2010-03-01

True Efficient Frontiers in Production Performance Assessment Using Machine Learning and Data Envelopment Analysis

Kerry R. Poitier
University of Miami, kpoitier@hotmail.com

UNIVERSITY OF MIAMI

TRUE EFFICIENT FRONTIERS IN PRODUCTION PERFORMANCE ASSESSMENT USING MACHINE LEARNING AND DATA ENVELOPMENT ANALYSIS

By

Kerry R. Poitier

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

June 2008

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

TRUE EFFICIENT FRONTIERS IN PRODUCTION PERFORMANCE
ASSESSMENT USING MACHINE LEARNING AND
DATA ENVELOPMENT ANALYSIS

Kerry R. Poitier

Approved:

Dr. Sohyung Cho Assistant Professor of Industrial Engineering	Dr. Terri A. Scandura Dean of the Graduate School
Dr. Shihab Asfour Professor & Associate Dean of Engineering	Dr. Murat Erkoc Assistant Professor of Industrial Engineering
Dr. Moiez Tapia Professor of Electrical and Computer Engineering	

POITIER, KERRY R. (Ph.D., Industrial Engineering)

True Efficient Frontiers in Production Performance Assessment using Machine Learning and Data Envelopment Analysis (June 2008)

Abstract of a dissertation at the University of Miami

Dissertation supervised by: Dr. Sohyung Cho, Ph.D.
No. of pages in text (199)

Data envelopment analysis (DEA) method which is based on a mathematical programming approach, and stochastic frontier functions (SFF) which is based on the econometric regression approach are two well-known tools for performance and efficiency analysis for profit and non-profit organizations, called decision making units (DMUs). While SFF accounts for both managerial and observational errors, DEA assumes that all of the errors are due to only managerial errors, which can be misleading to decision-makers and managers, if the data utilized is contaminated with statistical noise. The challenge therefore facing empirical or traditional DEA's methodology, is how to account for both managerial and observational errors if present in the analysis, so as to determine DEA's "true" or optimal frontiers.

The main objective of this dissertation is to determine DEA's "true" frontier in a totally nonparametric environment, by utilizing traditional DEA efficient frontiers, along with DEA inefficient frontiers. DEA is integrated with SFF, thus enabling the identification of efficient frontiers, and specifically, a machine learning technique called support vector machine (SVM) is employed to provide an adaptive way to estimate "true" frontiers for a

set of input-output data, considering both managerial and observational errors/deviations. A ratio based on statistics for managerial and observational errors is utilized to find the "true" frontiers that perform in between two extremes, and the methodology developed is applied to a real data set where frontiers generated by SVM are compared to ones obtained by the neural network (NN), and ordinary least squares (OLS) regression approaches.

The results showed that SVM outperformed NN and OLS regression by about 2-to-1 in estimating nonlinear functions for efficient and inefficient frontiers. Also, utlizing a ratio based on statistics for managerial and observational errors, SVM gave a better estimation of the "true" frontier for DEA than both NN and OLS.

The work in this research can prevent managers and decision-makers from committing grievous errors relative to the allocation and distribution of the funds and resources of their organizations, as well as, help organizations to plan a more realistic investment by providing reasonable sense of benchmarking to their peers (DMUs).

To all who faced insurmountable obstacles but

were determined to persevere

ACKNOWLEDGMENTS

My thanks to God Almighty for giving me the strength, wisdom, knowledge and understanding to carry out and complete this research and dissertation.

I would like to express my sincere gratitude to my advisor, Dr. Sohyung Cho, who introduced me to the ideas presented in this dissertation, and who motivated and encouraged me each step of the way, even at times when things seemed dismal. Also, I would like to thank him for his priceless guidance, invaluable input, and constructive feedback and criticisms, which resulted in the completion of this dissertation.

My heartfelt thanks to Dr. Shihab Asfour, Professor and Associate Dean of Engineering, for being like a father to me during my years at the University of Miami in the Department of Industrial Engineering. He has always been a pillar of strength, support and guidance for me during my difficult and good times, and for which I cannot say thank you enough.

Sincerest appreciation and thanks to Dr. Murat Erkoc and Dr. Moiez Tapia, my other doctoral committee members, for giving unselfishly of their time, invaluable suggestions, and their excellent feedback throughout the process.

I also wish to thank Dr. David Sumanth, who served as a mentor, and gave me much support and guidance during my initial years in the Department of Industrial Engineering.

My deepest gratitude goes to my mother, Marinette Poitier-Bethel, who has always been a solace in the time of storm, as well as, my brothers and sister for their undying support over the course of my degree.

My sincerest appreciation to all the faculty, staff and students in the Department of Industrial Engineering with special thanks to Christine Lee – the former secretary, Gisela Fernandez, and Augusto Roca for all the assistance rendered me throughout the years.

Finally, I would like to thank my dear colleague, Dr. Dwayne Leonard for his overwhelming input, moral support and suggestions as I worked towards completing this research and dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN(s)	Artificial neural network(s)
CCR	Charnes, Cooper, and Rhodes
COLS	Corrected ordinary least squares
CRS	Constant return to scale
DEA	Data envelopment analysis
DMU(s)	Decision making unit(s)
EA	Each
ERM	Empirical risk minimization
EUTT	“Extreme-Unbalanced-Two-Tailed”
LP	Linear programming
NN(s)	Neural network(s) or neural net(s)
OLS	Ordinary least squares
OLS1	Ordinary least squares regression Model 1
OLS2	Ordinary least squares regression Model 2
PDF	Probability density function
SFA	Stochastic frontier analysis
SFF(s)	Stochastic frontier function(s)
SLT	Statistical learning theory
SOS	Sum of squares
SRM	Structural risk minimization

SVM(s)	Support vector machine(s)
SVR	Support vector regression
VRS	Variable return to scale

NOMENCLATURE

Chapter One:

DMU_0	Targeted decision making unit
h_0	Efficiency score of targeted decision making unit
max	Maximize
min	Minimize
n	Number of decision making units
u_r	Weight of r^{th} output of targeted decision making unit
v_i	Weight of i^{th} input of targeted decision making unit
x_{ij}	i^{th} input of the j^{th} decision making unit
x_{i0}	i^{th} input of targeted decision making unit
y_{rj}	r^{th} output of the j^{th} decision making unit
y_{r0}	r^{th} output of targeted decision making unit

Chapter Two:

θ_j	Efficiency score of the j^{th} decision making unit
λ_j	Dual variable of the j^{th} decision making unit
m	Number of inputs
min	Minimize
n	Number of decision making units
s	Number of outputs

x_{ij}	i^{th} input of the j^{th} decision making unit
y_{rj}	r^{th} output of the j^{th} decision making unit
$\boldsymbol{u}$	Managerial errors/inefficiencies or technical inefficiencies
$\boldsymbol{v}$	Observational errors
$\boldsymbol{\lambda}$	Convex combination coefficient for determining DEA "true" frontier, $\boldsymbol{0 \leq \lambda \leq 1}$
$F_{Reg}(x)$	DEA average regression function
$F_{\lambda}(x)$	DEA's "true" or optimal frontier function
$F_{CCR}(x)$	Empirical or traditional DEA efficient frontier function
ε	Error or residual
$f(\varepsilon)$	Density function of ε
f^*	Standard normal density function
F^*	Standard normal distribution function
η	Relative variability of the two sources of random error that distinguishes firms from each other also written as $\boldsymbol{\sigma_u / \sigma_v}$
$f(x,w)$	Function for support vector regression
$g_i(x)$	Nonlinear transformation for support vector regression
b	Unknown coefficient to be determined for support vector regression
w_i	Unknown coefficient to be determined for support vector regression

$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	Dot product in m-dimensional feature space for support vector regression
$R(w)$	Regularized risk function for support vector regression
ζ	Regularization constant for support vector regression
$\lvert\zeta\rvert_{\varepsilon}$	ε-insensitive loss function for support vector regression
α_i	Lagrangian multiplier for support vector regression
α_i^*	Lagrangian multiplier for support vector regression
$K(x,y)$	Kernel function satisfying Mercer's condition
c	SVR parameter which determines trade-off between model flatness and degree to which deviations larger than ε are tolerated
$\boldsymbol{\varepsilon}$	SVR parameter which controls the width of $\boldsymbol{\varepsilon}$-insensitive zone
$\boldsymbol{u_i}$	Nonnegative technical inefficiency component of error term for i^{th} production unit (i.e. a half normal distribution)
$\boldsymbol{v_i}$	Two-sided noise component of error term for i^{th} production unit (i.e. a normal distribution)

<u>Chapter Four and Five:</u>

max	Maximize
min	Minimize
θ_j	Efficiency score of the j^{th} decision making unit
λ_j	Dual variable of the j^{th} decision making unit

θ_1	Efficiency score of the decision making unit 1 for illustrative example
θ_2	Efficiency score of the decision making unit 2 for illustrative example
θ_3	Efficiency score of the decision making unit 3 for illustrative example
θ_4	Efficiency score of the decision making unit 4 for illustrative example
λ_1	Dual variable for decision making unit 1
λ_2	Dual variable for decision making unit 2
λ_3	Dual variable for decision making unit 3
λ_4	Dual variable for decision making unit 4
x_{ij}	i^{th} input of the j^{th} decision making unit
y_{rj}	r^{th} output of the j^{th} decision making unit
$F_{(Eff)}$	Efficient frontier
F_{λ}	Efficient frontier for convex Combination coefficient
$F_{(Eff)}(X)$	General efficient frontier function
$F_{(Eff\text{-}SVM)}(X)$	Efficient frontier function for support vector machine
$F_{(Eff\text{-}NN)}(X)$	Efficient frontier function for neural network
$F_{(Eff\text{-}OLS1)}(X)$	Efficient frontier function for OLS Model 1
$F_{(Eff\text{-}OLS2)}(X)$	Efficient frontier function for OLS Model 2
$F_{(Ineff)}$	Inefficient frontier
$F_{(Ineff)}(X)$	General inefficient frontier function

$F_{(Ineff\text{-}SVM)}(X)$	Inefficient frontier function for support vector machine
$F_{(Ineff\text{-}NN)}(X)$	Inefficient frontier function for neural network
$F_{(Ineff\text{-}OLS1)}(X)$	Inefficient Frontier Function for OLS Model 1
$F_{(Ineff\text{-}OLS2)}(X)$	Inefficient Frontier Function for OLS Model 2
F_{Reg}	Average regression function
$F_{Reg}(X)$	General Average Regression Function
$F_{Reg\text{-}SVM}(X)$	Non-parametric Average Regression Function for SVM
$F_{Reg\text{-}NN}(X)$	Non-parametric Average Regression Function for NN
$F_{Reg\text{-}OLS1}(X)$	Parametric Average Regression Function for OLS Model 1
$F_{\lambda\text{-}SVM}(X)$	Efficient frontier function for convex combination coefficient for support vector machine
$F_{\lambda\text{-}NN}(X)$	Efficient frontier function for convex combination coefficient for neural network
$F_{\lambda\text{-}OLS1}(X)$	Efficient frontier function for convex combination coefficient for OLS Regression Model 1
$F_{\lambda\text{-}OLS2}(X)$	Efficient frontier function for convex combination coefficient for OLS Regression Model 2
$\boldsymbol{\lambda}$	Convex combination coefficient for determining DEA "true" frontier, $\boldsymbol{0 \leq \lambda \leq 1}$
$\boldsymbol{\lambda}_{SVM}$	Convex combination coefficient for determining DEA "true" frontier for support vector machine, $\boldsymbol{0 \leq \lambda \leq 1}$
$\boldsymbol{\lambda}_{NN}$	Convex combination coefficient for determining DEA "true" frontier for neural network, $\boldsymbol{0 \leq \lambda \leq 1}$

λ_{OLS1}	Convex combination coefficient for determining DEA "true" frontier for OLS regression model OLS1, $0 \leq \lambda \leq 1$
λ_{OLS2}	Convex combination coefficient for determining DEA "true" frontier for OLS regression model OLS2, $0 \leq \lambda \leq 1$
ω	Convex combination coefficient for determining average regression function, $0 \leq \omega \leq 1$
ω_{SVM}	Convex combination coefficient for determining average regression function for support vector machine, $0 \leq \omega_{SVM} \leq 1$
ω_{NN}	Convex combination coefficient for determining average regression function for neural network, $0 \leq \omega_{NN} \leq 1$
ω_{OLS1}	Convex combination coefficient for determining average regression function for OLS regression model OLS1, $0 \leq \omega_{OLS1} \leq 1$
R^2	R square value for ordinary least squares regression

1. CHAPTER ONE: INTRODUCTION

1.1 Overview

Evaluating the performance, productivity, and efficiency of organizations in both the public and private sector has become increasingly crucial to managers and decision-makers.

Traditionally, organizations have focused on various profitability measures to assess performance, productivity and efficiency of their operations such as single dimensional performance indicators and multiple ratios. Unfortunately, ratio analysis and performance indicators do not provide a significant amount of information when considering the effects of economies of scale, the identification of benchmarking, and the estimation of overall performance measures of organizations (Wu, Yang and Liang, 2006). Regression analysis has also been utilized and although producing many useful insights, it is subjected to the limitation that the estimated function represents the average as opposed to the best-practice input-output relationship (Chirikos and Sear, 2000).

In recent years, owing to the increase in global competition, as well as, the demand placed upon managers and decision-makers to further drive down costs, and increase productivity and efficiency, the use of a linear programming technique, known as Data

Envelopment Analysis (DEA) has become prevalent in numerous industries (Charnes, Cooper, Lewin and Seiford, 1994).

Data Envelopment Analysis or DEA as it is sometimes called is an extreme point, non-parametric, linear programming technique used to assess the relative efficiency of decision making units (DMUs) where the presence of multiple inputs and outputs makes comparisons difficult. The definition of a DMU is generic and flexible and generally refers to any entity that is to be evaluated in terms of its abilities to convert inputs to outputs. DEA was originally proposed by Charnes, Cooper and Rhodes (1978), and subsequently extended to allow variable return to scale (VRS) by Banker, Charnes and Cooper (1984).

Formally, DEA is a methodology, which focuses on frontiers rather than central tendencies such as statistical regression. Instead of attempting to fit a regression line or plane through the center of the data, a piecewise linear surface is floated on the data (Cooper, Seiford and Zhu, 2004). DEA identifies the best practices among decision making units (DMUs) for multiple inputs and multiple outputs, and compares each DMU with only the “best” DMUs; The “best” DMUs lying on an efficiency/production frontier and having an efficiency score of 1.00 (one). The inefficient DMUs, each having an efficiency score less than 1.00 (one) are enclosed by the efficiency/production envelope and exist beneath the frontier (Cooper, Seiford and Tone, 2000).

According to production and microeconomic theories, systems are more productive when producing more output(s) with the same or less input(s). The inefficiency identified by DEA model corresponds to the extent that the input variable(s) can possibly be decreased while producing the same level of output(s); Or the extent that the output(s) levels can possibly be increased while using the same level of input(s). It has been proposed that the efficiency of a target DMU can be calculated by solving a fractional mathematical model which compares the weighted multiple inputs with the weighted multiple outputs data (Cooper, Seiford and Zhu, 2004). This ratio which is to be maximized forms the objective function for the target DMU to be evaluated and for the Charnes, Cooper and Rhodes (CCR) DEA model introduced by Charnes, Cooper and Rhodes (1978) is written algebraically as:

$$\max h_0(u, v) = \sum_r u_r y_{r0} / \sum_i v_i x_{i0} \tag{1.1}$$

subject to:

$$\sum_r u_r y_{rj} / \sum_i v_i x_{ij} \leq 1 \quad \text{for } j = 1,\ldots,n$$

$$u_r,\; v_i \geq 0 \qquad \text{for all } i \text{ and } r$$

where h_0 is the efficiency score of DMU_0, the DMU to be targeted or evaluated; the y_{r0}'s and x_{i0}'s are the observed output and input values respectively of DMU_0; and the u_r's and v_i's the weights of the outputs and inputs respectively. The program determines a value h_0, the efficiency of the target DMU and the weights leading to the efficiency h_0. If the efficiency $h_0 = 1.00$, then the target DMU is efficient relative to the others but if h_0 is less than 1.00 then some other DMU(s) is more efficient than the target DMU being evaluated even when the weights are chosen to maximize DMU_0's efficiency. The fractional linear program DEA Model in Equation 1.1 is converted into a linear form so that linear programming may be applied thus making it easily solvable (Emrouznejad, 1995-2001).

The linear version of the constraints in Equation 1.1 may be written as follows:

$$\mathbf{max}\ \boldsymbol{h_0\ (u) = \sum_r u_r\, y_{r0}} \tag{1.2}$$

subject to:

$$\boldsymbol{\sum_i v_i\, x_{ij} = 1}$$

$$\boldsymbol{\sum_r u_r\, y_{rj} - \sum_i v_i\, x_{ij} \leq 1 \ \text{for}\ j = 1,\ldots,n}$$

$$\boldsymbol{u_r,\ v_i \geq 0 \qquad \text{for all}\ i\ \text{and}\ r}$$

For the objective function in Equation 1.1 it is necessary to observe that in maximizing a ratio or fraction, it is the relative magnitude of the numerator and denominator that are of importance and not their individual values. It is thus possible to achieve the same effect by setting the denominator equal to a constant and maximizing the numerator giving the resultant linear program as in Equation 1.2 (Emrouznejad, 1995-2001).

Table 1.1 contains the data for a simple 3-dimensional example adapted from a problem given by Dr. S. Cho for a 5 (five) manufacturing systems having 1 (one) input variable (i.e. Area), and 2 (two) ouput variables (i.e. Product 1 and Product 2) to show the application of DEA using Banxia DEA Software 3.0 (Banxia, 2007).

Figure 1.1 gives the DEA frontier plot for the data using Banxia for input-orientation and constant return to scale (CRS). Manufacturing systems 2 and 5 are on the frontier and are the efficient DMUs whilst manufacturing systems 1, 3, and 4 are the inefficient DMUs and lie beneath the frontier enclosed by the envelope.

System	Area (10,000 sq-ft)	Product 1 (100,000 EA)	Product 2 100,000 EA)
System 1	3	40	55
System 2	2.5	45	50
System 3	4	55	45
System 4	6	48	20
System 5	2.3	28	50

Table 1.1 Data for 5 (Five) Manufacturing Systems

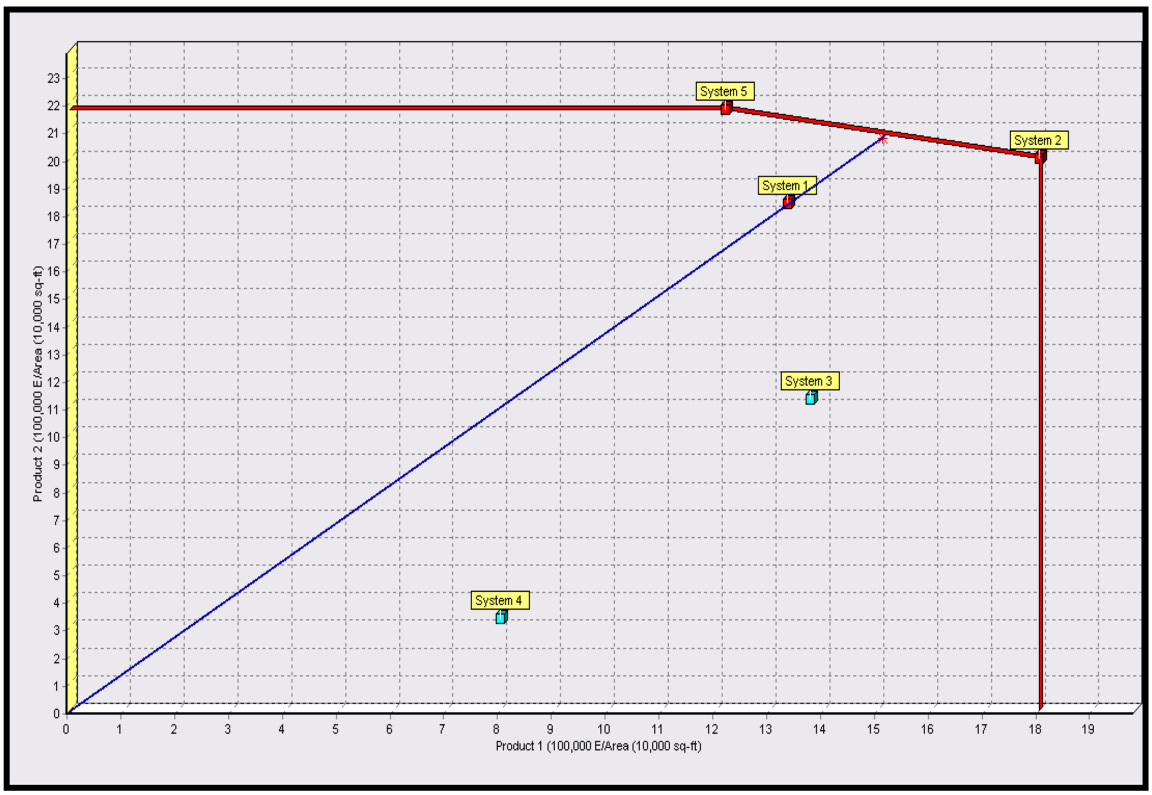


Figure 1.1 DEA Frontier for 5 (Five) Manufacturing Systems (Source: Banxia DEA S/W 3.0, 2007)

Table 1.2 gives the efficiency scores of the 5 (five) manufacturing systems and the errors/deviations of the inefficient systems, namely system 1, system 3, and system 4.

System	Efficiency Score	By DEA Deemed	Errors/Deviations
System 1	87.94% (i.e. 0.88)	Inefficient	1.00 – 0.88 = 0.12
System 2	100% (i.e. 1.00)	Efficient	0.00
System 3	76.39% (i.e. 0.76)	Inefficient	1.00 – 0.76 = 0.24
System 4	44.44% (i.e. 0.44)	Inefficient	1.00 – 0.44 = 0.56
System 5	100% (i.e. 1.00)	Efficient	0.00

Table 1.2 Efficiency Scores and Errors/Deviations for 5 (Five) Systems

Empirical or traditional DEA assumes that all of the errors or deviations from the frontier are due to managerial inefficiencies. To the contrary, research has suggested that there are usually two types of errors involved in the raw input-output data for organizations and institutions namely, managerial errors and observational errors (Aigner, Lovell and Schmidt, 1977; Greene, 1990; Schmidt, 1985 and Sueyoshi, 1991). For empirical DEA, the latter is usually taken to be zero leaving only the former, managerial errors to account for all the errors which can be misleading for decision-makers (Wang, 2003).

Despite DEA increasing popularity and use, a major challenge confronting it is the fact that the efficiency frontier calculated by DEA may be warped if the data are contaminated with statistical noise (Bauer, 1990). Should such errors actually exist in the analysis, the results obtained can mislead decision-makers to ‘over-prescribe’ or ‘under-prescribe’ resources to improve the performance and efficiency of the less efficient units since the efficient frontier obtained may not be the “true” frontier for DEA.

This challenge presents a need therefore to improve DEA's methodology so as to provide decision-makers and managers with a more accurate tool to assess and determine the "true" performance and efficiency for their organizations.

1.2 Purpose, Scope, and Objectives of Research

The main purpose of this research is to determine DEA's "true" or optimal frontier in a totally non-parametric environment by integrating DEA with stochastic frontier functions (SFFs). By utilizing support vector machine (SVM), this research provides an adaptive way to estimate "true" or optimal frontiers, for a set of input-output data considering both managerial and observational errors/deviations.

The scope of this research applies the methodology developed and provided to determine the "true" or optimal frontier for DEA to the data set from the original study 'Program Follow Through' by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for a total of 49 (forty-nine) data pairs in its entirety are used under DEA's assumption of variable return to scale. While only the single-input, single-output, 49 (forty-nine) decision making units, VRS case is utilized as an example for this research, the methodology developed may also be applied to multiple-inputs, multiple-outputs and VRS situations where the total decision making units are greater than 49 (forty-nine).

