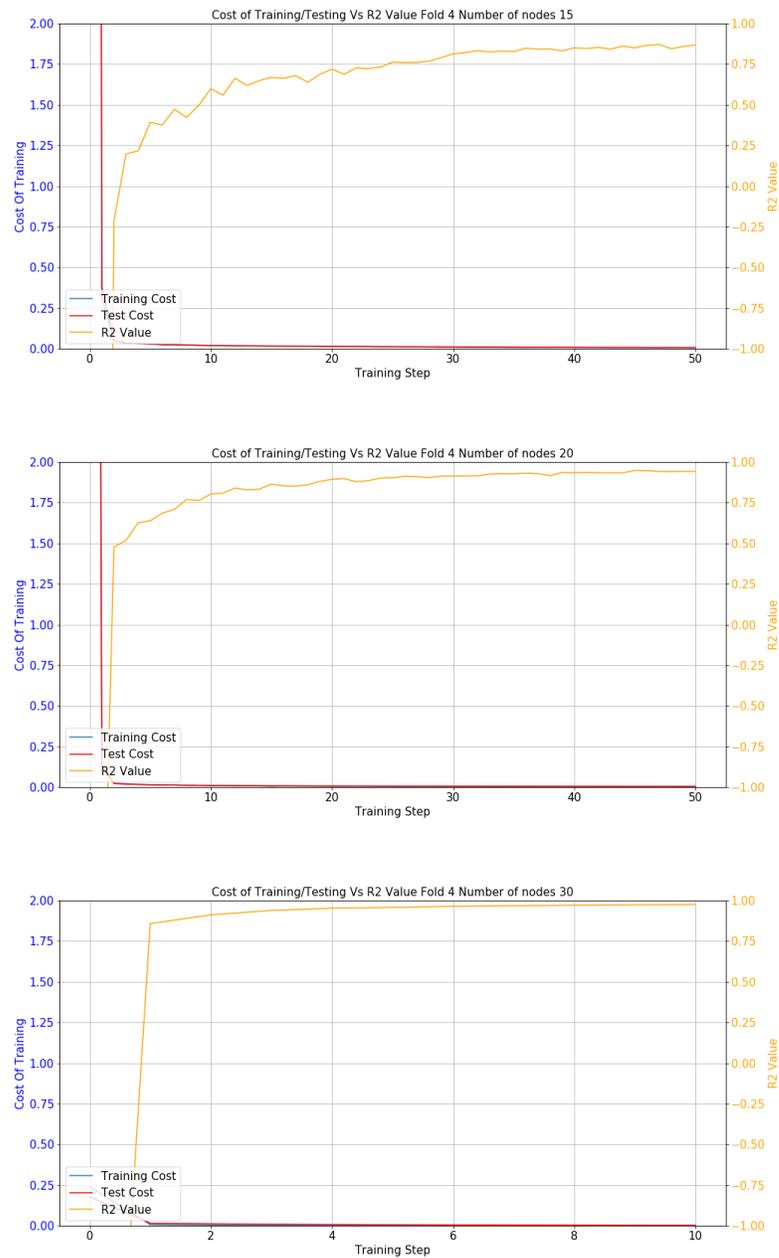


Figure 3.10: Example of Using Different Number of Kernel While Train an RFBN (Training Steps x 100)

3. Proposed Estimation Methods

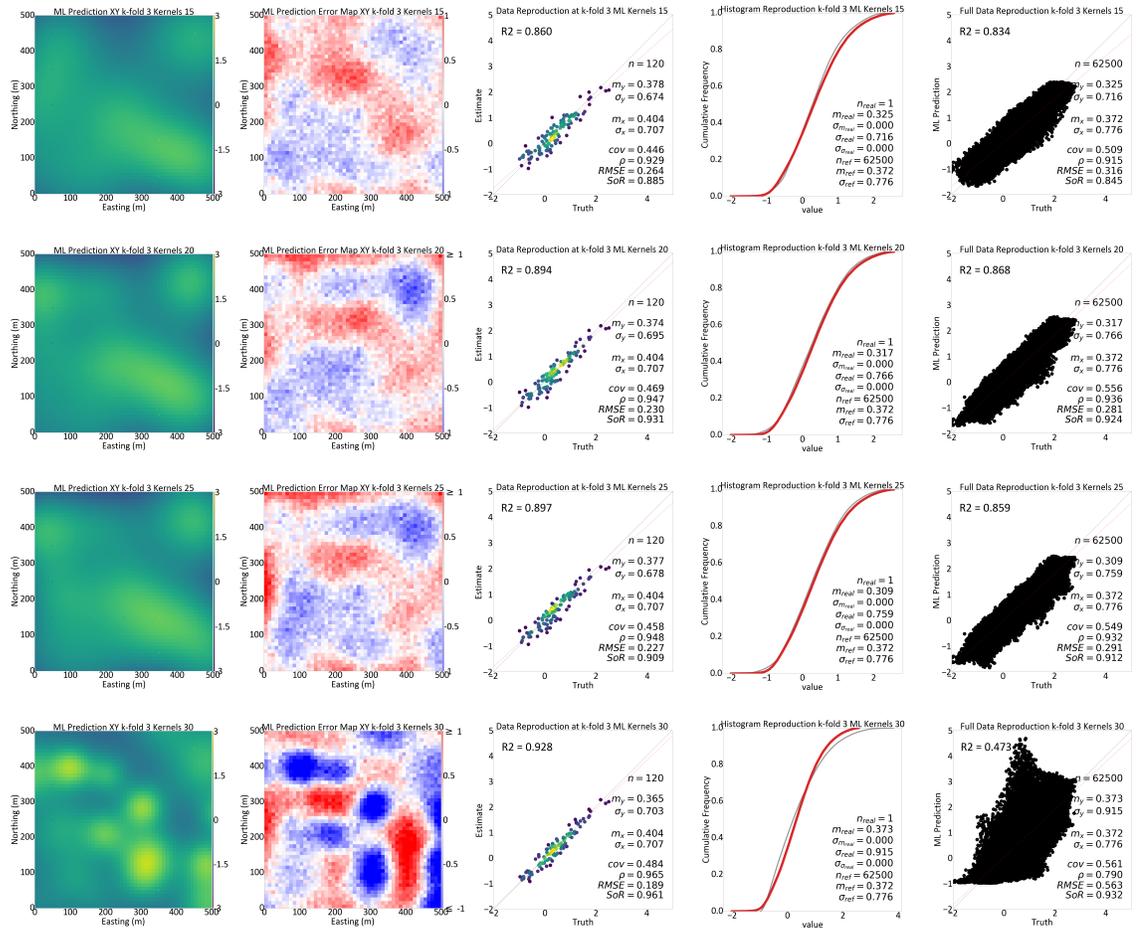


Figure 3.11: Example of Using Different Number of Nodes While Train an RBFN k-fold 3 Results

| Radial Basis Function K-fold 3 Kernel Comparison | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Kernels | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 15 | 0.264 | 0.860 | 0.316 | 0.833 | 0.716 | 0.325 |
| 20 | 0.230 | 0.894 | 0.281 | 0.868 | 0.766 | 0.317 |
| 25 | 0.227 | 0.897 | 0.291 | 0.859 | 0.759 | 0.309 |
| 30 | 0.372 | 0.928 | 0.563 | 0.473 | 0.915 | 0.372 |

Table 3.5: Summary Kernel Comparison Radial Basis Ensemble Using XYZ Data

Trial and error is not always suitable for determining the best estimation; another solution is to use an ensemble technique. Ensembling is a method of combining multiple estimations into one by averaging all estimates together. Using a range of kernels that go from too low to high and ensembling balances out the estimation, resulting in an improved estimation.

Table 3.5 confirms that using an ensemble estimate results in a better estimation than each node estimate alone.

| Radial Basis Function Neural Network Using XYZ Location Ensemble | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 1 | 0.201 | 0.937 | 0.280 | 0.870 | 0.728 | 0.325 |
| 2 | 0.201 | 0.935 | 0.255 | 0.892 | 0.729 | 0.317 |
| 3 | 0.198 | 0.922 | 0.274 | 0.875 | 0.747 | 0.331 |
| 4 | 0.208 | 0.904 | 0.304 | 0.846 | 0.753 | 0.309 |
| 5 | 0.240 | 0.947 | 0.298 | 0.852 | 0.698 | 0.370 |
| Average | 0.210 | 0.922 | 0.283 | 0.867 | 0.731 | 0.330 |

Table 3.6: Summary of Radial Basis Ensemble Using XYZ Data

The kriging statistical in Table 3.2 are compared to the ensemble results in Table 3.6 and the visual checks in 3.14.

From the visual and statistical check, the RBFN and kriging give similar results, which support M. W. Mak and C. K. Li "Elliptical Basis Function Networks and Radial Basis Function Networks for Speaker Verification: Comparative Study" findings. It is worth noting how similar the variogram reproduction is between the RBFN and the kriged estimate, even though we do not expect reproduction to the variogram and how both estimation methods produce similar estimation continuity (Figure 3.12).

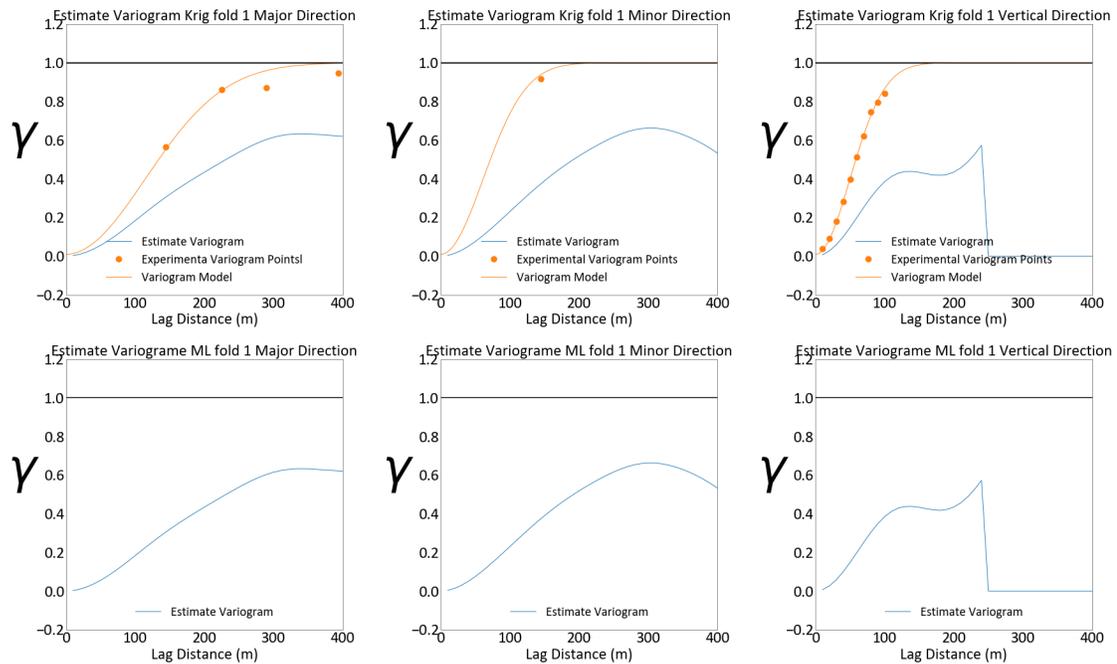


Figure 3.12: K-Fold Results Variogram Results From Radial Basis Neural Network Using XYZ Data

3. Proposed Estimation Methods

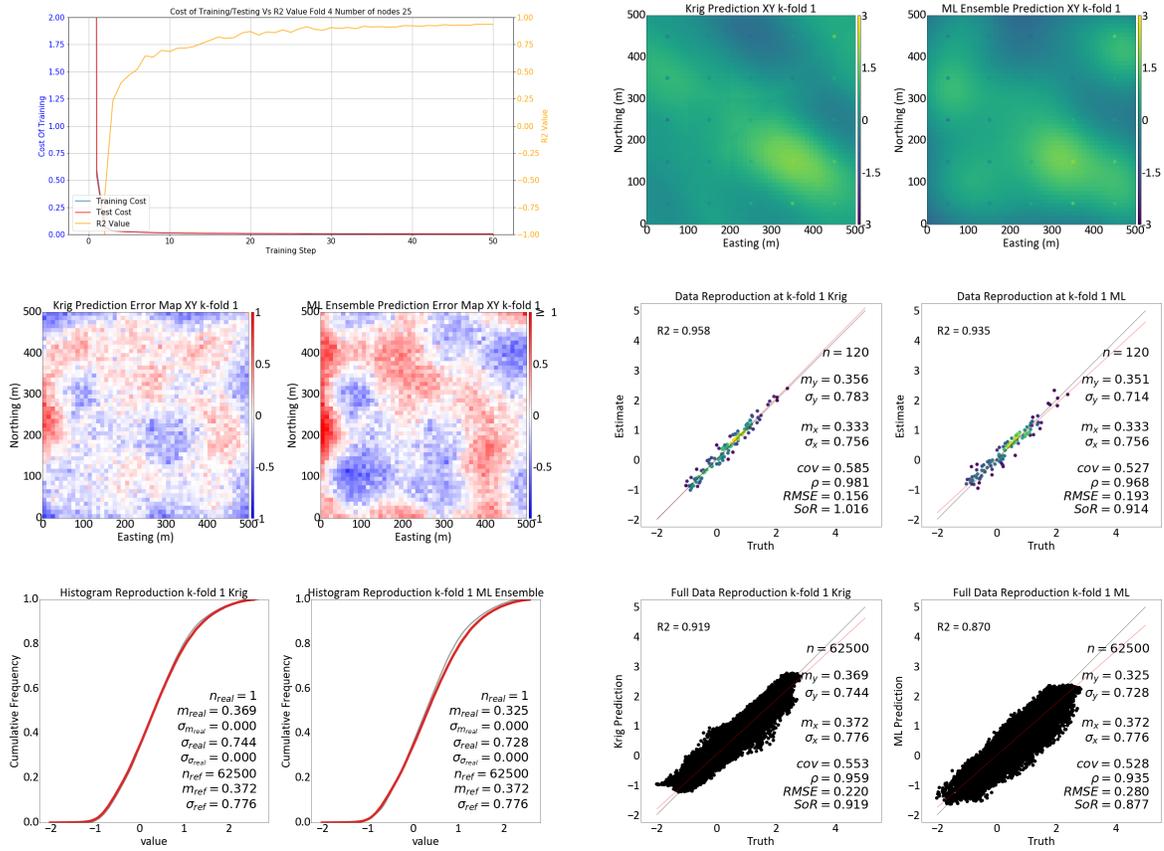


Figure 3.13: K-Fold Results From Radial Basis Neural Network Using XYZ Data)

3. Proposed Estimation Methods

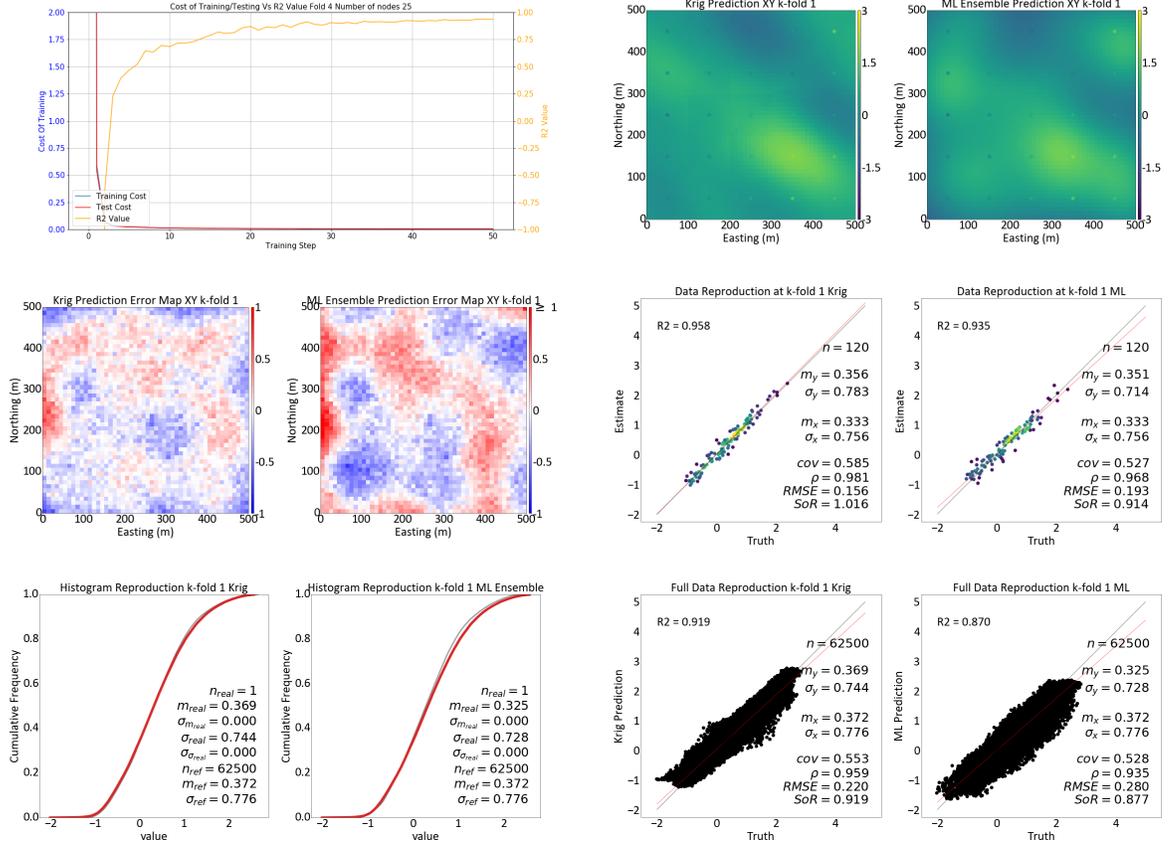


Figure 3.14: K-Fold Results From Radial Basis Neural Network Using XYZ Data

Using an ERBFN, instead of an RBFN, further improves the ML estimate. The purpose of converting the RBFN to an ERBFN is to include anisotropy into the estimation by introducing a covariance term into the estimation with Mahalanobis distance. The ERBFN has the same structure as the RBFN Figure 3.9; however, a covariance term for each node is learned, see Equation 3.2.

$$\begin{bmatrix} e^{-(r\sqrt{\Delta_1^T S_1^{-1} \Delta_1})} & \dots & e^{-(r\sqrt{\Delta_1^T S_n^{-1} \Delta_1})} \\ \vdots & \ddots & \vdots \\ e^{-(r\sqrt{\Delta_i^T S_1^{-1} \Delta_i})} & \dots & e^{-(r\sqrt{\Delta_i^T S_n^{-1} \Delta_i})} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad (3.2)$$

Looking at the ERBFN S_n^{-1} or the covariance/shape of each kernel is a new parameter that must be optimized by the neural network. The covariance term is different for each kernel allowing for multiple directions of anisotropy in a single estimation. Kernels are no longer forced to be circular. Figure 3.15 shows a simple data set simulated in 3D with 10000 data points and then sampled for 20% of data set (2000 data points), the sampled data is passed through a RBFN and a ERBFN with one node. Figure 3.16 shows that the ERBFN better reproduces the simulated data, R^2 equal to 0.948, as it is able to develop the correct distribution shape due to the Mahalanobis distance; whereas, the

RBFN, R^2 of 0.782, is limited to a traditional Gaussian structure using Euclidean distance.

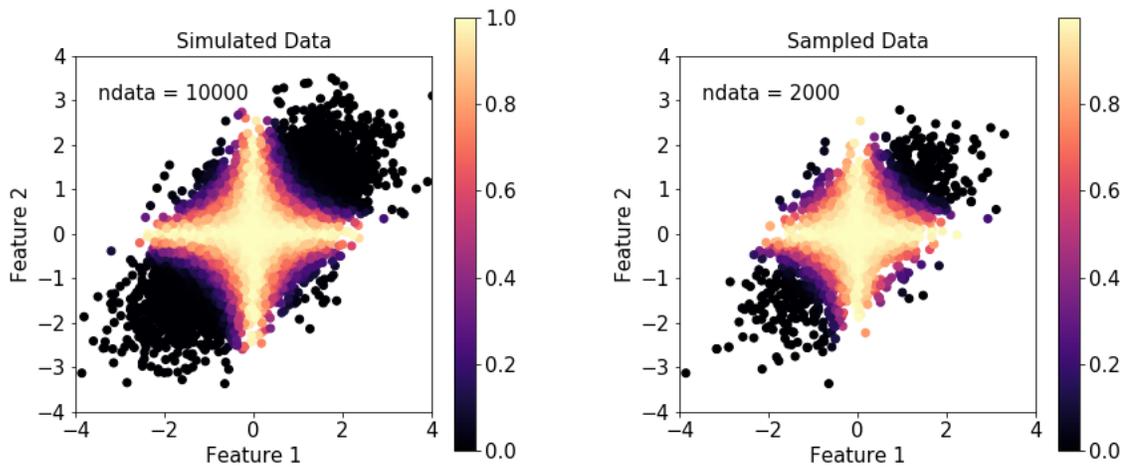


Figure 3.15: Simple Simulated and Sampled Data Example for RBFN to ERBFN

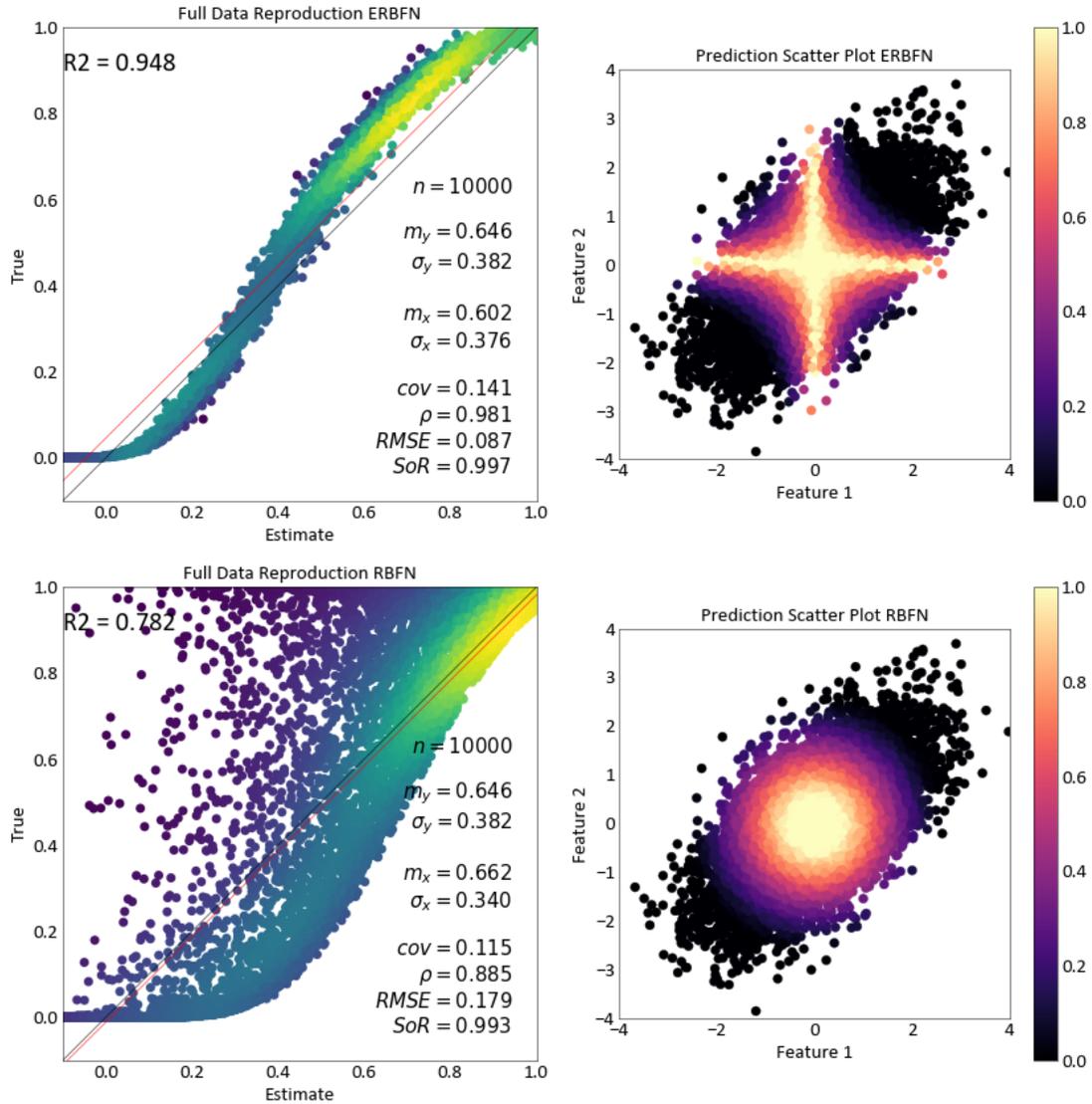


Figure 3.16: Results From Simple Simulated and Sampled Data Example for RBFN to ERBFN

The ERBFN outperforms the RBFN in this simulated example. The test case from above will now be considered.