The objectives of this research are to:

(1) Utilize traditional DEA non-parametric efficient frontier but at the same time, extend traditional DEA by introducing DEA non-parametric inefficient frontier in order to determine DEA's "true" or optimal frontier.

(2) Employ machine learning, in the form of SVM to more accurately estimate nonlinear functions for DEA efficient and inefficient frontiers and to carry out a comparison with neural network (NN) and ordinary least squares (OLS) regression models.

(3) Determine an average regression function non-parametrically from both efficient and inefficient frontier functions.

(4) Integrate DEA with SFF and introduce a simple ratio statistic based upon managerial and observational errors to determine DEA's "true" or optimal frontier between two extremes, those being best-in-practice frontiers and average frontiers.

(5) Develop and provide a simple methodology for accomplishing all of the previously mentioned objectives from (1) to (4).

(6) Apply the developed methodology to a real data set under VRS assumptions to compare frontiers generated by SVM to ones obtained by the ordinary least squares regression approach, as well as, the neural network approach.

1.3 Organization of Dissertation

This dissertation is divided into 6 (six) chapters, of which, Chapter One has been presented in the previous pages.

Chapter Two gives an overview of the literature review. The intention of this chapter is to briefly introduce DEA relative to its CRS and VRS models. The difference between DEA, which utilizes mathematical programming techniques, and the SFF approach, which utilizes econometric regression theory is presented along with studies and research work carried out, most of them incorporating DEA along with neural networks in an attempt to better estimate or predict efficiency frontiers or nonlinear functions. Critical to the literature review, the chapter gives a summary of the research by Wang (2003) which establishes the foundation of this research and dissertation, as well as, the characterization of managerial errors and observational errors relative to stochastic production frontiers. A brief theory and literature review on machine learning, in the form of support vector machine (SVM) and support vector regression (SVR) is also presented since machine learning plays a crucial role in this research.

The limitation of empirical or traditional DEA in accurately determining the “true” efficient frontier if the data is contaminated with statistical noise is key to the problem statement for this research and dissertation in Chapter Three. The 5 (five) manufacturing system example introduced in Chapter One is utilized to demonstrate the mistakes which can be made by managers and decision-makers if the data utilized to determine traditional

DEA frontier is indeed contaminated with statistical noise. As a result of the limitation of traditional DEA, the chapter concludes emphasizing the need to improve DEA's methodology to more accurately determine its "true" or optimal frontier and it recapitulates how this overall objective would be achieved in the research.

Chapter Four outlines the proposed methodology developed for determining DEA's "true" or optimal frontier for the research. Although, the emphasis of the research is on the integration of DEA with SFF relative to support vector machine, the methodology also includes the neural network and ordinary least squares regression models utilized for comparison purposes in the research. An illustrative example explaining traditional DEA efficient frontiers is given and is extended to introduce DEA inefficient frontiers which is a paramount contribution to the research. The chapter gives a detailed methodology on how efficient frontiers, inefficient frontiers and their functions for DEA are obtained for the research by employing machine learning in the form of SVM and NN, as well as, for two OLS models. How the average regression functions are achieved for SVM, NN, and OLS models is outlined, as well as, how a ratio based on statistics for managerial and observational errors to determine the "true" or optimal frontier is applied for SVM, NN, and OLS.

The methodology developed in the previous chapter is applied to a real data set from the original study 'Program Follow Through' by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students and is given in Chapter Five. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for a total of 49 (forty-nine) data pairs in its entirety are used under DEA's assumption of variable return to scale and the detailed results and analysis documented for SVM, NN, and the two OLS models in the research. By utilizing the probability density function (pdf) and the area under the curve statistics, benchmarking is included in the results and analysis, in order to assess the performance of the ratio statistic method utilized in the research.

Chapter Six of the dissertation gives a general summary of the findings of the research on how support vector machine, neural network, and ordinary least squares regression performed relative to each other. Future work, for example, in the form of applying the methodology to data sets consisting of multiple-inputs, multiple-outputs, larger number of decision making units, and other situations are suggested.

2. CHAPTER TWO: LITERATURE REVIEW

This chapter gives an overview of the literature review. The intention of the chapter is to briefly introduce DEA relative to its CRS and VRS models. The difference between DEA, which utilizes mathematical programming techniques, and the SFF approach, which utilizes econometric regression theory is presented along with studies and research work carried out, most of them incorporating DEA along with neural networks in an attempt to better estimate or predict efficiency frontiers or nonlinear functions. Critical to the literature review, the chapter gives a summary of the research by Wang (2003) which establishes the foundation of this research and dissertation, as well as, the characterization of managerial errors and observational errors relative to stochastic production frontiers. A brief theory and literature review on machine learning, in the form of support vector machine (SVM) and support vector regression (SVR) is also presented since machine learning plays a crucial role in this research.

2.1 Overview

Since, the introduction of Data Envelopment Analysis (DEA) by Charnes, Cooper and Rhodes (1978) it has proven to be a popular methodology for assessing and evaluating the efficiency and performance of decision making units (DMUs) within organizations producing multiple outputs from multiple inputs (Ruggiero, 2000). The linear programming modeling for the input-oriented constant return to scale (CRS) model introduced by Charnes, Cooper and Rhodes (1978), also referred to as the CCR (Charnes, Cooper and Rhodes, 1978) model is given by:

$$\min \theta_j \tag{2.1}$$

$$\sum_{j=1}^{n} \lambda_j x_{ij} = \theta x_{ij} \qquad i = 1,2,\ldots,m$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} = y_{rj} \qquad r = 1,2,\ldots.,s$$

$$\lambda_j \geq 0 \qquad j = 1,2,\ldots.,n$$

where θ_j is the efficiency score for the j**th** DMU, where there are n systems or DMUs, the j**th** DMU represents one of the n DMU under evaluation, x_{ij} and y_{rj} are the i**th** input and r**th** output for the j**th** DMU respectively, and λs are dual variables.

The CCR model, or CRS model as it is sometimes called, assumes constant return to scale economies, which means that doubling output exactly doubles inputs. The CCR model was extended to the variable return to scale (VRS) model by Banker, Charnes and Cooper (1984) also referred to as the BCC (Banker, Charnes and Cooper, 1984) model. The VRS input-oriented model is the same as the CRS model except for the fact that the sum of the λs is equal to 1 and is written as:

$$\min \theta_j \tag{2.2}$$

$$\sum_{j=1}^{n} \lambda_j x_{ij} = \theta x_{ij} \qquad i = 1,2,\ldots,m$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} = y_{rj} \qquad r = 1,2,\ldots,s$$

$$\sum_{j=1}^{n} \lambda_j = 1, \quad \lambda_j \geq 0 \qquad j = 1,2,\ldots n$$

where θ_j is the efficiency score for the j**th** DMU, where there are n systems or DMUs, the j**th** DMU represents one of the n DMU under evaluation, x_{ij} and y_{rj} are the i**th** input and r**th** output for the j**th** DMU respectively, and λs are dual variables.

DEA applications for assessing performance and efficiency has been utilized in many industries and organizations, such as hospitals, restaurants, US Air Force wings, universities, cities, courts, business firms, just to mention a few (Cooper, Seiford and Zhu, 2004; and Charnes, Lewin and Seiford, 1994). There is however, another competing paradigm on how to construct frontiers to evaluate and assess the performance and efficiency of DMUs, which is the Stochastic Frontier Function (SFF) approach (Bauer, 1990). DEA utilizes mathematical programming techniques, whilst the SFF approach utilizes econometric regression theory. The major advantage of the DEA approach is that no assumption has to be made about the functional form other than the concavity of the frontier functions (Bauer, 1990). To the contrary, the SFF approach imposes an explicit and possibly over restrictive, functional form for the data (Bauer,1990).

Econometricians favor the stochastic frontier model because it separates the error due to inefficiency and the error due to shock or noise. However, in the operation research field, DEA has been popular, where all the errors or deviations are assumed to be due to managerial/technical inefficiencies and is denoted by ***u*** (Wang, 2003). Bauer (1990) stated that the DEA frontier is very sensitive to the presence of outliers and statistical noise and as a result, the frontier derived from traditional DEA analysis may be incorrect if the data is contaminated by statistical noise. Research has also suggested that there are usually two types of errors involved in the raw input-output data for organizations and institutions namely, managerial errors denoted by ***u*** and observational errors denoted by ***v*** (Aigner, Lovell and Schmidt, 1977; Greene, 1990; Schmidt, 1985 and Sueyoshi, 1991).

Owing to the inability of DEA to be utilized to accurately predict the performance of other DMUs (Wu, Yang and Liang, 2006), in recent years, the artificial neural networks (ANNs) were introduced as good alternatives to assist in the estimation of efficiency frontiers for managers and decision-makers (Wang, 2003).

Athanassopoulos and Curram (1996) were the first to introduce the combination of neural networks and DEA for classification and/or prediction. In their study, DEA was used as a preprocessing methodology to screen training cases for forecasting the number of employees in the health care industry. After the selection of samples, the ANNs were trained as a tool to learn a nonlinear forecasting model.

In the application to the London underground efficiency analysis, Costa and Markellos (1997) carried out a comparison of ANNs with corrected ordinary least squares (COLS) and DEA. They concluded that ANNs perform better in regard of the decision-making, the impact of CRS versus VRS or congestion areas.

DEA was utilized by Pendharkar and Rodger (2003) as a data screening approach to create a sub-sample training data set that was 'approximately' monotonic, which is a key property assumed in certain forecasting and prediction problems. They concluded that the predictive power on an ANN that is trained on the 'efficient' training data subset is stronger than the predictive performance of an ANN that is trained on the 'inefficient training data subset.

Wu, Yang and Liang (2006) were the first to apply a DEA-neural network approach to assess and evaluate branch efficiency of a large Canadian bank. They concluded that DEA-NN approach produces a more robust frontier and identifies more efficient units as a result of the exploration of more good performances and patterns.

While all of the aforementioned studies lend significantly to the progress made in utilizing machine learning, such as neural networks to improve, estimate, and predict nonlinear functions and models relative to DEA, none of them have addressed the observational errors ($\boldsymbol{v}$) which are not accounted for in traditional DEA and which may result in the DEA's "true" frontier being inside the DEA envelope. Wang (2003) is the only known research to date which attempted to account for the observational error (***v***) associated with DEA.

Wang (2003) which establishes the foundation for this research work introduced an interesting concept which incorporated DEA, a mathematical approach, and stochastic frontier functions, which is based on the econometric regression approach to determine more accurately efficient frontiers. In the study, DEA, SFF, and neural networks were utilized to determine the "true" frontier. An average function, $F_{Reg}(x)$ was first determined at $\lambda = 0$; $u = 0$; where ***u*** is managerial error, using neural networks. Secondly, a maximum function $F_{CCR}(x)$ synonymous to the DEA frontier was determined at $\lambda = 1$; $v = 0$; where ***v*** is observational error, using neural networks. Wang (2003) concluded that the "true" frontier would exist between λ= 0 and λ = 1 as represented in Figure 2.1 and is determined by the distribution of the errors/residuals for the function:

$$\mathbf{F_\lambda(x) = \lambda F_{CCR}(x) + (1-\lambda)F_{Reg}(x) \qquad \text{for } 0 \leq \lambda \leq 1 \qquad (2.3)}$$

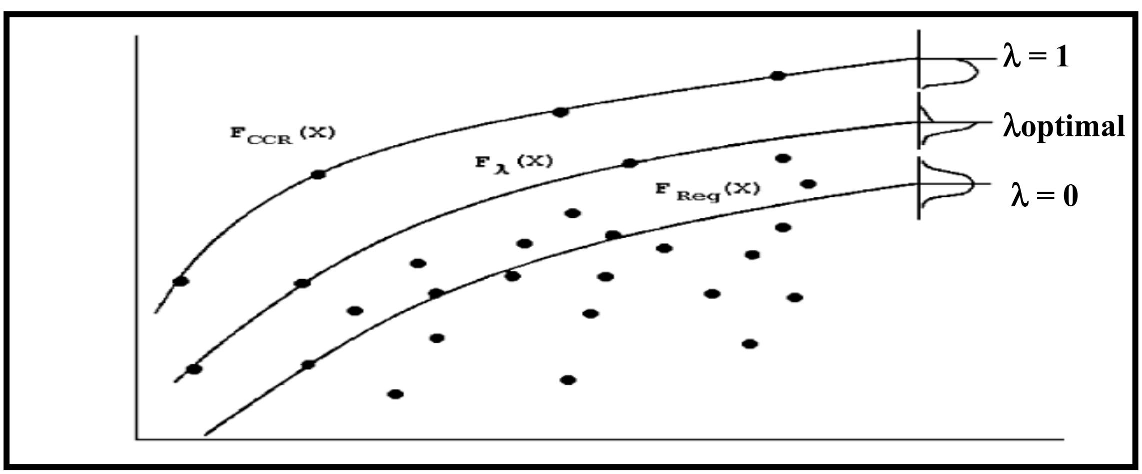


Figure 2.1 Diagram of Proposed Error(s) Distributions for Lambdas (Source: Wang, 2003)

Wang (2003) concluded that the optimal frontier is achieved at λ where the distribution of the residuals/errors depicts an "Extreme-Unbalanced-Two-Tailed – EUTT" distribution (i.e. a discontinuous distribution/function) as may be observed in Figure 2.2 but did not state a criterion for achieving this or for when the optimal λ is achieved. In the study, no assumptions were made explicitly about the distributional forms of the managerial errors (***u***) or the observational errors (***v***) although referral to the representation of ***u*** and ***v*** as a half non-negative normal and normal distributions respectively by Aigner, Lovell and Schmidt (1977) and Schmidt (1985) are mentioned. However, according to Aigner, Lovell and Schmidt (1977) the function or distribution is continuous and given by Equation 2.4 and not discontinuous such as the EUTT distribution presented by Wang (2003).

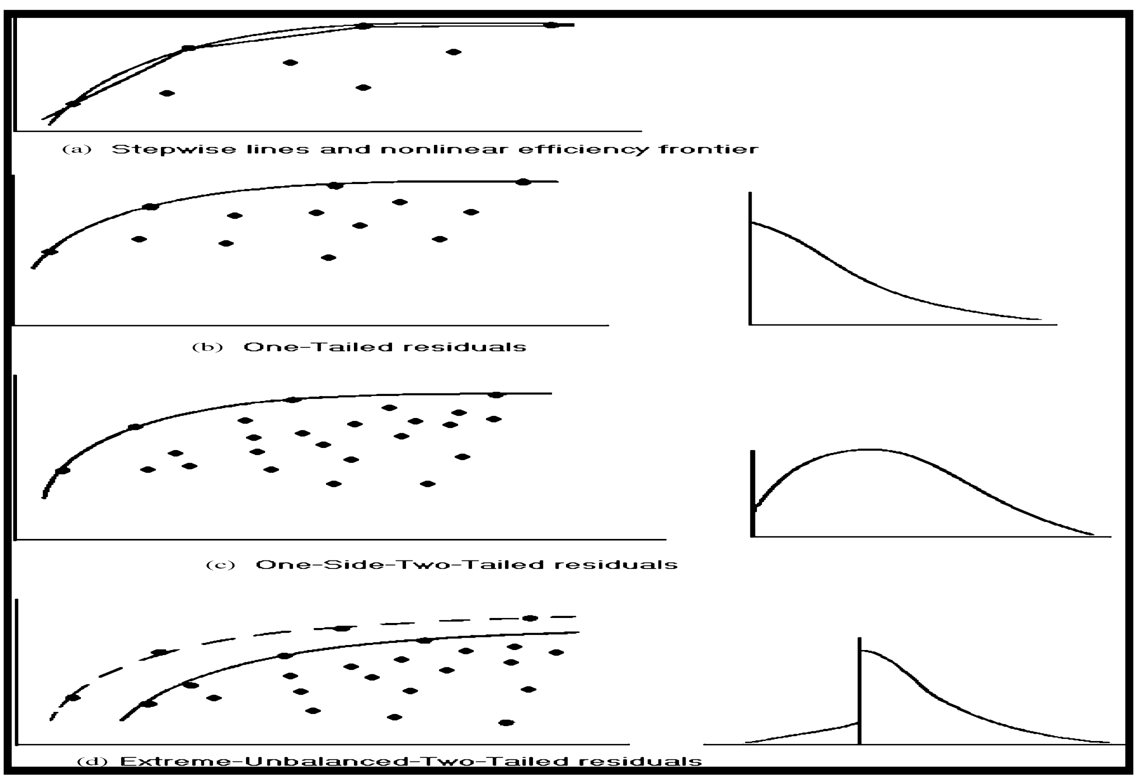


Figure 2.2 Diagram of Distributions of Errors/Residuals and EUTT Residuals (Source: Wang, 2003)

$$f(\varepsilon) = 2/\sigma\, f^{*}(\varepsilon/\sigma)[1 - F^{*}(\varepsilon\eta\sigma^{-1})] \qquad -\infty \leq \varepsilon \leq \infty \tag{2.4}$$

where $\sigma^2 = \sigma_u^2 + \sigma_v^2$, $\eta = \sigma_u / \sigma_v$, and $f^{*}(*)$ and $F^{*}(*)$ are the standard normal density and distribution functions respectively.

Although, Wang (2003) demonstrated that the combination of DEA, SFF, and neural networks may assist model developers in finding data envelopes which are based on the entire data set, rather than some extreme data points from which uncertainty information has been lost, it included no criterion for optimal **λ**. In fact, preliminary experiments carried out during this research showed the "EUTT" characteristics occurring at multiple **λs**. It should be pointed out that even though Wang (2003) suggested several methods for the pre-processing of the data (Wang, 1992) for the training of the neural network to obtain $F_{Reg}(x)$, lack of clarity on how to carry this step out specifically may give rise to training the neural network with incorrect central points. Due to these shortfalls and limitations, this research would attempt to determine "true" frontiers more accurately by utilizing a completely non-parametric environment, as well as, both efficient and in-efficient frontiers where the $F_{Reg}(x)$ is determined easily and with clarity from a combination of DEA non-parametric efficient frontier and DEA non-parametric inefficient frontier functions.

To date, there has been no literature found on any study, which incorporates support vector machine (SVM), along with DEA and SFF to assess the efficiency and performance of DMUs or organizations. Neither have any literature been found which incorporates DEA efficient frontiers with DEA inefficient frontiers in order to determine DEA "true" or optimal frontiers. While, the few studies of DEA along with artificial neural networks have improved the ability of managers and decision-makers to make better judgements and decisions. On the other hand, support vector machines (SVMs) are known to produce equally good, if not better, results than neural networks, while being computational cheaper and producing an actual mathematical function (Clarke, Griebsch and Simpson, 2005). Unlike Wang (2003) which did not provide a criterion for optimal λ for determining "true" frontiers, this research proposes a ratio based on statistics for managerial and observational errors to determine the optimal frontiers that perform in between two extremes: best-in-practice frontiers only considering managerial error and average DMUs.

2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM), based on statistical learning theory (SLT) was developed by Vapnik (1995), and is used for both classification and regression problems and tasks. Support vector machines (SVMs) as they are sometimes called provide non-linear and robust solutions, by mapping the input space into a higher dimensional feature space by utilizing kernel functions where the capacity of such systems are controlled by parameters that do not depend on the dimensionality of the feature space.

The support vector machine regression problem differs from the support vector machine classification problem in a few ways (Vapnik, 1995; and Cristianini and Shawe-Taylor, 2000). The objective of classification with SVM is to make binary decisions (i.e. to choose in which of two classes a given point should be classified). The objective of regression is to approximate a function of which the solution is a function that accepts a data point and return a continuous value (Musicant and Feinberg, 2004).

According to Gunn (1998), in empirical data modeling, data is finite and sampled with the sampling being non-uniform and therefore due to the high dimensional nature of the problem, the data will only form a sparse distribution in the input space. As a result, the problem is nearly always ill posed (Poggio, Torre and Koch, 1985). Neural network approaches have suffered difficulties with generalization, producing models that can overfit the data as a result of the optimization algorithms utilized for parameter selection and the statistical measures used to select the ‘best’ model (Gunn, 1998). SVM possesses great potential and performance largely due to the structural risk minimization (SRM) principles which has a greater generalization ability and is superior to the empirical risk minimization (ERM) principle as adopted in neural networks (Lint, Hoogendoorn and Zuylen, 2000). In SVM, the results guarantee global minima whereas ERM in the case of neural networks, can only locate local minima, where there may be several and not promised to include global minima (Wu, Wei, Su, Chang and Ho, 2003). SVMs provide excellent generalization capabilities, fast, robust to high input space dimension, low number of samples, provide sparse solutions where only the most relevant samples of the training data called support vectors are weighted, resulting in low computational cost and memory requirements (Durbha, King and Younan, 2006).

This research incorporates support vector regression (SVR) which is a powerful technique for predictive data analysis (Cherkassky and Mulier, 1998; and Vapnik, 1995) along with DEA. Support vector regression has been used for diverse application areas, some of which include drug discovery (Demiriz, Bennett, Breneman and Embrechts, 2001), civil engineering (Dibike, Velickov and Solomatine, 2000), sunspot frequency prediction (Collobert and Bengio, 2001), and benchmarking time series prediction tests (Muller, Smola, Ratsch, Scholkppf, Kohlmorgen and Vapnik, 1997; and Cao, 2003).

2.2.1 Brief Theory on Support Vector Regression (SVR)

There are two basic aims of SVR. Firstly, to find a function $f(\mathbf{x}, \mathbf{w})$ that has at most $\boldsymbol{\varepsilon}$ deviations from each of the targets of the training inputs, and secondly and at the same time, would like this function to be as flat as possible (Clarke, Griebsch and Simpson, 2005; and Smola, Scholkopf and Muller, 1998).

In SVR, the input vector **x** is first mapped onto a m-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space given by:

$$f(x,w) = \sum_{i=1}^{m} w_i g_i(x) + b \qquad \textbf{(2.5)}$$

where $g_i(\mathbf{x})$, $i = 1, \ldots\ldots m$ denotes the set of nonlinear transformations; b and w_i are unknown coefficients; and $\langle x, y \rangle$ is a dot product in m-dimensional feature space.

The following regularized risk function given by Equation 2.6 is used to compute the unknown coefficient b and w_i:

$$R(w) = \frac{\beta}{2}|w|^2 + \frac{1}{N}\sum_{i=1}^{N}|\zeta|_{\varepsilon} \tag{2.6}$$

where $\zeta = y_i - f(x_{i,}, w)$, $\beta \geq 0$ is a regularization constant to control the trade-off between model accuracy and complexity and

$$|\zeta|_{\varepsilon} = \begin{cases} 0 \\ |\zeta| - \varepsilon \end{cases} \tag{2.7}$$

is the ε-insensitive loss function (Vapnik, 1995). It has been shown that the regression estimate that minimizes the risk function (2.6) has the form (Vapnik, 1995):

$$\begin{aligned} f(x, \alpha) &= \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\langle g(x_i), g(x)\rangle \\ &= \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)K(x_i, x) + b \end{aligned} \tag{2.8}$$

Lagrangian multipliers $\alpha_{i,}\alpha_i^*$ satisfy conditions $\alpha_i, \alpha_i^* > 0,\ \alpha_i\alpha_i^* = 0$ and $K(x, y)$ is a kernel function, satisfying Mercer's condition which corresponds to a dot product in feature space given by Equation 2.9:

$$K(x, y) = \sum_i^m c_i g_i(x) g_i(y) \tag{2.9}$$

where $c_i > 0$ are positive coefficients.

The parameter c determines the trade-off between the model complexity (flatness) and the degree to which deviations larger than ε are tolerated in optimization formulation. If c is too large (infinity), the objective is to minimize the empirical risk only, without regard to model complexity part in the optimization formulation.

The parameter ε controls the width of the ε-insensitive zone, used to fit the training data. The value of ε can affect the number of support vectors used to construct the regression function. The bigger ε, the fewer support vectors are selected. On the other hand, bigger ε values result in more 'flat' estimates. Therefore, both c and ε values affect model complexity in a different way.