Similar to the RBFN, the number of nodes is not optimized. Figure 3.17 and Table 3.7 shows the kernel comparison of k-fold three. All folds produced similar results and k-fold three was chosen at random. In the location map there is visible evidence of modelling artifacts; however, looking at Figure 3.14 it can be seen that by ensembling the different kernel estimations the artifacts are smoothed out and are statistically better than the single kernel estimations, see Tables 3.7 and 3.8.

3. Proposed Estimation Methods

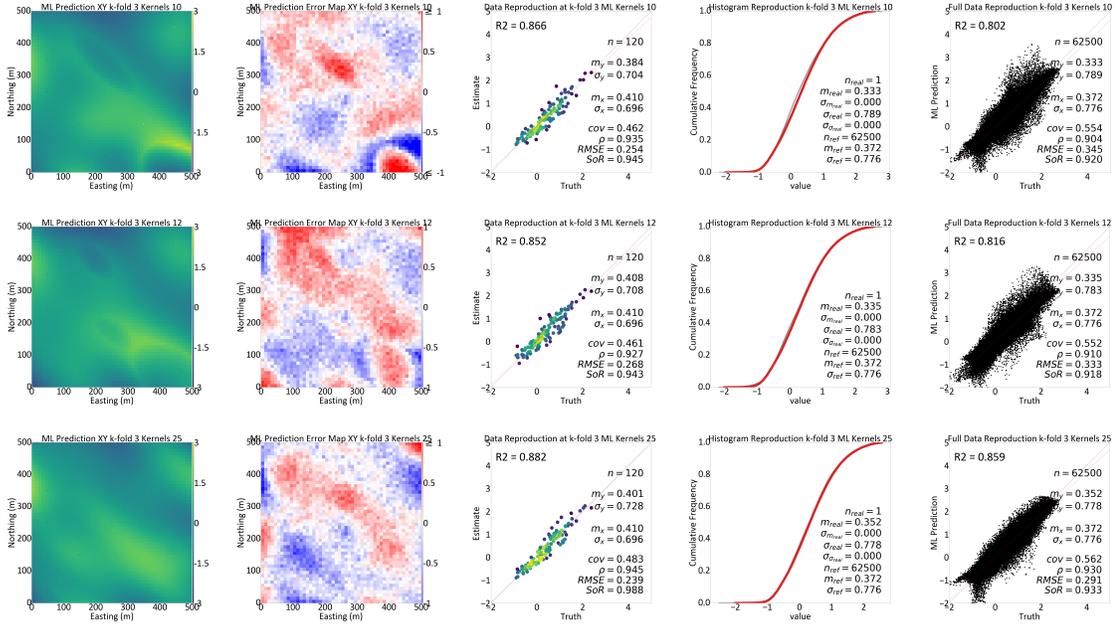


Figure 3.17: Example of Using Different Number of Nodes While Train an ERBFN k-fold 3 Results

| Elliptical Radial Basis Function K-fold 3 Kernel Comparison | | | | | | |
|---|---------------|-------------|-------------------|-----------------|-------|-------|
| Kernels | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 10 | 0.254 | 0.866 | 0.345 | 0.802 | 0.789 | 0.333 |
| 12 | 0.268 | 0.852 | 0.333 | 0.816 | 0.783 | 0.335 |
| 25 | 0.239 | 0.882 | 0.291 | 0.859 | 0.778 | 0.352 |

Table 3.7: Summary Kernel Comparison Elliptical Radial Basis Ensemble Using XYZ Data

| Elliptical Radial Basis Function Neural Network Using XYZ Location | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.776 | 0.372 |
| 1 | 0.219 | 0.920 | 0.269 | 0.879 | 0.752 | 0.357 |
| 2 | 0.224 | 0.926 | 0.262 | 0.886 | 0.748 | 0.365 |
| 3 | 0.223 | 0.897 | 0.271 | 0.878 | 0.763 | 0.340 |
| 4 | 0.234 | 0.900 | 0.275 | 0.874 | 0.746 | 0.339 |
| 5 | 0.223 | 0.915 | 0.306 | 0.844 | 0.767 | 0.358 |
| Average | 0.225 | 0.912 | 0.277 | 0.872 | 0.755 | 0.352 |

Table 3.8: Summary of Elliptical Radial Basis Ensemble Using XYZ Data

The results of RBFN and the ERBFN, are similar, 0.283 RMSE with 0.867 R^2 and 0.277 RMSE and 0.872 R^2 , respectively for the full model. The example data set has relatively simple anisotropy. When comparing the ERBFN to the kriged model, the kriged model still out preferences the ML algorithm with a kriging RMSE and R^2 of 0.233 and 0.910, respectively.

3. Proposed Estimation Methods

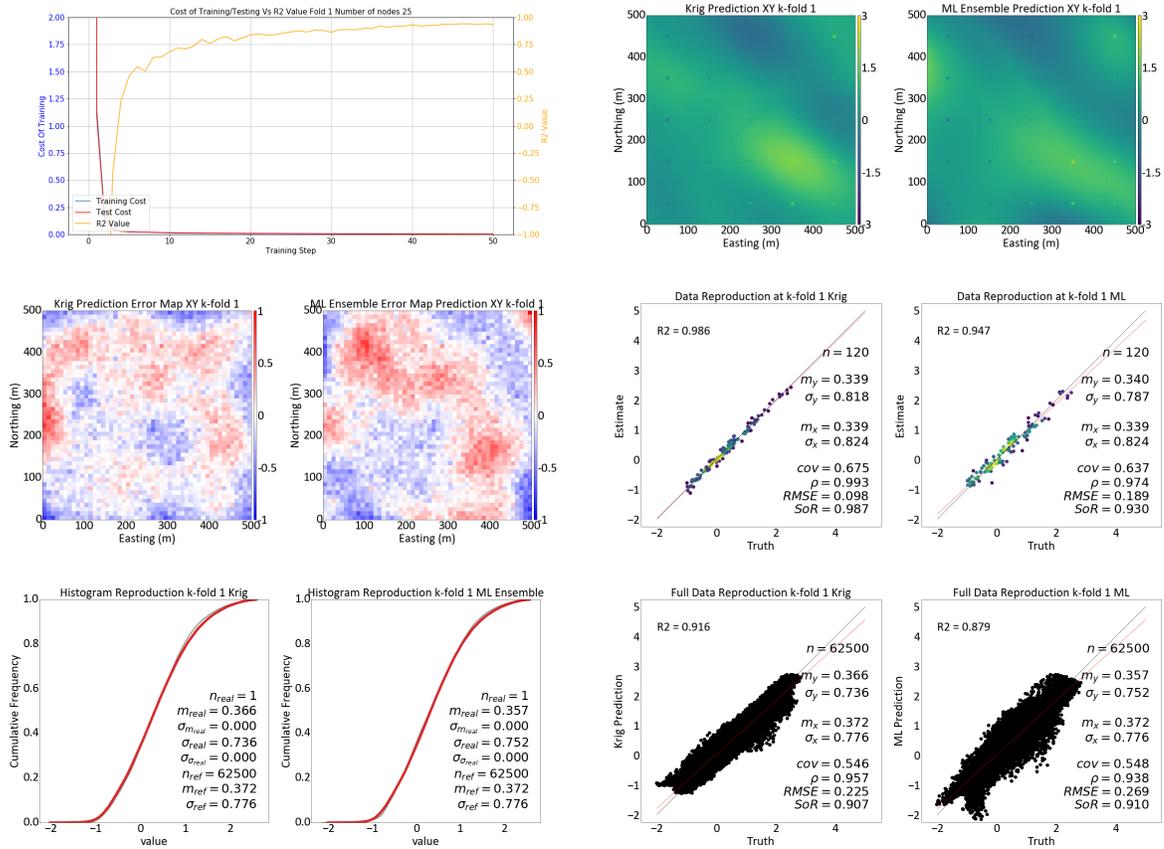


Figure 3.18: K-Fold Results From Elliptical Radial Basis Neural Network Using XYZ Data(Training Steps x 100)

Figure 3.19 shows a simulated data set that resembles a doughnut and drilled at a spacing of 35% of the variogram range in the X and Y direction and 5% of the range in the Z direction.

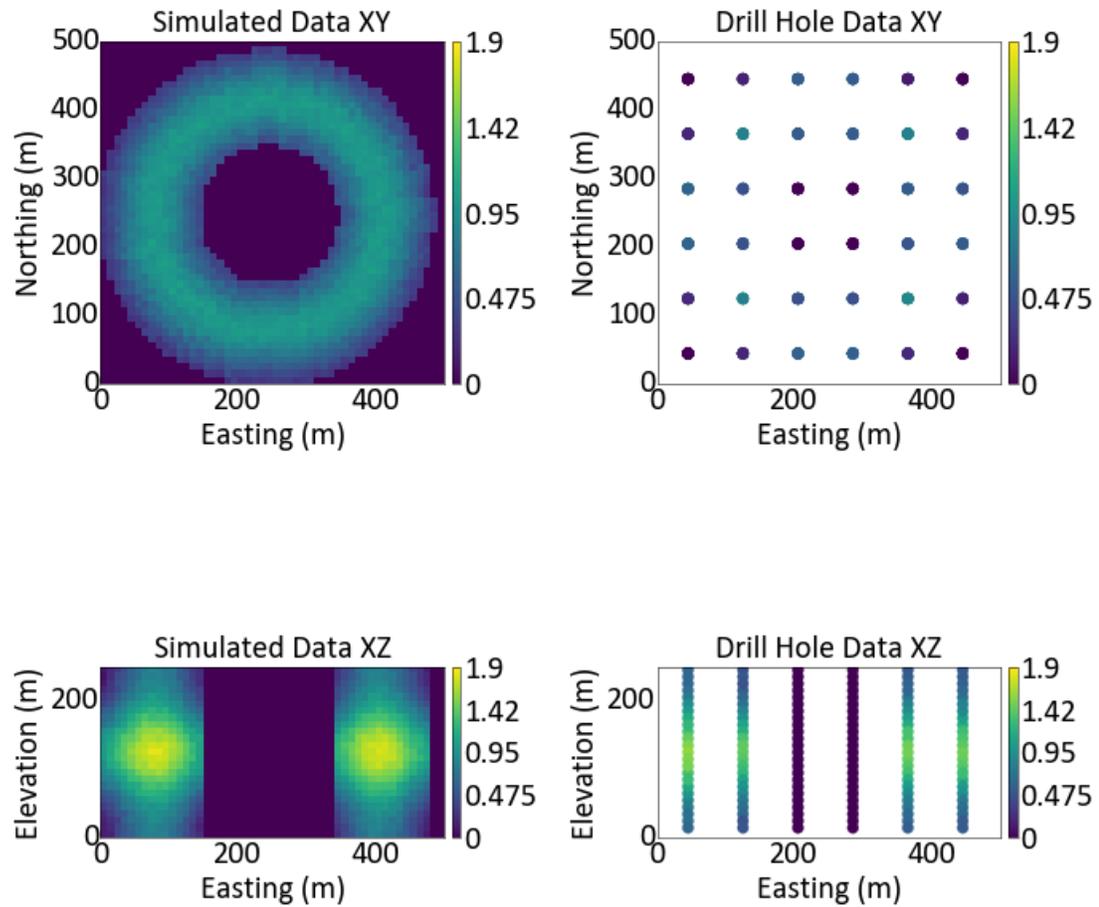


Figure 3.19: Complex Structure Example, Doughnut Data

Estimations are generated using a fivefold validation method for both kriging and machine learning. Estimations are generated using an omnidirectional variogram for the kriging workflow and the XYZ locations for the ML features.

Figure 3.20 show the results from one fold with the statistical results of all folds in Table 3.9. From the k-fold results, it appears as if the kriging is slightly better; however, looking at the kriged prediction plot, there is evidence of artifacts. Looking at the full model data reproduction, the ERBFN outperforms the kriged model in terms of both RMSE and R^2 . For this complex scenario, the ERBFN estimation is better than the kriged estimation.

3. Proposed Estimation Methods

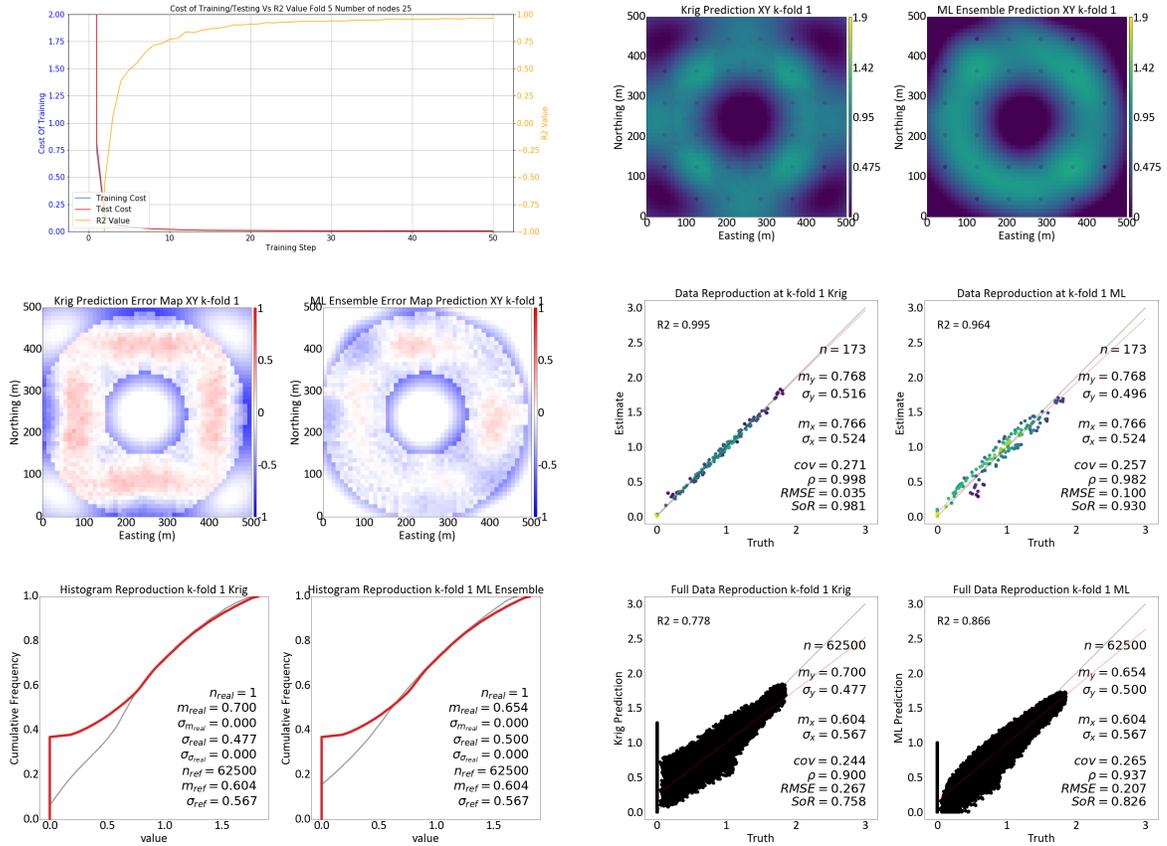


Figure 3.20: K-Fold Results From Elliptical Radial Basis Neural Network Using Doughnut Data Set (Training Steps x 100)

| Elliptical Radial Basis Function Neural Network Using Doughnut Data Set | | | | | | |
|---|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.518 | 0.604 |
| 1 | 0.100 | 0.964 | 0.207 | 0.866 | 0.500 | 0.654 |
| 2 | 0.084 | 0.972 | 0.214 | 0.857 | 0.499 | 0.657 |
| 3 | 0.093 | 0.967 | 0.216 | 0.854 | 0.496 | 0.661 |
| 4 | 0.099 | 0.964 | 0.221 | 0.847 | 0.497 | 0.664 |
| 5 | 0.099 | 0.964 | 0.210 | 0.862 | 0.500 | 0.653 |
| Average | 0.095 | 0.966 | 0.214 | 0.858 | 0.498 | 0.658 |

Table 3.9: Summary of Elliptical Radial Basis Ensemble Using Doughnut Data Set

| Simple Kriging Using Doughnut Data Set | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.518 | 0.604 |
| 1 | 0.035 | 0.995 | 0.267 | 0.778 | 0.477 | 0.700 |
| 2 | 0.033 | 0.996 | 0.268 | 0.776 | 0.477 | 0.699 |
| 3 | 0.030 | 0.997 | 0.267 | 0.778 | 0.479 | 0.700 |
| 4 | 0.032 | 0.996 | 0.267 | 0.778 | 0.480 | 0.699 |
| 5 | 0.033 | 0.996 | 0.269 | 0.774 | 0.477 | 0.701 |
| Average | 0.033 | 0.996 | 0.268 | 0.777 | 0.478 | 0.700 |

Table 3.10: Summary of Simple Kriging Using Doughnut Data Set

A machine learning neural network algorithm is a regression, with no data reproduction at the known locations. In real estimation scenario data reproduction is required. Figure 3.21 shows the drill hole reproduction for kriging and the ERBFN.

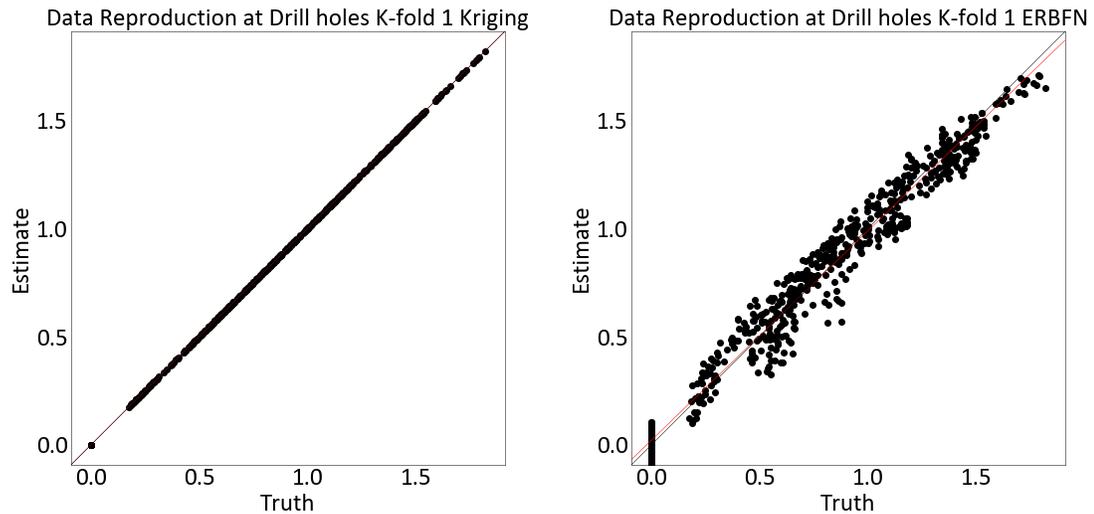


Figure 3.21: Drill Hole Data Reproduction for ERBFN and Kriging

The estimates at data locations could be replaced with the correct value; however, this would lead to artifacts. A method of combining both ML and Geostatistics estimation methods is the focus of the next section.

3.2 Combining Machine Learning and Geostatistical Estimation

Recall from Chapter 2 the geostatistical method using exhaustive secondary data, Equation 2.42 intrinsic collocated cokriging (ICCK); this estimation method takes advantage of super secondary data and the correlation between the super secondary and the primary data. In Figure 3.22, the machine learning and geostatistical estimation workflow can be seen. The workflow involves making a ML estimation to generate a super secondary variable for an ICCK estimation. At first glance this

seems strange, using an estimation to make an estimation; however, machine learning estimate is used to enforce non-stationarity feature into the ICCK estimation. Due to known data not being reproduced exactly in the machine learning method the correlation between the estimation is close to one, giving fairly similar weights to the primary and secondary data in the ICCK workflow.

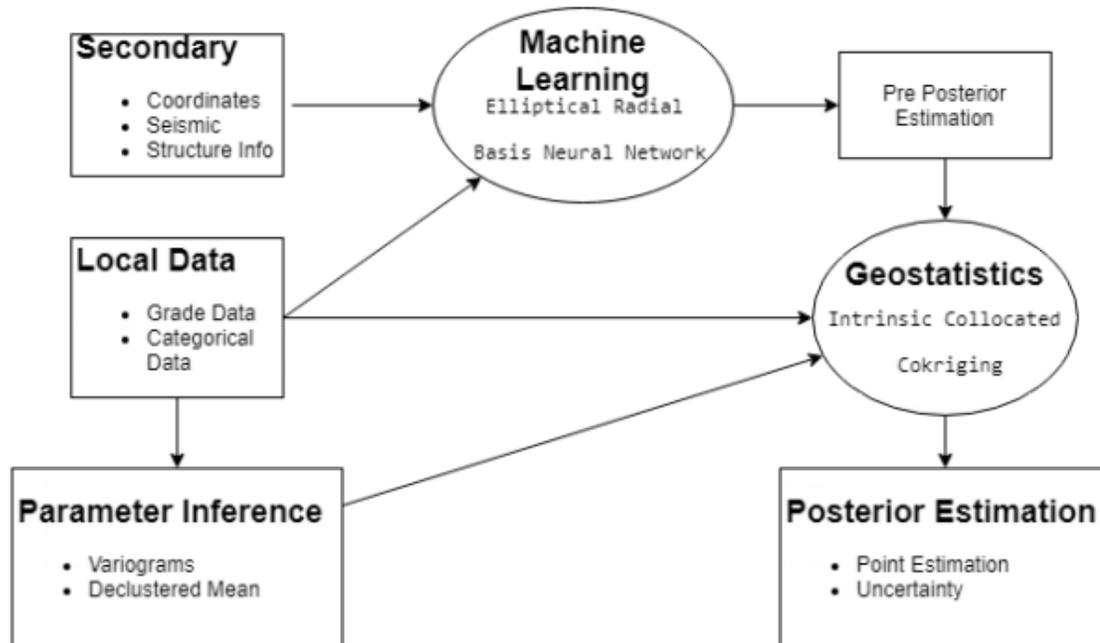


Figure 3.22: Machine Learning And Geostatistical Estimation Work Flow

As demonstrated above, the ERBFN helps model non-stationary features and provides a reasonable estimation. Also, from above, it can be seen that kriging is better when deposits are not complex and reproduce data exactly; thus combining these two estimation methods using ICCK should allow for data reproduction and model non-stationary features more effectively. Figure 3.23 shows the results with a summary of the results in Table 3.11. The results and summary table illustrate that combining the estimation methods further improves the results of the estimation, with the R^2 being closer to one and the $RMSE$ being lower in the full model results. Figure 3.24 confirms data reproduction. Another advantage of combining these estimations is that by using the Geostatistical framework it is possible to simulate multiple realizations. In Pierre Goovaerts' book *Geostatistics For Natural Resources Evaluation* and A. G. Journel/Phaedon C. Kyriakidis' *Evaluation of Mineral Reserves: a Simulation Approach* the method of simulating models for uncertainty and the purpose of simulating models for uncertainty is described.