Figure 2.3 gives a simple illustration of a non-linear SVR solution with ε-insensitive zone. The constraints of the problem allow the regression function to lie inside the ε-tube giving no penalty to those samples inside the ε-tube.

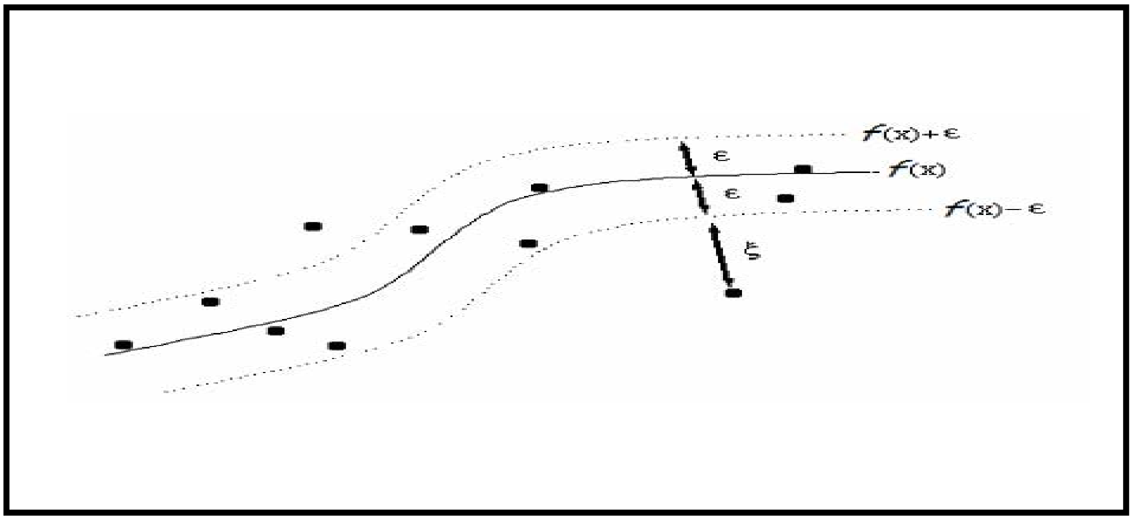


Figure 2.3 Simple Example of Non-linear SVR with Epsilon-Insensitive Zone (Source: Pozdnoukhov and Kanevski, 2006)

Table 2.1 shows a few examples of valid kernel functions that can be used satisfying Mercer's condition.

NAME	
Linear	$K(x,y) = x^T y$
Polynomial	$K(x,y) = \langle x.y \rangle^d$
Gaussian radial base function (RBF)	$K(x,y) = \exp\left\{-\frac{\lVert x-y \rVert^2}{2\sigma}\right\}$

Table 2.1 Examples of Some Valid Kernels

2.3 The Stochastic Production Frontiers and Errors

The stochastic production frontier by Aigner, Lovell and Schmidt (1977), Battese and Corra (1977), and Meeusen and Van den Broeck (1977) is motivated by the idea that deviations from the frontier may not be entirely under the control of the production unit under study. These models allow for technical inefficiency but also accommodate the possibility that the relative performance of a production unit may also be affected by random shock such as measurement errors and other factors, such as weather, machinery performance, even luck etc. outside of its control (Kebede, 2001).

An appropriate formulation of a stochastic frontier model in terms of a general production function for the i^{th} production unit is given by Wang (2003):

$$\boldsymbol{y_i = f(x_i, \beta) - v_i - u_i = f(x_i, \beta) - \varepsilon_i} \qquad \textbf{(2.10)}$$

where $\boldsymbol{v_i}$ is the two-sided noise component, and $\boldsymbol{u_i}$ is the nonnegative technical inefficiency component of the error term. The noise component $\boldsymbol{v_i}$ is assumed to be independently and identically distributed (iid) and symmetric, distributed independently of $\boldsymbol{u_i}$. Thus, the error term $\boldsymbol{\varepsilon_i} = \boldsymbol{v_i} + \boldsymbol{u_i}$ is not symmetric since $\boldsymbol{u_i \geq 0}$.

Bauer (1990), Bravo-Ureta and Pinheiro (1993), and Coelli (1995) observed that most applied papers describe estimation of stochastic frontier models with errors composed of a normal and half-normal random variable, where the errors $\boldsymbol{v_i}$ and $\boldsymbol{u_i}$ are assumed to follow a normal distribution and half-normal distribution respectively. Other assumptions for the errors in stochastic frontier models include normal-exponential, normal-gamma, and exponential-truncated normal (Kebede, 2001).

The distribution function of the sum of a symmetric normal random variable and a half-normal random variable as utilized by Aigner, Lovell and Schmidt (1977) in estimating stochastic frontier models is given by Equation 2.4. The density function is continuous and is asymmetric around zero. The density function of $\boldsymbol{u_i} \geq \mathbf{0}$ for three different values of the standard deviation parameter is given in Figure 2.4.

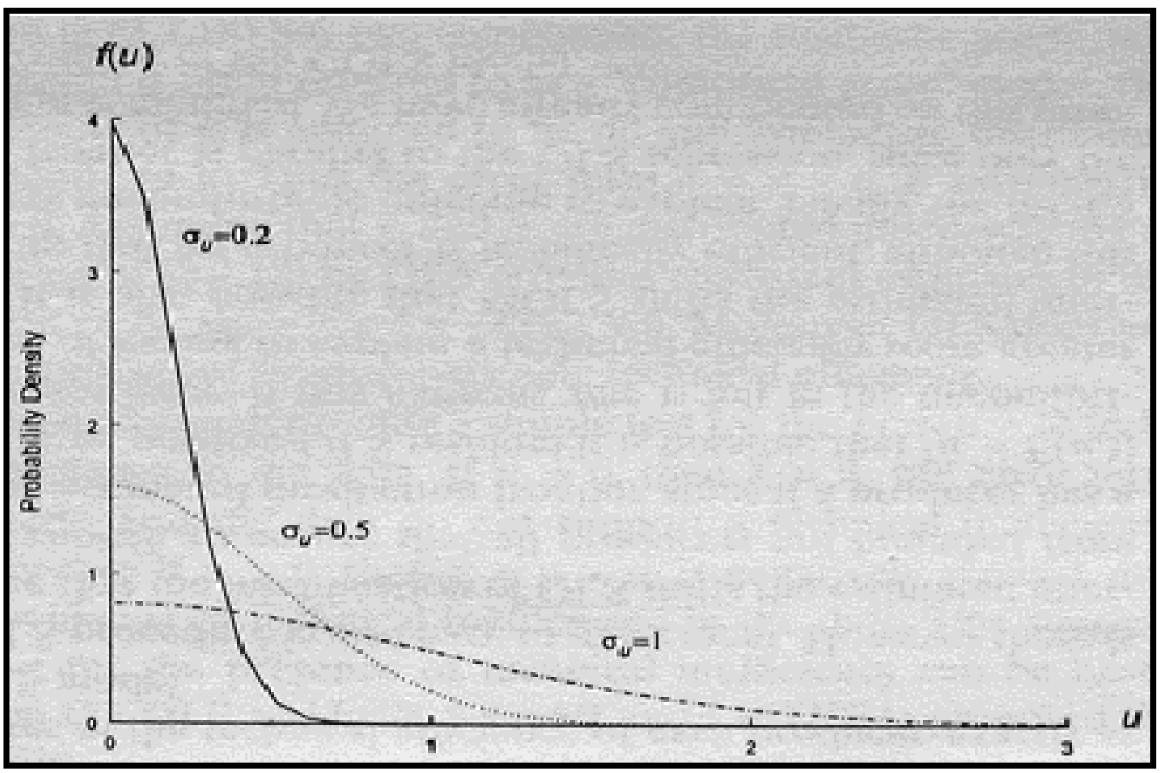


Figure 2.4 Diagram of Half Normal Distributions for Three Different SD Parameter

In a study on stochastic frontier analysis by PWC Consulting (2001), the observational errors v_i and the managerial errors u_i were assumed to be normal and half-normal respectively and are illustrated individually in Figure 2.5 along with the summation of both.

- Observational error (***v***) may be positive or negative with equal likelihood. Therefore, they can be captured by symmetric distribution
- Managerial error (***u***) always causes costs to be higher. Therefore, it can be captured by an asymmetric (half-normal) distribution
- Sum of the two distributions. Its degree of symmetry reveals the relative importance of the observational error

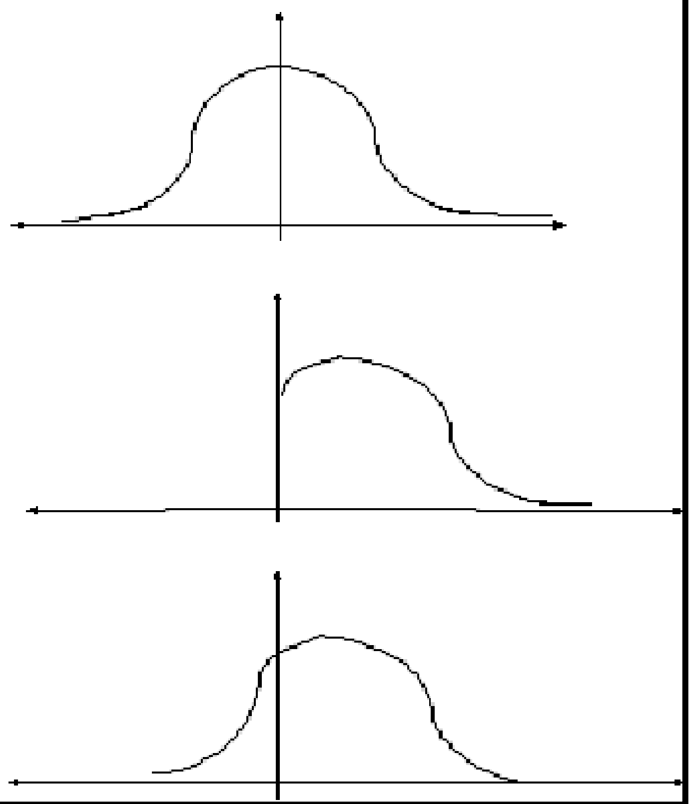


Figure 2.5 Diagram of Normal and Half Normal Distribution Summation (Source:PWC Consulting, 2001)

2.4 Summary of Chapter Two

In this chapter, the literature review for the research and dissertation was outlined. While the entire literature review is important, this section recaps briefly some portions which are significant to the research.

The initial DEA model by Charnes, Cooper and Rhodes (1978), the CRS model or CCR model as it is sometimes called, which assumes constant return to scale was differentiated from the extended variable return to scale (VRS) model, the model assumed for applying the methodology developed in this research. We saw that while doubling the output exactly doubled the inputs for the CRS case, in the VRS case, outputs and inputs have a nonlinear increasing or decreasing relationship.

We learnt that DEA which utilizes a mathematical programming technique, and assumes that all the errors and deviations are due to managerial/technical inefficiencies is popular in the operations research arena. Whereas, the competing paradigm, SFF which separates the error due to inefficiency and the error due to shock or noise is favored by econometricians. Research suggested that there are usually two types of errors involved in the raw input-output data for organizations and institutions namely, managerial errors denoted by $\boldsymbol{u}$ and observational errors denoted by $\boldsymbol{v}$ (Aigner, Lovell and Schmidt, 1977; Greene, 1990; Schmidt, 1985 and Sueyoshi, 1991).

While several studies were presented which attempted to improve, estimate, and predict nonlinear functions and models relative to DEA by incorporating machine learning, in the form of neural networks, Wang (2003) which establishes the foundation of this research was mentioned to have introduced an interesting concept, where DEA was incorporated with SFF and neural networks to determine DEA "true" optimal frontier. In the study, Wang (2003) firstly determined an average function, $F_{Reg}(x)$ at $\lambda = 0$; $\boldsymbol{u} = 0$; where $\boldsymbol{u}$ is managerial error, using neural networks, followed by the determining of a maximum function $F_{CCR}(x)$ synonymous to the DEA frontier was determined at $\lambda = 1$; $\boldsymbol{v} = 0$; where $\boldsymbol{v}$ is observational error, using neural networks. Wang (2003) concluded that the "true" frontier existed between $\lambda = 0$ and $\lambda = 1$ and was achieved when distribution of the residuals/errors depicted an "Extreme-Unbalanced-Two-Tailed – EUTT" discontinuous distribution. We observed that while Wang (2003) made no assumptions explicitly about the distributional forms of the managerial errors ($\boldsymbol{u}$) or the observational errors ($\boldsymbol{v}$) referral to the representation of $\boldsymbol{u}$ and $\boldsymbol{v}$ as a half non-negative normal and normal distributions respectively by Aigner, Lovell and Schmidt (1977) and Schmidt (1985) were mentioned, however, according to Aigner, Lovell and Schmidt (1977) the function or distribution is continuous.

In the chapter, a brief review on SVM was given since in this research it is to be utilized to estimate and predict nonlinear functions for efficient and inefficient frontiers. It was pointed out, that while the few studies of DEA along with artificial neural networks have improved the ability of managers and decision-makers to make better judgements and decisions. On the other hand, support vector machines (SVMs) were known to produce equally good, if not better, results than neural networks, while being computational cheaper and producing an actual mathematical function (Clarke, Griebsch and Simpson, 2005). In the chapter it was stated, that neural network approaches have suffered difficulties with generalization, producing models that can overfit the data as a result of the optimization algorithms utilized for parameter selection and the statistical measures used to select the "best" model (Gunn, 1998). On the other hand, SVM possessed great potential and performance largely due to the structural risk minimization (SRM) principles which has a greater generalization ability and is superior to the empirical risk minimization (ERM) principle as adopted in neural networks (Lint, Hoogendoorn and Zuylen, 2000). For SVM, the results guaranteed global minima whereas (empirical risk minimization (ERM) in the case of neural networks, can only locate local minima, where there may be several and not promised to include global minima (Wu, Wei, Su, Chang and Ho, 2003).

The chapter concluded with a very important topic area to this research, by looking at stochastic production frontiers and their error components. It outlined that the stochastic production frontier according to Aigner, Lovell and Schmidt (1977), Battese and Corra (1977), and Meeusen and Van den Broeck (1977) was motivated by the idea that deviations from the frontier may not be entirely under the control of the production unit under study. Their models while they allowed for technical inefficiency also accommodated the possibility that the relative performance of a production unit may also be affected by random shock such as measurement errors and other factors, such as weather, machinery performance, even luck etc. outside of its control (Kebede, 2001). The observational errors $\boldsymbol{v_i}$ was characterized to be the two-sided noise component, and the managerial error $\boldsymbol{u_i}$ the nonnegative technical inefficiency component of the error term. The noise component $\boldsymbol{v_i}$ was assumed to be independently and identically distributed (iid) and symmetric, distributed independently of $\boldsymbol{u_i}$. Thus, they concluded that the error term $\boldsymbol{\varepsilon_i} = \boldsymbol{v_i} + \boldsymbol{u_i}$ was not symmetric since $\boldsymbol{u_i} \geq \mathbf{0}$.

3. CHAPTER THREE: PROBLEM STATEMENT

This chapter briefly outlines the limitation of empirical or traditional DEA in accurately determining the "true" efficient frontier if the data is contaminated with statistical noise. The 5 (five) manufacturing system example introduced in Chapter One is utilized to demonstrate the mistakes which can be made by managers and decision-makers if the data utilized to determine traditional DEA frontier is indeed contaminated with statistical noise. As a result of the limitation of traditional DEA, the chapter concludes emphasizing the need to improve DEA's methodology to more accurately determine its "true" or optimal frontier and recapitulates how this overall objective would be achieved in the research.

3.1 Overview

For decades keen interest has centered on improving existing performance and efficiency measurement methods, as well as, creating new methods to assess the performance, productivity, and efficiency of organizations.

The most commonly used efficiency measurement methods are Ordinary Least Squares (OLS), Corrected Ordinary Least Squares (COLS), Stochastic Frontier Analysis (SFA), and Data Envelopment Analysis (DEA) as illustrated in Figure 3.1. The first three are parametric in nature requiring an assumption for the functional form, meaning that mis-specification of the functional form can result in catastrophic results, but the latter, DEA is non-parametric and requires no assumption of the functional form (Sarafidis, 2002). Of the entire group, SFA and DEA are the two competing paradigms on efficiency analysis (Wang, 2003).

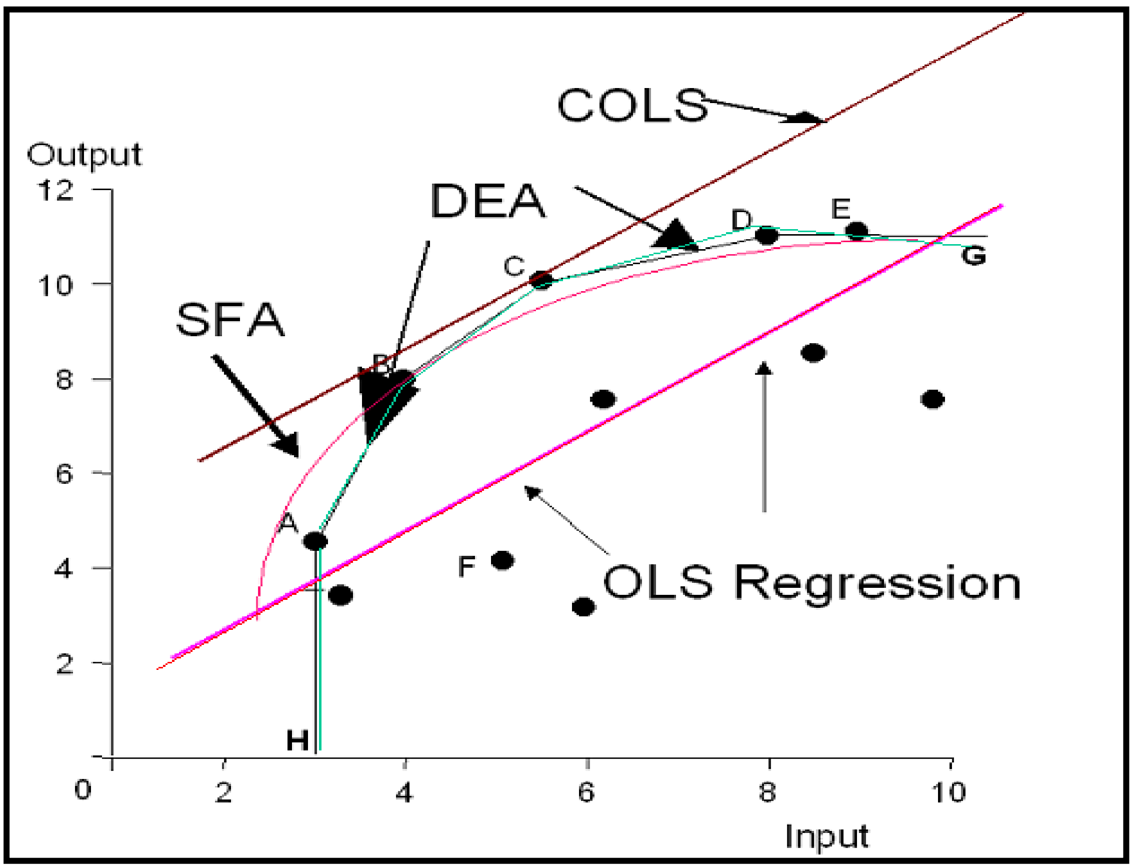


Figure 3.1 Diagram Contrasting Alternative Efficiency Assessment Methods (Source: Thanassoulis, 2007)

More recently, DEA has become increasingly popular as result of its multiple inputs/outputs capability and more importantly because of its non-parametric approach, where no assumption has to be made of the functional form compared to the parametric approach of SFA. Unlike SFA, which accounts for errors due to shock or noise (i.e.. observational errors) as well as, errors due to inefficiency (i.e. managerial errors), empirical DEA assumes that all the errors or deviations from the frontier are due to managerial inefficiencies. To the contrary, research has suggested that there are usually two types of errors involved in the raw input-output data for organizations and institutions namely, managerial errors and observational errors (Aigner, Lovell and Schmidt, 1977; Greene, 1990; Schmidt, 1985 and Sueyoshi, 1991). For empirical DEA, the latter is usually taken to be zero leaving only the former, managerial errors to account for all the errors which can be misleading for decision-makers (Wang, 2003).

Despite DEA increasing popularity and use, a major challenge confronting it, is the fact, that the efficiency frontier calculated by DEA may be warped if the data are contaminated with statistical noise (Bauer, 1990). Should such errors actually exist in the analysis, the results obtained can mislead decision-makers to ‘over-prescribe’ or ‘under-prescribe’ resources to improve the performance and efficiency of what may be thought to be the less efficient units, since the efficient frontier obtained may not be the DEA’s “true” frontier. In other words, the “true” efficiency frontier may exist inside/below the traditional DEA’s envelope/frontier and those units which are deemed efficient may already be over-resourced, whilst those deemed as inefficient may be more efficient than the results convey.

The possibility therefore for managers and decision-makers to thus draw incorrect conclusions from empirical or traditional DEA can lead them into committing grievous errors in how they allocate and distribute the funds and resources of their organizations, which can only result in reduced efficiency, effectiveness, and profitability.

3.2 The 5 (Five) Manufacturing System DEA Example Revisited

Revisiting the DEA example for the 5 (five) manufacturing systems in Figure 1.1. Utilizing traditional DEA, managers and decision-makers would conclude that Systems 2 (two) and 5 (five) are on the efficient frontier, whilst Systems 1 (one), 3 (three) and 4 (four) are inefficient and are below the frontier. Some of the actions, which may be taken as a result, include reducing the area for System 1 (one) by 12.1% from 3 (10,000 sq-ft) to 2.64 (10,000 sq-ft). Secondly, reducing the area of System 3 by 23.6%, as well as targeting an increase in the output of Product 2 (two) by System 3 (three) by approximately 35.8% to 61.11 (100,000 EA). Lastly, reducing the area of System 4 (four) by 55.6% to 2.67 (10,000 sq-ft) as well as targeting an increase in the output of Product 2 (two) by 166.7% to 53.33 (100,000 EA).

Traditional DEA gives the total potential improvement as may be viewed in Figure 3.2 to include a 31.06% reduction in the input (i.e. area) as well as, a possible increase in the output of Product 2 (two) of 68.94%.

In this example, if the efficiency frontier is indeed inside the envelope or below the frontier and not as illustrated by empirical DEA then these decisions would be incorrect and have drastic consequences for organizations and their decision-makers. This could result in the possibility that Systems 2 (two) and 5 (five) do not exist on the "true" frontier and may already be over-resourced and instead, one or of the other systems or a combination of the others may be efficient and exist on the "true" frontier. This simple example cannot emphasize enough the need for managers and decisions-makers to be equipped with more accurate methods so as to better assess and determine "true" production/efficiency frontiers.