3. Proposed Estimation Methods

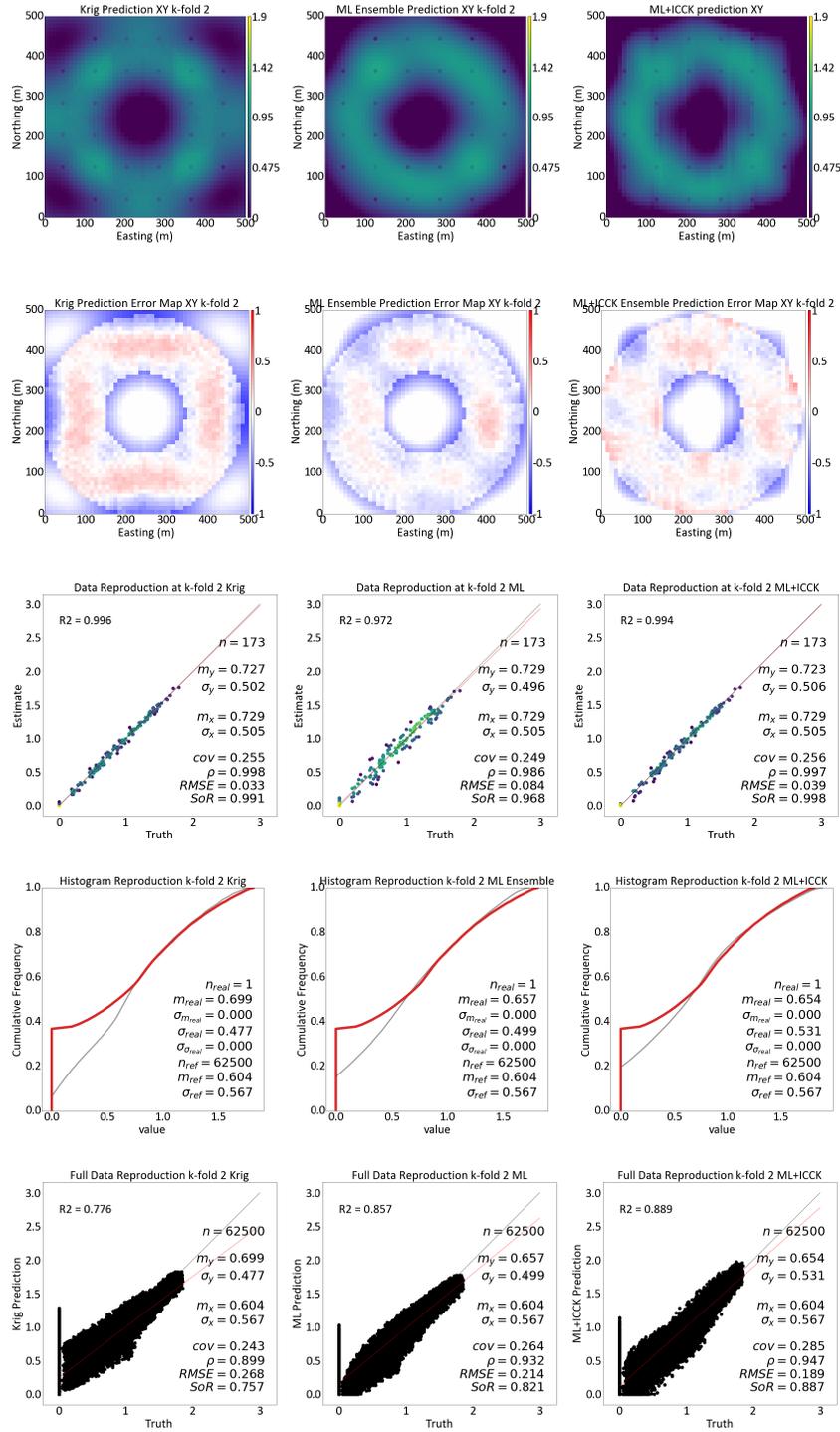


Figure 3.23: K-Fold Results From ICCK Plus ERBFN Using Doughnut Data Set

| ERBFN + ICCK Using Doughnut Data Set | | | | | | |
|--------------------------------------|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.518 | 0.604 |
| 1 | 0.038 | 0.995 | 0.189 | 0.889 | 0.536 | 0.652 |
| 2 | 0.039 | 0.994 | 0.189 | 0.889 | 0.531 | 0.654 |
| 3 | 0.031 | 0.996 | 0.189 | 0.889 | 0.529 | 0.653 |
| 4 | 0.037 | 0.995 | 0.194 | 0.882 | 0.526 | 0.659 |
| 5 | 0.040 | 0.994 | 0.192 | 0.885 | 0.537 | 0.652 |
| Average | 0.037 | 0.995 | 0.191 | 0.887 | 0.532 | 0.654 |

Table 3.11: Summary of ICCK Plus ERBFN Using Doughnut Data Set

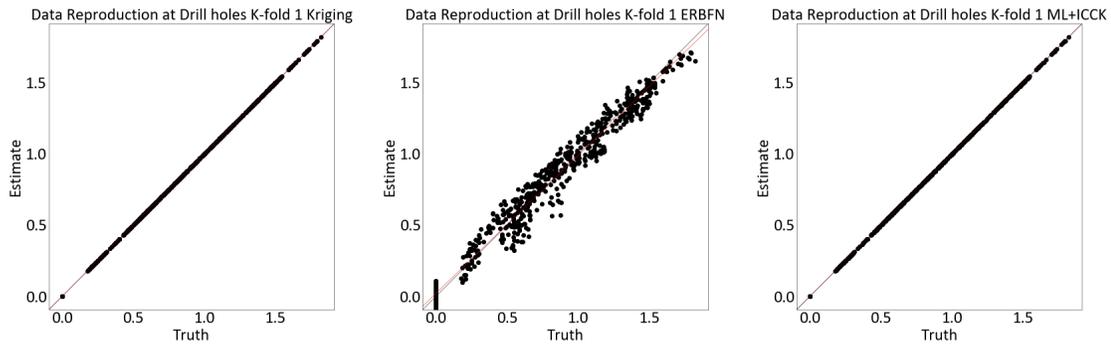


Figure 3.24: Drill Hole Data Reproduction for ICCK Plus ERBFN

In the next example (Figure 3.25) ordinary intrinsic collocated cokriging is demonstrated (OICCK). OICCK reproduces the secondary data when there is no local conditioning. There is a slight deviation between the OICCK estimate and the secondary data if the OICCK search range is greater than the variogram range. If there is conditioning data in the search range, a small amount of weight is given to conditioning data to estimate the unbiased conditional mean.

Figure 3.25 shows the conditioning data, secondary data, ICCK estimate, and the OICCK estimate. A variogram range that is 20% of the domain and a search radius that is 50% domain for both estimate types is used with a mean of two for ICCK. In Figure 3.25 as the easting increases for the estimates, the influence of the conditioning data decreases. In ICCK, as the influence of conditioning data decrease, more weight is transferred to the mean. In OICCK, as the influence of conditioning data decrease, more weight is given to the secondary data. For the proposed workflow in Figure 3.22, if the estimate location is outside conditioning data range, using OICCK to generate the posterior estimate results in the ML estimate.

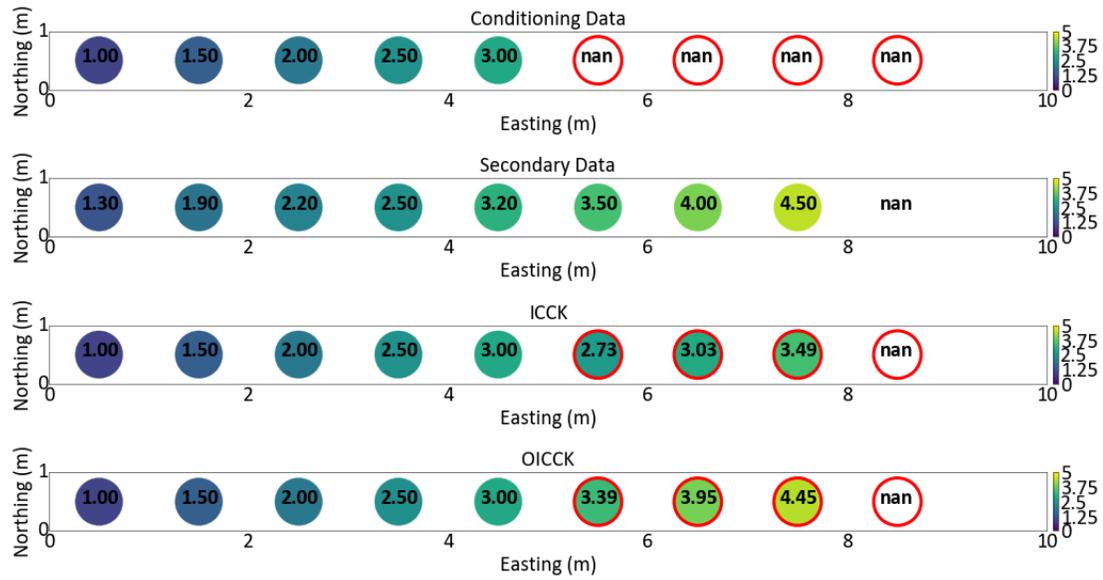


Figure 3.25: Small Example Demonstrating the Difference Between ICCK and OICCK (Estimates Are Made in Red Circles)

3.3 Sensitivity Studies

The first sensitivity study shows that, the ICCK+ML estimation method behaves similarly to kriging as the number of data increase. In Figure 3.26 a simulated data set that is Gaussian in nature with a variogram that has a major direction of 135° with a major range that is 40% of the width and minor and vertical range that is 15% of the width. This data set was then sampled one-hundred times randomly with a vertical sampling that is 4% of the elevation. Kriging, ML, and ICCK+ML estimation generated using a fivefold validation method and all relevant statistics calculated.

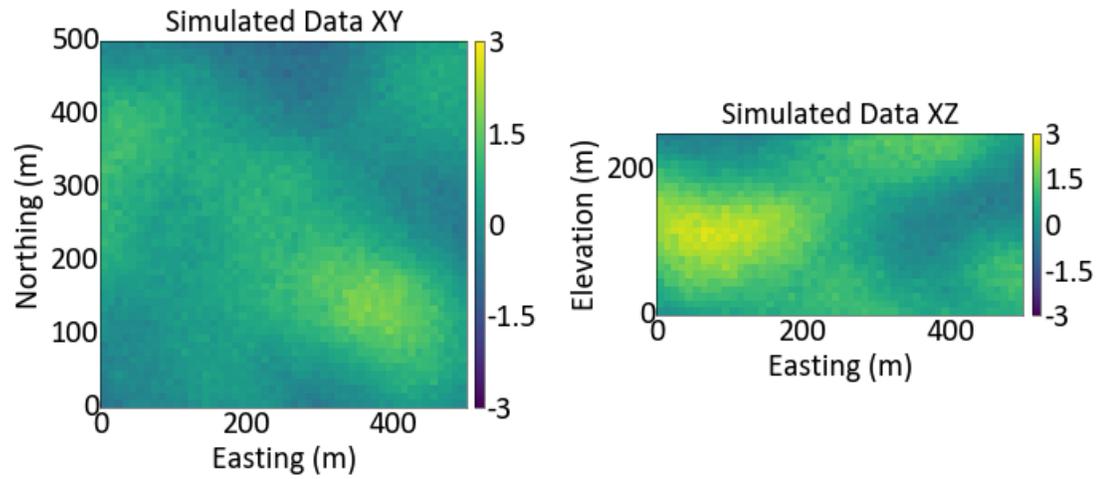


Figure 3.26: Simulated Data Set for Number of Data Sensitivity Study

Figure 3.27, 3.28, 3.29, and 3.30 show examples with 100,75,50,and 25 drill holes. The summary graph of the RMSE and R^2 is shown in Figure 3.31 with a table of the summary statistics in Tables 3.12, 3.13, and 3.14 for examples ranging from 100 to 5 drill holes.

3. Proposed Estimation Methods

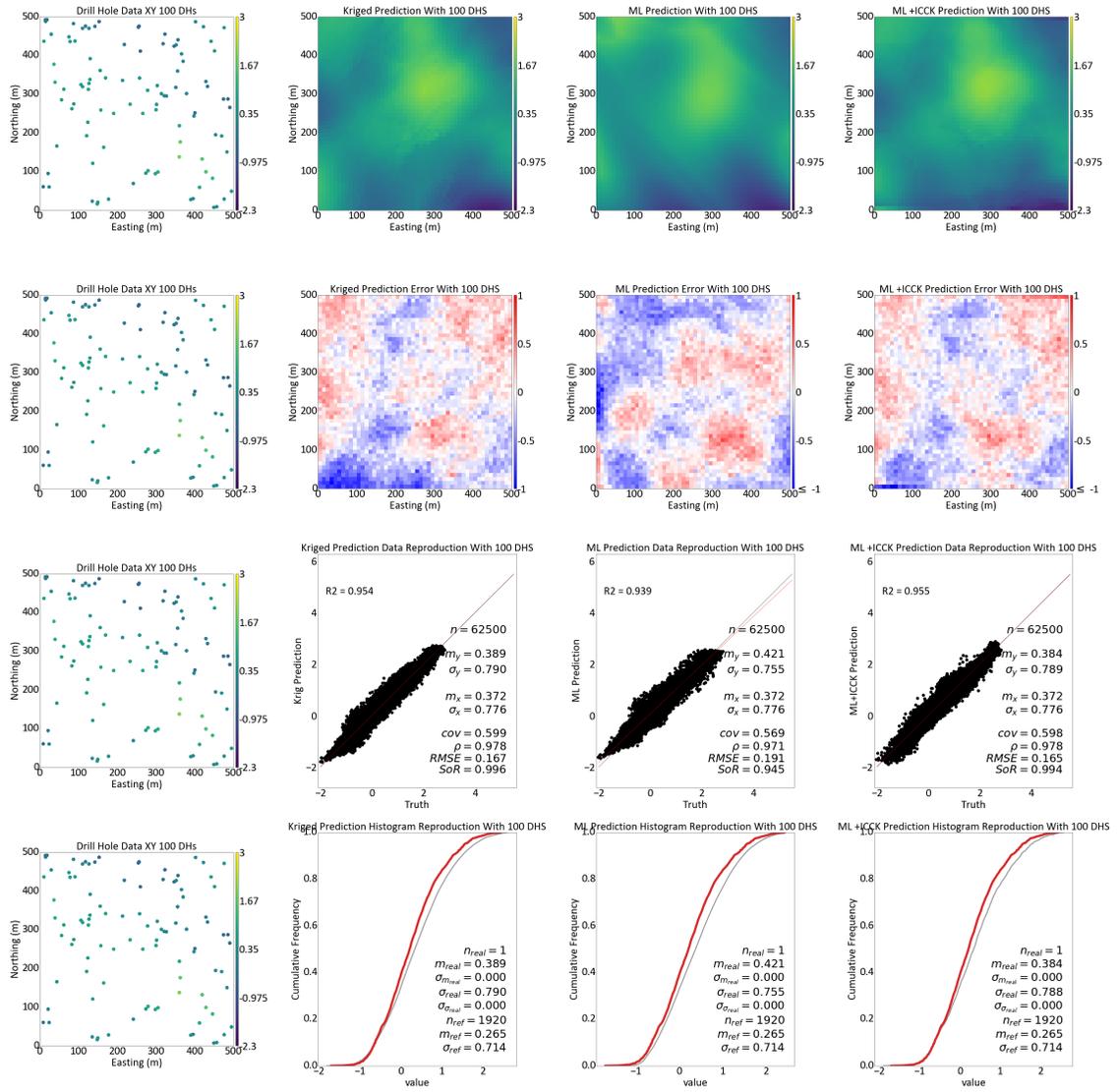


Figure 3.27: Sensitivity Study 100 Drill Holes

3. Proposed Estimation Methods

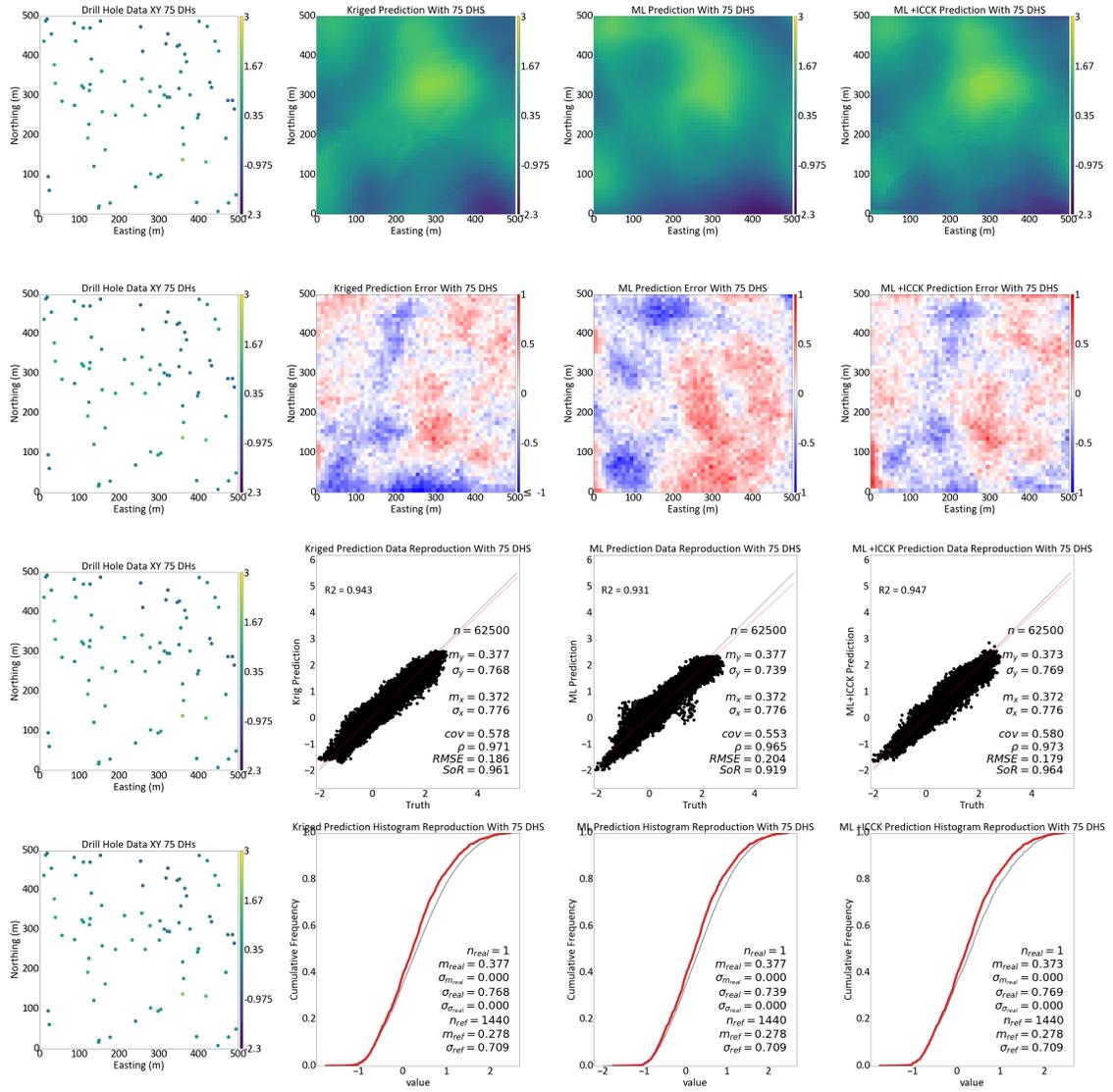


Figure 3.28: Sensitivity Study 75 Drill Holes

3. Proposed Estimation Methods

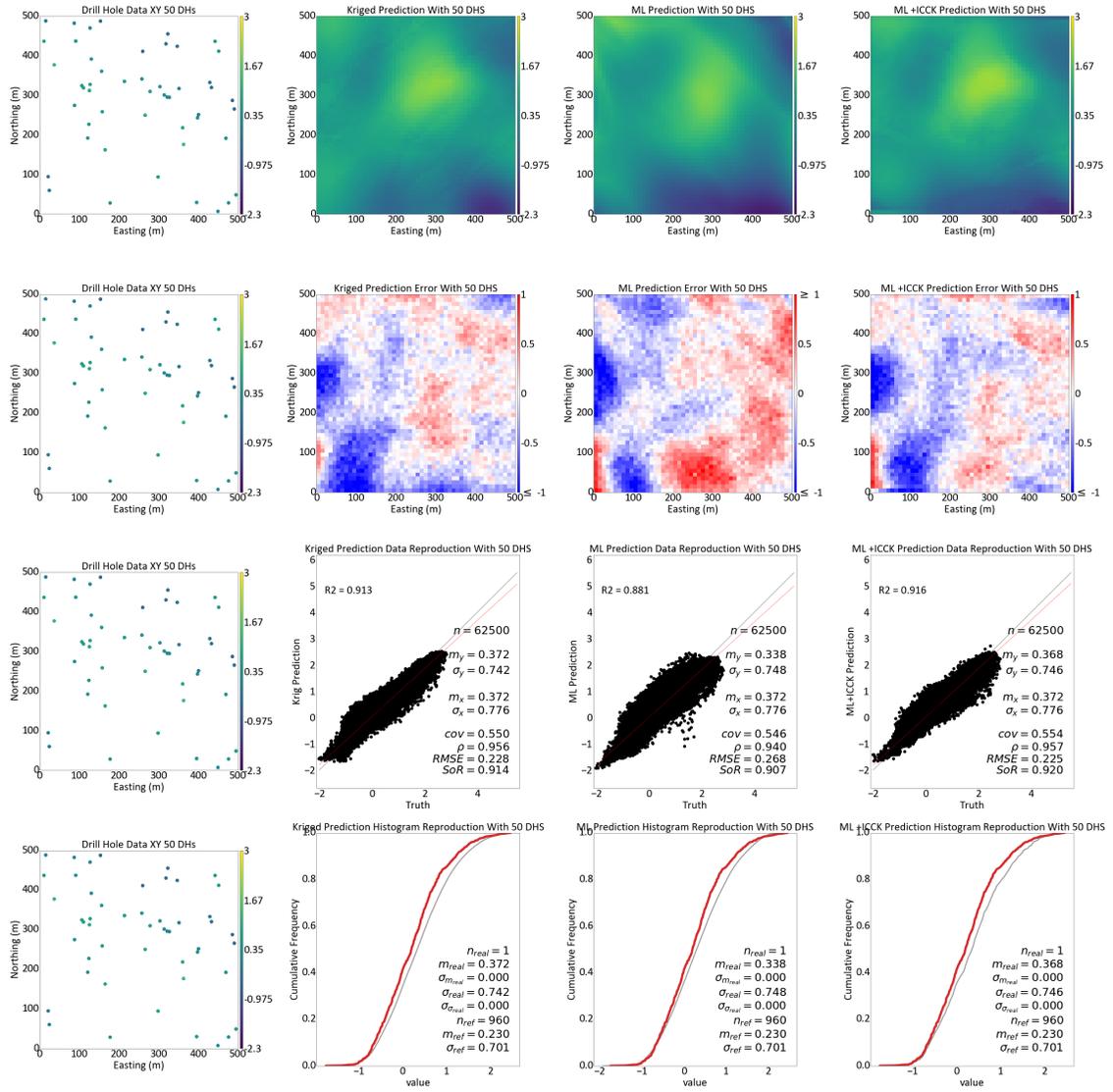


Figure 3.29: Sensitivity Study 50 Drill Holes

3. Proposed Estimation Methods

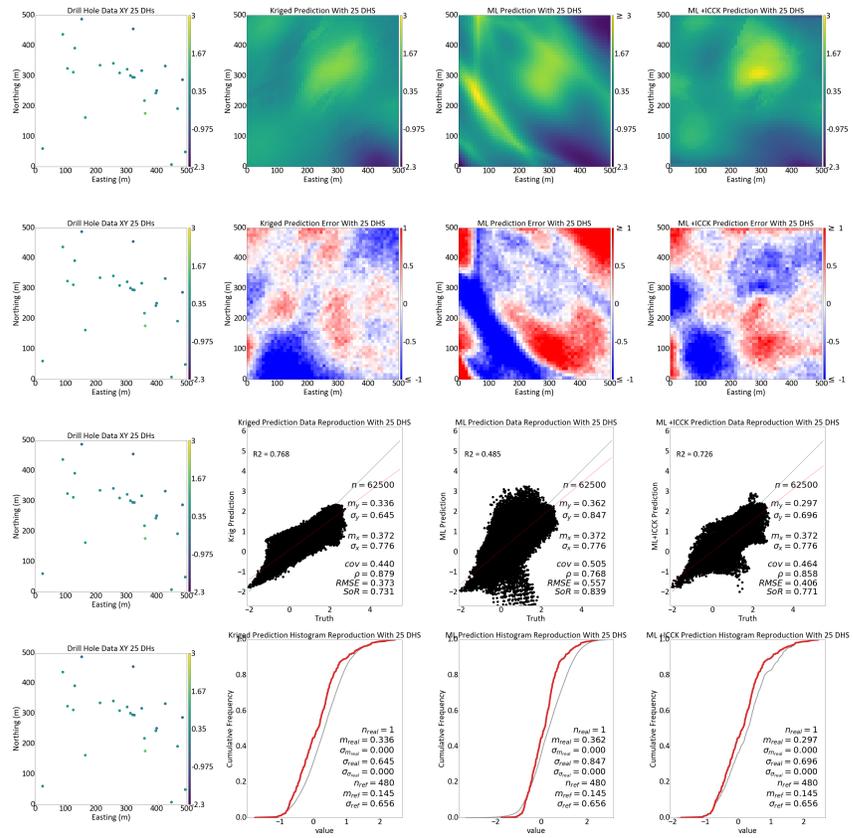


Figure 3.30: Sensitivity Study 25 Drill Holes

3. Proposed Estimation Methods

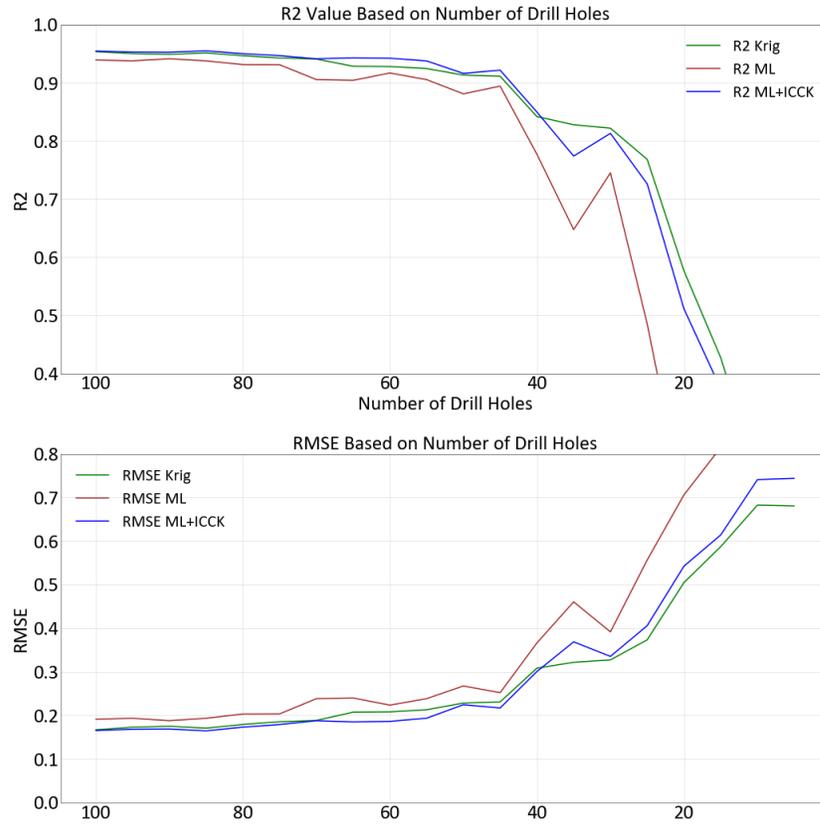


Figure 3.31: Summary Graphs of RMSE and R^2 Values for Number of Data Sensitivity Study