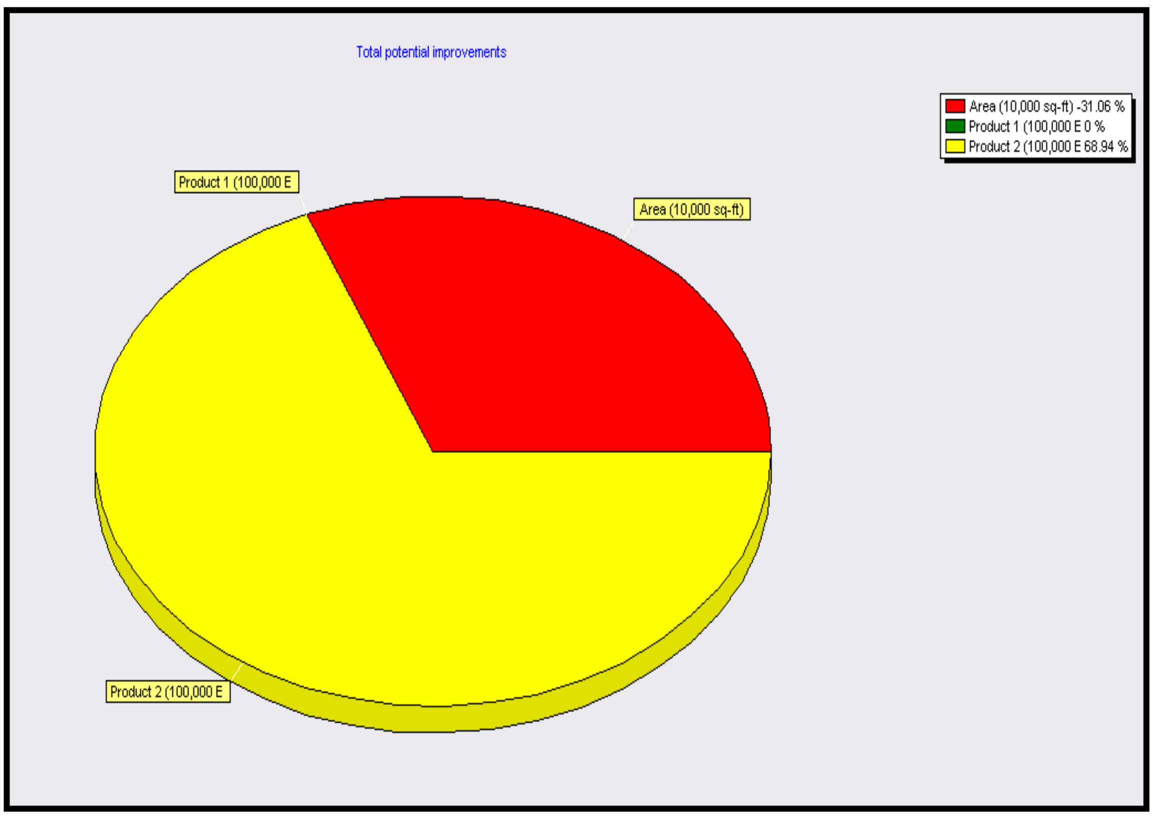


Figure 3.2 Total Potential Improvements for 5 (Five) Manufacturing System DEA Example (Source: Banxia DEA S/W 3.0, 2007)

3.3 How Research Plans to Deal with Problem

To avoid the possible dilemma therefore posed by empirical DEA, the overlying objective of this research is to provide a more accurate tool for managers and decision-makers to assess the performance, productivity, and efficiency of their organizations, and to enable them to determine efficiency/production frontiers more accurately.

Although, Wang (2003) which establishes the foundation for this research work, demonstrated that the combination of DEA, SFF, and neural networks may assist model developers in finding data envelopes which are based on the entire data set, rather than some extreme data points from which uncertainty information has been lost, it included no criterion for optimal λ. In fact, preliminary experiments carried out during this research showed the "EUTT" characteristics occurring at multiple λs. It should be pointed out that even though Wang (2003) suggested several methods for the pre-processing of the data (Wang, 1992) for the training of the neural network to obtain $F_{Reg}(x)$, lack of clarity on how to carry this step out specifically may give rise to training the neural network with incorrect central points. Due to these shortfalls and limitations, this research would attempt to determine "true" frontiers more accurately by utilizing a completely non-parametric environment, as well as, both efficient and inefficient frontiers where the $F_{Reg}(x)$ is determined easily and with clarity from a combination of DEA non-parametric efficient frontier and DEA non-parametric inefficient frontier functions.

To date, there has been no literature found on any study, which incorporates support vector machine (SVM), along with DEA and SFF to assess the efficiency and performance of DMUs or organizations as would be performed in this research and dissertation. Neither have any literature been found which incorporates DEA efficient frontiers with DEA inefficient frontiers in order to determine DEA "true" or optimal frontiers. While, the few studies of DEA along with artificial neural networks have improved the ability of managers and decision-makers to make better judgements and decisions. On the other hand, support vector machines (SVMs) are known to produce equally good, if not better, results than neural networks, while being computational cheaper and producing an actual mathematical function (Clarke, Griebsch and Simpson, 2005). Unlike Wang (2003) which did not provide a criterion for optimal λ for determining "true" frontiers, this research proposes a ratio based on statistics for managerial and observational errors to determine the optimal frontiers that perform in between two extremes: best-in-practice frontiers only considering managerial error and average DMUs.

Recapping the research and dissertation purpose, scope, and objectives as outlined in Chapter One. They are as follows:

- The main purpose of this research is to determine DEA's "true" or optimal frontier in a totally non-parametric environment by integrating DEA with stochastic frontier functions (SFFs). By utilizing support vector machine (SVM), this research provides an adaptive way to estimate “true” or optimal frontiers, for a set of input-output data considering both managerial and observational errors/deviations.

- The scope of this research applies the methodology developed and provided to determine the “true” or optimal frontier for DEA to the data set from the original study ‘Program Follow Through’ by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for a total of 49 (forty-nine) data pairs in its entirety are used under DEA’s assumption of variable return to scale. While only the single-input, single-output, 49 (forty-nine) decision making units, VRS case is utilized as an example for this research, the methodology developed may also be applied to multiple-inputs, multiple-outputs and VRS situations where the total decision making units are greater than 49 (forty-nine).

- The objectives of this research are to:

(1) Utilize traditional DEA non-parametric efficient frontier but at the same time, extend traditional DEA by introducing DEA non-parametric inefficient frontier in order to determine DEA’s “true” or optimal frontier.

(2) Employ machine learning, in the form of SVM to more accurately estimate nonlinear functions for DEA efficient and inefficient frontiers and to carry out a

comparison with neural network (NN) and ordinary least squares (OLS) regression models.

(3) Determine an average regression function non-parametrically from both efficient and inefficient frontier functions.

(4) Integrate DEA with SFF and introduce a simple ratio statistic based upon managerial and observational errors to determine DEA's "true" or optimal frontier between two extremes, those being best-in-practice frontiers and average frontiers.

(5) Develop and provide a simple methodology for accomplishing all of the previously mentioned objectives from (1) to (4).

(6) Apply the developed methodology to a real data set under VRS assumptions to compare frontiers generated by SVM to ones obtained by the ordinary least squares regression approach, as well as, the neural network approach.

3.4 Summary of Chapter Three

The chapter began by presenting some of the most commonly used efficiency measurement methods. We learnt that OLS, COLS, and SFA are parametric in nature and require an assumption of the functional form, meaning that mis-specification of the functional form can result in catastrophic results, whereas, DEA is non-parametric and requires no assumption of the functional form (Sarafidis, 2002).

It was pointed out, that despite DEA increasing popularity and use, a major challenge confronting it, is the fact, that the efficiency frontier calculated by DEA may be warped if the data are contaminated with statistical noise (Bauer, 1990). Should such errors actually exist in the analysis, the results obtained can mislead decision-makers to 'over-prescribe' or 'under-prescribe' resources to improve the performance and efficiency of their organizations. The 5 (five) manufacturing system example was revisited and the results obtained by traditional DEA were used to illustrate how decision-makers and managers can make drastic decisions if indeed the efficient frontier is incorrect.

The chapter concluded by recapping the main purpose and overlying objective of the research and dissertation, along with its scope, and its itemized objectives, which are as follows:

- The main purpose of this research is to determine DEA's "true" or optimal frontier in a totally non-parametric environment by integrating DEA with stochastic frontier functions (SFFs). By utilizing support vector machine (SVM), this research provides an adaptive way to estimate "true" or optimal frontiers, for a set of input-output data considering both managerial and observational errors/deviations.

- The scope of this research applies the methodology developed and provided to determine the “true” or optimal frontier for DEA to the data set from the original study ‘Program Follow Through’ by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for a total of 49 (forty-nine) data pairs in its entirety are used under DEA’s assumption of variable return to scale. While only the single-input, single-output, 49 (forty-nine) decision making units, VRS case is utilized as an example for this research, the methodology developed may also be applied to multiple-inputs, multiple-outputs and VRS situations where the total decision making units are greater than 49 (forty-nine).

- The objectives of this research are to:

  (1) Utilize traditional DEA non-parametric efficient frontier but at the same time, extend traditional DEA by introducing DEA non-parametric inefficient frontier in order to determine DEA’s “true” or optimal frontier.

  (2) Employ machine learning, in the form of SVM to more accurately estimate nonlinear functions for DEA efficient and inefficient frontiers and to carry out a comparison with neural network (NN) and ordinary least squares (OLS) regression models.

(3) Determine an average regression function non-parametrically from both efficient and inefficient frontier functions.

(4) Integrate DEA with SFF and introduce a simple ratio statistic based upon managerial and observational errors to determine DEA's "true" or optimal frontier between two extremes, those being best-in-practice frontiers and average frontiers.

(5) Develop and provide a simple methodology for accomplishing all of the previously mentioned objectives from (1) to (4).

(6) Apply the developed methodology to a real data set under VRS assumptions to compare frontiers generated by SVM to ones obtained by the ordinary least squares regression approach, as well as, the neural network approach.

4. CHAPTER FOUR: METHODOLOGY

In this chapter, the proposed methodology for this research is developed and presented for determining DEA's "true" or optimal frontier, from a totally non-parametric environment by utilizing efficient, as well as, inefficient frontiers, and residuals' distributions. Although, the emphasis of the research is on the integration of DEA with SFF relative to support vector machine, the methodology also includes the neural network and ordinary least squares regression models to be utilized for comparison purposes in the research. An illustrative example explaining traditional DEA efficient frontiers is given and is extended to introduce DEA inefficient frontiers, which is a paramount contribution to the research. Included inside the detailed methodology is also how efficient frontiers, inefficient frontiers and their functions for DEA are obtained in the research by employing machine learning in the form of SVM and NN, as well as, for two OLS models. It also explains in detail, how the average regression functions are achieved for SVM, NN, and the OLS models OLS1 and OLS2, as well as, how a ratio based on statistics for managerial and observational errors to determine the "true" or optimal frontier is applied for SVM, NN, and OLS.

4.1 Overview

The methodology for determining DEA's "true" frontier from a totally non-parametric environment by utilizing efficient, as well as, inefficient frontiers, and residuals' distributions is the cornerstone for this research.

The methodology is applied to the data set from the original study 'Program Follow Through' by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students and may be viewed in Appendix A1. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), with a total of 49 (forty-nine) data pairs in its entirety are used under DEA's assumption of variable return to scale.

DMUs on the DEA's efficient frontier, $F_{(Eff)}$ for the 49 (forty-nine) data pairs are determined under the VRS case by linear programming as in Equation 2.2 or by DEA software, with the efficiency score criterion for these DMUs being 1.00 (one) or 100% (one-hundred percent). The DMUs on the DEA's inefficient frontier, $F_{(Ineff)}$ for the 49 (forty-nine) data pairs are determined under the VRS case by modifying the linear programming in Equation 2.2, where the efficiency score criterion for these DMUs is also denoted by 1.00 (one) or 100% (one-hundred percent).

4.1.1 Illustrative Example for Calculating Efficient and Inefficient Frontiers

While the literature, linear programming, and software for determining DEA's efficient frontier is prevalent, determining DEA's inefficient frontier is not. This research and its methodology is the only known instance of which DEA's inefficient frontier is introduced and will be calculated.

The following example for the 4 (four) DMUs, each with a single input and a single output, is utilized to demonstrate easily how to calculate DEA's efficient and inefficient frontiers. The example's objective is input-oriented and aims to obtain the maximum output from the least amount of input in the efficient frontier case, and aims to obtain the minimum output from the greatest amount of input in the inefficient frontier case. The data for the example is given in Table 4.1.

DMU	Input	Output
DMU1	1	2
DMU2	3	7
DMU3	4	6
DMU4	2	6

Table 4.1 Input-Output Data for 4 (four) DMUs DEA Example

Efficiency in traditional measure is expressed as:

$$\boldsymbol{Efficiency = \frac{Output}{Input}}$$

Therefore the efficiencies of the 4 (four) DMUs as a ratio of unit output to unit input are:

$$\textit{Efficiency of DMU1} = \frac{2}{1} = 2.00$$

$$\textit{Efficiency of DMU2} = \frac{7}{3} = 2.33$$

$$\textit{Efficiency of DMU3} = \frac{6}{4} = 1.50$$

$$\textit{Efficiency of DMU4} = \frac{6}{2} = 3.00$$

From these results, DMU4 is the most efficient, since each unit of its input produces 3 units of output. Also, DMU3 is the least efficient, since for each unit of its input, it only produces 1.50 units of output. Hence, DMU4 is utilized as the base or reference DMU for determining the efficiency score of DMUs for efficient frontier, and DMU3 as the base or reference DMU for determining the efficiency score of DMUs for inefficient frontier. Therefore, in the CRS case as mentioned in Section 2.1, the efficiency scores for the DMUs considering efficient frontier or traditional DEA frontier are calculated as follows:

<u>For DMU1:</u>

$$\textit{Efficient frontier score} = \frac{DMU1}{DMU4} = \frac{2*2}{1*6} = \frac{4}{6} = 0.67$$

For DMU2:

$$Efficient\ \ frontier\ \ score = \frac{DMU2}{DMU4} = \frac{7*2}{3*6} = \frac{14}{18} = 0.78$$

For DMU3:

$$Efficient\ \ frontier\ \ score = \frac{DMU3}{DMU4} = \frac{6*2}{4*6} = \frac{12}{24} = 0.50$$

For DMU4:

$$Efficient\ \ frontier\ \ score = \frac{DMU4}{DMU4} = \frac{6*2}{2*6} = \frac{12}{12} = 1.00$$

Utilizing Equation 2.1 and Lingo Version 10.0 for the CRS case, the linear programming to determine the efficiency score of the DMUs for efficient frontier is written as follows and gives the following objective functions, which are the same as those calculated previously and may be viewed in Appendix B.

For DMU1 - CRS Case Linear Programming (Using Lingo 10.0):

$\mathbf{min} = \theta_1$;
$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 2;$
$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 1\theta_1;$
$\lambda_1 >= 0; \lambda_2 >= 0; \lambda_3 >= 0; \lambda_4 >= 0;$
$Objective\ \ function\ \theta_1\ \ given\ \ for\ efficent\ frontier = 0.6667 = 0.67$

For DMU2 - CRS Case Linear Programming (Using Lingo 10.0):

$$\mathbf{min} = \theta_2 \;\;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 7;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 3\theta_2;$$
$$\lambda_1 >= 0;\ \lambda_2 >= 0;\ \lambda_3 >= 0;\ \lambda_4 >= 0;$$

Objective function θ_2 ***given for efficient frontier = 0.7778 = 0.78***

For DMU3 - CRS Case Linear Programming (Using Lingo 10.0):

$$\mathbf{min} = \theta_3 \;\;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 4\theta_3;$$
$$\lambda_1 >= 0;\ \lambda_2 >= 0;\ \lambda_3 >= 0;\ \lambda_4 >= 0;$$

Objective function θ_3 ***given for efficient frontier = 0.5000 = 0.50***

For DMU4 - CRS Case Linear Programming (Using Lingo 10.0):

$$\mathbf{min} = \theta_4 \;\;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 2\theta_4;$$
$$\lambda_1 >= 0;\ \lambda_2 >= 0;\ \lambda_3 >= 0;\ \lambda_4 >= 0;$$

Objective function θ_4 ***given for efficient frontier = 1.0000 = 1.00***

Hence, it is concluded that only DMU4 exist on the DEA's efficient frontier, $F_{(Eff)}$ since it is the only DMU determined under the CRS case to fulfill the efficiency score criterion, that being 1.00 (one) or 100% (one-hundred percent).

Utilizing Equation 2.2 and Lingo Version 10.0 for the VRS case, the linear programming to determine the efficiency score of the DMUs for efficient frontier is written as follows and gives the following objective functions as also may be viewed in Appendix C.

For DMU1 - VRS Case Linear Programming (Using Lingo 10.0):

$$\min = \theta_1 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 2;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 1\theta_1;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$
$$\text{Objective function } \theta_1 \text{ given for efficent frontier} = 1.0000 = 1.00$$

For DMU2 - VRS Case Linear Programming (Using Lingo 10.0):

$$\min = \theta_2 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 7;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 3\theta_2;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$
$$\text{Objective function } \theta_2 \text{ given for efficient frontier} = 1.0000 = 1.00$$

For DMU3 - VRS Case Linear Programming (Using Lingo 10.0):

$$\min = \theta_3 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 4\theta_3;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$
$$\text{Objective function } \theta_3 \text{ given for efficient frontier} = 0.5000 = 0.50$$

For DMU4 - VRS Case Linear Programming (Using Lingo 10.0):

$$\min = \theta_4 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 >= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 <= 2\theta_4;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$
$$\text{Objective function } \theta_4 \text{ given for efficient frontier} = 1.0000 = 1.00$$

Hence, it is concluded that DMU1, DMU2, and DMU4 exist on the DEA's efficient frontier, $F_{(Eff)}$ since they are the only DMUs determined under the VRS case to fulfill the efficiency score criterion, that being 1.00 (one) or 100% (one-hundred percent).

In the CRS case as mentioned in Section 2.1, the efficiency scores for the DMUs considering inefficient frontier using DMU3 as the base or reference DMU are calculated as follows:

For DMU1:

$$\textit{Inefficient frontier score} = \frac{DMU1}{DMU3} = \frac{2*4}{1*6} = \frac{8}{6} = 1.33$$

For DMU2:

$$\textit{Inefficient frontier score} = \frac{DMU2}{DMU3} = \frac{7*4}{3*6} = \frac{28}{18} = 1.56$$

For DMU3:

$$\textit{Inefficient frontier score} = \frac{DMU3}{DMU3} = \frac{6*4}{4*6} = \frac{24}{24} = 1.00$$

For DMU4:

$$\textit{Inefficient frontier score} = \frac{DMU4}{DMU3} = \frac{6*4}{2*6} = \frac{24}{12} = 2.00$$

Therefore, under the CRS condition, the efficiency scores of DMUs for inefficient frontier are determined by modifying Equation 2.1 and is given by:

$$\max \theta_j \tag{4.1}$$

$$\sum_{j=1}^{n} \lambda_j x_{ij} \geq \theta x_{ij} \qquad i = 1,2,...,m$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} \leq y_{rj} \qquad r = 1,2,....,s$$

$$\lambda_j \geq 0 \qquad j = 1,2,....,n$$

Utilizing Equation 4.1 and Lingo Version 10.0 for the CRS case, the linear programming to determine the efficiency score of the DMUs for inefficient frontier is written as follows and gives the following objective functions, which are the same as those calculated previously and may be viewed in Appendix D.

For DMU1 - CRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_1 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 2;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 1\theta_1;$$
$$\lambda_1 >= 0; \lambda_2 >= 0; \lambda_3 >= 0; \lambda_4 >= 0;$$
$$Objective\ \ function\ \theta_1\ \ given\ \ for\, inefficent\ frontier = 1.3333 = 1.33$$

For DMU2 - CRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_2 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 7;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 3\theta_2;$$
$$\lambda_1 >= 0; \lambda_2 >= 0; \lambda_3 >= 0; \lambda_4 >= 0;$$
$$Objective\ \ function\ \theta_2\ \ given\ \ for\, inefficient\ frontier = 1.5555 = 1.56$$

For DMU3 - CRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_3 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 4\theta_3;$$
$$\lambda_1 >= 0; \lambda_2 >= 0; \lambda_3 >= 0; \lambda_4 >= 0;$$
$$Objective\ \ function\ \theta_3\ \ given\ \ for\, inefficient\ frontier = 1.0000 = 1.00$$

For DMU4 - CRS Case Linear Programming (Using Lingo 10.0):

$\mathbf{max} = \boldsymbol{\theta_4}$;
$\boldsymbol{2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 6;}$
$\boldsymbol{1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 2\theta_4;}$
$\boldsymbol{\lambda_1 >= 0; \lambda_2 >= 0; \lambda_3 >= 0; \lambda_4 >= 0;}$
Objective function θ_4 given for inefficient frontier = 2.0000 = 2.00

Hence, it is concluded that only DMU3 exist on the DEA's inefficient frontier, $F_{(\text{Ineff})}$ since it is the only DMU determined under the CRS case to fulfill the efficiency score criterion, that being 1.00 (one) or 100% (one-hundred percent).

Likewise, under the VRS condition, the efficiency scores of DMUs for inefficient frontier are determined by modifying Equation 2.2 and is given by:

$$\max \theta_j \tag{4.2}$$

$$\sum_{j=1}^{n} \lambda_j x_{ij} \geq \theta x_{ij} \qquad i = 1,2,\ldots,m$$

$$\sum_{j=1}^{n} \lambda_j y_{rj} \leq y_{rj} \qquad r = 1,2,\ldots.,s$$

$$\sum_{j=1}^{n} \lambda_j = 1, \quad \lambda_j \geq 0 \qquad j = 1,2,\ldots.n$$

Utilizing Equation 4.2 and Lingo Version 10.0 for the VRS case, the linear programming to determine the efficiency score of the DMUs for inefficient frontier is written as follows and gives the following objective functions as also may be viewed in Appendix E.

For DMU1 - VRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_1 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 2;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 1\theta_1;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$

Objective function θ_1 given for inefficent frontier = 1.0000 = 1.00

For DMU2 - VRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_2 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 7;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 3\theta_2;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$

Objective function θ_2 given for inefficient frontier = 1.3333 = 1.33

For DMU3 - VRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_3 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 4\theta_3;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$

Objective function θ_3 given for inefficient frontier = 1.0000 = 1.00

For DMU4 - VRS Case Linear Programming (Using Lingo 10.0):

$$\max = \theta_4 \;;$$
$$2\lambda_1 + 7\lambda_2 + 6\lambda_3 + 6\lambda_4 <= 6;$$
$$1\lambda_1 + 3\lambda_2 + 4\lambda_3 + 2\lambda_4 >= 2\theta_4;$$
$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1;$$

Objective function θ_4 given for inefficient frontier = 2.0000 = 2.00

Hence, it is concluded that DMU1 and DMU3 exist on the DEA's inefficient frontier, $F_{(Ineff)}$ since they are the only DMUs determined under the VRS case to fulfill the efficiency score criterion, that being 1.00 (one) or 100% (one-hundred percent).

As this research only examines the VRS case on the selected data set in Appendix A1, those DMUs on the efficient frontier $F_{(Eff)}$, and inefficient frontier $F_{(Ineff)}$ are expanded utilizing both Minitab 12.23 and Datafit Version 8.2 to plot and generate DEA's efficient and inefficient piecewise frontiers so as to further generate a total set of 20 (twenty) DMUs each for efficient frontier $F_{(Eff)}$ expanded data set, and inefficient frontier $F_{(Ineff)}$ expanded data set to allow for the training and testing of both neural networks and support vector machines. Each additional data pair, which is included in the expanded data set obtained from the actual DEA's efficient and inefficient frontiers using Datafit 8.2 is added into Equation 2.2 and Equation 4.2 with the original data used to initially calculate efficient frontier and inefficient frontier. Each data pair included in the expanded data set to 20 (twenty) DMUs must fulfill the efficiency score criterion of 1.00 and monotonicity.

Datafit 8.2 is an easy to use curve-fitting software and would fit a curve or produce a line plot, connecting all the data pairs of a set to be examined. It also has a feature which allows the user to position the pointer anywhere along the fitted curve or line plot and would give the xy data coordinates or pair at the pointer's position.

Therefore, for the illustrated example, the DMUs which are on the efficient frontier $F_{(Eff)}$ and the inefficient frontier $F_{(Ineff)}$ in the VRS case are as follows:

$F_{(Eff)}$ = {DMU1, DMU2, DMU4}

$F_{(Ineff)}$ = {DMU1, DMU3}

The 3 (three) DMUs on the efficient frontier $F_{(Eff)}$ are used in Minitab and Datafit to plot the DEA's piecewise efficient frontier, so as to allow the set to be expanded to a total of 20 (twenty) DMUs, including the original 3 (three) DMUs on the efficient frontier. The 2 (two) DMUs on the inefficient frontier $F_{(Ineff)}$ are used in Minitab and Datafit to plot the DEA's piecewise inefficient frontier, so as to allow the set to be expanded to a total of 20 (twenty) DMUs, including the original 2 (two) DMUs on the inefficient frontier. It is these expanded data sets, each consisting of 20 (twenty) DMUs total each, for both the efficient frontier $F_{(Eff)}$ and inefficient frontier $F_{(Ineff)}$ which are used to train and test both neural networks and support vector machines.