3. Proposed Estimation Methods

| Number of Data Sensitivity Analysis For Simple Kriging | | | | |
|--|-------------------|-----------------|-------|-------|
| Number of Drill Holes | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | 0.601 | 0.371 |
| 100 | 0.167 | 0.954 | 0.790 | 0.389 |
| 95 | 0.173 | 0.950 | 0.789 | 0.393 |
| 90 | 0.175 | 0.949 | 0.791 | 0.393 |
| 85 | 0.171 | 0.952 | 0.786 | 0.391 |
| 80 | 0.179 | 0.947 | 0.779 | 0.385 |
| 75 | 0.186 | 0.943 | 0.768 | 0.377 |
| 70 | 0.189 | 0.941 | 0.762 | 0.378 |
| 65 | 0.207 | 0.928 | 0.751 | 0.374 |
| 60 | 0.208 | 0.928 | 0.749 | 0.377 |
| 55 | 0.213 | 0.925 | 0.749 | 0.379 |
| 50 | 0.228 | 0.913 | 0.742 | 0.372 |
| 45 | 0.231 | 0.911 | 0.744 | 0.375 |
| 40 | 0.308 | 0.842 | 0.688 | 0.338 |
| 35 | 0.322 | 0.828 | 0.684 | 0.339 |
| 30 | 0.327 | 0.822 | 0.678 | 0.329 |
| 25 | 0.373 | 0.768 | 0.645 | 0.336 |
| 20 | 0.505 | 0.577 | 0.542 | 0.240 |
| 15 | 0.587 | 0.427 | 0.453 | 0.205 |
| 10 | 0.682 | 0.226 | 0.384 | 0.199 |
| 5 | 0.680 | 0.231 | 0.313 | 0.275 |

Table 3.12: Summary of Kriging Number of Data Sensitivity Analysis

| Number of Data Sensitivity Analysis Machine Learning | | | | |
|--|-------------------|-----------------|-------|--------|
| Number of Drill Holes | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | 0.601 | 0.371 |
| 100.000 | 0.191 | 0.939 | 0.755 | 0.421 |
| 95.000 | 0.193 | 0.938 | 0.765 | 0.380 |
| 90.000 | 0.188 | 0.941 | 0.765 | 0.396 |
| 85.000 | 0.194 | 0.938 | 0.770 | 0.382 |
| 80.000 | 0.203 | 0.931 | 0.766 | 0.387 |
| 75.000 | 0.204 | 0.931 | 0.739 | 0.377 |
| 70.000 | 0.238 | 0.906 | 0.756 | 0.363 |
| 65.000 | 0.240 | 0.904 | 0.757 | 0.337 |
| 60.000 | 0.224 | 0.917 | 0.757 | 0.363 |
| 55.000 | 0.238 | 0.905 | 0.752 | 0.352 |
| 50.000 | 0.268 | 0.881 | 0.748 | 0.338 |
| 45.000 | 0.252 | 0.894 | 0.737 | 0.392 |
| 40.000 | 0.366 | 0.777 | 0.751 | 0.362 |
| 35.000 | 0.461 | 0.648 | 0.788 | 0.271 |
| 30.000 | 0.392 | 0.745 | 0.784 | 0.378 |
| 25.000 | 0.557 | 0.485 | 0.847 | 0.362 |
| 20.000 | 0.706 | 0.172 | 0.814 | 0.256 |
| 15.000 | 0.815 | -0.103 | 0.735 | 0.129 |
| 10.000 | 1.206 | -1.417 | 0.957 | 0.021 |
| 5.000 | 1.205 | -1.414 | 0.819 | -0.190 |

Table 3.13: Summary of Kriging Number of Data Sensitivity Analysis

| Number of Data Sensitivity Analysis ML+ICCK | | | | |
|---|-------------------|-----------------|-------|-------|
| Number of Drill Holes | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | 0.601 | 0.371 |
| 100.000 | 0.165 | 0.955 | 0.789 | 0.384 |
| 95.000 | 0.168 | 0.953 | 0.788 | 0.380 |
| 90.000 | 0.169 | 0.953 | 0.790 | 0.381 |
| 85.000 | 0.165 | 0.955 | 0.780 | 0.388 |
| 80.000 | 0.173 | 0.950 | 0.775 | 0.382 |
| 75.000 | 0.179 | 0.947 | 0.769 | 0.373 |
| 70.000 | 0.188 | 0.941 | 0.768 | 0.382 |
| 65.000 | 0.185 | 0.943 | 0.767 | 0.377 |
| 60.000 | 0.186 | 0.942 | 0.763 | 0.377 |
| 55.000 | 0.194 | 0.938 | 0.764 | 0.381 |
| 50.000 | 0.225 | 0.916 | 0.746 | 0.368 |
| 45.000 | 0.217 | 0.922 | 0.756 | 0.372 |
| 40.000 | 0.301 | 0.850 | 0.706 | 0.344 |
| 35.000 | 0.369 | 0.774 | 0.726 | 0.280 |
| 30.000 | 0.335 | 0.813 | 0.743 | 0.325 |
| 25.000 | 0.406 | 0.726 | 0.696 | 0.297 |
| 20.000 | 0.542 | 0.511 | 0.619 | 0.224 |
| 15.000 | 0.614 | 0.374 | 0.557 | 0.191 |
| 10.000 | 0.741 | 0.088 | 0.510 | 0.149 |
| 5.000 | 0.744 | 0.080 | 0.440 | 0.197 |

Table 3.14: Summary of ML+ICCK Number of Data Sensitivity Analysis

From the results, the ICCK+ML and kriging estimations behave very similarly in terms of $RMSE$ and R^2 , with the machine learning being slightly inferior to the other two methods. Looking at mean reproduction, the SK, ML, and ML+ICCK all start to have poor reproduction when there are less than 40 drill holes. Similarly, the machine learning algorithm also begins to demonstrate artifacts below 40 drill holes.

ML+ICCK method behaves similarly to kriging in a Gaussian stationary system, the next examples explore the sensitivity of the ML+ICCK and kriging in a non-stationary example and non-Gaussian examples.

The next example looks at using the ML and ICCK method to predict locations where there is no conditioning data within the variogram range. With simple kriging, if there are no conditioning data within the variogram range, the estimation is set to the mean; however, this is likely not the correct solution. Using the ML estimate as a secondary data with ICCK results in a prediction that is similar to the ML prediction where there is no conditioning data, and not just estimates the mean. In Figure 3.32, a similar doughnut data set was simulated and sampled with drill holes missing on the left of the estimation area. An omnidirectional variogram that has a range of 17.5% of the Easting hence everything from zero to about two-hundred east is outside the range of the conditioning data.

Figure 3.33, shows kriging gives an estimate of the mean at a location outside of the range of conditioning data. The ML estimation results in almost a doughnut; however, the estimation has

artifacts. Looking at the hybrid method of estimation, the estimation is now artifact-free, and best resembles the simulated data set. Looking at Tables 3.15 and 3.16 it can be seen that the hybrid methods of estimation result in the best estimate in terms of minimizing root mean squared error and R^2 . The mean and distribution reproduction between the hybrid methods is similar to the kriging having a slight underestimation of 0.03 and the ML+ICCK having a slight over mean estimation of 0.04. In this scenario, it appears as if ML plus geostatistical hybrid method outperforms kriging.

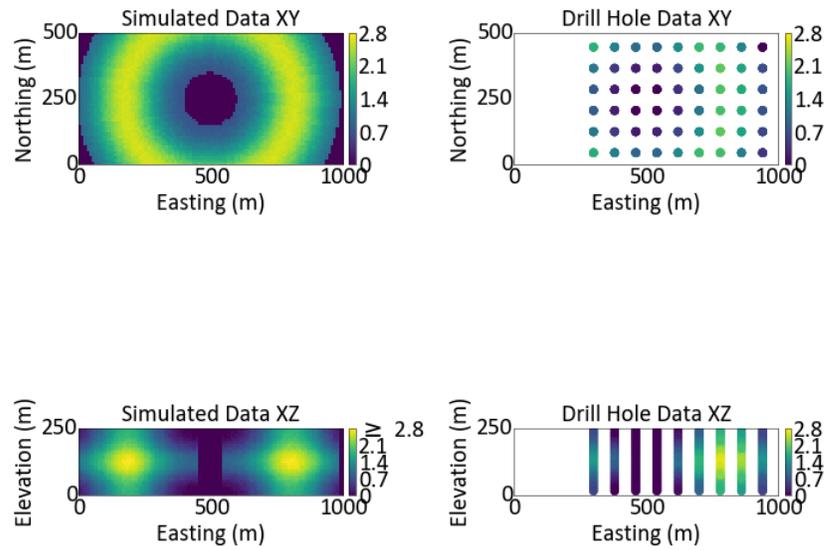


Figure 3.32: Doughnut Data Set With Missing Drill Holes

3. Proposed Estimation Methods

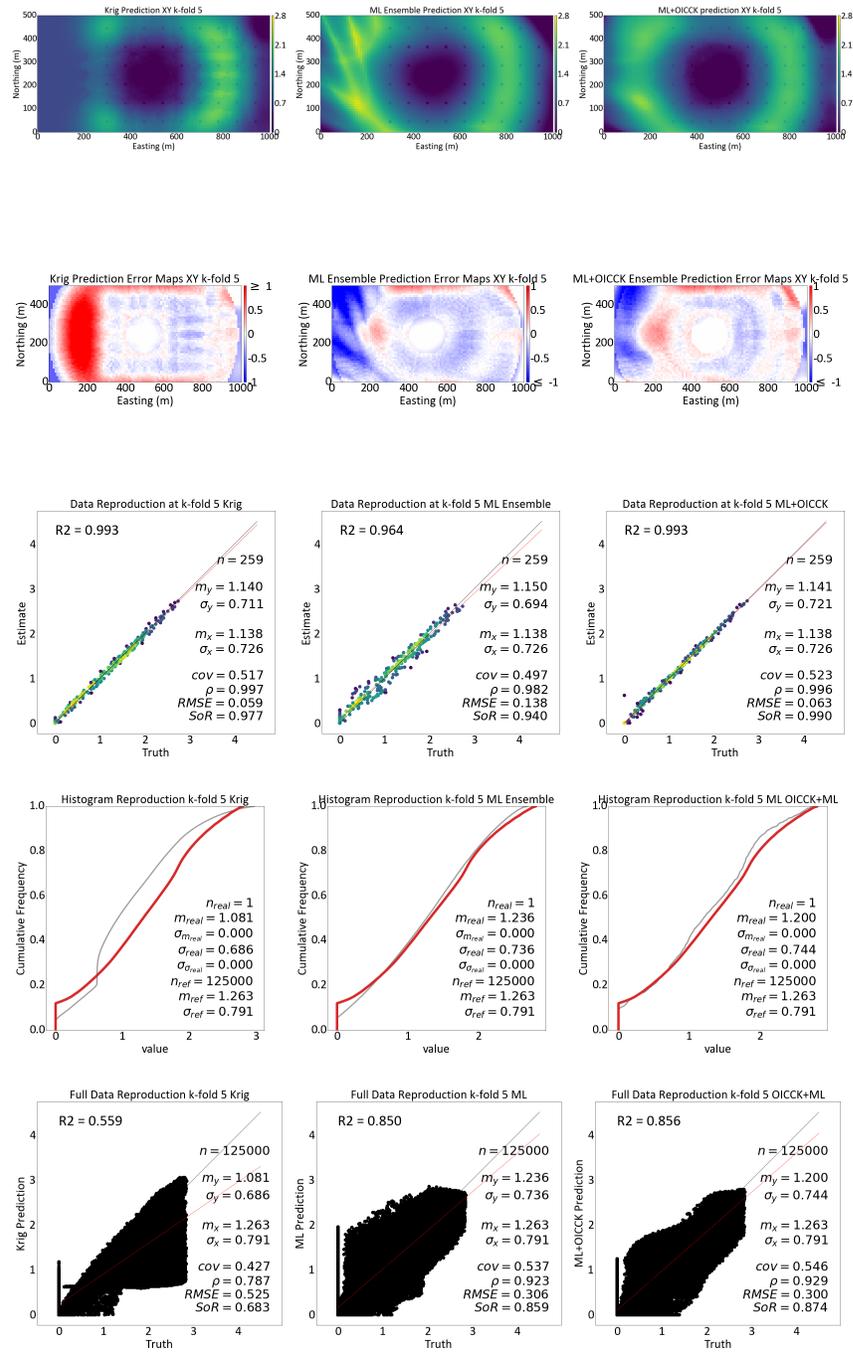


Figure 3.33: K-Fold Results From OICCK Plus ERFBN Using Doughnut Data Set Missing Drill Holes

3. Proposed Estimation Methods

| OICCK+ ML Using Doughnut Data Set Missing Left | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.75 | 1.112 |
| 1 | 0.058 | 0.994 | 0.533 | 0.545 | 0.769 | 1.034 |
| 2 | 0.061 | 0.993 | 0.325 | 0.831 | 0.742 | 1.189 |
| 3 | 0.060 | 0.994 | 0.385 | 0.763 | 0.750 | 1.147 |
| 4 | 0.052 | 0.995 | 0.332 | 0.823 | 0.776 | 1.197 |
| 5 | 0.063 | 0.993 | 0.300 | 0.856 | 0.744 | 1.200 |
| Average | 0.059 | 0.994 | 0.375 | 0.764 | 0.756 | 1.154 |

Table 3.15: Summary of OICCK Plus ERBFN Using Doughnut Data Set Missing Drill Holes

| Kriging Using Doughnut Data Set Missing Left | | | | | | |
|--|---------------|-------------|-------------------|-----------------|-------|-------|
| Fold | RMSE - K-Fold | R2 - K-Fold | RMSE - Full Model | R2 - Full Model | Sigma | Mean |
| Truth | N/A | N/A | N/A | N/A | 0.75 | 1.112 |
| 1 | 0.060 | 0.993 | 0.525 | 0.558 | 0.685 | 1.081 |
| 2 | 0.066 | 0.992 | 0.525 | 0.559 | 0.684 | 1.080 |
| 3 | 0.062 | 0.994 | 0.526 | 0.558 | 0.684 | 1.081 |
| 4 | 0.061 | 0.993 | 0.524 | 0.560 | 0.684 | 1.081 |
| 5 | 0.059 | 0.993 | 0.525 | 0.559 | 0.686 | 1.081 |
| Average | 0.062 | 0.993 | 0.525 | 0.559 | 0.685 | 1.081 |

Table 3.16: Summary of Kriging Using Doughnut Data Set Missing Drill Holes

Lognormal data sets are simulated, with a mean of one and standard deviations from 0.2 to 2 incrementing by 0.2. Each simulated data set is drilled and then used for estimation. Figure 3.34 shows the cumulative distribution function for each simulated data set with an example of a drilling pattern and simulated data set in Figure 3.35. The purpose of this study is not the variogram, the data set is drilled at a spacing that is 37.5% of the major direction variogram range and 50% of the minor direction variogram range. 6% of the variogram range, in the verticle direction, is used as the sample spacing.

An example estimation and visual check can be seen in Figures 3.36. Figure 3.37 shows a visual summary of the results. Tables 3.17, 3.18, and 3.19 shows a statistical summary of the results. From the tables and graphs, the machine learning and the ML+ICCK estimates perform similarly in terms of R^2 and $RMSE$. All estimates perform similarly in terms of $RMSE$; however, the R^2 when the standard deviation is low is best in the kriged estimate. As the data becomes more skewed the machine learning estimate and the ML+ICCK becomes a better estimation. The ML estimation becomes a better estimation at a standard deviation of about 1.80 and the ML+ICCK becomes a better estimate at a standard deviation of 1.60. It would be reasonable to say based on the trend of the R^2 and $RMSE$ that all three estimations would continually get worse as the distribution becomes more skewed; however, the ML estimate and the ML+ICCK estimate should remain a better estimate then kriging. As for mean reproduction, all three estimation techniques behave similarly and reproduce the mean reasonably. These results are reasonable as kriging is not optimal when the data are not Gaussian.

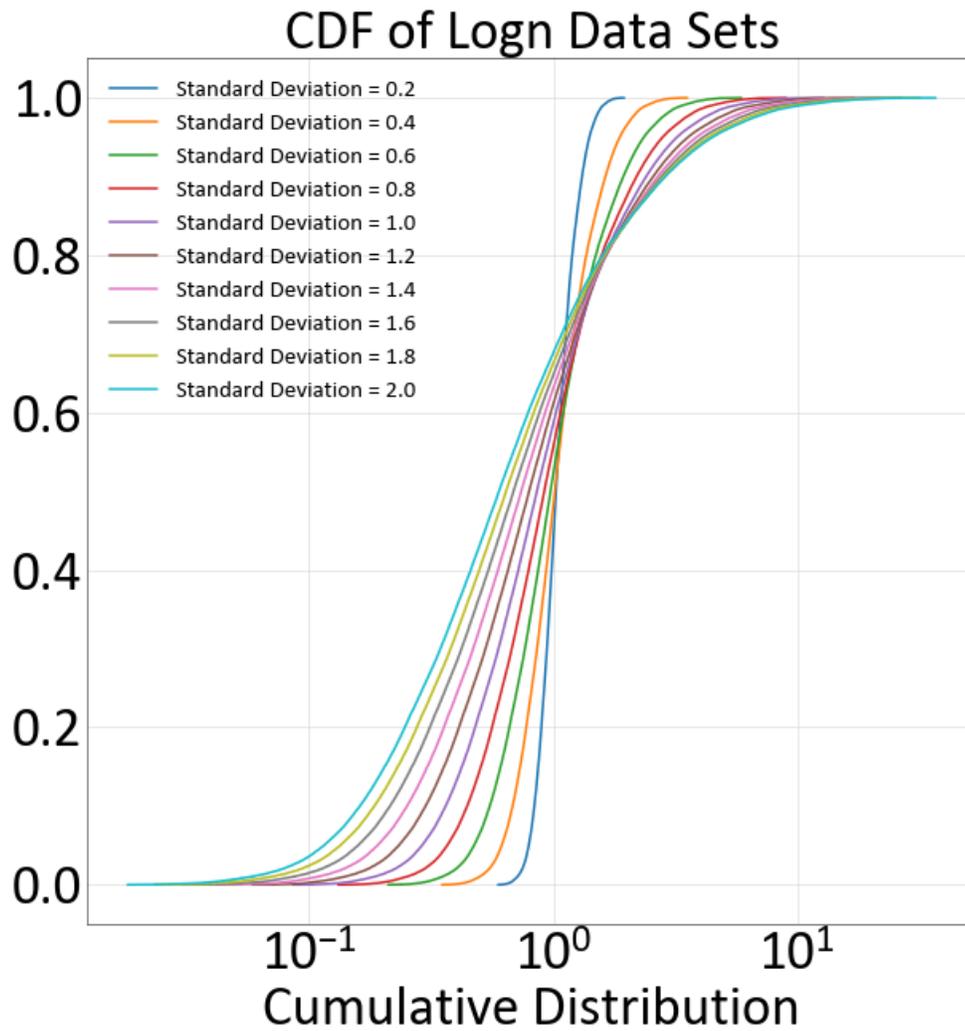


Figure 3.34: Cumulative Distribution Functions of Lognormal Data Sets

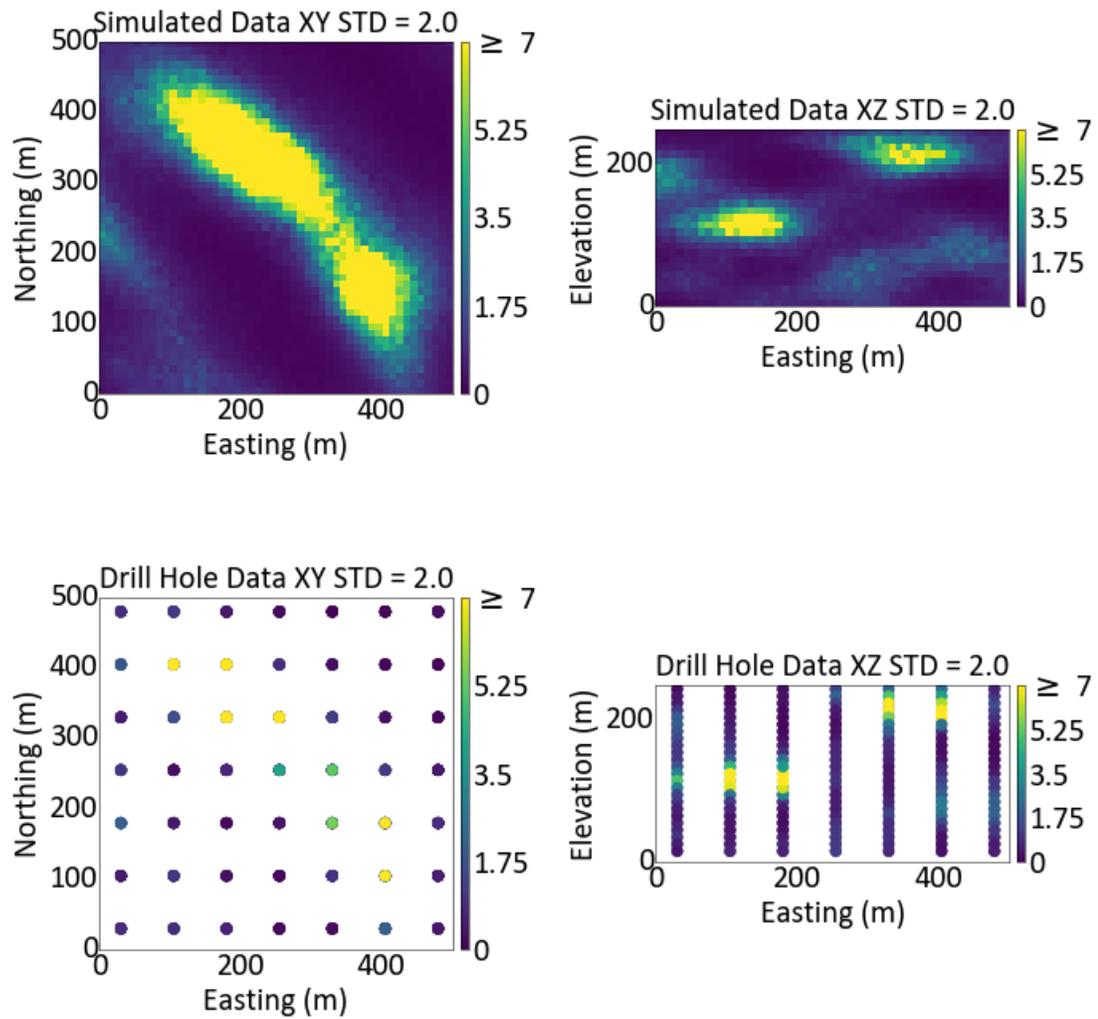


Figure 3.35: Example of Lognormal Data Set

3. Proposed Estimation Methods

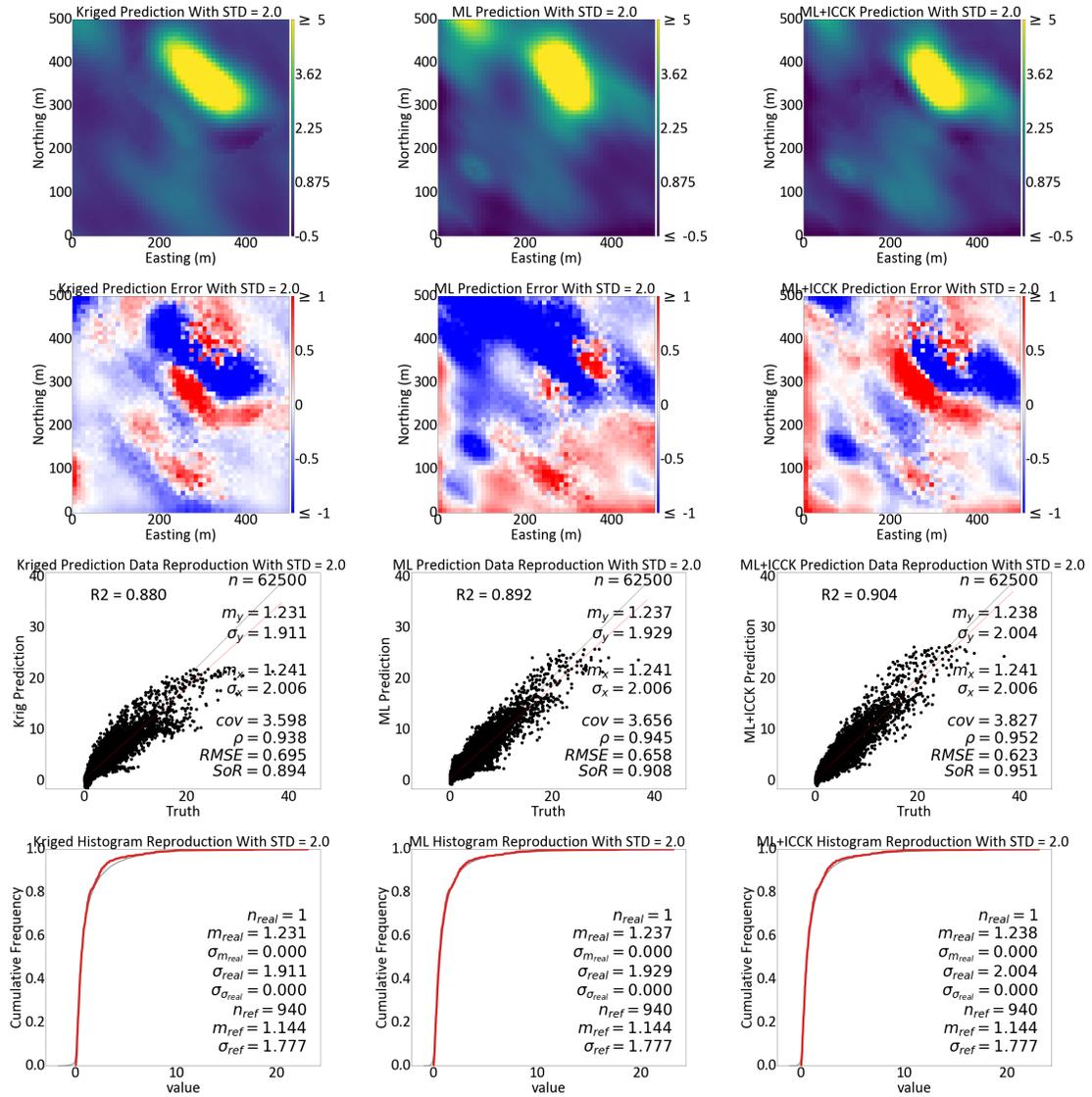


Figure 3.36: Example of Lognormal Estimation Standard Deviation = 2.0

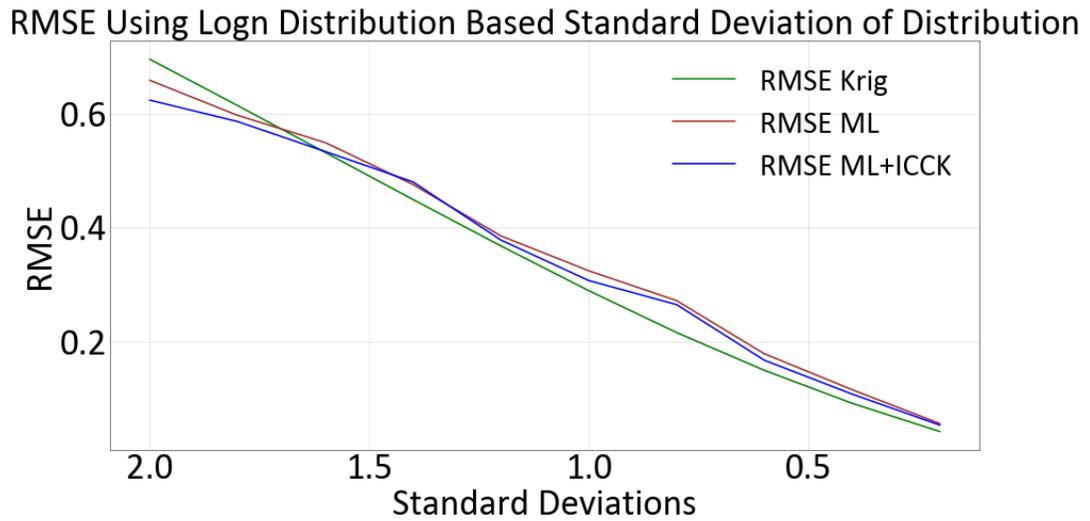
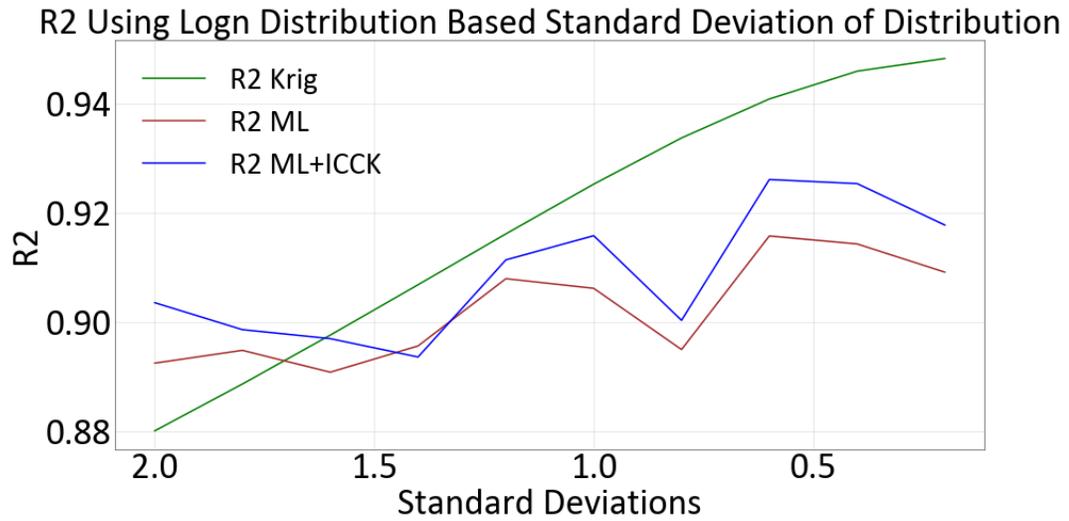


Figure 3.37: Lognormal Sensitivity Summary Graphs

| Lognormal Sensitivity Study - Kriging | | | | | |
|---------------------------------------|-------|-------------------|-----------------|------------|-------|
| Truth | | RMSE - Full Model | R2 - Full Model | Estimation | |
| Sigma | Mean | | | Sigma | Mean |
| 0.200 | 1.047 | 0.043 | 0.948 | 0.186 | 1.052 |
| 0.400 | 1.084 | 0.092 | 0.946 | 0.392 | 1.095 |
| 0.600 | 1.113 | 0.150 | 0.941 | 0.608 | 1.129 |
| 0.800 | 1.134 | 0.216 | 0.934 | 0.825 | 1.156 |
| 1.000 | 1.150 | 0.290 | 0.925 | 1.035 | 1.177 |
| 1.200 | 1.161 | 0.368 | 0.916 | 1.236 | 1.193 |
| 1.400 | 1.169 | 0.449 | 0.907 | 1.424 | 1.206 |
| 1.600 | 1.175 | 0.531 | 0.898 | 1.599 | 1.216 |
| 1.800 | 1.179 | 0.614 | 0.889 | 1.761 | 1.224 |
| 2.000 | 1.182 | 0.695 | 0.880 | 1.911 | 1.231 |

Table 3.17: Summary of Kriging Using Lognormal Data Sets

| Lognormal Sensitivity Study - Machine Learning | | | | | |
|--|-------|-------------------|-----------------|------------|-------|
| Truth | | RMSE - Full Model | R2 - Full Model | Estimation | |
| Sigma | Mean | | | Sigma | Mean |
| 0.200 | 1.048 | 0.054 | 0.918 | 0.186 | 1.056 |
| 0.400 | 1.085 | 0.108 | 0.925 | 0.390 | 1.101 |
| 0.600 | 1.100 | 0.168 | 0.926 | 0.603 | 1.127 |
| 0.800 | 1.154 | 0.265 | 0.900 | 0.837 | 1.147 |
| 1.000 | 1.137 | 0.307 | 0.916 | 1.014 | 1.172 |
| 1.200 | 1.186 | 0.378 | 0.911 | 1.278 | 1.202 |
| 1.400 | 1.168 | 0.480 | 0.894 | 1.513 | 1.224 |
| 1.600 | 1.168 | 0.533 | 0.897 | 1.583 | 1.216 |
| 1.800 | 1.189 | 0.586 | 0.899 | 1.736 | 1.211 |
| 2.000 | 1.180 | 0.623 | 0.904 | 2.004 | 1.238 |

Table 3.18: Summary of Machine Learning Using Lognormal Data Sets

| Lognormal Sensitivity Study - ML+ICCK | | | | | |
|---------------------------------------|-------|-------------------|-----------------|------------|-------|
| Truth | | RMSE - Full Model | R2 - Full Model | Estimation | |
| Sigma | Mean | | | Sigma | Mean |
| 0.200 | 1.048 | 0.057 | 0.909 | 0.181 | 1.049 |
| 0.400 | 1.085 | 0.116 | 0.914 | 0.382 | 1.105 |
| 0.600 | 1.100 | 0.179 | 0.916 | 0.587 | 1.113 |
| 0.800 | 1.154 | 0.272 | 0.895 | 0.819 | 1.164 |
| 1.000 | 1.137 | 0.324 | 0.906 | 0.986 | 1.194 |
| 1.200 | 1.186 | 0.385 | 0.908 | 1.241 | 1.230 |
| 1.400 | 1.168 | 0.475 | 0.896 | 1.455 | 1.209 |
| 1.600 | 1.168 | 0.549 | 0.891 | 1.531 | 1.193 |
| 1.800 | 1.189 | 0.596 | 0.895 | 1.719 | 1.201 |
| 2.000 | 1.180 | 0.658 | 0.892 | 1.929 | 1.237 |

Table 3.19: Summary of ML+ICCK Using Lognormal Data Sets

ML+ICCK estimation appears to work more effectively than kriging in non-Gaussian scenarios. The next step is to explore how the drill hole spacing to variogram range affects a non-Gaussian distributed scenario.

Multiple datasets are simulated from a lognormal distribution with a mean of one and a standard deviation of 2. Each dataset was simulated with a different variogram range and drilled at the same spacing to study how drill hole spacing to variogram range affects the estimation techniques. The data spacing to variogram ratio starts at 50% of the variogram range; the variogram range is then decreased uniformly until the ratio of data spacing to variogram range is equal to 150%. Figure 3.38 and 3.39 shows two examples of the datasets used.

3. Proposed Estimation Methods

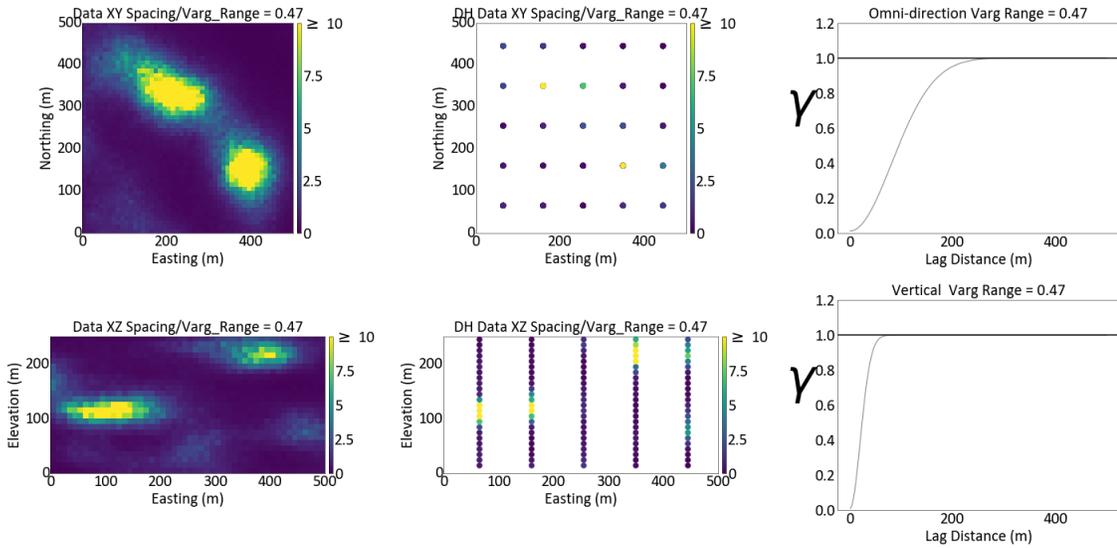


Figure 3.38: Example of Lognormal Data Set DH/Varg Range = 0.46

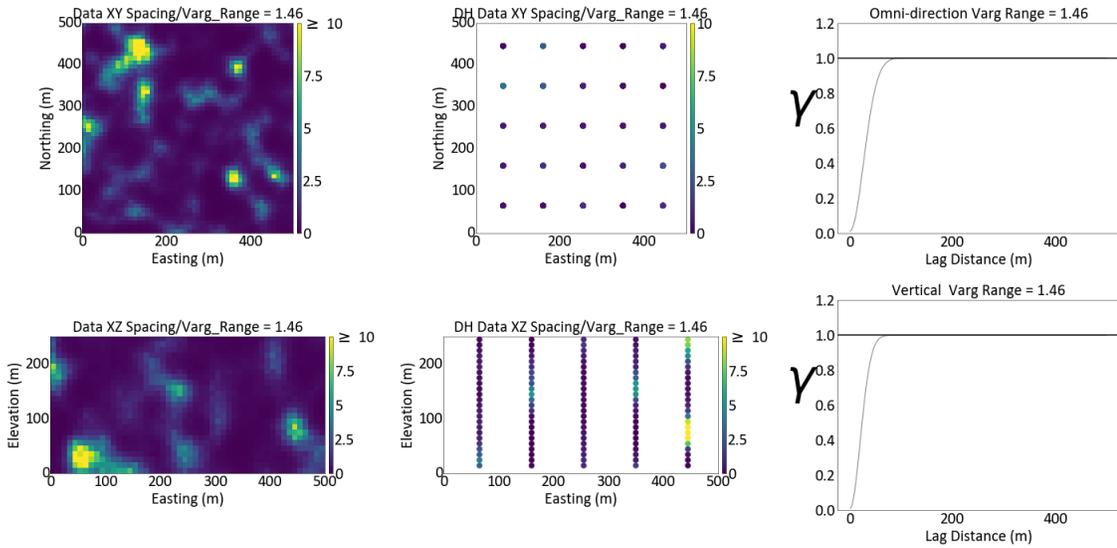


Figure 3.39: Example of Lognormal Data Set DH/Varg Range = 1.46

With each dataset, estimation is performed with the appropriate variograms. Figure 3.40 shows the results from the Drill Holes Spacing/Variogram Range = 0.46. Figure 3.41 shows a summary graph of the $RMSE$ and R^2 base on the Drill Holes Spacing/Variogram Range. Tables 3.20,3.21,3.22 show summary tables for all with all relevant statistics. From the results it appears as if based on $RMSE$ and R^2 that ML+ICCK method provides the better estimation until a drill hole spacing/variogram range of 110% is reached, this is likely due to the lack of data being used in the machine learning estimation causing issues in the ML+ICCK workflow. At a ratio of 100%, the estimation starts to have visual artifacts due to the drill hole spacing being greater than the variogram range.

3. Proposed Estimation Methods

As before, it appears as if the ML+ICCK method produces a better estimation and has the same sensitivity to drill hole spacing to variogram range as kriging.

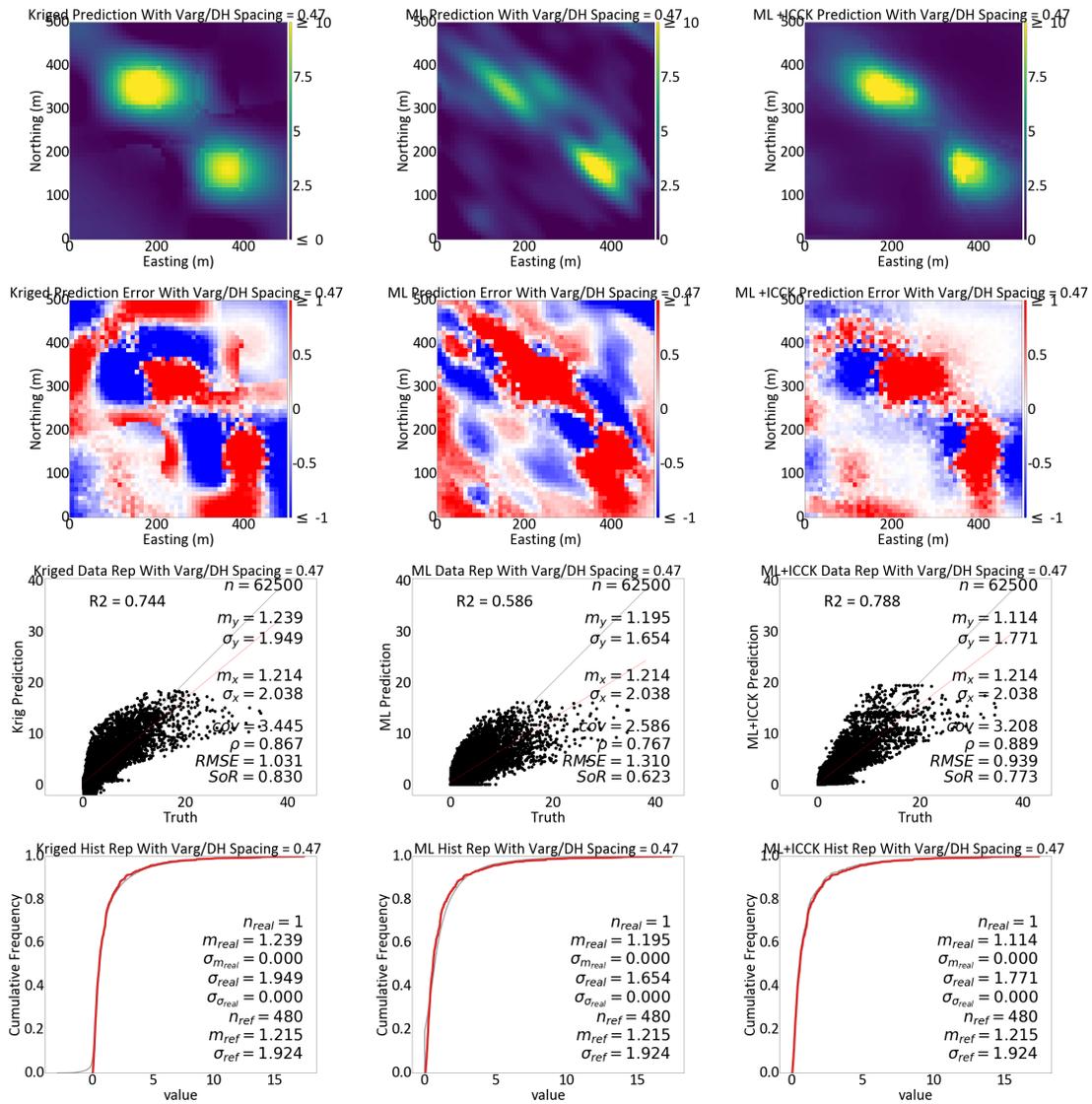
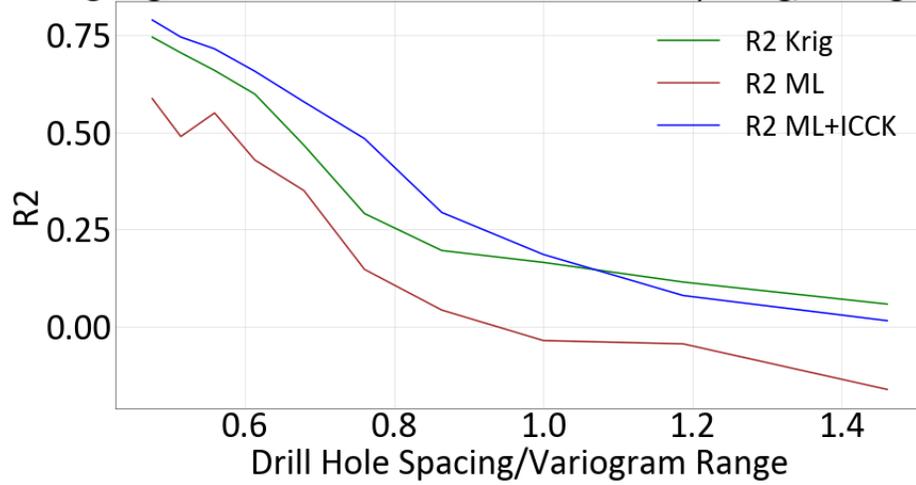


Figure 3.40: Example of Lognormal Estimation Drill Holes Spacing/Variogram Range = 0.46

R2 Using Logn Distribution Based On Drill Hole Spacing/Variogram Range



RMSE Using Logn Distribution Based On Drill Hole Spacing/Variogram Range

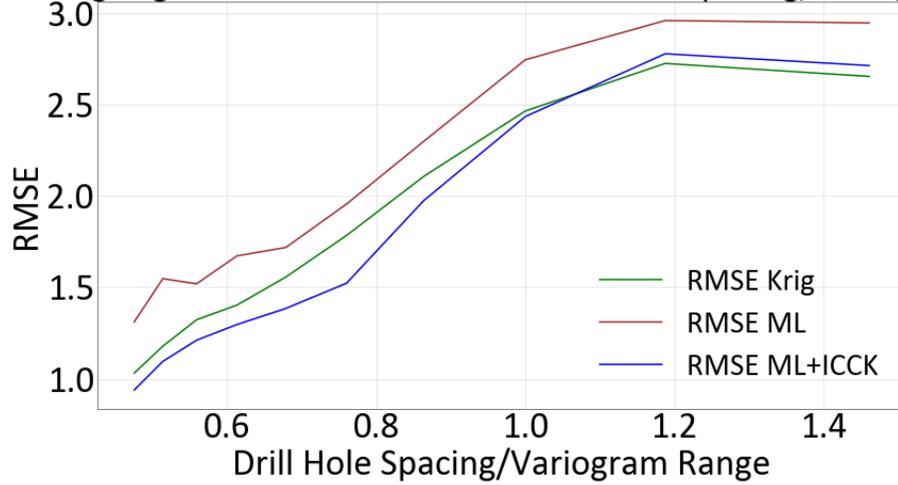


Figure 3.41: Summary Graph for Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study

3. Proposed Estimation Methods

| Drill Holes Spacing/Variogram Range Sensitivity Study - Kriging | | | | | | |
|---|-------|-------|-------------------|-----------------|------------|-------|
| Drill Hole Spacing/ Variogram Range | Truth | | RMSE - Full Model | R2 - Full Model | Estimation | |
| | Sigma | Mean | | | Sigma | Mean |
| 0.475 | 2.178 | 1.273 | 1.031 | 0.744 | 1.949 | 1.239 |
| 0.514 | 2.311 | 1.311 | 1.179 | 0.704 | 2.046 | 1.281 |
| 0.559 | 2.428 | 1.359 | 1.323 | 0.658 | 2.110 | 1.332 |
| 0.613 | 2.470 | 1.390 | 1.403 | 0.597 | 2.088 | 1.367 |
| 0.679 | 2.528 | 1.410 | 1.557 | 0.467 | 2.051 | 1.390 |
| 0.760 | 2.648 | 1.393 | 1.784 | 0.291 | 2.033 | 1.378 |
| 0.864 | 2.794 | 1.354 | 2.107 | 0.196 | 1.988 | 1.340 |
| 1.000 | 2.782 | 1.292 | 2.464 | 0.166 | 1.784 | 1.281 |
| 1.188 | 2.382 | 1.139 | 2.724 | 0.115 | 1.319 | 1.133 |
| 1.462 | 2.006 | 0.980 | 2.653 | 0.059 | 0.912 | 0.978 |

Table 3.20: Summary of Kriging Using Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study

| Drill Holes Spacing/Variogram Range Sensitivity Study - ML | | | | | | |
|--|-------|-------|-------------------|-----------------|------------|-------|
| Drill Hole Spacing/ Variogram Range | Truth | | RMSE - Full Model | R2 - Full Model | Estimation | |
| | Sigma | Mean | | | Sigma | Mean |
| 0.475 | 2.178 | 1.273 | 1.270 | 0.514 | 1.792 | 1.061 |
| 0.514 | 2.311 | 1.311 | 1.310 | 0.450 | 1.765 | 1.249 |
| 0.559 | 2.428 | 1.359 | 1.360 | 0.478 | 2.065 | 1.241 |
| 0.613 | 2.470 | 1.390 | 1.390 | 0.337 | 2.013 | 1.180 |
| 0.679 | 2.528 | 1.410 | 1.800 | 0.253 | 1.941 | 1.270 |
| 0.760 | 2.648 | 1.393 | 2.840 | - 0.050 | 2.181 | 1.159 |
| 0.864 | 2.794 | 1.354 | 2.170 | - 0.050 | 2.015 | 1.161 |
| 1.000 | 2.782 | 1.292 | 2.410 | - 0.077 | 2.076 | 1.332 |
| 1.188 | 2.382 | 1.139 | 3.150 | - 0.182 | 2.059 | 1.020 |
| 1.462 | 2.006 | 0.980 | 3.093 | - 0.279 | 1.750 | 0.774 |

Table 3.21: Summary of ML Using Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study

| Drill Holes Spacing/Variogram Range Sensitivity Study - ML+ICCK | | | | | | |
|---|-------|-------|-------------------|-----------------|------------|-------|
| Drill Hole Spacing/ Variogram Range | Truth | | RMSE - Full Model | R2 - Full Model | Estimation | |
| | Sigma | Mean | | | Sigma | Mean |
| 0.475 | 2.178 | 1.273 | 0.939 | 0.788 | 1.771 | 1.114 |
| 0.514 | 2.311 | 1.311 | 1.096 | 0.744 | 1.850 | 1.203 |
| 0.559 | 2.428 | 1.359 | 1.213 | 0.713 | 1.965 | 1.178 |
| 0.613 | 2.470 | 1.390 | 1.297 | 0.656 | 1.975 | 1.225 |
| 0.679 | 2.528 | 1.410 | 1.386 | 0.577 | 1.733 | 1.157 |
| 0.760 | 2.648 | 1.393 | 1.523 | 0.483 | 1.582 | 1.060 |
| 0.864 | 2.794 | 1.354 | 1.976 | 0.293 | 1.536 | 1.019 |
| 1.000 | 2.782 | 1.292 | 2.433 | 0.187 | 1.286 | 0.876 |
| 1.188 | 2.382 | 1.139 | 2.777 | 0.081 | 0.974 | 0.718 |
| 1.462 | 2.006 | 0.980 | 2.713 | 0.016 | 0.708 | 0.605 |

Table 3.22: Summary of ML+ICCK Using Lognormal Drill Holes Spacing/Variogram Range Sensitivity Study

This sensitivity study explores data with a trend. The data set initially has a major variogram

3. Proposed Estimation Methods

that is 2.75 times larger than the minor, the major to minor range ratio is then increased until the ratio is equal to 5.25 times. Each dataset is drilled the same. The minor and down-hole variogram range is set to the truth, so that variogram is modelled only in the major direction. Using the truth minimizes error introduced in the variogram modelling step, and more effectively demonstrates the difficulty modelling trends.