4.1.2 Machine Learning

In order to train and test neural networks and support vector machines satisfactorily, there must be a sufficient amount of data, which give rise to the need for the expanded data set mentioned in Section 4.1.1. It is common that for the training and testing of neural networks and support vector machines, that the expanded data set mentioned be preprocessed or scaled so as to facilitate the nonlinear estimation process for both techniques. The reason for scaling the data, is that the use of very high or low numbers, or series with a few very high or very low outliers, can cause underflow or overflow problems, with the computer stopping, or as Judd (1998) points out, the computer continuing by assigning a value of zero to the values being minimized. For this research, the expanded data sets consisting of 20 (twenty) DMUS each for both efficient frontier $F_{(Eff)}$ and inefficient frontier $F_{(Ineff)}$ are subsequently scaled before being split randomly into a 70% (seventy percent) set for training both neural networks and support vector machines, and a 30% (thirty percent) set for testing both neural networks and support vector machines. The same 70% (seventy percent) set which is used to train neural networks to obtain a function, is the exact same set used to train support vector machines to obtain a function. Likewise, the same 30% (thirty percent) set used to test the neural networks' function, is the exact same set used to test the support vector machines' function.

This research allows the software used to determine the optimal parameters for the support vector machine model. For the neural network architecture however, the number of hidden nodes are determined by a more common heuristic for smaller sample sizes, that being twice the number of input nodes+1 (Bhattacharyya and Pendharkar, 1998; and Pendharkar, 2001). Subsequent to this, the software is allowed to determine the optimal hidden activation and output activation functions for the neural network architecture as determined by the number of hidden node heuristic.

4.1.3 Ordinary Least Squares Regression Models for Comparison

In order to assist in the evaluations of the non-parametric neural network and support vector machine models for the research, two ordinary least squares regression models are created.

The first model, OLS1 utilizes the same expanded data sets for both efficient frontier $F_{(Eff)}$ and inefficient frontier $F_{(Ineff)}$ as mentioned in Section 4.1.2. The same 70% (seventy percent) randomly selected scaled data pairs used to train the neural network and support vector machine is the exact same 70% (seventy percent) used to create the regression functions for both efficient and inefficient frontiers. The same 30% (thirty percent) randomly selected scaled data pairs used to test the neural network and support vector machine are the exact same 30% used to test the OLS1 regression functions for efficient and inefficient frontiers.

The second regression model, OLS2 utilizes the conventional approach initially, by fitting a regression function to all original 49 (forty-nine) data pairs. For the original 49 (forty-nine) data pairs in Appendix A1, the x-input values are scaled between 0.000 and 1.000, and the y-output values between 0.200 and 0.800. The model is then extended to include the efficient frontier $F_{(Eff)}$ as for NN, SVM, and OLS1. The same 70% (seventy percent) randomly selected scaled data pairs used to train the neural network and support vector machine are the exact same 70% (seventy percent) used to obtain the regression function for the efficient frontier $F_{(Eff)}$. The same 30% (thirty percent) randomly selected scaled data pairs used to test the neural network and support vector machine are the exact same 30% (thirty percent) used to test the OLS2 regression function for the efficient frontier $F_{(Eff)}$. The OLS2 model does not include the inefficient frontier as NN, SVM, and OLS1.

4.1.4 Frontier Analysis

4.1.4.1 Determining the Average Regression Function $F_{Reg}(X)$ Non-parametrically

This research utilizes a completely non-parametric approach to determine DEA's "true" frontier. In order to achieve this, an average regression function $F_{Reg}(X)$ is obtained from functions obtained by training and testing neural networks and support vector machines for non-parametric efficient frontier and non-parametric inefficient frontier. The average function $F_{Reg}(X)$ is achieved by varying the ω in Equation 4.3 until a normal or approximate normal distribution of the residuals is realized (i.e. where the sum of the residuals on the left hand side and the right hand side equal 0). The ω at which this is achieved, is the optimal ω for $F_{Reg}(X)$ and determines the average regression function $F_{Reg}(X)$ found non-parametrically at $\lambda = 0$ for the research and as illustrated in Figure 4.1.

$$F_{Reg}(X) = \omega F_{(Eff)}(X) + (1-\omega)F_{(Ineff)}(X) \qquad \text{for } 0 \leq \omega \leq 1 \tag{4.3}$$

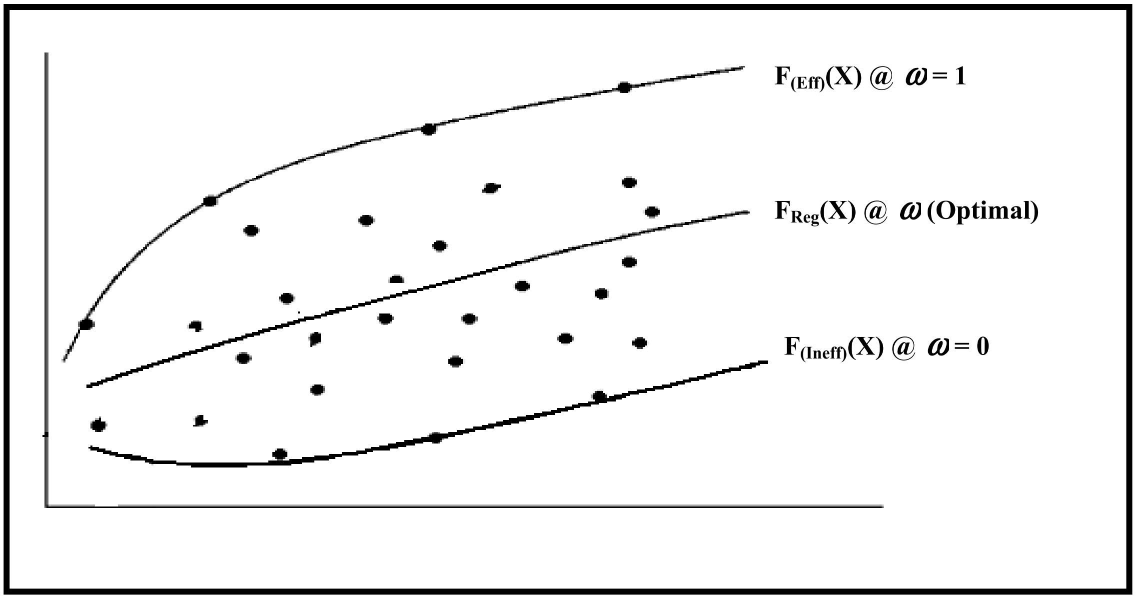


Figure 4.1 Simplified Diagram to Calculate $F_{Reg}(X)$ Non-parametrically

4.1.4.2 Determining the Average Regression Function $F_{Reg}(X)$ Parametrically

The two ordinary least squares regression models to be used for comparison purposes compute the average regression function $F_{Reg}(X)$ in Equation 4.3 parametrically. In order to achieve this, for the first model OLS1, an average regression function $F_{Reg}(X)$ is obtained from ordinary least squares regression functions obtained with the exact same expanded data sets used for the training and testing of neural networks and support vector machines for non-parametric efficient frontier and non-parametric inefficient frontier. The average function $F_{Reg}(X)$ for model OLS1 is achieved parametrically by varying the ω in Equation 4.3 until a normal or approximate normal distribution of the residuals is realized (i.e. where the sum of the residuals on the left hand side and the right hand side equal 0). The ω at which this is achieved, is the optimal ω for $F_{Reg}(X)$ and determines the average regression function $F_{Reg}(X)$ found parametrically at $\lambda = 0$ for model OLS1 in this research and as illustrated in Figure 4.2.

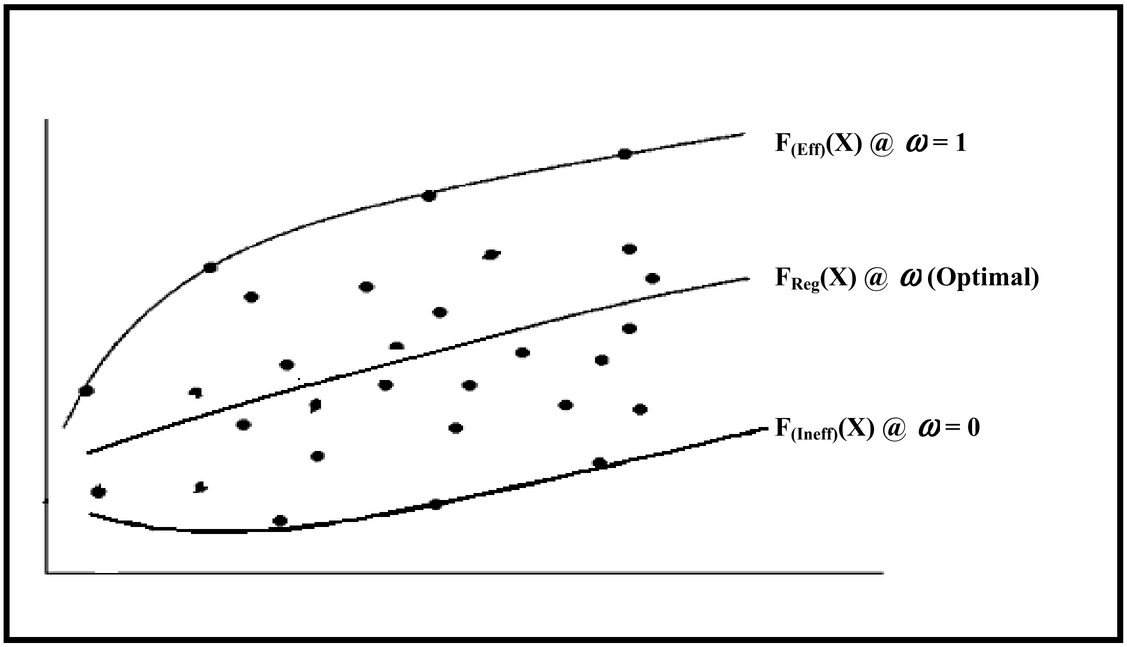


Figure 4.2 Simplified Diagram to Calculate $F_{Reg}(X)$ Parametrically for Model OLS1

For the second model OLS2, an average regression function $F_{Reg}(X)$ is obtained by fitting an ordinary least squares regression function for the original 49 (forty-nine) data pairs of which the x-input values are scaled between 0.000 and 1.000, and the y-ouput values between 0.200 and 0.800. Equation 4.3 for efficient frontier, and inefficient frontier are not utilized in this model for determining $F_{Reg}(X)$. This $F_{Reg}(X)$ determines the average regression function $F_{Reg}(X)$ found parametrically at $\lambda = 0$ for model OLS2 in this research and as illustrated in Figure 4.3.

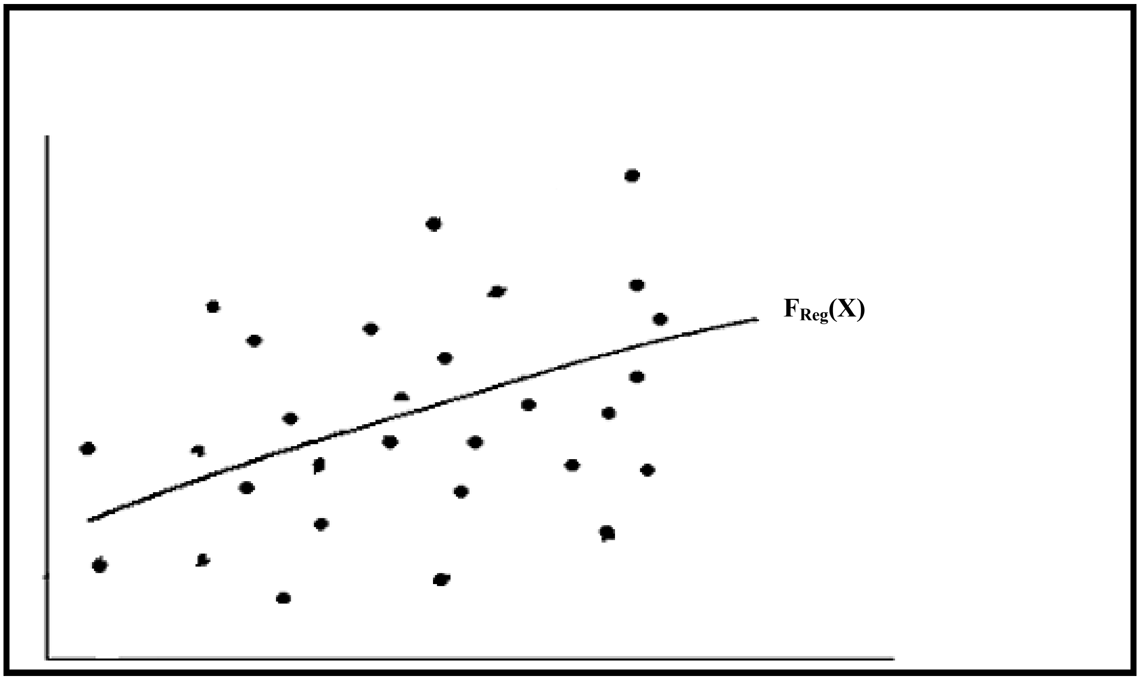


Figure 4.3 Simplified Diagram to Calculate $F_{Reg}(X)$ Parametrically for Model OLS2

4.1.4.3 Determining DEA's "true" Frontier Or Optimal Frontier at F_λ

DEA's "true" frontier is determined by varying the λ in the combined function in Equation 4.4 between the average function $F_{Reg}(X)$ at $\lambda = 0$, and the efficient frontier function $F_{(Eff)}(X)$ at $\lambda = 1$ for both neural networks and support vector machines and a ratio based on the statistics for the residuals (i.e. managerial and observational errors) is utilized to determine the optimal frontier. The ordinary least squares regression models created for comparison purposes also utilize Equation 4.4 and the same methodology as for neural networks and support vector machines in determining DEA's "true" frontier.

$$\mathbf{F_\lambda(X) = \lambda F_{(Eff)}(X) + (1-\lambda)F_{Reg}(X) \qquad \text{for } 0 \leq \lambda \leq 1 \qquad (4.4)}$$

4.2 Research Method for Determining Efficient Frontier $F_{(Eff)}$ DMUs

The low dimensional raw data set, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for the 49 (forty-nine) DMUs in Appendix A1 are utilized in DEA software or Equation 2.2 under the variable return to scale (VRS) assumption to determine the DMUs on the efficient frontier $F_{(Eff)}$. Those DMUs on the efficient frontier $F_{(Eff)}$ fulfilling the efficiency score criterion of being 1.00 (one) or 100% (one-hundred percent).

Since the goal is to train and test neural networks and support vector machines which require a large data set, the limitation imposed by the small data set of 49 data pairs for the DMUs, makes it necessary for the data set to be expanded. The DMUs determined to be on the efficient frontier by DEA's software or Equation 2.2 are used inside Minitab 12.23 and Datafit Version 8.2, so as to enable the efficient frontier data set to be expanded to a total of 20 (twenty) data pairs or DMUs. Each additional data pair or DMU obtained from the actual plotted DEA's piecewise efficient frontier by Minitab and Datafit is imputted inside the DEA's software or Equation 2.2 along with the original 49 (forty-nine) data pairs for the DMUs and is only added to the expanded data set if the efficiency score criterion of 1.00 (one) and monotonicity are met.

4.3 Research Method for Determining Inefficient Frontier $F_{(Ineff)}$ DMUs

The low dimensional raw data set, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for the 49 (forty-nine) DMUs in Appendix A1 are utilized in Equation 4.2 under the variable return to scale (VRS) assumption to determine the DMUs on the inefficient frontier $F_{(Ineff)}$. Those DMUs on the inefficient frontier $F_{(Ineff)}$ fulfilling the efficiency score criterion of being 1.00 (one) or 100% (one-hundred percent).

Since the goal is to train and test neural networks and support vector machines which require a large data set, the limitation imposed by the small data set of 49 data pairs for the DMUs, makes it necessary for the data set to be expanded. The DMUs determined to be on the inefficient frontier by Equation 4.2 are used inside Minitab 12.23 and Datafit Version 8.2, so as to enable the inefficient frontier data set to be expanded to a total of 20 (twenty) data pairs or DMUs. Each additional data pair or DMU obtained from the actual plotted DEA’s piecewise inefficient frontier by Minitab and Datafit is imputted inside Equation 4.2 along with the original 49 (forty-nine) data pairs for the DMUs and is only added to the expanded data set if the efficiency score criterion of 1.00 (one) and monotonicity are met.

4.4 Research Method for Machine Learning

The major challenges facing researchers when carrying out studies with machine learning software include the difficult and time-consuming “trial and error” process of choosing the right architecture of the neural network and in the support vector machines circumstance, determining the optimal values to use for the parameters.

This research utilizes Statistica Version 8.0 Machine Learning Module, which includes Statistica Automated Neural Networks (SANN), and Statistica Support Vector Machines (SSVM). Statistica, is a comprehensive application capable of designing a wide range of neural network architectures and support vector machine models by employing both, widely-utilized and highly-specialized training algorithms. It alleviates the challenges facing researchers which require the determination of the correct neural network architecture by specifically doing it all for the user. In the case of neural networks, Statistica includes traditional learning algorithms, such as backward propagation and sophisticated training algorithms such as Conjugate Gradient Descent and Levenberg-Marquardt iterative procedures, just to mention a few. It has an Intelligent Problem Solver that utilizes heuristics and sophisticated optimization strategies for determining the best neural network architecture and walks the researcher through a step-by-step analysis. The Intelligent Problem Solver compares different neural network types, such as linear, radial basis function, multi-layer perceptron, and bayesian networks, and determines the number of hidden nodes, as well as, chooses the smoothing factor in the case of radial basis function networks. In the case of support vector machines, the Intelligent Problem Solver will determine the optimal parameters to use for the data set.

As mentioned inside Section 4.1.2, the expanded data set in Section 4.2 and Section 4.3, for both efficient frontier DMUs and inefficient frontier DMUs require scaling or preprocessing for the successful training and testing of both neural network and support vector machine. As suggested by Wang (1992), the raw x-input values are scaled between 0.000 and 1.000, and the raw y-output values between 0.200 and 0.800.

The raw x-input values of the expanded data set totaling the 20 (twenty) DMUs or data pairs for efficient frontier $F_{(Eff)}$ in Section 4.2 are placed with the original raw x-input values for the original 49 (forty-nine) DMUs and are scaled between 0.000 and 1.000. The raw y-input values of the expanded data set totaling the 20 (twenty) DMUs or data pairs for efficient frontier $F_{(Eff)}$ are placed with the original raw x-input values for the original 49 (forty-nine) DMUs and are scaled between 0.200 and 0.800

The raw x-input values of the expanded data set totaling the 20 (twenty) DMUs or data pairs for inefficient frontier $F_{(Ineff)}$ in Section 4.3 are placed with the original raw x-input values for the original 49 (forty-nine) DMUs and are scaled between 0.000 and 1.000. The raw y-input values of the expanded data set totaling the 20 (twenty) DMUs or data pairs for inefficient frontier $F_{(Ineff)}$ are placed with the original raw x-input values for the original 49 (forty-nine) DMUs and are scaled between 0.200 and 0.800.

As mentioned in Section 4.1.2 for the neural network architecture, the number of hidden nodes is determined by the common heuristic used for smaller sample size that being twice the number of the input nodes+1.

Utilizing Statistica, the expanded scaled data set of Section 4.2 totaling the 20 (twenty) DMUs or data pairs for efficient frontier $F_{(Eff)}$ are randomly split into a 70% (seventy percent) set for training both neural network and support vector machines, and a 30% (thirty percent) set for testing both neural networks and support vector machines. The optimal neural network model in accordance with the hidden node heuristic for the expanded scaled data set, along with the optimal support vector machine model given by Statistica are then compared utilizing a sum of squares algorithm. The optimal neural network model for efficient frontier $F_{(Eff)}$ gives the function $F_{(Eff\text{-}NN)}$ (X) for efficient frontier at $\omega_{NN} = 1$, and the optimal support vector machine model for efficient frontier $F_{(Eff)}$ gives the function $F_{(Eff\text{-}SVM)}$ (X) for efficient frontier at $\omega_{SVM} = 1$.

Utilizing Statistica, the expanded scaled data set of Section 4.3 totaling the 20 (twenty) DMUs or data pairs for inefficient frontier $F_{(Ineff)}$ are randomly split into a 70% (seventy percent) set for training both neural networks and support vector machines, and a 30% (thirty percent) set for testing both neural networks and support vector machines. The optimal neural network model in accordance with the hidden node heuristic for the expanded scaled data set, along with the optimal support vector machine model given by Statistica are then compared utilizing a sum of squares algorithm. The optimal neural network model for inefficient frontier $F_{(Ineff)}$ gives the function $F_{(Ineff\text{-}NN)}$ (X) for inefficient frontier at ω_{NN}= 0, and the optimal support vector machine model for inefficient frontier $F_{(Ineff)}$ gives the function $F_{(Ineff\text{-}SVM)}$ (X) for inefficient frontier at ω_{SVM}= 0.

4.5 Research Method for OLS Regression Models for Comparison

Utilizing Minitab Version 12.23, the exact same 70% (seventy percent) scaled and randomly split data pairs used for training both neural network and support vector machines for the efficient frontier $F_{(Eff)}$ in Section 4.4 are used to create a linear and quadratic regression model. The model with the highest R^2 value is selected as the optimal regression model for efficient frontier $F_{(Eff)}$ and gives the function $F_{(Eff\text{-}OLS1)}$ (X) for efficient frontier at ω_{OLS1} = 1 for the OLS1 regression model, as well as, the function $F_{(Eff\text{-}OLS2)}$ (X) for efficient frontier at ω_{OLS2} = 1 for the OLS2 regression model. The exact same 30% (thirty percent) scaled and randomly split data pairs used for testing both neural network and support vector machines in for the efficient frontier $F_{(Eff)}$ in Section 4.4 are the exact same 30% (thirty percent) used to test the regression functions $F_{(Eff\text{-}OLS1)}$ (X) and $F_{(Eff\text{-}OLS2)}$ (X) for regression models OLS1 and OLS2 for efficient frontier $F_{(Eff)}$.

Utilizing Minitab Version 12.23, the exact same 70% (seventy percent) scaled and randomly split data pairs used for training both neural network and support vector machines for the inefficient frontier $F_{(Ineff)}$ in Section 4.4 are used to create a linear and quadratic regression model for the OLS1 regression model. The model with the highest R^2 value is selected as the optimal regression model for inefficient frontier $F_{(Ineff)}$ and gives the function $F_{(Ineff\text{-}OLS1)}$ (X) for inefficient frontier at $\omega_{OLS1} = 0$ for the OLS1 regression model. No inefficient frontier function is determine for the second regression model OLS2 as its average function is determined in the traditional regression manner as mentioned inside Section 4.1.4.2. The exact same 30% (thirty percent) scaled and randomly split data pairs used for testing both neural network and support vector machines for the inefficient frontier $F_{(Ineff)}$ in Section 4.4 are the exact same 30% (thirty percent) used to test the regression functions $F_{(Ineff\text{-}OLS1)}$ (X) for regression models OLS1 inefficient frontier $F_{(Ineff)}$. Utilizing a sum of squares algorithm as in Section 4.5 for neural networks and support vector machines, the regression model OLS1 is compared with the neural network and support vector machine models in Section 4.5.