Figure 3.42 shows the simulated data, drilled data, and variograms can be seen for the 2.75 major/minor variogram range and in Figure 3.43 the 5.25 major/minor variogram range.

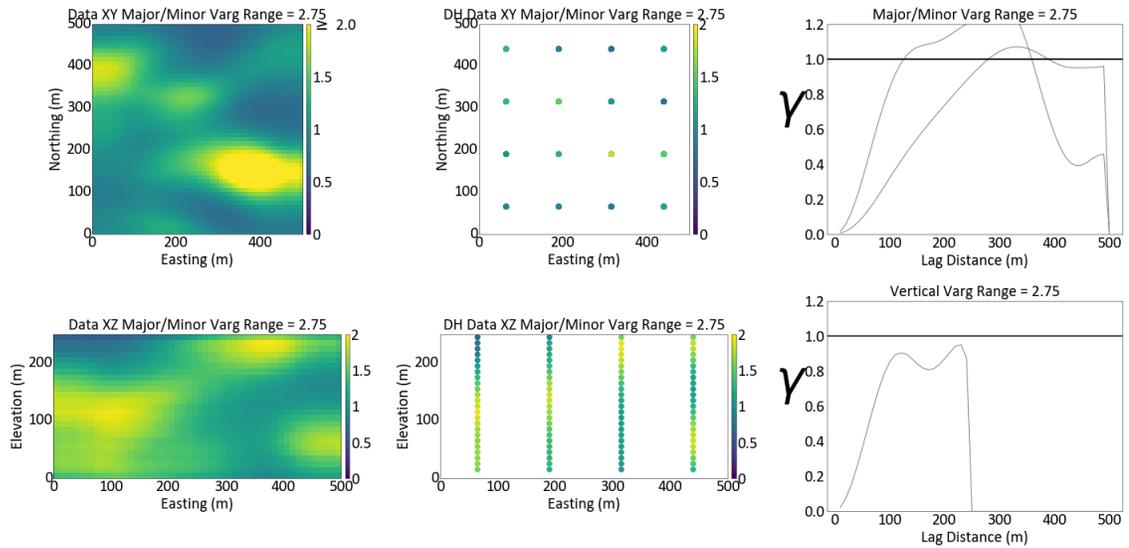


Figure 3.42: Example of Simulated and Drilled Data For Major/Minor Vargiogram Range Sensitivity Study = 2.75

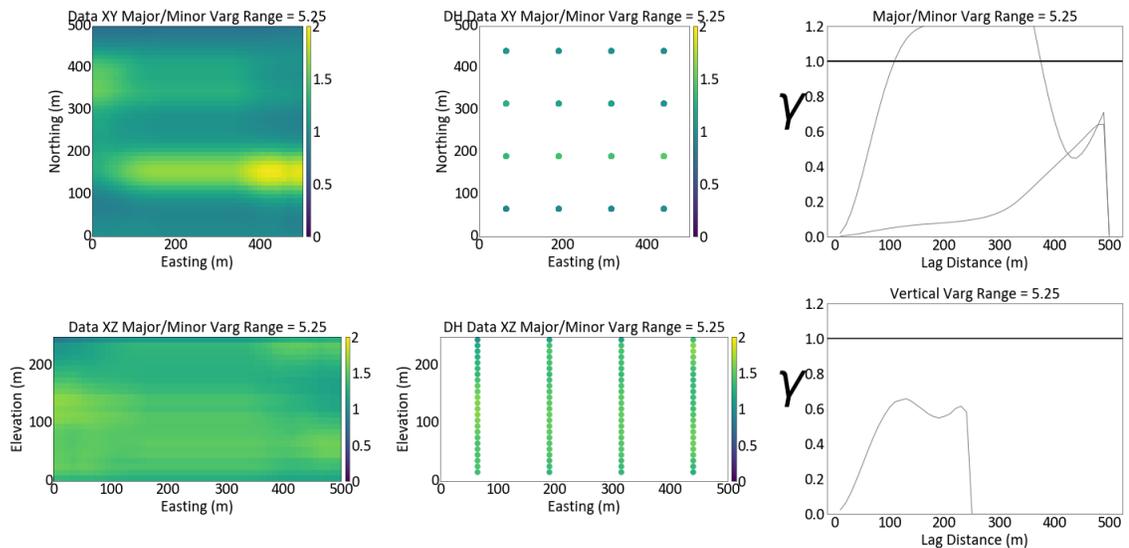


Figure 3.43: Example of Simulated and Drilled Data For Major/Minor Vargiogram Range Sensitivity Study = 5.25

3. Proposed Estimation Methods

Figure 3.44 shows the results with the largest trend. From the results, all methods accurately reproduce the mean; however, the distribution of data is reproduced appears best in the machine learning methods.

In terms of $RMSE$ and R^2 the machine learning method outperforms kriging with the hybrid method doing the best in all cases except where the major to minor variogram ratio is only 2.75 (Figure 3.45). A summary of all relevant statistics can also be seen in Tables 3.23, 3.24, and 3.25. From the results it would be reasonable to say that the machine learning and machine learning plus intrinsic collocated cokriging methods outperform kriging.

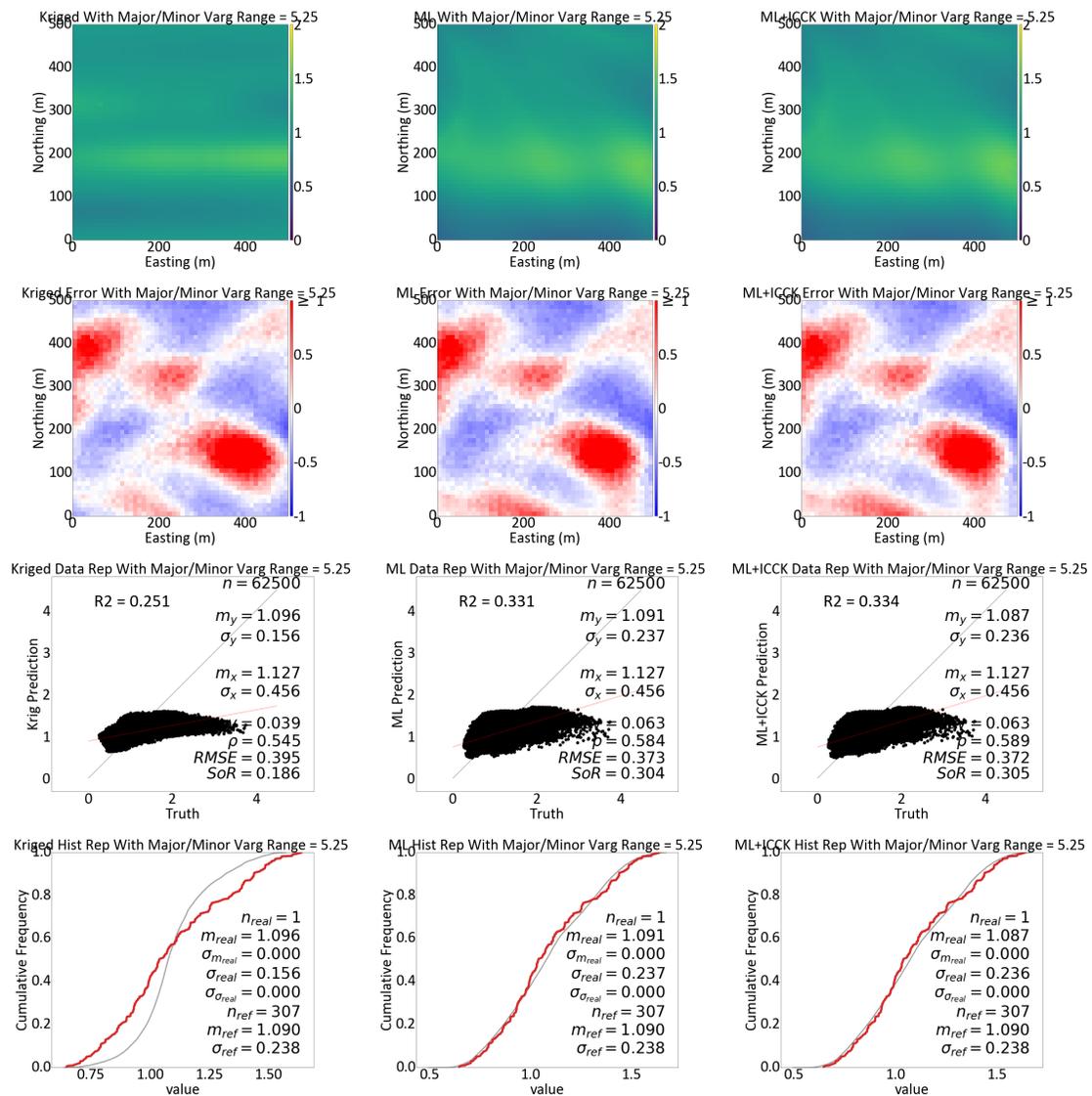


Figure 3.44: Example of Major/Minor Variogram Range Sensitivity Study = 5.25

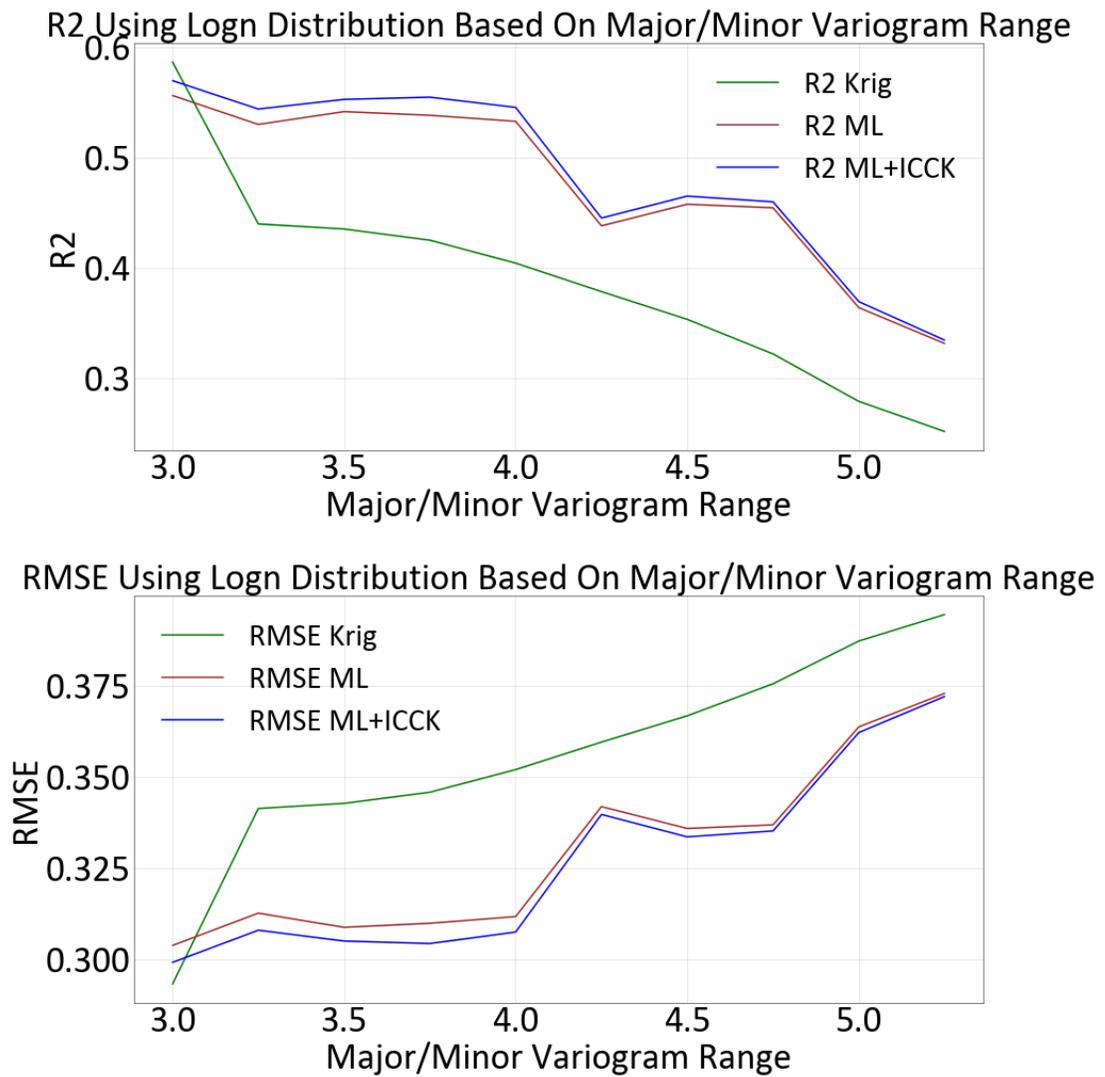


Figure 3.45: Summary Graph for Increasing Major/Minor Variogram Range Sensitivity Study

3. Proposed Estimation Methods

| Summary of Increasing Major/Minor Vargiogram Range Sensitivity Study - Kriging | | | | | | |
|--|-------|-------|-------------------|-----------------|------------|-------|
| Truth | | | RMSE - Full Model | R2 - Full Model | Estimation | |
| Major/Minor Varg Range | Sigma | Mean | | | Sigma | Mean |
| 2.75 | 0.121 | 1.091 | 0.293 | 0.587 | 0.276 | 1.107 |
| 3.00 | 0.110 | 1.080 | 0.341 | 0.440 | 0.216 | 1.088 |
| 3.25 | 0.107 | 1.080 | 0.343 | 0.435 | 0.213 | 1.089 |
| 3.50 | 0.101 | 1.082 | 0.346 | 0.425 | 0.208 | 1.091 |
| 3.75 | 0.091 | 1.086 | 0.352 | 0.404 | 0.199 | 1.095 |
| 4.00 | 0.082 | 1.088 | 0.360 | 0.378 | 0.190 | 1.097 |
| 4.50 | 0.075 | 1.089 | 0.367 | 0.353 | 0.180 | 1.098 |
| 4.75 | 0.068 | 1.089 | 0.376 | 0.322 | 0.172 | 1.097 |
| 5.00 | 0.060 | 1.089 | 0.387 | 0.279 | 0.162 | 1.097 |
| 5.25 | 0.056 | 1.089 | 0.395 | 0.251 | 0.156 | 1.096 |

Table 3.23: Summary of Kriging Increasing Major/Minor Vargiogram Range Sensitivity Study

| Summary of Increasing Major/Minor Vargiogram Range Sensitivity Study - ML | | | | | | |
|---|-------|-------|-------------------|-----------------|------------|-------|
| Truth | | | RMSE - Full Model | R2 - Full Model | Estimation | |
| Major/Minor Varg Range | Sigma | Mean | | | Sigma | Mean |
| 2.750 | 0.123 | 1.093 | 0.304 | 0.557 | 0.353 | 1.092 |
| 3.000 | 0.110 | 1.079 | 0.313 | 0.530 | 0.327 | 1.080 |
| 3.250 | 0.103 | 1.072 | 0.309 | 0.542 | 0.318 | 1.070 |
| 3.500 | 0.100 | 1.080 | 0.310 | 0.539 | 0.312 | 1.087 |
| 3.750 | 0.090 | 1.080 | 0.312 | 0.533 | 0.300 | 1.076 |
| 4.000 | 0.084 | 1.086 | 0.342 | 0.438 | 0.283 | 1.087 |
| 4.500 | 0.074 | 1.091 | 0.336 | 0.458 | 0.267 | 1.081 |
| 4.750 | 0.068 | 1.089 | 0.337 | 0.454 | 0.258 | 1.090 |
| 5.000 | 0.059 | 1.092 | 0.364 | 0.364 | 0.237 | 1.089 |
| 5.250 | 0.056 | 1.090 | 0.373 | 0.331 | 0.237 | 1.091 |

Table 3.24: Summary of ML Increasing Major/Minor Vargiogram Range Sensitivity Study

| Summary of Increasing Major/Minor Vargiogram Range Sensitivity Study - ML+ICCK | | | | | | |
|--|-------|-------|-------------------|-----------------|------------|-------|
| Truth | | | RMSE - Full Model | R2 - Full Model | Estimation | |
| Major/Minor Varg Range | Sigma | Mean | | | Sigma | Mean |
| 2.750 | 0.123 | 1.093 | 0.299 | 0.570 | 0.347 | 1.094 |
| 3.000 | 0.110 | 1.079 | 0.308 | 0.544 | 0.328 | 1.076 |
| 3.250 | 0.103 | 1.072 | 0.305 | 0.553 | 0.319 | 1.067 |
| 3.500 | 0.100 | 1.080 | 0.304 | 0.555 | 0.309 | 1.088 |
| 3.750 | 0.090 | 1.080 | 0.307 | 0.546 | 0.296 | 1.078 |
| 4.000 | 0.084 | 1.086 | 0.340 | 0.445 | 0.281 | 1.079 |
| 4.500 | 0.074 | 1.091 | 0.334 | 0.465 | 0.265 | 1.080 |
| 4.750 | 0.068 | 1.089 | 0.335 | 0.460 | 0.259 | 1.088 |
| 5.000 | 0.059 | 1.092 | 0.362 | 0.369 | 0.235 | 1.091 |
| 5.250 | 0.056 | 1.090 | 0.372 | 0.334 | 0.236 | 1.087 |

Table 3.25: Summary of ML+ICCK Increasing Major/Minor Vargiogram Range Sensitivity Study

The final important sensitivity study is run time. Figures 3.46 and 3.47 show the run time for

training and prediction. From the runtime graphs the only thing that effects run time is the number of nodes and input features, conceptually this makes sense. Increasing the number of training data, and prediction locations does not change the number of equations to solve just the matrix size. From the examples demonstrated five batches where used to training the data and the prediction grid was split 50 times due to memory issues. Theoretically the prediction time could be sped up 50x and training time could be sped up 5x with more memory. For these examples a NVIDIA GeForce GTX 760 was used.

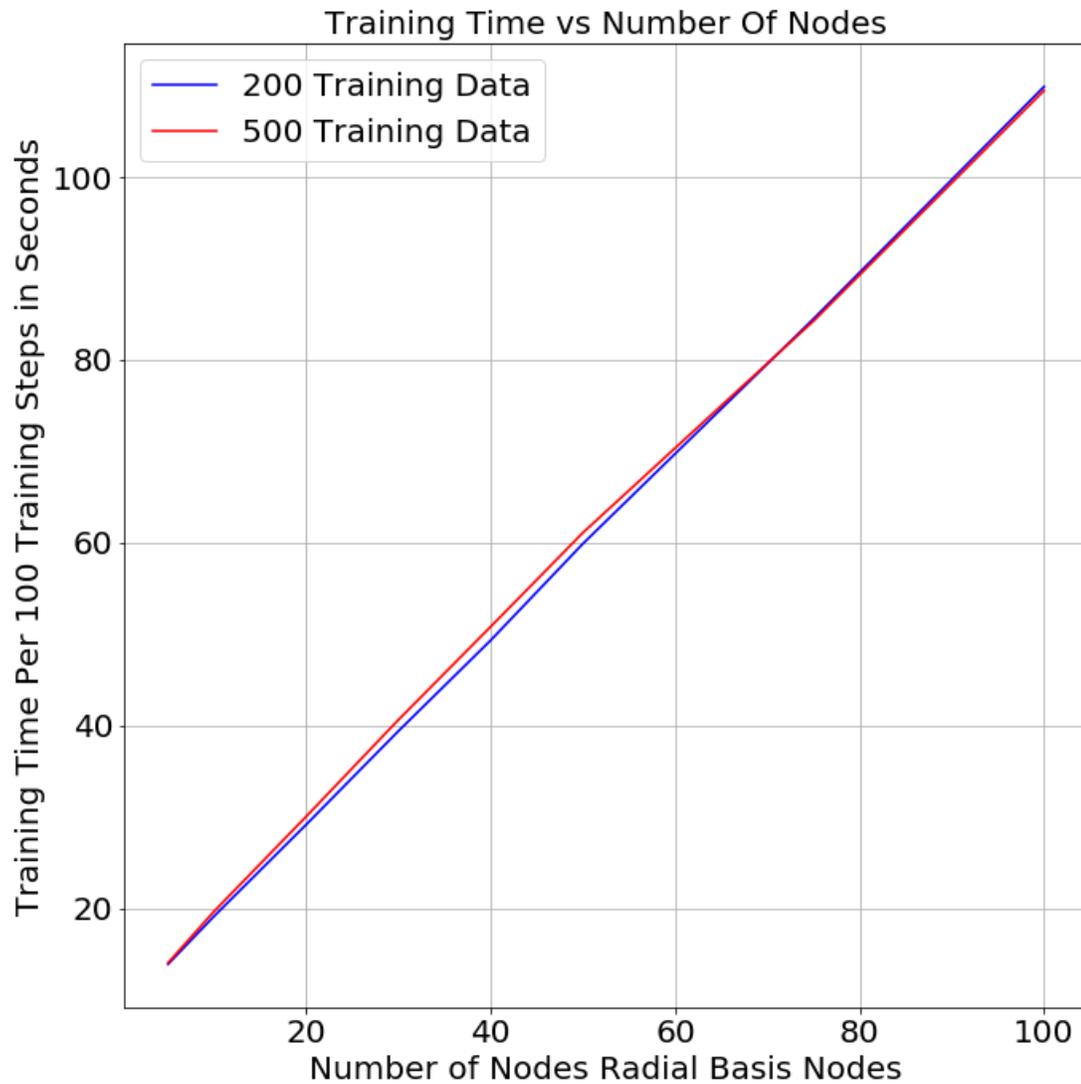


Figure 3.46: Run time vs Number of Nodes for 100 Training Steps

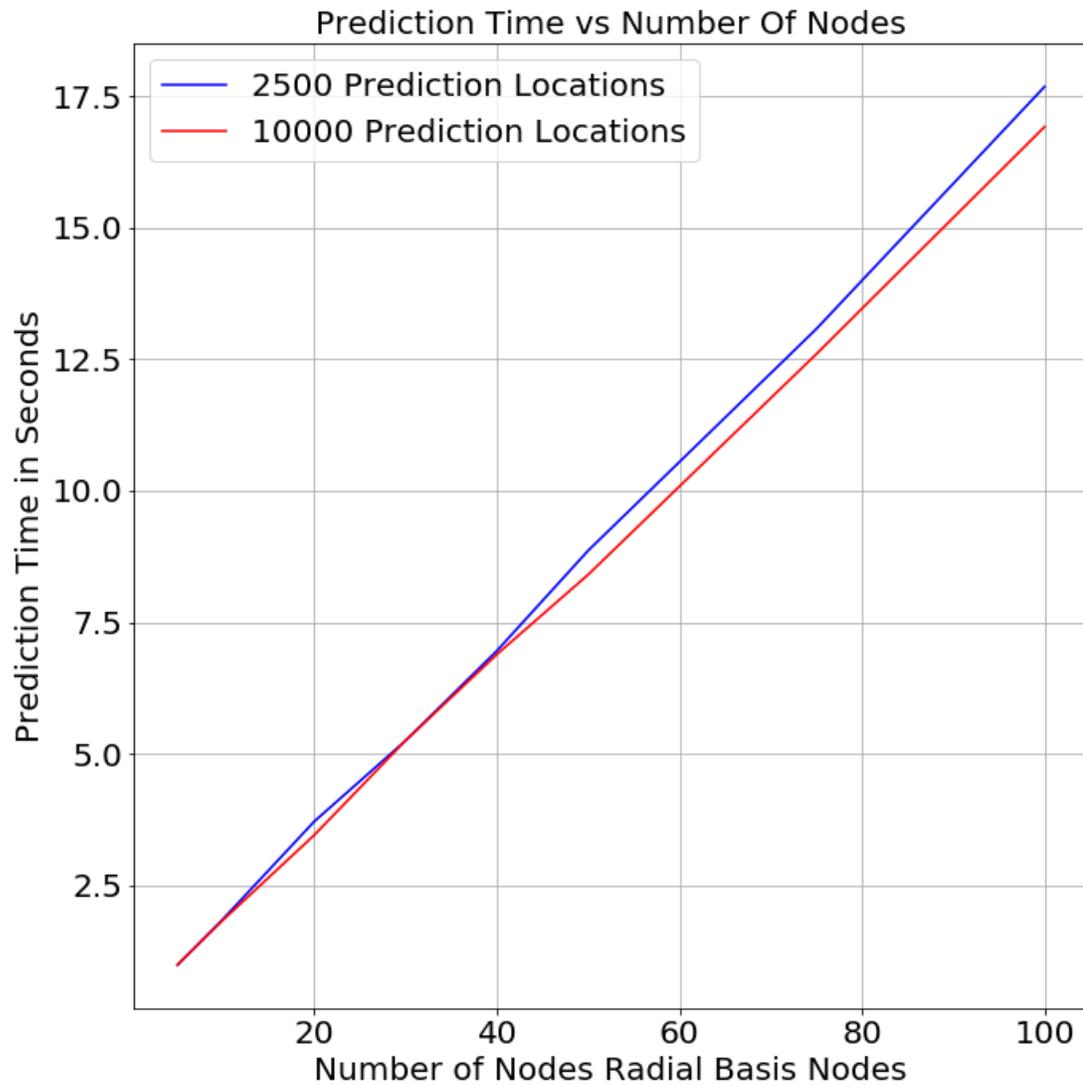


Figure 3.47: Run Time vs Number of Nodes for Prediction (Note: A small time discrepancy between the two examples is likely due to a slight time deviation in computer overhead)

Using simulated data to test new estimations method is an useful way to prove the validity of the method; however, the best way to prove the validity of an estimation technique is with real data. The next Chapter focuses on exploring a real dataset.