4.6 Research Method for Frontier Analysis

4.6.1 Determining Average Regression Function F_{Reg}(X) Non-parametrically

This research utilizes a completely non-parametric approach to determine DEA's "true" frontier. In order to achieve this, an average regression function for neural network and support vector machine are obtained by combining the functions obtained by training and testing neural networks and support vector machines for non-parametric efficient frontier and non-parametric inefficient frontier as in Section 4.4. The average function for neural network $F_{Reg\text{-}NN}$(X) is achieved non-parametrically by varying the ω_{NN} in Equation 4.5 until a normal or approximate normal distribution of the residuals is realized. To assist in determining this, this is achieved when the sum of the residuals for all the data points of the left hand half and the right hand half of the distribution are equal zero or approximately zero. The ω_{NN} at which this is achieved, is the optimal ω_{NN} for $F_{Reg\text{-}NN}(X)$ and determines the average regression function obtained non-parametrically for neural network $F_{Reg\text{-}NN}(X)$ at $\lambda = 0$ for the research and as illustrated in Figure 4.4.

$$\mathbf{F_{Reg\text{-}NN}(X) = \omega_{NN} F_{(Eff\text{-}NN)}(X) + (1-\omega_{NN}) F_{(Ineff\text{-}NN)}(X) \quad \text{for } 0 \leq \omega_{NN} \leq 1} \qquad \textbf{(4.5)}$$

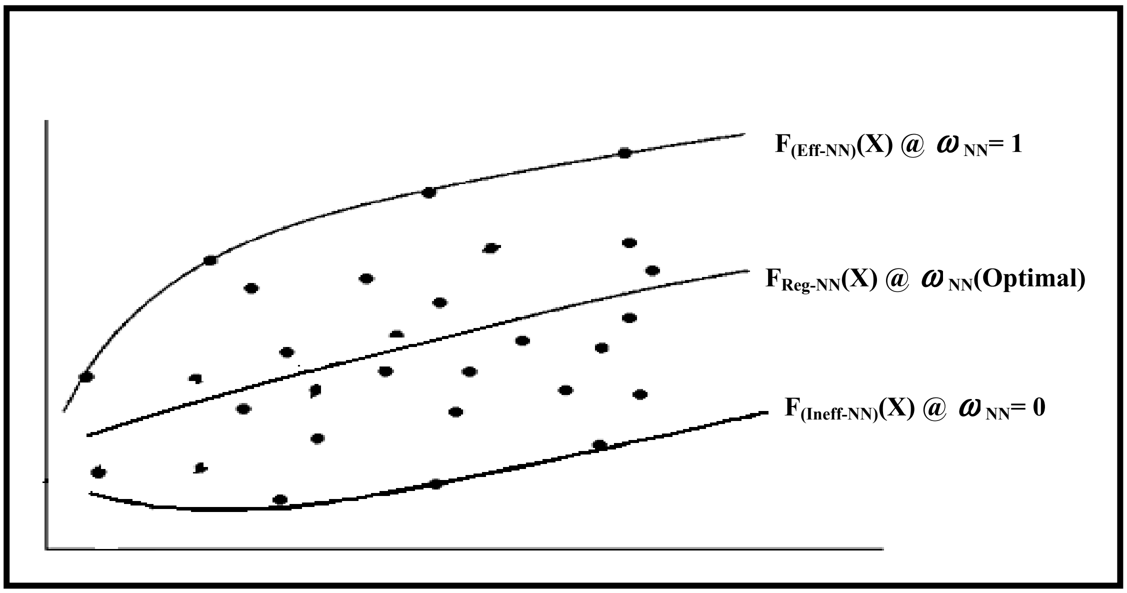


Figure 4.4 Simplified Diagram to Calculate $F_{Reg\text{-}NN}(X)$ Non-parametrically

The average function for support vector machine $F_{Reg\text{-}SVM}(X)$ is achieved non-parametrically by varying the ω_{SVM} in Equation 4.6 until a normal or approximate normal distribution of the residuals is realized. To assist in determining this, this is achieved when the sum of the residuals for all the data points of the left hand half and the right hand half of the distribution are equal zero or approximately zero. The ω_{SVM} at which this is achieved, is the optimal ω_{SVM} for $F_{Reg\text{-}SVM}(X)$ and determines the average regression function obtained non-parametrically for support vector machine $F_{Reg\text{-}SVM}(X)$ at $\lambda = 0$ for the research and as illustrated in Figure 4.5.

$$\mathbf{F_{Reg\text{-}SVM}(X) = \omega_{SVM} F_{(Eff\text{-}SVM)}(X) + (1-\omega_{SVM}) F_{(Ineff\text{-}SVM)}(X) \quad for\ 0 \leq \omega_{SVM} \leq 1} \qquad \textbf{(4.6)}$$

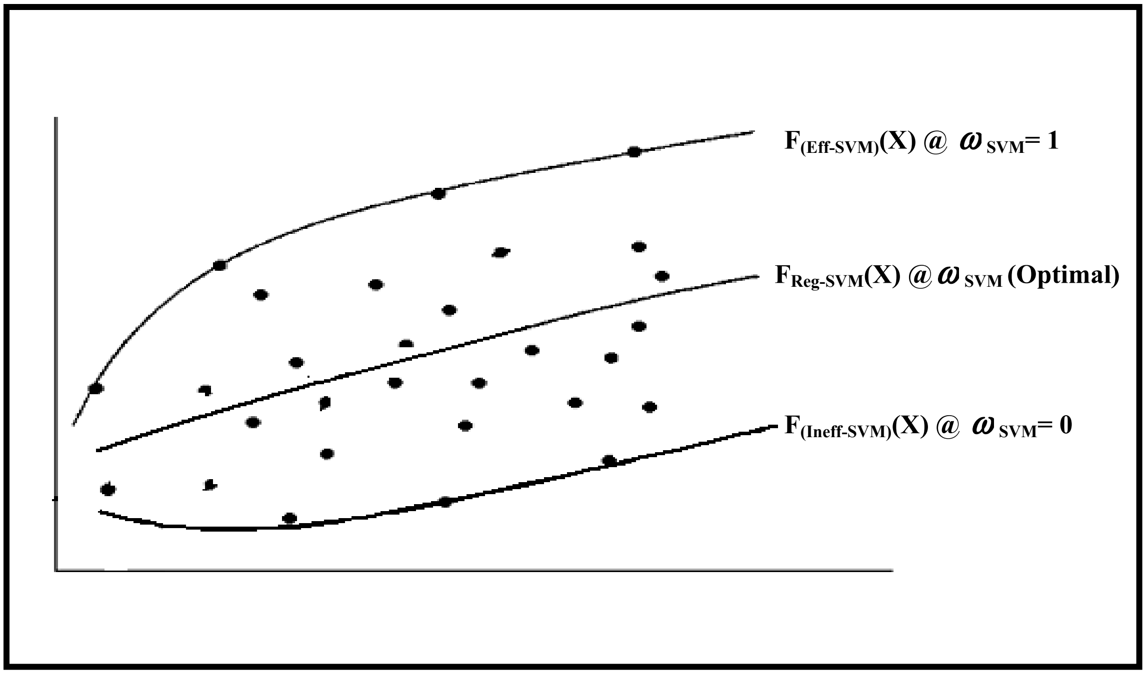


Figure 4.5 Simplified Diagram to Calculate $F_{Reg\text{-}SVM}(X)$ Non-parametrically

4.6.2 Determining Average Regression Function $F_{Reg}(X)$ Parametrically

As the two ordinary least squares regression models OLS1 and OLS2 created for comparison purposes are parametric in nature, the average function $F_{Reg}(X)$ for each of the models are referred to as being obtained parametrically as well.

For the first model OLS1, the average function $F_{Reg\text{-}OLS1}(X)$ is achieved parametrically by varying the ω_{OLS1} in Equation 4.7 until a normal or approximate normal distribution of the residuals is realized. To assist in determining this, this is achieved when the sum of the residuals for all the data points of the left hand half and the right hand half of the distribution are equal zero or approximately zero. The ω_{OLS1} at which this is achieved, is the optimal ω_{OLS1} for $F_{Reg\text{-}OLS1}(X)$ and determines the average regression function obtained parametrically for regression model OLS1 $F_{Reg\text{-}OLS1}(X)$ at $\lambda = 0$ for the research and as illustrated in Figure 4.6.

$$\mathbf{F_{Reg\text{-}OLS1}(X) = \omega_{OLS1} F_{(Eff\text{-}OLS1)}(X) + (1-\omega_{OLS1}) F_{(Ineff\text{-}OLS1)}(X) \quad for\ 0 \leq \omega_{OLS1} \leq 1} \quad \mathbf{(4.7)}$$

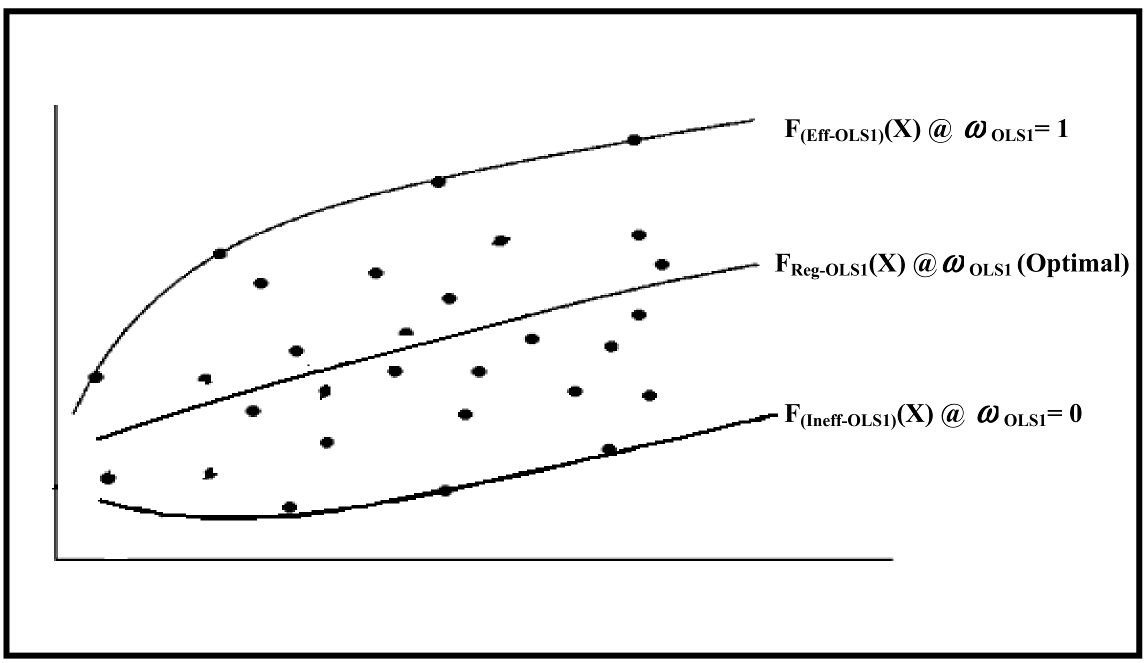


Figure 4.6 Simplified Diagram to Calculate $F_{Reg\text{-}OLS1}(X)$ Parametrically

For the second model OLS2, the average function $F_{Reg\text{-}OLS2}(X)$ is achieved parametrically by utilizing Minitab Version 12.23 to fit a linear and a quadratic regression curve to the original 49 (forty-nine) data pairs for the DMUs which are scaled as mentioned in Section 4.1.4.2**.** Whichever of the linear or quadratic model gives the highest R^2 value determines the average regression function obtained parametrically for regression model OLS2 $F_{Reg\text{-}OLS2}(X)$ at $\lambda = 0$ for the research and as illustrated in Figure 4.7.

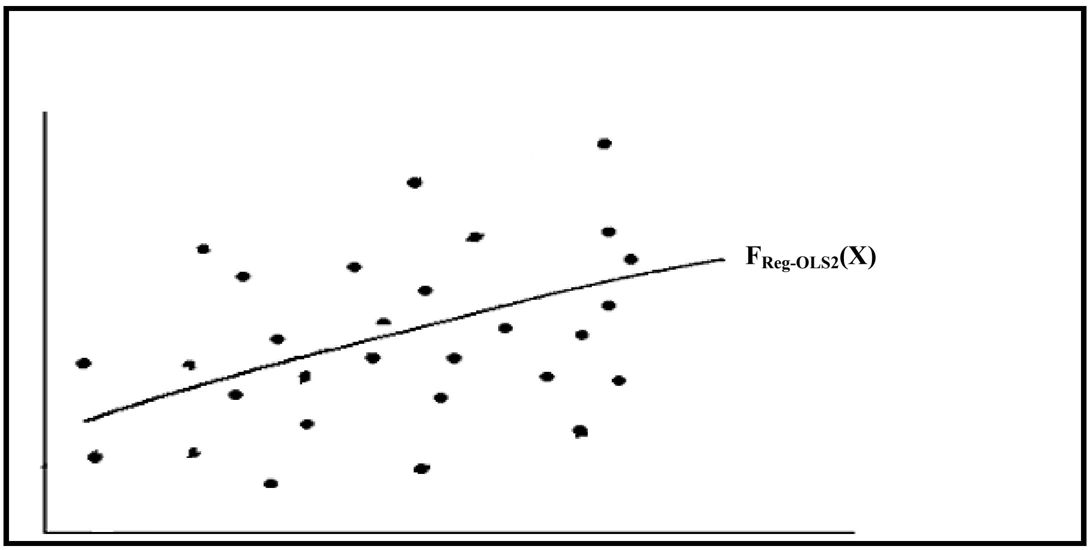


Figure 4.7 Simplified Diagram to Calculate $F_{Reg\text{-}OLS2}(X)$ Parametrically

4.6.3 Determining DEA's "true" Frontier Or Optimal Frontier at F_λ

DEA's "true" frontier using neural network is determined by combining the function in Section 4.4 for neural network $F_{(Eff\text{-}NN)}(X)$ for efficient frontier at $\lambda_{NN} = 1$ with the average function obtained non-parametrically for neural network $F_{Reg\text{-}NN}(X)$ at $\lambda_{NN} = 0$ in Section 4.6.1 and as is represented in Equation 4.8.

$$\mathbf{F_{\lambda\text{-}NN}(X) = \lambda_{NN}\, F_{(Eff\text{-}NN)}(X) + (1\text{-}\lambda_{NN})F_{Reg\text{-}NN}(X) \qquad for\ 0 \le \lambda_{NN} \le 1} \tag{4.8}$$

DEA's "true" frontier using support vector machine is determined by combining the function in Section 4.4 for support vector machine $F_{(Eff\text{-}SVM)}$ (X) for efficient frontier at λ_{SVM}= 1 with the non-parametric average function for support vector machine $F_{Reg\text{-}SVM}$(X) at λ_{SVM} = 0 in Section 4.6.1 and as is represented in Equation 4.9.

$$\mathbf{F_{\lambda\text{-}SVM}(X) = \lambda_{SVM}\, F_{(Eff\text{-}SVM)}(X) + (1\text{-}\lambda_{SVM})F_{Reg\text{-}SVM}(X) \qquad for\ 0 \le \lambda_{SVM} \le 1} \tag{4.9}$$

Determining DEA's "true" frontier for the two regression models created for comparison purposes utilizes similar equations to Equation 4.8 and Equation 4.9 for neural network and support vector machine.

For regression model OLS1, DEA's "true" frontier is determined by combining the function in Section 4.5 $F_{(Eff\text{-}OLS1)}$ (X) for efficient frontier at λ_{OLS1}= 1 with the parametric average function $F_{Reg\text{-}OLS1}$(X) at λ_{OLS1} = 0 in Section 4.6.2 and as is represented in Equation 4.10.

$$\mathbf{F_{\lambda\text{-}OLS1}(X) = \lambda_{OLS1}\, F_{(Eff\text{-}OLS1)}(X) + (1\text{-}\lambda_{OLS1})F_{Reg\text{-}OLS1}(X) \qquad for\ 0 \le \lambda_{OLS1} \le 1} \tag{4.10}$$

For regression model OLS2, DEA's "true" frontier is determined by combining the function in Section 4.5 $F_{(Eff\text{-}OLS2)}$ (X) for efficient frontier at λ_{OLS2}= 1 with the parametric average function $F_{Reg\text{-}OLS2}(X)$ at $\lambda_{OLS2} = 0$ in Section 4.6.2 and as is represented in Equation 4.11.

$$\mathbf{F_{\lambda\text{-}OLS2}(X) = \lambda_{OLS2}\, F_{(Eff\text{-}OLS2)}(X) + (1\text{-}\lambda_{OLS2}) F_{Reg\text{-}OLS2}(X) \quad \text{for } 0 \leq \lambda_{OLS2} \leq 1 \qquad (4.11)}$$

Under the assumptions that the managerial errors (***u***) are a positive half-normal distribution, and the observational errors (***v***) are a normal distribution. The λ value in Equation 4.8, Equation 4.9, Equation 4.10, and Equation 4.11 for neural network, support vector machine, OLS1 regression model, and OLS2 regression model respectively are reduced in steps of 0.01 from λ= 1 to λ = 0 and a ratio based on the statistics of the residuals (i.e. managerial and observational errors) is utilized to determine DEA's "true" or optimal frontier for the neural network and support vector machine methodologies, as well as, the regression models OLS1 and OLS2 used in this research for comparison purposes. Figure 4.8 illustrates how the ratio is to be applied to the managerial and observational errors.

- Observational error (v) may be positive or negative with equal likelihood. Therefore, they can be captured by symmetric distribution
- Managerial error (u) always causes costs to be higher. Therefore, it can be captured by an asymmetric (half-normal) distribution
- Sum of the two distributions. Its degree of symmetry reveals the relative importance of the observational error

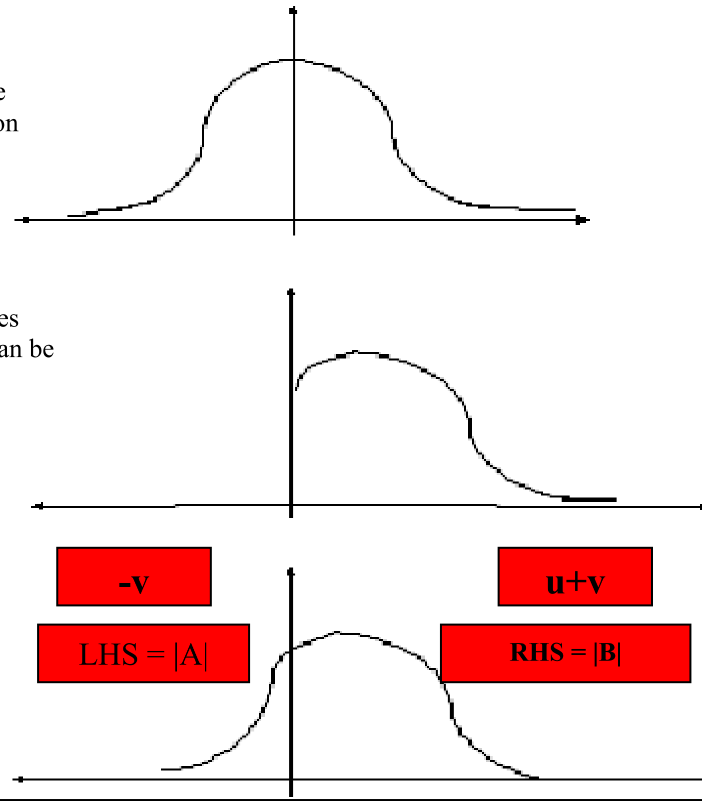


Figure 4.8 Application of Ratio to Managerial and Observational Errors

As λ is varied in steps of 0.01, for each value of λ, the ratio of **|A|/|B|** is plotted relative to λ for the neural network, support vector machine, OLS1, and OLS2 case. Starting at λ=1.00, the plot should initially depict an almost constant curve or slope then eventually a profound shift in slope. The ratio statistic should also be constant or repetitive starting from λ=1.00 then its constant or repetitive characteristic should change. When the shift is observed, the λ value at which it occurs is noted as the optimal lambda, thus conveying that DEA's optimal frontier or "true" frontier has been achieved. Therefore for the residuals at λ = 0 (i.e. managerial errors (***u***) = 0; at an approximate normal/normal distribution) for the ratio statistic used, |A|/|B| should be equal to approximately 1.00 (one) for the neural network, support vector machine, OLS1, and OLS2 case.

4.7 Summary of Chapter Four

In this chapter, a detailed methodology was developed and provided for determining DEA "true" or optimal frontiers non-parametrically for neural networks and support vector machines, as well as, parametrically for two ordinary least squares regression models, OLS1 and OLS2. While the focus of the research includes utilizing SVM to estimate or predict nonlinear functions for DEA efficient and inefficient frontiers, the neural network, and OLS models were incorporated into the research for comparison purposes.

An illustrative example was given, introducing DEA inefficient frontier and explaining how the efficiency score is determined for each DMU by modifying Equation 2.1 and Equation 2.2 for the CRS and VRS cases respectively. We learnt that the efficiency score of those DMUs being on the inefficient frontier must fulfill a criterion score of 1.00.

Owing to the small size of the real data set to be utilize on the proposed methodology. Also, since SVM and NN require a reasonable amount of data pairs to be trained and tested, a detailed outline on how the data set for efficient and inefficient frontiers are to be expanded utilizing Datafit 8.2 and Minitab 12.23 was given. The importance of allowing Statistic 8.0 initially to randomly select 70% (seventy percent) of the expanded data set to train the NN and 30% (thirty percent) for testing the NN was pointed out. The importance of utilizing the exact same 70% (seventy percent) and 30% (thirty percent) data set for the training and testing respectively of SVM, and the OLS models for both efficient and inefficient frontiers was emphasized.

The general Equation 4.3 was given for calculating the average regression function $F_{Reg}(X)$ non-parmetrically for NN and SVM, as well as, the OLS model OLS1 whereas, for OLS2, the average regression function was specified to be calculated utilizing the traditional OLS regression method on the original 49 (forty-nine) data pairs after they are scaled. The equations for calculating the average regression function specifically for NN, SVM, and OLS1 were given by Equations 4.5, 4.6, and 4.7 respectively.

The chapter concluded by giving Equation 4.4 as a general equation for calculating DEA's "true" frontier for NN, SVM, OLS1, and OLS2 utilizing a ratio based on the statistics for managerial error and observational errors. The equations for calculating DEA's "true" or optimal frontier specifically for NN, SVM, OLS1, and OLS2 were given by Equations 4.8, 4.9, 4.10, and 4.11 respectively.

5. CHAPTER FIVE: ANALYSIS AND RESULTS

This chapter contains the results for applying a real data set to the proposed methodology developed in Chapter Four for determining DEA's "true" or optimal frontier. Although, the central focus of the research incorporates DEA with SFF and utilizes support vector machines, this chapter also includes the results for neural network, and two OLS models, OLS1 and OLS2 for comparison purposes. The chapter also includes a detailed analysis of all four models, those being, SVM, NN, OLS1, and OLS2.

5.1 Overview

The methodology outlined in Chapter Four is applied to the data set from the original study 'Program Follow Through' by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students and may be viewed in Appendix A1. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), with a total of 49 (forty-nine) data pairs in its entirety are used under DEA's assumption of variable return to scale. The scatter plot for the data set may be viewed in Figure 5.1.