CHAPTER 4

JA EXTENSION COPPER PORPHYRY DEPOSIT CASE STUDY

The focus of Chapter 4 is on the practical application of the proposed estimation techniques in Chapter 3 using the JA extension copper-molybdenum porphyry deposit. Simple kriged estimates are generated then compared to the machine learning and hybrid estimates with k-fold validation. The comparison between estimates done with four-folds. K-fold 0 is the average of all estimates and is considered the final estimate. The goal of this study is to review the three estimation methods in a practical setting and provide modelling recommendations based on the results.

4.1 Available Data

The JA extension copper-molybdenum porphyry deposit is located near the Highland Valley in British Columbia and consists of 136 drill holes with 3245 copper samples. The estimation domain in the X-direction ranges from 34400-36275, in the Y-Direction ranges from 27400-28600, in the Z-direction ranges from 600-1200. Figure 4.1 is a location map of the available data in an XY(left) and XZ(right) slices. The ore is deposited along veins, fracture and faults similar to many porphyry deposits (Daniels, 2015). The copper (Cu) variable is a mass percentage. More information on the JA extension copper-molybdenum porphyry deposit and the geological setting is available in Northcote (2019). This case study will focus on the copper variable. From the Location maps and data spacing CDF, it is evident that the data spacing is fairly regular in the low-grade areas and densely sampled in the high-grade areas. The drill holes have a composite length of 10m.

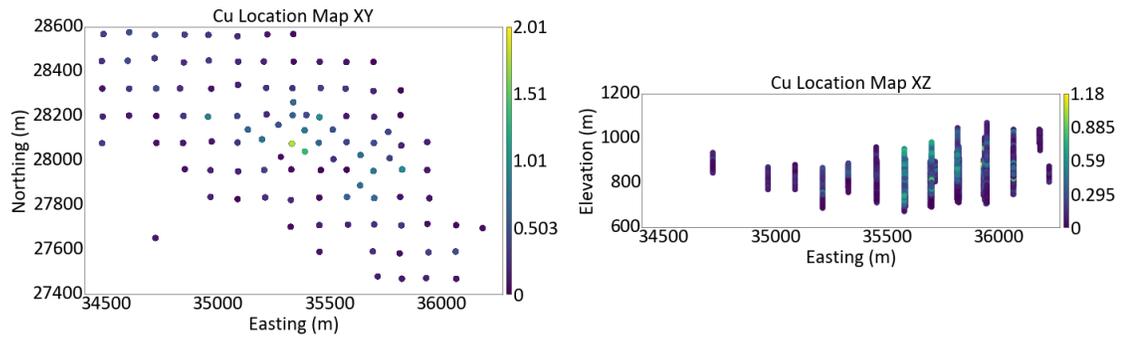


Figure 4.1: JA extension Copper Porphyry Deposit Location Map of Cu Data XY view, left, XZ, right

4.2 Modelling Parameters

The data is not evenly spaced and must be de-clustered before calculating reference statistics. Figure 4.2 is a de-clustered mean size vs cell size(left) and a probability density function of the data spacing (right). The de-clustered cell size is 250 units, the 95th percentile (Silva, 2018). The average data spacing is 191m, 10.2% of the X direction and 15.8% of the Y direction.

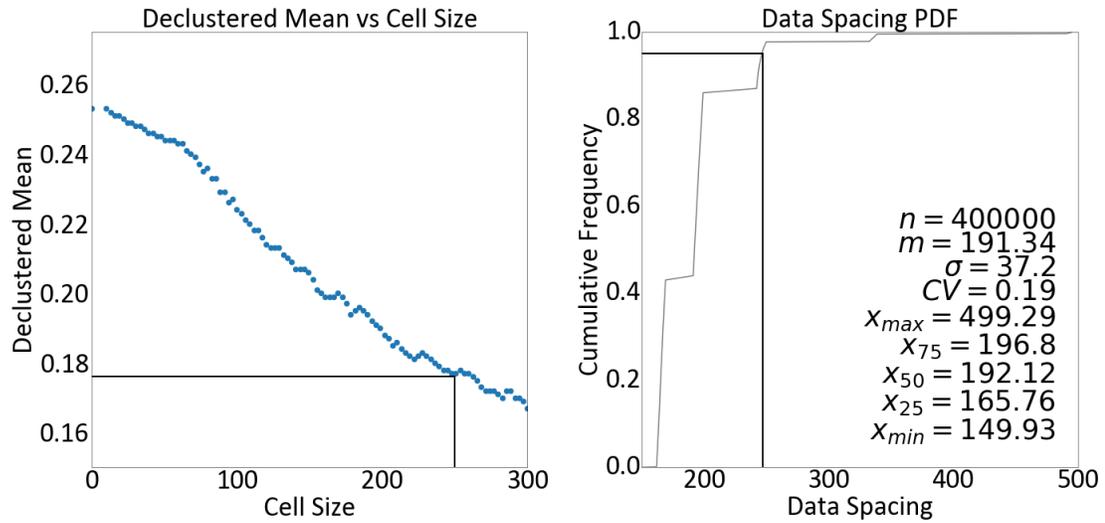


Figure 4.2: JA extension Copper Porphyry Deposit Data Spacing Study. The 95th Percentile of the Data Spacing of the PDF Will be Used as the Declustering Cell Size

Figure 4.3 is the histogram from the Cu data. From the histogram, the de-clustered mean of the data is 0.176, with a standard deviation of 0.192. The distribution of the data is positively skewed with a coefficient of variation (CV) is 0.9; at a CV of 2.5, it is recommended that the geological domains of the high and low data be split (Rossi & Deutsch, 2016). Using one domain for modelling this deposit is reasonable.

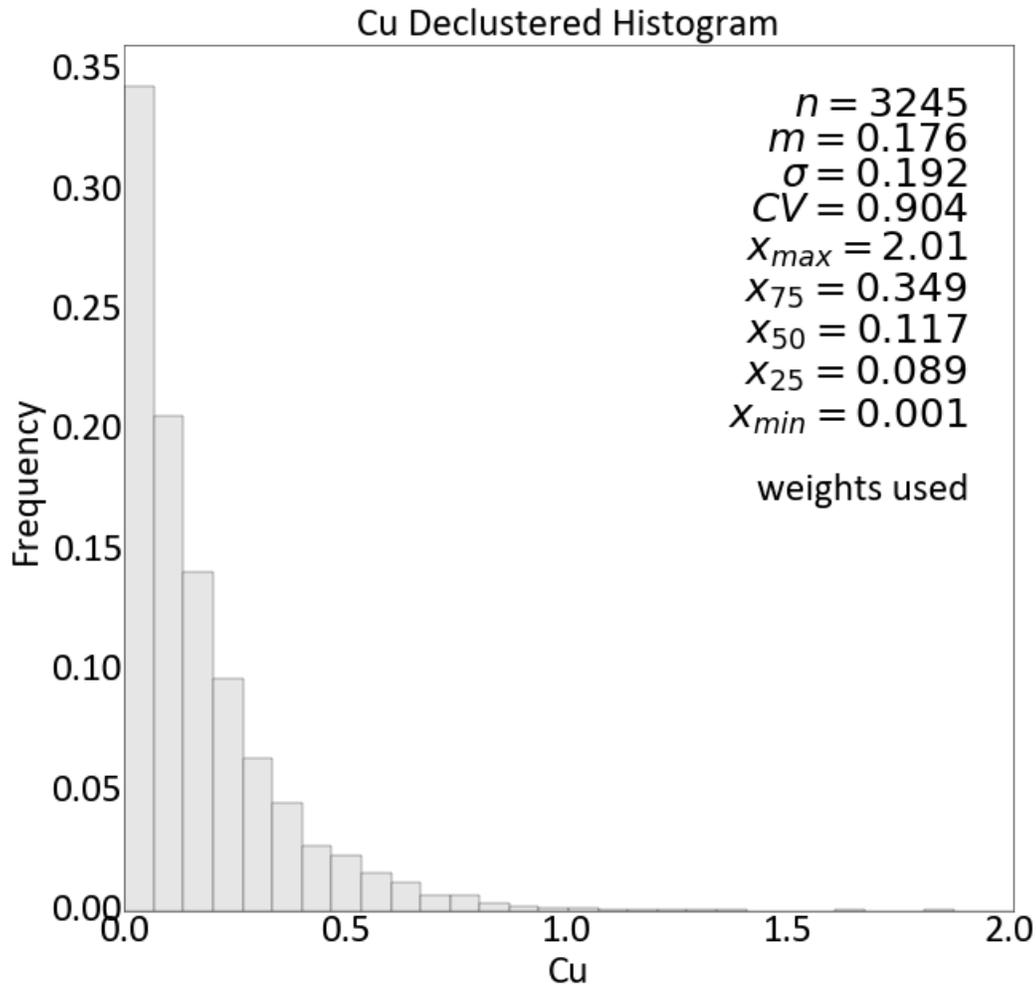


Figure 4.3: JA extension Copper Porphyry Deposit De-clustered Histogram of Cu Data

The major features of the deposit appear to have an azimuth 110° east of north dipping slightly at -10° . The experimental variogram points are calculated using a traditional semivariogram with a 100 unit lag distance, the 5th percentile of data spacing, on the plane of greatest continuity (Silva, 2018). 90° to the plane of greatest continuity points are calculated based on the composite length.

Figure 4.4 shows the experimental variogram points and variogram models in three directions. Equation 4.1 is the variogram model. In the short-range, the variogram is modelled fairly isotropically due to the nature of porphyry deposits. Both the minor and the direction perpendicular to the dip have the same range. The major to minor variogram range is 1.12:1. More information on the traditional semivariogram can be found in Rossi and Deutsch (2016).

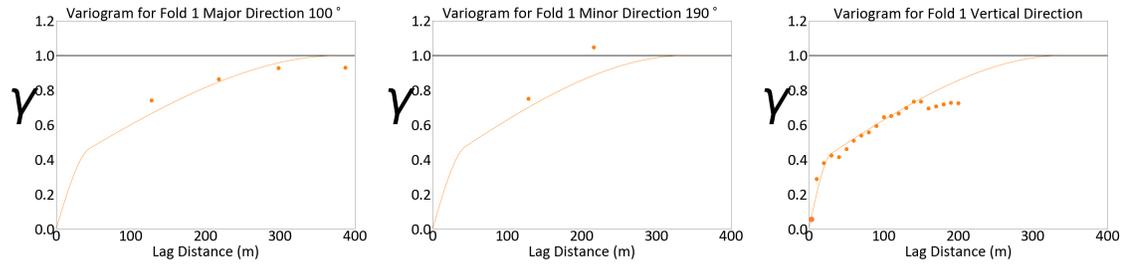


Figure 4.4: JA extension Copper Porphyry Deposit Cu Variograms

$$\gamma(\mathbf{h}) = 0.01Nugget + 0.35\gamma Sph(\mathbf{h}) + 0.64\gamma Sph(\mathbf{h}) \tag{4.1}$$

$$a_{hmax} = 45 \qquad a_{hmax} = 380$$

$$a_{hmin} = 45 \qquad a_{hmin} = 340$$

$$a_{vert} = 30 \qquad a_{vert} = 340$$

Table 4.1 is the grid definition the estimate is made on.

| Direction | Origin | Number of Blocks | Block Size |
|--------------|--------|------------------|------------|
| Northing(Y) | 27400 | 80 | 15 |
| Easting(X) | 34400 | 125 | 15 |
| Elevation(Z) | 600 | 40 | 15 |

Table 4.1: JA extension Copper Porphyry Deposit Modelling Grid Definition

The first method to consider is simple kriging. The kriging parameters include the variograms modelled above, a minimum of 8 and a maximum of 100 data to consider for each estimate, an unbiased mean from the k-fold sets equal to 0.180, and a 1000m search ellipse. Each estimate has a discretization of 5x5x5. The second method to consider is the ERBFN. X, Y, and Z are divided by 100 and used as the features and nodes ranging from 10 to 300 for the ensemble estimation. The final estimate technique to considers is the ML+OICCK. The OICCK parameters are the same as the SK parameter except the mean is not required, and the ML estimate is the collocated data with correlation 0.72. All estimation parameters are determined; the next section explores the results of each estimate.

4.3 Results and Analysis

Figure 4.5 shows one fold with the three different estimations results and Figure 4.6 shows the final estimate. From the k-fold results, all estimation methods produce similar results with the ML smoothest. For this fold the mean is overestimated in all three estimate types. Looking at Tables 4.2, 4.3, and 4.4 the kriged method overestimates the final results, whereas the ML and ML+OICCK methods do not. K-folding the data results in a small mean bias increasing the mean of the data slightly. Due to the distribution of the data, the random sampling for the k-fold validation is likely

to remove more data from the lower values. Due to the assumption of first-order stationarity, kriging attempts to match k-fold mean. In the final estimate and the k-fold example, the histogram reproduction looks best in ML and ML+OICCK. The final estimate is the average of all folds together; hence, the data reproduction *RMSE* value is the error introduced into the estimate when the known location was not apart of the estimate. There was less error introduced into the ML+OICCK estimate than any of the other two estimations techniques with a *RMSE* of 0.091 and R^2 of 0.832. ML did the worst with a *RMSE* of 0.160 and R^2 of 0.576. The kriged estimate is slightly worse than the ML+OICCK with a *RMSE* of 0.100 and R^2 of 0.832. There is evidence in all three estimates of artifacts; this is likely due to the plotting limits.

4. JA Extension Copper Porphyry Deposit Case Study

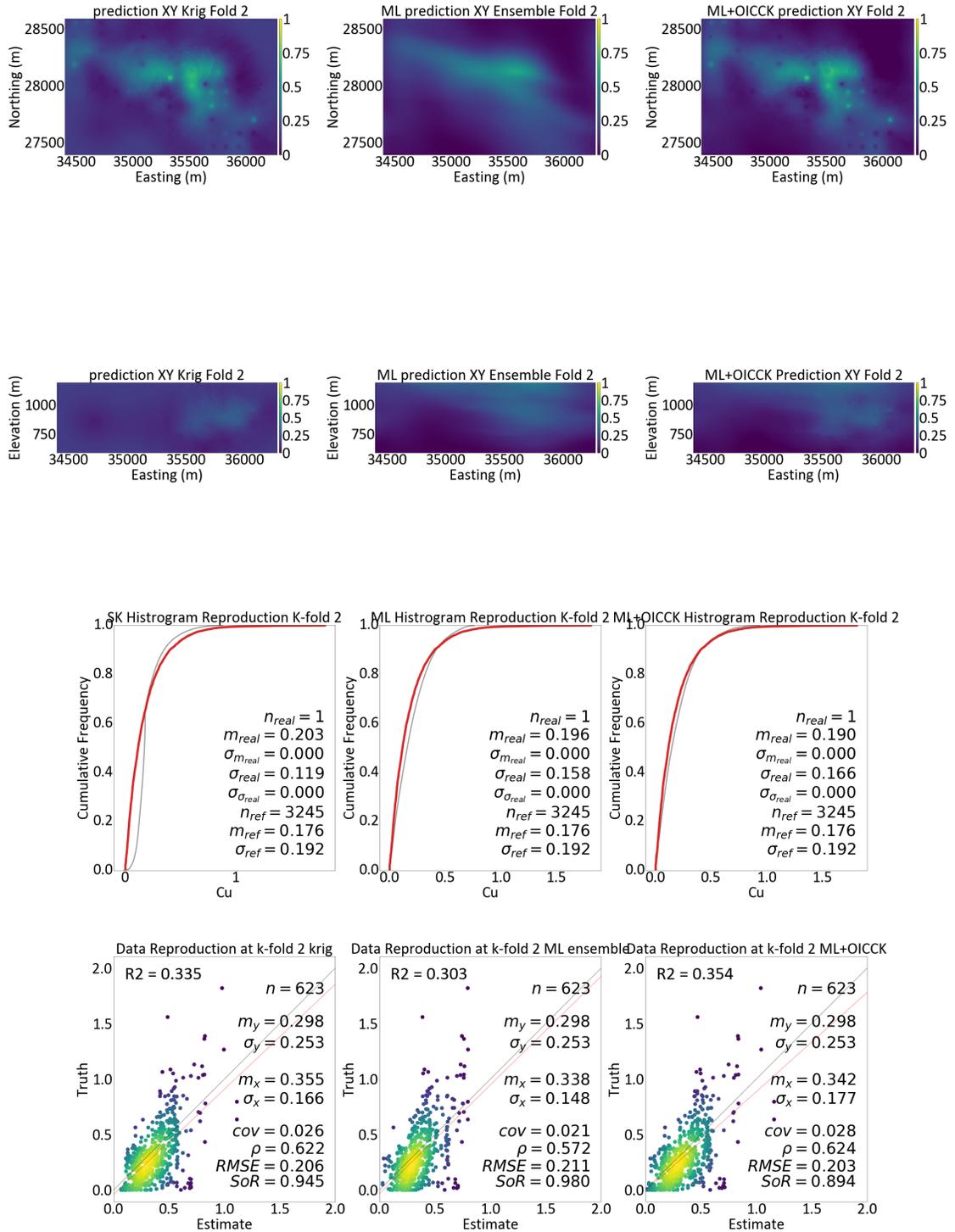


Figure 4.5: JA extension Copper Porphyry Deposit K-Fold 2 Results

4. JA Extension Copper Porphyry Deposit Case Study

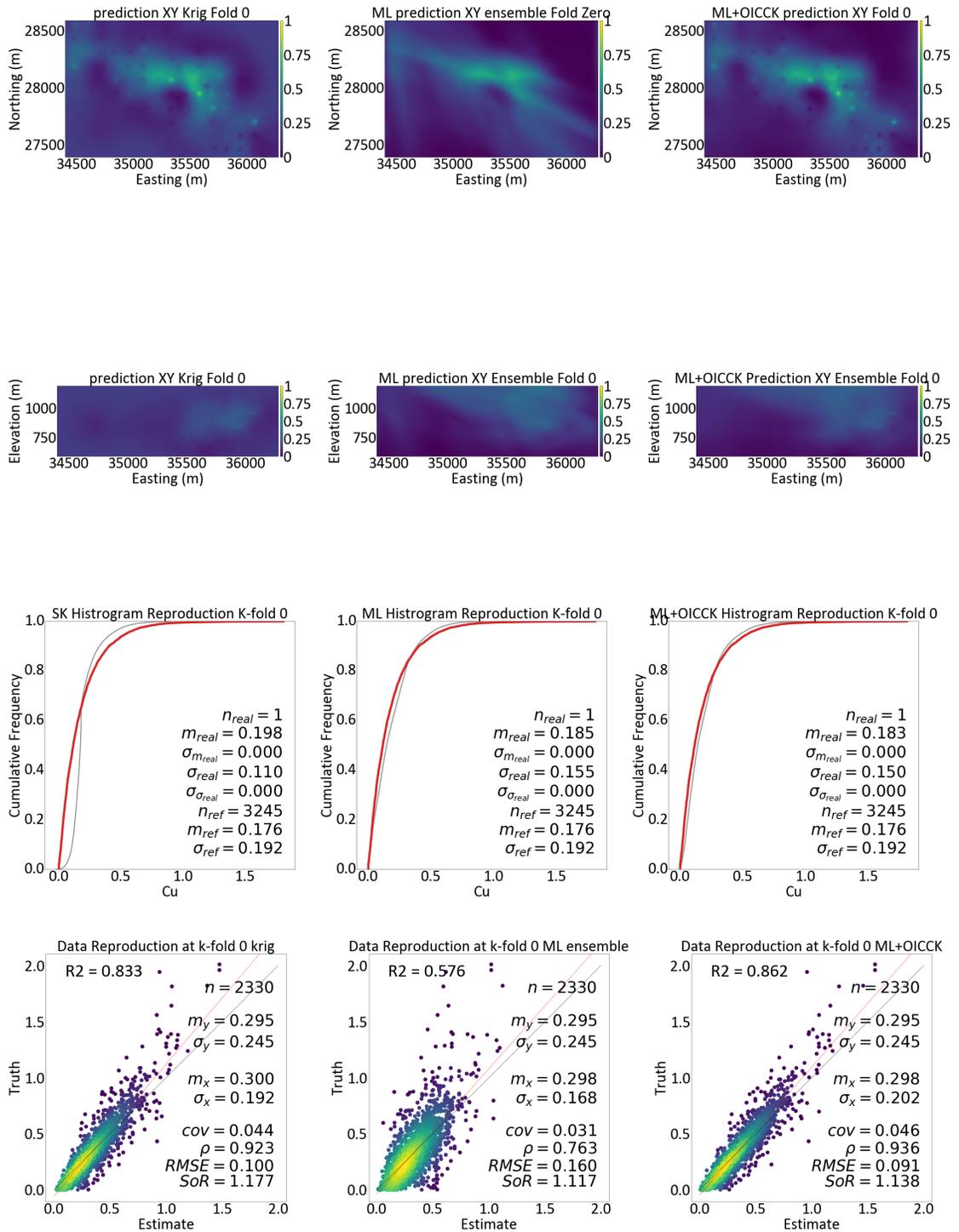


Figure 4.6: JA extension Copper Porphyry Deposit Final Estimate Results

| Summary of the JA extension Copper Porphyry Deposit SK | | | | | | |
|--|-------|-------|-------|-------|----------|-------|
| Truth | | | RMSE | R2 | Estimate | |
| Fold | Sigma | Mean | | | Sigma | Mean |
| 0-Final Estimate | 0.192 | 0.176 | 0.100 | 0.832 | 0.110 | 0.198 |
| 1 | 0.203 | 0.212 | 0.273 | 0.070 | 0.108 | 0.192 |
| 2 | 0.203 | 0.212 | 0.206 | 0.335 | 0.119 | 0.203 |
| 3 | 0.203 | 0.212 | 0.143 | 0.280 | 0.114 | 0.198 |
| 4 | 0.203 | 0.212 | 0.136 | 0.188 | 0.114 | 0.201 |

Table 4.2: JA extension Copper Porphyry Deposit SK Summary

| Summary of the JA extension Copper Porphyry Deposit ML | | | | | | |
|--|-------|-------|-------|-------|----------|-------|
| Truth | | | RMSE | R2 | Estimate | |
| Fold | Sigma | Mean | | | Sigma | Mean |
| 0-Final Estimate | 0.192 | 0.176 | 0.160 | 0.576 | 0.155 | 0.185 |
| 1 | 0.203 | 0.212 | 0.274 | 0.062 | 0.176 | 0.144 |
| 2 | 0.203 | 0.212 | 0.211 | 0.303 | 0.196 | 0.158 |
| 3 | 0.203 | 0.212 | 0.149 | 0.222 | 0.140 | 0.176 |
| 4 | 0.203 | 0.212 | 0.138 | 0.136 | 0.143 | 0.192 |

Table 4.3: JA extension Copper Porphyry Deposit ML Summary

| Summary of the JA extension Copper Porphyry Deposit ML+OICCK | | | | | | |
|--|-------|-------|-------|-------|----------|-------|
| Truth | | | RMSE | R2 | Estimate | |
| Fold | Sigma | Mean | | | Sigma | Mean |
| 0-Final Estimate | 0.192 | 0.176 | 0.091 | 0.862 | 0.150 | 0.183 |
| 1 | 0.203 | 0.212 | 0.277 | 0.040 | 0.149 | 0.173 |
| 2 | 0.203 | 0.212 | 0.203 | 0.354 | 0.166 | 0.190 |
| 3 | 0.203 | 0.212 | 0.146 | 0.249 | 0.151 | 0.178 |
| 4 | 0.203 | 0.212 | 0.134 | 0.205 | 0.150 | 0.192 |

Table 4.4: JA extension Copper Porphyry Deposit ML+OICCK Summary

Tables 4.5, 4.6, and 4.7 shows the total tonnage above a cut-off grade ranging from 0.0 – 0.50%Cu. A bulk density of $2.60t/m^3$ is used to calculate total tonnes similar to Graden (2013). From the tables, the ML and ML+OICCK estimates show similar results; however, the kriging results has more lower grade tonnes. The kriged estimate has 29% of the total tonnage above a cut-off grade of 0.20%Cu; whereas, the ML estimate and the hybrid estimate have 0.39% and 0.36% of the total tonnage above a cut-off grade of 0.20%Cu. The difference between the tonnages at a cut-off grade of 0.20%Cu is significant. This difference is likely due to modelling the high grade and low grade Cu zones together.