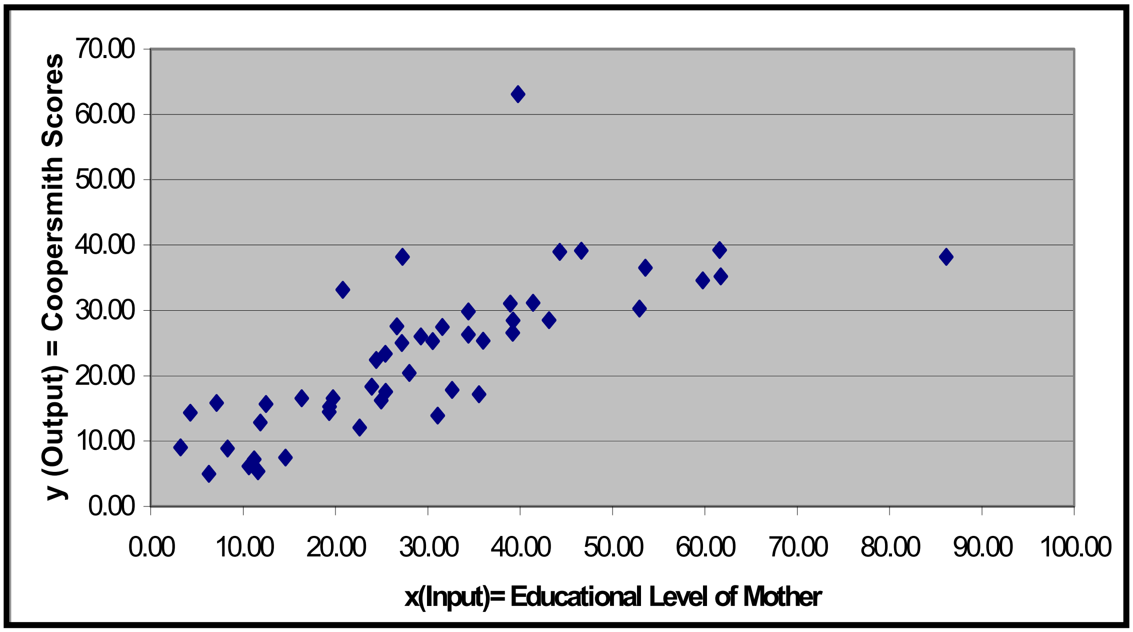


Figure 5.1 Scatter Plot for Original 49 DMUs Input-Output Low Dimensional Data Pairs

Section 5.2 assigns the efficiency scores for the original 49 DMUs for traditional DEA or efficiency frontier under the VRS case. The DMUs on the DEA's efficient frontier, $F_{(Eff)}$ for the 49 (forty-nine) data pairs are determined under the VRS case by linear programming as in Equation 2.2 and by DEA software, with the efficiency score criterion for these DMUs being 1.00 (one) or 100% (one-hundred percent). Using Minitab Version 12.23 and Datafit Version 8.2, the DMU set on the efficient frontier is extended in order to achieve the expanded data set of 20 DMUs total, so as to facilitate the training and testing of neural network and support vector machine, as well as, for the creation of two parametric ordinary least squares regression models, OLS1 and OLS2 in Section 5.5 for comparison purposes.

The efficiency scores for the 49 DMUs based upon DEA's inefficient frontier and under the VRS case are determined in Section 5.3. The DMUs on the DEA's inefficient frontier, $F_{(Ineff)}$ for the 49 (forty-nine) data pairs are also given under the VRS case by modifying the linear programming in Equation 2.2 and as given in Equation 4.2, where the efficiency score criterion for these DMUs is also denoted by 1.00 (one) or 100% (one-hundred percent). Utilizing Minitab Version 12.23 and Datafit Version 8.2, the DMU set on the inefficient frontier is extended in order to achieve the expanded data set of 20 DMUs total for the inefficient frontier, so as to enabled the training and testing of neural network and support vector machine in Section 5.4, as well as, for the creation of the parametric ordinary least squares regression model OLS1 in Section 5.5 for comparison purposes.

In Section 5.6.1, the average functions for neural network $F_{Reg\text{-}NN}(X)$, and support vector machine $F_{Reg\text{-}SVM}(X)$ are determined non-parametrically by varying the ω in Equation 4.5 and Equation 4.6 respectively until a normal or approximate normal distribution of the residuals is realized (i.e. sum of residuals = 0). This is determined to occur at an optimal ω which is also synonymous to or set to $\lambda = 0$ and is used in the determination of DEA's "true" frontier in Section 5.6.3.

In Section 5.6.2, the average functions for the two parametric ordinary least squares regression models OLS1 and OLS2 are determined. For OLS1, the average function $F_{Reg\text{-}OLS1}(X)$ is determined parametrically by varying the ω in Equation 4.7 until a normal or approximate normal distribution of the residuals is realized (i.e. sum of residuals = 0). This is determined to occur at an optimal ω which is also synonymous to or set to $\boldsymbol{\lambda = 0}$ and is used in the determination of DEA's "true" frontier in Section 5.6.3. For OLS2, the average function $F_{Reg-OLS2}(X)$ is determined parametrically by fitting an optimal regression curve to the original 49 (forty-nine) data pairs which have been scaled for the research.

In Section 5.6.3, DEA's "true" frontier for the neural network and support vector machine methodologies utilized for this research are determined. The λ value in both Equation 4.8 and Equation 4.9 are reduced in steps of 0.01, from $\lambda = 0$ to $\lambda = 1$ and a ratio based on the statistics for the residuals (i.e. managerial and observational errors) is utilized to determine DEA's "true" or optimal frontier for neural network and support vector machine.

The "true" frontier for the two parametric ordinary least squares regression models OLS1 and OLS2, are also determined for comparison purposes. As performed for neural network and support vector machine, the λ value in both Equation 4.10 and Equation 4.11 are reduced in steps of 0.01, from $\lambda = 0$ to $\lambda = 1$ and a ratio based on the statistics for the residuals (i.e. managerial and observational errors) is utilized to determine DEA's "true" or optimal frontier for both parametric models.

The chapter concludes with employing the probability density function and the area under the curve statistics, in order to perform benchmarking to assess the performance of the ratio statistic method used for the research.

5.2 Efficiency Scores of All DMUs and DMUs On the Efficient Frontier $F_{(Eff)}$

The low dimensional raw data set, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for the 49 (forty-nine) DMUs in Appendix A1 were entered into both EMS DEA's software and Equation 2.2 under the variable return to scale (VRS) assumption to determine the efficiency scores of all 49 DMUs and the DMUs on the efficient frontier $F_{(Eff)}$. Those DMUs on the efficient frontier $F_{(Eff)}$ fulfilling the efficiency score criterion of being 1.00 (one) or 100% (one-hundred percent) are recorded in Table 5.1 and the efficiency scores for all 49 DMUs may be viewed in Appendix F1.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	Efficiency Scores
DMU15	4.29	14.33	1.00
DMU44	39.79	63.11	1.00
DMU48	3.24	9.02	1.00

Table 5.1 DMUs On Efficient Frontier $F_{(Eff)}$ and Efficiency Scores for VRS Case

Since the goal is to train and test neural networks and support vector machines with DMUs on the efficient frontier, which requires a large data set, the three DMUs on the efficient frontier in Table 5.1, those being DMU15, DMU44, and DMU44 require expansion to a total of twenty DMUs. In other words, 17 (seventeen) additional data pairs or DMUs has to be created to expand the existing three DMUs to a total set of twenty DMUs on the efficient frontier $F_{(Eff)}$.

In order to enable the three DMUs on the efficient frontier in Table 5.1 to be expanded to a total of twenty DMUs, Minitab Version 12.23 and Datafit Version 8.2 curve fitting and plotting softwares were utilized. The x-input and y-output data pairs for DMU15, DMU44, and DMU48 were entered into each software so as to produce the DEA piecewise efficient frontier for the three DMUs so as to allow for the expanded data set.

The DEA piecewise efficient frontier for the three DMUs' data may be viewed in Figure 5.2.

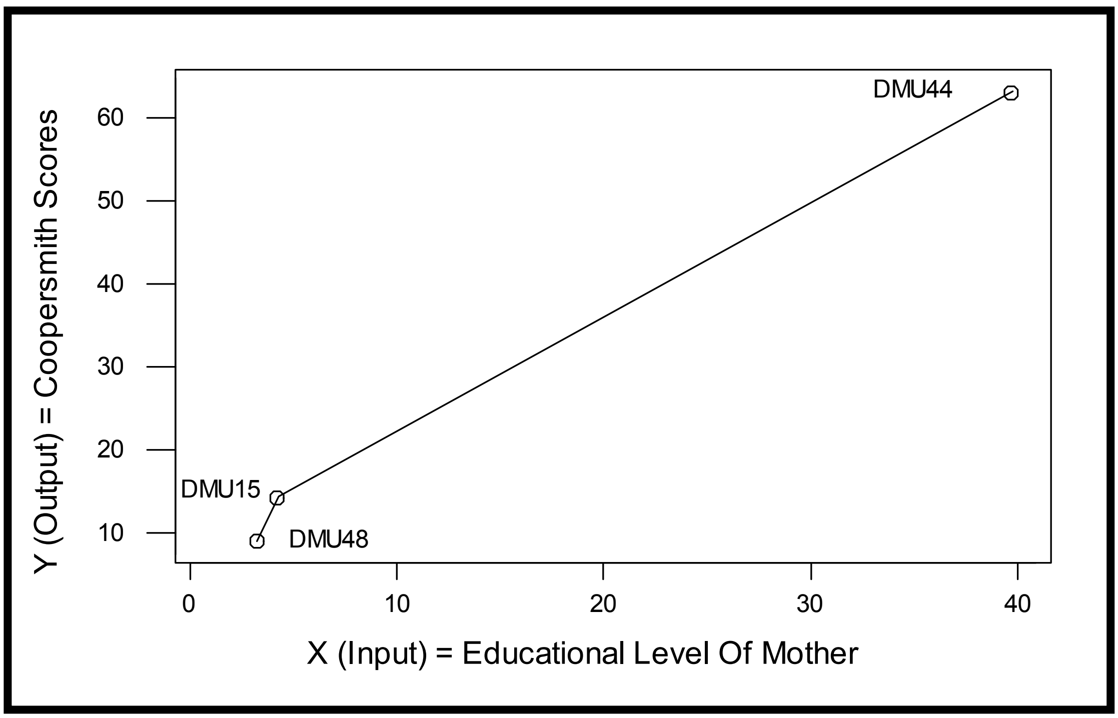


Figure 5.2 DEA Piecewise Efficient Frontier For 3 DMUs On Efficient Frontier

Using Figure 5.2 and Datafit Version 8.2 pointer co-ordinate feature, the three DMU set on the efficient frontier was expanded to a total of twenty DMUs, making certain that for each of the (17) seventeen additional DMU or data pair added, each met the efficiency score criterion of 1.00 when placed into Equation 2.2 relative to the original 49 (forty-nine) DMUs. Also making certain that monotonicity was maintained for the complete expanded data set consisting of the total twenty DMUs on the efficient frontier and as is in Table 5.2.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	Efficiency Scores	Original data Or Expanded data
DMU15	4.29	14.33	1.00	Original data
DMU44	39.79	63.11	1.00	Original data
DMU48	3.24	9.02	1.00	Original data
$DMU1_{Eff\text{-}expanded}$	3.75	11.98	1.00	Expanded data
$DMU2_{Eff\text{-}expanded}$	5.75	16.35	1.00	Expanded data
$DMU3_{Eff\text{-}expanded}$	7.08	18.29	1.00	Expanded data
$DMU4_{Eff\text{-}expanded}$	8.06	19.55	1.00	Expanded data
$DMU5_{Eff\text{-}expanded}$	10.54	22.93	1.00	Expanded data
$DMU6_{Eff\text{-}expanded}$	12.15	25.22	1.00	Expanded data
$DMU7_{Eff\text{-}expanded}$	15.27	29.42	1.00	Expanded data
$DMU8_{Eff\text{-}expanded}$	18.12	33.42	1.00	Expanded data
$DMU9_{Eff\text{-}expanded}$	20.28	36.31	1.00	Expanded data
$DMU10_{Eff\text{-}expanded}$	22.55	39.49	1.00	Expanded data
$DMU11_{Eff\text{-}expanded}$	23.67	40.99	1.00	Expanded data
$DMU12_{Eff\text{-}expanded}$	25.03	42.88	1.00	Expanded data
$DMU13_{Eff\text{-}expanded}$	26.67	45.09	1.00	Expanded data
$DMU14_{Eff\text{-}expanded}$	28.70	47.88	1.00	Expanded data
$DMU15_{Eff\text{-}expanded}$	30.75	50.69	1.00	Expanded data
$DMU16_{Eff\text{-}expanded}$	33.37	54.29	1.00	Expanded data
$DMU17_{Eff\text{-}expanded}$	37.01	59.30	1.00	Expanded data

Table 5.2 Expanded Data Set Of N = 20 DMUs/Data Pairs On Efficient Frontier $F_{(Eff)}$

5.3 Efficiency Scores of All DMUs and DMUs On the Inefficient Frontier $F_{(Ineff)}$

The low dimensional raw data set, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), for the 49 (forty-nine) DMUs in Appendix A1 were entered into Equation 4.2 under the variable return to scale (VRS) assumption to determine the efficiency scores of all 49 DMUs and the DMUs on the inefficient frontier $F_{(Ineff)}$. Those DMUs on the inefficient frontier $F_{(Ineff)}$ fulfilling the efficiency score criterion of being 1.00 (one) or 100% (one-hundred percent) are recorded in Table 5.3 and the efficiency scores for all 49 DMUs may be viewed in Appendix F2**.**

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	Efficiency Scores
DMU1	86.13	38.16	1.00
DMU5	11.62	5.37	1.00
DMU32	6.30	4.99	1.00
DMU36	31.08	13.91	1.00

Table 5.3 DMUs On Inefficient Frontier $F_{(Ineff)}$ and Efficiency Scores for VRS Case

Since the goal is to train and test neural networks and support vector machines with DMUs on the inefficient frontier which requires a large data set, the four DMUs on the efficient frontier in Table 5.3, those being DMU1, DMU5, DMU32, and DMU36 require expansion to a total of twenty DMUs. In other words, 16 (sixteen) additional data pairs or DMUs has to be created to expand the existing four DMUs to a total set of twenty DMUs on the inefficient frontier $F_{(Ineff).}$

In order to enable the four DMUs on the inefficient frontier in Table 5.3 to be expanded to a total of twenty DMUs, Minitab Version 12.23 and Datafit Version 8.2 curve fitting and plotting softwares were utilized. The x-input and y-output data pairs for DMU1, DMU5, DMU32, and DMU36 were entered into each software so as to produce the DEA piecewise inefficient frontier for the four DMUs so as to allow for the expanded data set. The DEA piecewise inefficient frontier for the four DMUs' data may be viewed in Figure 5.3.

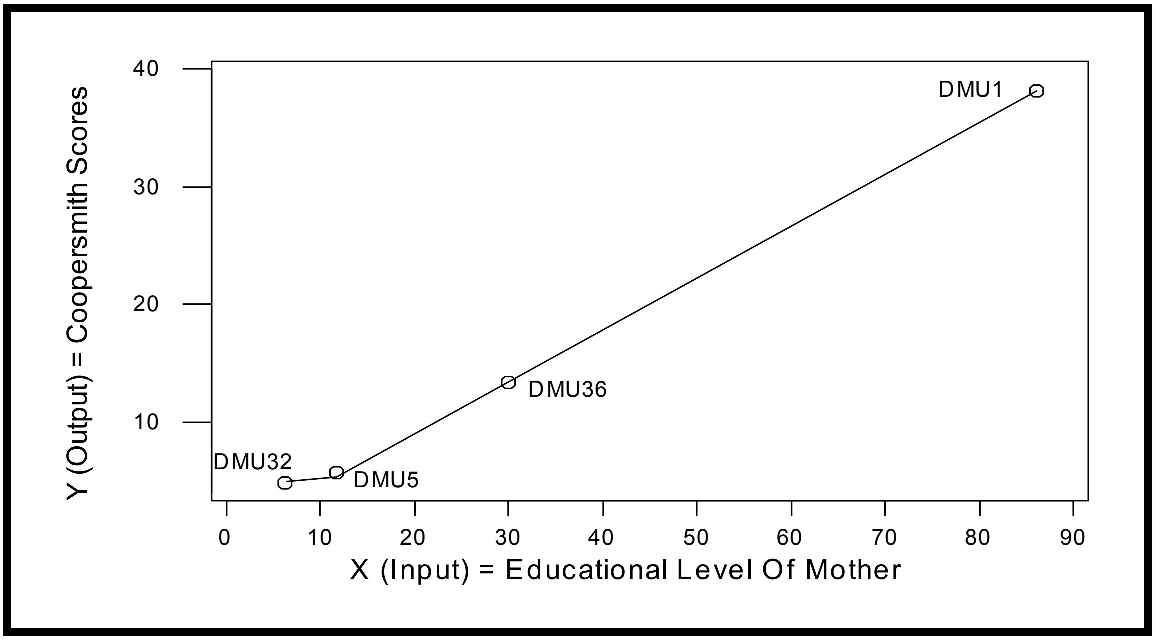


Figure 5.3 DEA Piecewise Inefficient Frontier For 4 DMUs On Inefficient Frontier

Using Figure 5.3 and Datafit Version 8.2 pointer coordinate feature, the four DMU set on the inefficient frontier was expanded to a total of twenty DMUs, making certain that for each of the (16) sixteen additional DMU or data pair added, each met the efficiency score criterion of 1.00 when placed into Equation 4.2 relative to the original 49 (forty-nine) DMUs. Also making certain that monotonicity was maintained for the complete expanded data set consisting of the total twenty DMUs on the inefficient frontier and as is in Table 5.4.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	Efficiency Scores	Original data Or Expanded data
DMU1	86.13	38.16	1.00	Original data
DMU5	11.62	5.37	1.00	Original data
DMU32	6.30	4.99	1.00	Original data
DMU36	31.08	13.91	1.00	Original data
$DMU1_{Ineff\text{-}expanded}$	7.39	5.05	1.00	Expanded data
$DMU2_{Ineff\text{-}expanded}$	8.73	5.13	1.00	Expanded data
$DMU3_{Ineff\text{-}expanded}$	13.53	6.20	1.00	Expanded data
$DMU4_{Ineff\text{-}expanded}$	15.74	7.17	1.00	Expanded data
$DMU5_{Ineff\text{-}expanded}$	19.77	8.94	1.00	Expanded data
$DMU6_{Ineff\text{-}expanded}$	24.56	11.03	1.00	Expanded data
$DMU7_{Ineff\text{-}expanded}$	32.53	14.54	1.00	Expanded data
$DMU8_{Ineff\text{-}expanded}$	35.55	15.87	1.00	Expanded data
$DMU9_{Ineff\text{-}expanded}$	37.90	16.91	1.00	Expanded data
$DMU10_{Ineff\text{-}expanded}$	39.43	17.58	1.00	Expanded data
$DMU11_{Ineff\text{-}expanded}$	42.12	18.76	1.00	Expanded data
$DMU12_{Ineff\text{-}expanded}$	47.11	20.96	1.00	Expanded data
$DMU13_{Ineff\text{-}expanded}$	52.10	23.15	1.00	Expanded data
$DMU14_{Ineff\text{-}expanded}$	59.58	26.46	1.00	Expanded data
$DMU15_{Ineff\text{-}expanded}$	68.03	30.18	1.00	Expanded data
$DMU16_{Ineff\text{-}expanded}$	78.57	34.82	1.00	Expanded data

Table 5.4 Expanded Data Set Of N = 20 DMUs/Data Pairs On Inefficient Frontier $F_{(Ineff)}$

5.4 Machine Learning

5.4.1 Preprocessing and Scaling of Data for NN and SVM

As mentioned in Section 4.4, the expanded data set of N = 20 DMUs/data pairs on the efficient frontier $F_{(Eff)}$ in Table 5.2 in Section 5.2 were scaled so as to facilitate the satisfactory training and testing of both neural network and support vector machine. The x-input values of the of the 17 (seventeen) additional data pairs on the efficient frontier $F_{(Eff)}$ were combined with the original 49 (forty-nine) x-input values and scaled between 0.000 (zero) and 1.000 (one). The y-input values of the of the 17 (seventeen) additional data pairs on the efficient frontier $F_{(Eff)}$ were combined with the original 49 (forty-nine) y-input values and scaled between 0.200 and 0.800. The resultant expanded scaled 20 (twenty) data pairs for the efficient frontier $F_{(Eff)}$ may be viewed in Table 5.5.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	Efficiency Scores	Original data Or Expanded data
DMU15	0.013	0.296	1.00	Original data
DMU44	0.441	0.800	1.00	Original data
DMU48	0.000	0.242	1.00	Original data
$DMU1_{Eff\text{-}expanded}$	0.006	0.272	1.00	Expanded data
$DMU2_{Eff\text{-}expanded}$	0.030	0.317	1.00	Expanded data
$DMU3_{Eff\text{-}expanded}$	0.046	0.337	1.00	Expanded data
$DMU4_{Eff\text{-}expanded}$	0.058	0.350	1.00	Expanded data
$DMU5_{Eff\text{-}expanded}$	0.088	0.385	1.00	Expanded data
$DMU6_{Eff\text{-}expanded}$	0.107	0.409	1.00	Expanded data
$DMU7_{Eff\text{-}expanded}$	0.145	0.452	1.00	Expanded data
$DMU8_{Eff\text{-}expanded}$	0.180	0.493	1.00	Expanded data
$DMU9_{Eff\text{-}expanded}$	0.206	0.523	1.00	Expanded data
$DMU10_{Eff\text{-}expanded}$	0.233	0.556	1.00	Expanded data
$DMU11_{Eff\text{-}expanded}$	0.246	0.572	1.00	Expanded data
$DMU12_{Eff\text{-}expanded}$	0.263	0.591	1.00	Expanded data
$DMU13_{Eff\text{-}expanded}$	0.283	0.614	1.00	Expanded data
$DMU14_{Eff\text{-}expanded}$	0.307	0.643	1.00	Expanded data
$DMU15_{Eff\text{-}expanded}$	0.332	0.672	1.00	Expanded data
$DMU16_{Eff\text{-}expanded}$	0.363	0.709	1.00	Expanded data
$DMU17_{Eff\text{-}expanded}$	0.407	0.761	1.00	Expanded data

Table 5.5 Expanded Scaled Data Set Of 20 DMUs/Data Pairs On Efficient Frontier $F_{(Eff)}$

As mentioned in Section 4.4, the expanded data set of N = 20 DMUs/data pairs on the inefficient frontier $F_{(Ineff)}$ in Table 5.4 in Section 5.3 were scaled so as to facilitate the satisfactory training and testing of both neural network and support vector machine. The x-input values of the of the 16 (sixteen) additional data pairs on the inefficient frontier $F_{(Ineff)}$ were combined with the original 49 (forty-nine) x-input values and scaled between 0.000 (zero) and 1.000 (one). The y-input values of the of the 16 (sixteen) additional data pairs on the inefficient frontier $F_{(Ineff)}$ were combined with the original 49 (forty-nine) y-input values and scaled between 0.200 and 0.800. The resultant expanded scaled 20 (twenty) data pairs for the inefficient frontier $F_{(Ineff)}$ may be viewed in Table 5.6.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	Efficiency Scores	Original data Or Expanded data
DMU1	1.000	0.542	1.00	Original data
DMU5	0.101	0.204	1.00	Original data
DMU32	0.037	0.200	1.00	Original data
DMU36	0.336	0.292	1.00	Original data
$DMU1_{Ineff\text{-}expanded}$	0.050	0.201	1.00	Expanded data
$DMU2_{Ineff\text{-}expanded}$	0.066	0.201	1.00	Expanded data
$DMU3_{Ineff\text{-}expanded}$	0.124	0.212	1.00	Expanded data
$DMU4_{Ineff\text{-}expanded}$	0.151	0.223	1.00	Expanded data
$DMU5_{Ineff\text{-}expanded}$	0.199	0.241	1.00	Expanded data
$DMU6_{Ineff\text{-}expanded}$	0.257	0.262	1.00	Expanded data
$DMU7_{Ineff\text{-}expanded}$	0.353	0.299	1.00	Expanded data
$DMU8_{Ineff\text{-}expanded}$	0.390	0.312	1.00	Expanded data
$DMU9_{Ineff\text{-}expanded}$	0.418	0.323	1.00	Expanded data
$DMU10_{Ineff\text{-}expanded}$	0.437	0.330	1.00	Expanded data
$DMU11_{Ineff\text{-}expanded}$	0.469	0.342	1.00	Expanded data
$DMU12_{Ineff\text{-}expanded}$	0.529	0.365	1.00	Expanded data
$DMU13_{Ineff\text{-}expanded}$	0.589	0.387	1.00	Expanded data
$DMU14_{Ineff\text{-}expanded}$	0.680	0.422	1.00	Expanded data
$DMU15_{Ineff\text{-}expanded}$	0.782	0.460	1.00	Expanded data
$DMU16_{Ineff\text{-}expanded}$	0.909	0.508	1.00	Expanded data

Table 5.6 Expanded Scaled Data Set Of 20 DMUs/Data Pairs On Inefficient Frontier $F_{(Ineff)}$

5.4.2 Efficient Frontiers and their Functions for NN and SVM

5.4.2.1 Efficient Frontier and Function for NN

Using the common heuristic for determining the neural network architecture in Section 4.4 for the single input and single output data pairs used in this research, the neural network architecture was determined to be 1-3-1, that being, 1(one) input node, 3 (three) hidden nodes, and 1 (one) output node.