The final estimation check is a swath plot. A swath plot looks at the average grade is a strip of the domain. Figures 4.7, 4.8 and 4.9 are swath plots from all three estimation techniques compared to the drill hole data. All three estimation methods appear to do reasonably and reproduce the swaths accurately. The estimations are much smoother than the truth in the swath plot, as expected. In the Z direction swath plot, it appears as if the hybrid and ML estimates better reproduce the

4. JA Extension Copper Porphyry Deposit Case Study

| Cut-off Grade K-fold 0 Copper Porphyry Deposit Kriging | | | | |
|--|----------------|------------|--------------------|----------------------------|
| Cut-off Grade | % Total Tonnes | Mean (%Cu) | Millions of Tonnes | Metal (Millions of Tonnes) |
| 0.00 | 1.00 | 0.20 | 3,508 | 696.6 |
| 0.10 | 0.92 | 0.21 | 3,236 | 677.0 |
| 0.20 | 0.29 | 0.33 | 1,002 | 326.6 |
| 0.30 | 0.12 | 0.43 | 434.7 | 188.7 |
| 0.40 | 0.06 | 0.53 | 213.7 | 112.9 |
| 0.50 | 0.03 | 0.62 | 101.6 | 63.12 |

Table 4.5: JA Extension Copper Porphyry Deposit Kriging Cut-off Grade K-fold 0

| Cut-off Grade K-fold 0 Copper Porphyry Deposit ML | | | | |
|---|----------------|------------|--------------------|----------------------------|
| Cut-off Grade | % Total Tonnes | Mean (%Cu) | Millions of Tonnes | Metal (Millions of Tonnes) |
| 0.00 | 1.00 | 0.19 | 3,510 | 650.1 |
| 0.10 | 0.64 | 0.26 | 2,252 | 593.8 |
| 0.20 | 0.39 | 0.34 | 1,356 | 463.7 |
| 0.30 | 0.20 | 0.43 | 690.1 | 297.4 |
| 0.40 | 0.09 | 0.53 | 326.8 | 173.1 |
| 0.50 | 0.05 | 0.62 | 163.4 | 100.5 |

Table 4.6: JA Extension Copper Porphyry Deposit ML Cut-off Grade K-fold 0

| Cut-off Grade K-fold 0 Copper Porphyry Deposit ML+OICCK | | | | |
|---|----------------|------------|--------------------|----------------------------|
| Cut-off Grade | % Total Tonnes | Mean (%Cu) | Millions of Tonnes | Metal (Millions of Tonnes) |
| 0.00 | 1.00 | 0.18 | 3,510 | 642.7 |
| 0.10 | 0.64 | 0.25 | 2,259 | 573.9 |
| 0.20 | 0.36 | 0.34 | 1,276 | 432.0 |
| 0.30 | 0.17 | 0.44 | 597.9 | 265.1 |
| 0.40 | 0.09 | 0.54 | 299.6 | 162.9 |
| 0.50 | 0.05 | 0.63 | 160.6 | 101.0 |

Table 4.7: JA Extension Copper Porphyry Deposit ML+OICCK Cut-off Grade K-fold 0

average grade. Base on the k-fold results and the final estimate, it appears as if the best estimation technique is the ML+OICCK, it results in the minimum $RMSE$ and the maximum R^2 value. The ML learning estimate behaves similar to the SK estimate and could be used as a variogram free modelling tool by hard replacing the data values; however, the recommend modelling method is the hybrid estimation technique.

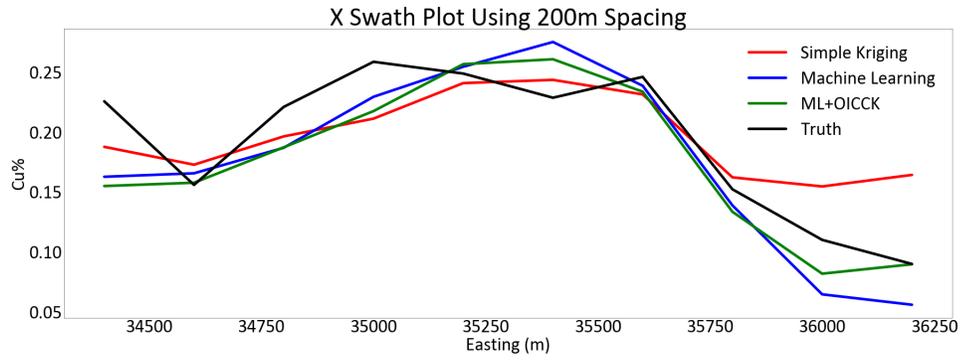


Figure 4.7: JA extension Copper Porphyry Deposit Cu Swath Plot X

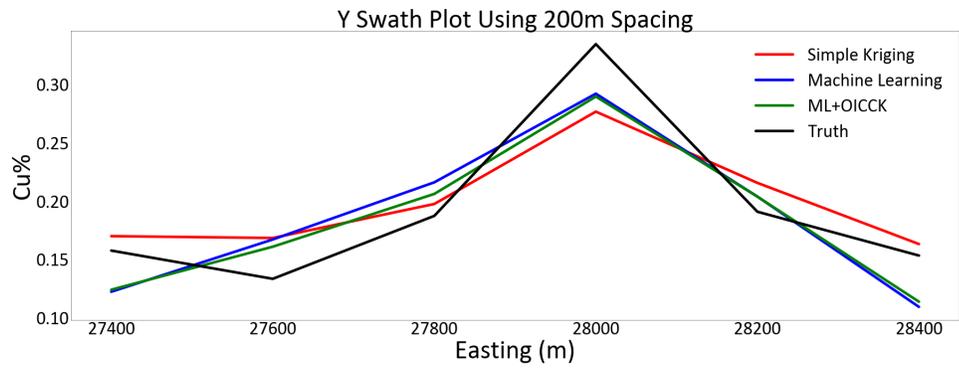


Figure 4.8: JA extension Copper Porphyry Deposit Cu Swath Plot Y

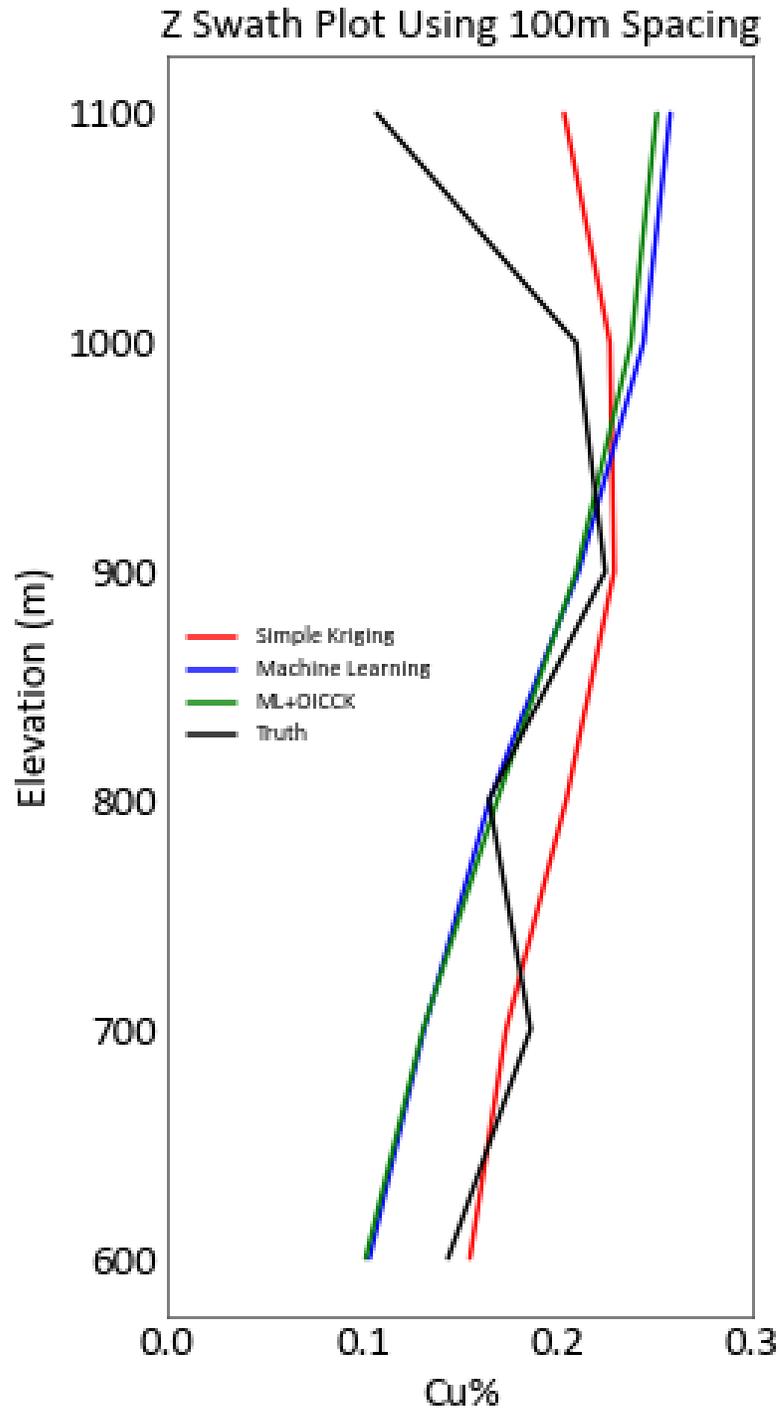


Figure 4.9: JA extension Copper Porphyry Deposit Cu Swath Plot Z

CHAPTER 5

CONCLUSION

Making a spatial prediction for resource modelling is an important task. In current practice, when making a prediction, the assumption of first and second-order stationarity is required. The assumption of stationarity is subjective. The machine learning algorithm presented produces similar results to kriging and does not require the assumptions of stationarity. The machine learning method minimizes human interaction, perhaps minimizing error and bias. The hybrid algorithm leads to an estimate that outperforms the traditional spatial estimation methods in scenarios complex with geological domains. The benefits and downfalls of these estimation techniques are highlighted, multiple simulated examples are explored, and a practical example is demonstrated.

5.1 Topics Covered and Contribution

Domain selection is subjective and traditionally requires expert knowledge. Poorly selected domains result in poor estimates. In some cases, domain selection is not always straight forward. Chapter 2, is a simple summary kriging and machine learning. The estimation domain must be determined, and variograms modelled while kriging. When done correctly, kriging is considered the best unbiased linear estimate. Machine learning presents many techniques for estimation; however, once the correct algorithm is selected, the assumption of stationarity is not required. The type and availability of data determine the machine learning algorithm chosen. Neural networks and k-means are the machine learning algorithms explored. Hyperparameters for neural networks include the number of hidden layers, activation functions, learning rates, and input features. For k-means, the main parameter is the number of nodes. The final estimation technique present in Chapter 2 is ordinary intrinsic collocated cokriging. For OICCK, the parameter selection is similar to simple kriging without the requirement of the mean. OICCK takes advantage of exhaustive secondary data to make a better estimate.

In Chapter 3, two estimation techniques are proposed. Neural networks with different depths and widths are explored and compared to kriging and found to be unsuccessful. The first successful ML estimate takes advantage of radial basis functions to generate a smooth spatial estimate using coordinates as input features. By switching from Euclidean distance to Mahalanobis distance, the radial basis neural network estimate is improved. Considering the covariance with the Mahalanobis distance better considers anisotropy. The radial basis nodes can now take an elliptical form. The elliptical radial basis function network appears to outperform kriging in estimation scenarios that demonstrate complex anisotropy. In more traditional spatial problems, the ML algorithm per-

forms slightly worse than SK; however, the ML algorithm does not require first and second-order stationary. The second estimation technique proposed is a hybrid geostatistical and machine learning hybrid that takes advantage of OICCK. In the OICCK+ML hybrid estimate, the ML estimate is used as the exhaustive secondary data and enforces the complex anisotropy in the estimate. The hybrid estimate appears to outperform kriging in data that exhibit non-stationary features such as log normality, trends, and complex features. The ML and hybrid estimation techniques behave similarly to kriging in terms of data spacing. Using the hybrid algorithm enforces data reproduction and allows for simulations.

Chapter 4 is a case study that explores the a copper porphyry deposit. The copper resources are estimated with the two proposed estimate techniques and compared to kriging. From the predictions maps, all estimates look similar. The ML estimate is the smoothest. In terms of histogram reproduction and mean reproduction, the hybrid and ML estimates perform better. The hybrid estimate outperforms SK in the k-fold 0 data reproduction, and the ML estimate performs the worst due to the nature of regression algorithms. Inspecting the swath plot shows that all estimates perform similarly in terms of average grade except in the Z direction the ML and hybrid estimate appear to reproduce the mean in the swaths better.

The major contributions from this thesis include an ML method of stationary free estimation that performs similarly to SK and a hybrid estimation technique that appears to outperforms the SK in complex spatial problems. An integrated machine learning and geostatistical algorithm for modelling spatial data the more effectively take into account complex anisotropy while minimizing human interaction has been demonstrated and shown effective. The goals and topic set forward in Chapter 1 have been addressed and accomplished.

5.2 Limitations and Future Work

Despite the developments made in this thesis, there are still several limitations with the machine learning method and the hybrid method. Currently, the machine learning method does not reproduce data at the known locations; although replacing the estimate at the known location is possible, it can cause artifacts; furthermore, simulation is currently not possible with the proposed machine learning technique. Simulation is an essential step in quantifying uncertainty in a geological estimate. A less significant limitation in the ML algorithm is that the initial anisotropy cannot be specified. Specifying the anisotropy would likely reduce the time required to train the algorithm. Incorporating a learning stage to determine the optimal number of nodes would potentially speed up the algorithm and produce a better estimate. If a node learning stage is implementing the need to ensemble the estimate would be removed. In the realm of machine learning, a vast amount of possibilities are available to explore when considering making mineral resource estimates. Further investigation into ML estimation techniques is warranted as only neural networks, and k-means

are explored in this dissertation.

Using the hybrid estimation solves the issues with data reproduction and simulation; however, it comes at the cost of assuming first and second-order stationarity, and a variogram model is required. Although using the hybrid estimation appears to produce better results than SK in complex domains, further research into auto variogram modelling using machine learning is warranted. If variograms are auto modelled, the ML+OICCK estimation method could be fully integrated, further reducing the human iteration.

5.3 Recommendations

The goals of this dissertation have been discussed and implemented in theoretical terms, simulated examples, and practical examples. Two estimation methods are explored. The first method explored is a machine learning method that produces similar results to simple kriging while not assuming first and second-order stationarity. The second method explored is a hybrid machine learning method that takes advantage of a machine learning estimate to enforce complex anisotropy and deal with complex data more effectively. If a deterministic estimation is the goal and modelling variogram is difficult, the elliptical radial basis function network is recommended. If a single best estimate is required or a stochastic model is desired, the ordinary intrinsic collocated cokriging model using the elliptical radial basis function network estimate as the secondary data is recommended to evaluate a resource and quantify uncertainty.

REFERENCES

- A Syakur, M., K Khotimah, B., M S Rochman, E., & Dwi Satoto, B. (2018, 04). Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IOP Conference Series: Materials Science and Engineering*, 336, 012017. doi: 10.1088/1757-899X/336/1/012017
- Babak, O., & Deutsch, C. (2009, 03). An intrinsic model of coregionalization that solves variance inflation in collocated cokriging. *Computers & Geosciences*, 35, 603-614. doi: 10.1016/j.cageo.2008.02.025
- Babak, O., & Deutsch, C. V. (2007). Comparison of cokriging with an lmc versus the intrinsic model..
- Boisvert, J. (2010). *Geostatistics with locally varying anisotropy* (Unpublished doctoral dissertation). University of Alberta.
- Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1). doi: 10.1186/s13174-018-0087-2
- Daniels, E. B. (2015). *Prediction of local uncertainty for resource evaluation* (Unpublished master's thesis). University of Alberta.
- Deutsch, C. V., & Journel, A. G. (1992). *Gslib: geostatistical software library and users guide*. Oxford University Press.
- Deutsch, J., & Deutsch, C. (2012, Sept). A new version of kt3d with test cases [Computer software manual]. CCG.
- Dey, A. (2016). Machine learning algorithms: A review. *International Journal of Computer Science and Information Technologies*, 7(3), 1174-1179.
- Dorffner, G. (2001). *Artificial neural networks: international conference, vienna, austria, august 21-25, 2001 ; proceedings*. Springer.
- Elboher, E., & Werman, M. (2012). Efficient and accurate gaussian image filtering using running sums. *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*. doi: 10.1109/isda.2012.6416657
- Farnoush, F. (2017). *Learning activation functions in deep neural network* (Unpublished master's thesis). UNIVERSITÉ DE MONTRÉAL.
- FuquaSchoolofBusiness. (2019). What's a good value for r-squared? *Statistical forecasting:notes on regression and time series analysis*. Retrieved from <https://people.duke.edu/~rnau/rsquared.htm>
- Graden, R. (2013). *Ni 43-101 technical report teck highland valley copper* [NI 43-101 Technical Report].
- Gringarten, E., & Deutsch, C. (1999). Methodology for variogram interpretation and modeling

- for improved reservoir characterization. *Proceedings of SPE Annual Technical Conference and Exhibition*. doi: 10.2523/56654-ms
- Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*.
- Machine learning: a brief breakdown*. (2018, Feb). Retrieved from <https://quantdare.com/machine-learning-a-brief-breakdown/>
- Maklin, C. (2019, Jul). *K-means clustering python example*. Towards Data Science. Retrieved from <https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203>
- Manavazhahan, M. (2017). *A study of activation functions for neural networks* (Unpublished master's thesis). University of Arkansas, Fayetteville.
- Manchuk, J. G. (2017). Isotropic covariance functions and fitting with newton method. *Centre for Computational Geostatistics (CCG) Guidebook Series Vol. 19*.
- Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8), 1246–1266. Retrieved from http://cg.ensmp.fr/bibliotheque/public/MATHERON_Publication_02396.pdf doi: 10.2113/gsecongeo.58.8.1246
- McLennan, J. A. (2007). *The decision of stationarity* (Unpublished doctoral dissertation). University of Alberta.
- Musumeci, F., Rottondi, C., Nag, A., Macaluso, I., Zibar, D., Ruffini, M., & Tornatore, M. (2019, Secondquarter). An overview on application of machine learning techniques in optical networks. *IEEE Communications Surveys Tutorials*, 21(2), 1383-1408. doi: 10.1109/COMST.2018.2880039
- Ng, A., & Katanforoosh, K. (2018, Oct). Cs229 lecture notes. *Index of /notes*. Retrieved from <http://cs229.stanford.edu/notes>
- Nordhaus, W. D. (2001, Sep). *The progress of computing* (Cowles Foundation Discussion Papers No. 1324). Cowles Foundation for Research in Economics, Yale University. Retrieved from <https://ideas.repec.org/p/cwl/cwldpp/1324.html>
- Northcote, B. (2019). Exploration and mining in the south central region, british columbia. *Exploration and Mining in British Columbia*.
- Ortiz, J., & Deutsch, C. (2002, 02). Calculation of uncertainty in the variogram. *Math. Geol.*, 34, 169-183. doi: 10.1023/A:1014412218427
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12.
- Polyak, B. (1964, 12). Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4, 1-17. doi: 10.1016/0041-5553(64)90137-5
- Qu, J. (2018). *A practical framework to characterize non-stationary regionalized variables* (PhD). University of Alberta.
- Raschka, S. (2018, Nov). *Model evaluation, model selection, and algorithm selection in machine learning*

- (Tech. Rep.). University of Wisconsin–Madison. Retrieved from <https://arxiv.org/pdf/1811.12808.pdf>
- Reich, Y., & Barai, S. (1999). Evaluating machine learning models for engineering problems. *Artificial Intelligence in Engineering*, 13(3), 257–272. doi: 10.1016/s0954-1810(98)00021-1
- Roser, M., & Ritchie, H. (2019). Technological progress. *Our World in Data*. (<https://ourworldindata.org/technological-progress>)
- Rossi, M. E., & Deutsch, C. V. (2016). *Mineral resource estimation*. SPRINGER.
- Ruder, S. (2017). An overview of gradient descent optimization algorithms. *arXiv:1609.04747*.
- Rusu, C., & Rusu, V. (n.d.). Radial basis functions versus geostatistics in spatial interpolations. *IFIP International Federation for Information Processing Artificial Intelligence in Theory and Practice*, 119–128. doi: 10.1007/978-0-387-34747-9_13
- Silva, D. (2018). *Enhanced geologic modeling of multiple categorical variables* (Unpublished doctoral dissertation). University of Alberta.
- Stulp, F., & Sigaud, O. (2015). Many regression algorithms, one unified model: A review. *Neural Networks*, 69, 60–79. doi: 10.1016/j.neunet.2015.05.005
- Tensorboard: Graph visualization | tensorflow core | tensorflow*. (n.d.). Retrieved from https://www.tensorflow.org/guide/graph_viz
- Tensorflow*. (n.d.). Retrieved from <https://www.tensorflow.org/>
- Todeschini, R., Ballabio, D., Consonni, V., Sahigara, F., & Filzmoser, P. (2013). Locally centred mahalanobis distance: A new distance measure with salient features towards outlier detection. *Analytica Chimica Acta*, 787, 1 - 9. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0003267013005618> doi: <https://doi.org/10.1016/j.aca.2013.04.034>
- Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. A., Elkhatib, Y., ... Al-Fuqaha, A. I. (2017). Unsupervised machine learning for networking: Techniques, applications and research challenges. *CoRR*, *abs/1709.06599*. Retrieved from <http://arxiv.org/abs/1709.06599>
- Wang, H., Raj, B., & Xing, E. P. (2017). On the origin of deep learning. *CoRR*, *abs/1702.07800*. Retrieved from <http://arxiv.org/abs/1702.07800>
- Wilde, B. J. (2011). Programs to aid the decision of stationarity. *Centre for Computational Geostatistics (CCG) Guidebook Series Vol. 14*.
- Xiaoharper. (n.d.). *How to choose algorithms - azure machine learning studio*. Retrieved from <https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice#the-machine-learning-algorithm-cheat-sheet>

APPENDIX A

APPENDICES

All Notebooks used to generate these results are available in the github repository, Search: Mineral-Resource-Estimates-with-Machine-Learning-and-Geostatistics