Utilizing Statistica 8.0 as mentioned in Section 4.4, the 20 (twenty) expanded scaled data pairs on the efficient frontier $F_{(Eff)}$ in Table 5.5 were entered into the Statistica Automated Neural Network Module. The interactive interface was set to randomly select 70% (seventy percent – 14 data pairs) of the expanded scaled data set for training the neural network, and randomly select 30% (thirty percent – 6 data pairs) of the expanded scaled data set for testing the neural network. The input node was set at 1 (one), the hidden node at 3 (three), and the output node at 1 (one). The software was ran to generate the optimal neural network model for the randomly selected training and testing data pairs and the specifications entered and shortly thereafter, outputted the optimal model or function for the efficient frontier for neural network
$F_{(Eff\text{-}NN)}$ which was saved. The optimal model and function along with its parameters given by Statistica 8.0 for efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ were as follows:

1. The neural network architecture was a three layer, 1-3-1 multi-layer perceptron model.
2. The training algorithm which gave the optimal neural network was BFGS (i.e. Quasi-Newton Back Propagation by Broyden, Fletcher, Goldfarb, and Shanno (BFGS)) (Hagen, Demuth, and Beale, 1996; and Haykin, 2000).
3. The hidden activation was Tanh, and the output activation Tanh.
4. The error function was sum of squares (SOS).

The 14 (fourteen) data pairs or 70% (seventy percent) of the data pairs in Table 5.5 which were randomly selected by SANN to train the neural network for the efficient frontier function $F_{(Eff\text{-}NN)}(X)$ are given in Table 5.7. The 6 (six) data pairs or 30% (thirty percent) of the data pairs in Table 5.5 which were randomly selected by SANN to test the neural network for efficient frontier function $F_{(Eff\text{-}NN)}(X)$ are given in Table 5.8.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)
DMU48	0.000	0.242
$DMU1_{Eff\text{-}expanded}$	0.006	0.272
DMU15	0.013	0.296
$DMU2_{Eff\text{-}expanded}$	0.030	0.317
$DMU4_{Eff\text{-}expanded}$	0.058	0.350
$DMU5_{Eff\text{-}expanded}$	0.088	0.385
$DMU7_{Eff\text{-}expanded}$	0.145	0.452
$DMU8_{Eff\text{-}expanded}$	0.180	0.493
$DMU10_{Eff\text{-}expanded}$	0.233	0.556
$DMU11_{Eff\text{-}expanded}$	0.246	0.572
$DMU12_{Eff\text{-}expanded}$	0.263	0.591
$DMU13_{Eff\text{-}expanded}$	0.283	0.614
$DMU14_{Eff\text{-}expanded}$	0.307	0.643
DMU44	0.441	0.800

Table 5.7 SANN's Randomly Selected (14) Fourteen Data Pairs for Training NN Efficient Frontier

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)
$DMU3_{Eff\text{-}expanded}$	0.046	0.337
$DMU6_{Eff\text{-}expanded}$	0.107	0.409
$DMU9_{Eff\text{-}expanded}$	0.206	0.523
$DMU15_{Eff\text{-}expanded}$	0.332	0.672
$DMU16_{Eff\text{-}expanded}$	0.363	0.709
$DMU17_{Eff\text{-}expanded}$	0.407	0.761

Table 5.8 SANN's Randomly Selected (6) Six Data Pairs for Testing NN Efficient Frontier

The actual output values and the predicted output values given by the neural network efficient frontier function $F_{(Eff\text{-}NN)}(X)$ on the training and test tests in Table 5.7 and Table 5.8 are given in Table 5.9 and 5.10 respectively. The actual values and the predicted values given by the neural network efficient frontier function $F_{(Eff\text{-}NN)}(X)$ for the original 49 (forty-nine) DMUs or data pairs may be viewed in Appendix G1.

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
DMU48	0.000	0.242	**0.265216**
$DMU1_{Eff\text{-}expanded}$	0.006	0.272	**0.273979**
DMU15	0.013	0.296	**0.284096**
$DMU2_{Eff\text{-}expanded}$	0.030	0.317	**0.308160**
$DMU4_{Eff\text{-}expanded}$	0.058	0.350	**0.346182**
$DMU5_{Eff\text{-}expanded}$	0.088	0.385	**0.384732**
$DMU7_{Eff\text{-}expanded}$	0.145	0.452	**0.452875**
$DMU8_{Eff\text{-}expanded}$	0.180	0.493	**0.492839**
$DMU10_{Eff\text{-}expanded}$	0.233	0.556	**0.553951**
$DMU11_{Eff\text{-}expanded}$	0.246	0.572	**0.569474**
$DMU12_{Eff\text{-}expanded}$	0.263	0.591	**0.590233**
$DMU13_{Eff\text{-}expanded}$	0.283	0.614	**0.615333**
$DMU14_{Eff\text{-}expanded}$	0.307	0.643	**0.646204**
DMU44	0.441	0.800	**0.781291**

Table 5.9 Predicted Outputs for NN Efficient Frontier Function $F_{(Eff\text{-}NN)}(X)$ on Training Set

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
$DMU3_{Eff\text{-}expanded}$	0.046	0.337	**0.330134**
$DMU6_{Eff\text{-}expanded}$	0.107	0.409	**0.408086**
$DMU9_{Eff\text{-}expanded}$	0.206	0.523	**0.522507**
$DMU15_{Eff\text{-}expanded}$	0.332	0.672	**0.678476**
$DMU16_{Eff\text{-}expanded}$	0.363	0.709	**0.716471**
$DMU17_{Eff\text{-}expanded}$	0.407	0.761	**0.759974**

Table 5.10 Predicted Outputs for NN Efficient Frontier Function $F_{(Eff\text{-}NN)}(X)$ on Test Set

The sum of the squared errors for the output values for the efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ on the training set was **0.001151.**

The sum of the squared errors for the output values for the efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ on the test set was **0.000147.**

The total sum of the squared errors for the output values for the efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ on both training and test set was **0.001298.**

5.4.2.2 Efficient Frontier and Function for SVM

Utilizing Statistica 8.0 as mentioned in Section 4.4, the 14 (fourteen) randomly selected data pairs by SANN in Table 5.7 used for training the efficient frontier function for neural network, were entered into the Statistica Support Vector Machine Module for the training of SVM. Likewise, the 6 (six) randomly selected data pairs by SANN in Table 5.8 used for testing the efficient frontier function for neural network, were entered into the Statistica Support Vector Machine Module for the testing of SVM. The interactive interface C value was set between a minimum of 0 (zero) and a maximum of 100 (one-hundred) and a step increase set at 1.00 (one), the ε value was set between a minimum of 0 (zero) and a maximum of 5 (five), the Nu value was set to 0 (zero) since performing epsilon support vector machine, and the V-fold cross-validation feature selected. The software was ran 6 (six) times for a gamma value of 0 (zero), 1 (one), 2 (two), 3 (three), 4 (four), and 5 (five) to generate the optimal support vector machine model for the randomly selected training and testing data pairs and the specifications entered. It outputted an optimal model for each of the gamma values, with the optimal model or function among the 6 (six) for the efficient frontier $F_{(Eff\text{-}SVM)}$ being that with a gamma value of 5 (five). This model or function was saved. The optimal model and function along with its parameters given by Statistica 8.0 for efficient frontier for the support vector function $F_{(Eff\text{-}SVM)}(X)$ were as follows:

1. The support vector type was Regression Type 1, also referred to as epsilon-svm.
2. The optimal kernel type was a radial basis function of gamma value 5 (five).
3. The number of support vectors was 13 (thirteen) with 2 (two) bounded.
4. The optimal C value was 68 (sixty-eight).
5. The optimal ε (epsilon) value was 0.0015

The actual output values and the predicted output values given by the support vector machine efficient frontier function $F_{(Eff\text{-}SVM)}(X)$ on the training and test tests in Table 5.7 and Table 5.8 are given in Table 5.11 and 5.12 respectively. The actual values and the predicted values given by the support vector machine efficient frontier function $F_{(Eff\text{-}SVM)}(X)$ for the original 49 (forty-nine) DMUs or data pairs may be viewed in Appendix G2.

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
DMU48	0.000	0.242	**0.261021**
$DMU1_{Eff\text{-}expanded}$	0.006	0.272	**0.272002**
DMU15	0.013	0.296	**0.284363**
$DMU2_{Eff\text{-}expanded}$	0.030	0.317	**0.312203**
$DMU4_{Eff\text{-}expanded}$	0.058	0.350	**0.351388**
$DMU5_{Eff\text{-}expanded}$	0.088	0.385	**0.386447**
$DMU7_{Eff\text{-}expanded}$	0.145	0.452	**0.448653**
$DMU8_{Eff\text{-}expanded}$	0.180	0.493	**0.490546**
$DMU10_{Eff\text{-}expanded}$	0.233	0.556	**0.556823**
$DMU11_{Eff\text{-}expanded}$	0.246	0.572	**0.572635**
$DMU12_{Eff\text{-}expanded}$	0.263	0.591	**0.592837**
$DMU13_{Eff\text{-}expanded}$	0.283	0.614	**0.616080**
$DMU14_{Eff\text{-}expanded}$	0.307	0.643	**0.643839**
DMU44	0.441	0.800	**0.802128**

Table 5.11 Predicted Outputs for SVM Efficient Frontier Function $F_{(Eff\text{-}SVM)}(X)$ on Training Set

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
$DMU3_{Eff\text{-}expanded}$	0.046	0.337	**0.335544**
$DMU6_{Eff\text{-}expanded}$	0.107	0.409	**0.406928**
$DMU9_{Eff\text{-}expanded}$	0.206	0.523	**0.523152**
$DMU15_{Eff\text{-}expanded}$	0.332	0.672	**0.673527**
$DMU16_{Eff\text{-}expanded}$	0.363	0.709	**0.712169**
$DMU17_{Eff\text{-}expanded}$	0.407	0.761	**0.767217**

Table 5.12 Predicted Outputs for SVM Efficient Frontier Function $F_{(Eff\text{-}SVM)}(X)$ on Test Set

The sum of the squared errors for the output values for the efficient frontier for the support vector machine function $F_{(Eff\text{-}SVM)}(X)$ on the training set was **0.000555.**

The sum of the squared errors for the output values for the efficient frontier for the support vector machine function $F_{(Eff\text{-}SVM)}(X)$ on the test set was **0.000057.**

The total sum of the squared errors for the output values for the efficient frontier for the support vector machine function $F_{(Eff\text{-}SVM)}(X)$ on both training and test set was **0.000613.**

5.4.3 Inefficient Frontiers and their Functions for NN and SVM

5.4.3.1 Inefficient Frontier and Function for NN

Using the common heuristic for determining the neural network architecture in Section 4.4 for the single input and single output data pairs used in this research, the neural network architecture was determined to be 1-3-1, that being, 1(one) input node, 3 (three) hidden nodes, and 1 (one) output node.

Utilizing Statistica 8.0 as mentioned in Section 4.4, the 20 (twenty) expanded scaled data pairs on the inefficient frontier $F_{(Ineff)}$ in Table 5.6 were entered into the Statistica Automated Neural Network Module. The interactive interface was set to randomly select 70% (seventy percent – 14 data pairs) of the expanded scaled data set for training the neural network, and randomly select 30% (thirty percent – 6 data pairs) of the expanded scaled data set for testing the neural network. The input node was set at 1 (one), the hidden node at 3 (three), and the output node at 1 (one). The software was ran to generate the optimal neural network model for the randomly selected training and testing data pairs and the specifications entered and shortly thereafter, outputted the optimal model or function for the inefficient frontier for neural network $F_{(Ineff\text{-}NN)}$ which was saved. The optimal model and function along with its parameters given by Statistica 8.0 for inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ were as follows:

1. The neural network architecture was a three layer, 1-3-1 multi-layer perceptron model.
2. The training algorithm which gave the optimal neural network was BFGS (i.e. Quasi-Newton Back Propagation by Broyden, Fletcher, Goldfarb, and Shanno (BFGS)) (Hagen, Demuth, and Beale, 1996; and Haykin, 2000).
3. The hidden activation was Tanh, and the output activation Identity.
4. The error function was sum of squares (SOS)

The 14 (fourteen) data pairs or 70% (seventy percent) of the data pairs in Table 5.6 which were randomly selected by SANN to train the neural network for the inefficient frontier function $F_{(Ineff\text{-}NN)}(X)$ are given in Table 5.13. The 6 (six) data pairs or 30% (thirty percent) of the data pairs in Table 5.6 which were randomly selected by SANN to test the neural network for inefficient frontier function $F_{(Ineff\text{-}NN)}(X)$ are given in Table 5.14.

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)
DMU32	0.037	0.200
$DMU1_{Ineff\text{-}expanded}$	0.050	0.201
$DMU2_{Ineff\text{-}expanded}$	0.066	0.201
DMU5	0.101	0.204
$DMU4_{Ineff\text{-}expanded}$	0.151	0.223
$DMU5_{Ineff\text{-}expanded}$	0.199	0.241
DMU36	0.336	0.292
$DMU7_{Ineff\text{-}expanded}$	0.353	0.299
$DMU9_{Ineff\text{-}expanded}$	0.418	0.323
$DMU10_{Ineff\text{-}expanded}$	0.437	0.330
$DMU11_{Ineff\text{-}expanded}$	0.469	0.342
$DMU12_{Ineff\text{-}expanded}$	0.529	0.365
$DMU13_{Ineff\text{-}expanded}$	0.589	0.387
DMU1	1.000	0.542

Table 5.13 SANN's Randomly Selected (14) Fourteen Data Pairs for Training NN Inefficient Frontier

DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)
$DMU3_{Ineff\text{-}expanded}$	0.124	0.212
$DMU6_{Ineff\text{-}expanded}$	0.257	0.262
$DMU8_{Ineff\text{-}expanded}$	0.390	0.312
$DMU14_{Ineff\text{-}expanded}$	0.680	0.422
$DMU15_{Ineff\text{-}expanded}$	0.782	0.460
$DMU16_{Ineff\text{-}expanded}$	0.909	0.508

Table 5.14 SANN's Randomly Selected (6) Six Data Pairs for Testing NN Inefficient Frontier

The actual output values and the predicted output values given by the neural network inefficient frontier function $F_{(Ineff\text{-}NN)}(X)$ on the training and test tests in Table 5.13 and Table 5.14 are given in Table 5.15 and 5.16 respectively. The actual values and the predicted values given by the neural network inefficient frontier function $F_{(Ineff\text{-}NN)}(X)$ for the original 49 (forty-nine) DMUs or data pairs may be viewed in Appendix G3.

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
DMU32	0.037	0.200	**0.191623**
$DMU1_{Ineff\text{-}expanded}$	0.050	0.201	**0.195637**
$DMU2_{Ineff\text{-}expanded}$	0.066	0.201	**0.200625**
DMU5	0.101	0.204	**0.211717**
$DMU4_{Ineff\text{-}expanded}$	0.151	0.223	**0.227980**
$DMU5_{Ineff\text{-}expanded}$	0.199	0.241	**0.244038**
DMU36	0.336	0.292	**0.292035**
$DMU7_{Ineff\text{-}expanded}$	0.353	0.299	**0.298190**
$DMU9_{Ineff\text{-}expanded}$	0.418	0.323	**0.322060**
$DMU10_{Ineff\text{-}expanded}$	0.437	0.330	**0.329128**
$DMU11_{Ineff\text{-}expanded}$	0.469	0.342	**0.341112**
$DMU12_{Ineff\text{-}expanded}$	0.529	0.365	**0.363808**
$DMU13_{Ineff\text{-}expanded}$	0.589	0.387	**0.386716**
DMU1	1.000	0.542	**0.540999**

Table 5.15 Predicted Outputs for NN Inefficient Frontier Function $F_{(Ineff\text{-}NN)}(X)$ on Training Set

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
$DMU3_{Ineff\text{-}expanded}$	0.124	0.212	**0.219138**
$DMU6_{Ineff\text{-}expanded}$	0.257	0.262	**0.263987**
$DMU8_{Ineff\text{-}expanded}$	0.390	0.312	**0.311715**
$DMU14_{Ineff\text{-}expanded}$	0.680	0.422	**0.421629**
$DMU15_{Ineff\text{-}expanded}$	0.782	0.460	**0.460596**
$DMU16_{Ineff\text{-}expanded}$	0.909	0.508	**0.508112**

Table 5.16 Predicted Outputs for NN Inefficient Frontier Function $F_{(Ineff\text{-}NN)}(X)$ on Test Set

The sum of the squared errors for the output values for the inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ on the training set was **0.000198.**

The sum of the squared errors for the output values for the inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ on the test set was **0.000055.**

The total sum of the squared errors for the output values for the inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ on both training and test set was **0.000254.**

5.4.3.2 Inefficient Frontier and Function for SVM

Utilizing Statistica 8.0 as mentioned in Section 4.4, the 14 (fourteen) randomly selected data pairs by SANN in Table 5.13 used for training the inefficient frontier function for neural network, were entered into the Statistica Support Vector Machine Module for the training of SVM. Likewise, the 6 (six) randomly selected data pairs by SANN in Table 5.14 used for testing the inefficient frontier function for neural network, were entered into the Statistica Support Vector Machine Module for the testing of SVM. The interactive interface C value was set between a minimum of 0 (zero) and a maximum of 100 (one-hundred) and a step increase set at 1.00 (one), the ε value was set between a minimum of 0 (zero) and a maximum of 5 (five), the Nu value was set to 0 (zero) since performing epsilon support vector machine, and the V-fold cross-validation feature selected. The software was ran 6 (six) times for a gamma value of 0 (zero), 1 (one), 2 (two), 3 (three), 4 (four), and 5 (five) to generate the optimal support vector machine model for the randomly selected training and testing data pairs and the specifications entered. It outputted an optimal model for each of the gamma values, with the optimal model or function among the 6 (six) for the inefficient frontier $F_{(Ineff\text{-}SVM)}$ being that with a gamma value of 3 (three). This model or function was saved. The optimal model and function along with its parameters given by Statistica 8.0 for inefficient frontier for the support vector function $F_{(Ineff\text{-}SVM)}(X)$ were as follows:

The support vector type was Regression Type 1, also referred to as epsilon-svm.

The optimal kernel type was a radial basis function of gamma value 3 (three).

The number of support vectors was 12 (twelve) with 1 (one) bounded.

The optimal C value was 69 (sixty-nine).

The optimal ε (epsilon) value was 0.0000

The actual output values and the predicted output values given by the support vector machine inefficient frontier function $F_{(\text{Ineff-SVM})}(X)$ on the training and test tests in Table 5.13 and Table 5.14 are given in Table 5.17 and 5.18 respectively. The actual values and the predicted values given by the support vector machine inefficient frontier function $F_{(\text{Ineff-SVM})}(X)$ for the original 49 (forty-nine) DMUs or data pairs may be viewed in Appendix G4.

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
DMU32	0.037	0.200	**0.197237**
$DMU1_{\text{Ineff-expanded}}$	0.050	0.201	**0.198730**
$DMU2_{\text{Ineff-expanded}}$	0.066	0.201	**0.201081**
DMU5	0.101	0.204	**0.208066**
$DMU4_{\text{Ineff-expanded}}$	0.151	0.223	**0.221818**
$DMU5_{\text{Ineff-expanded}}$	0.199	0.241	**0.238183**
DMU36	0.336	0.292	**0.292339**
$DMU7_{\text{Ineff-expanded}}$	0.353	0.299	**0.299141**
$DMU9_{\text{Ineff-expanded}}$	0.418	0.323	**0.324465**
$DMU10_{\text{Ineff-expanded}}$	0.437	0.330	**0.331637**
$DMU11_{\text{Ineff-expanded}}$	0.469	0.342	**0.343504**
$DMU12_{\text{Ineff-expanded}}$	0.529	0.365	**0.365292**
$DMU13_{\text{Ineff-expanded}}$	0.589	0.387	**0.387081**
DMU1	1.000	0.542	**0.543215**

Table 5.17 Predicted Outputs for SVM Inefficient Frontier Function $F_{(\text{Ineff-SVM})}(X)$ on Training Set

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
$DMU3_{Ineff\text{-}expanded}$	0.124	0.212	**0.213896**
$DMU6_{Ineff\text{-}expanded}$	0.257	0.262	**0.260479**
$DMU8_{Ineff\text{-}expanded}$	0.390	0.312	**0.313704**
$DMU14_{Ineff\text{-}expanded}$	0.680	0.422	**0.421623**
$DMU15_{Ineff\text{-}expanded}$	0.782	0.460	**0.463142**
$DMU16_{Ineff\text{-}expanded}$	0.909	0.508	**0.514007**

Table 5.18 Predicted Outputs for SVM Inefficient Frontier Function $F_{(Ineff\text{-}SVM)}(X)$ on Test Set

The sum of the squared errors for the output values for the inefficient frontier for the support vector machine function $F_{(Ineff\text{-}SVM)}(X)$ on the training set was **0.000047.**

The sum of the squared errors for the output values for the inefficient frontier for the support vector machine function $F_{(Ineff\text{-}SVM)}(X)$ on the test set was **0.000055.**

The total sum of the squared errors for the output values for the inefficient frontier for the support function $F_{(Ineff\text{-}SVM)}(X)$ on both training and test set was **0.000102.**

5.4.3.3 Analysis of Efficient/Inefficient Frontiers and Functions for NN and SVM

The summary of the sum of squared errors for the efficient frontier functions and the inefficient frontier functions for NN and SVM may be viewed in Table 5.19.

	NN Train	NN Test	NN Total	SVM Train	SVM Test	SVM Total
Efficient frontier	0.001151	0.000147	**0.001298**	0.000555	0.000057	**0.000613**
Inefficient frontier	0.000198	0.000055	**0.000254**	0.000047	0.000055	**0.000102**
Efficient Frontier + Inefficient frontier	**0.001349**	**0.000202**	**0.001552**	**0.000602**	**0.000112**	**0.000715**

Table 5.19 Summary of SOS Errors for Efficient and Inefficient Frontier for NN and SVM

The sum of squared errors on the training and test sets for the efficient frontier function, inefficient frontier function, and efficient frontier function + inefficient frontier function for support vector machine is less than half that for neural network. Support vector machine outperformed neural network by more than 2-to-1 in estimating a nonlinear function and both efficient and inefficient frontiers for the VRS case in this research.

5.5 OLS Regression Models for Comparison Purposes

5.5.1 Efficient Frontier and Functions for OLS Regression Models OLS1 and OLS2

Utilizing Minitab 12.23, as mentioned in Section 4.5, the 14 (fourteen) randomly selected data pairs by SANN in Table 5.7 used for training the efficient frontier function for both neural network and support vector machine, were entered into a Minitab spreadsheet and a linear and quadratic regression model for the 14 (fourteen) data pairs created so as to determine the efficient frontier function for model OLS1 and model OLS2. The linear and quadratic models created are shown in Figure 5.4 and Figure 5.5 respectively.

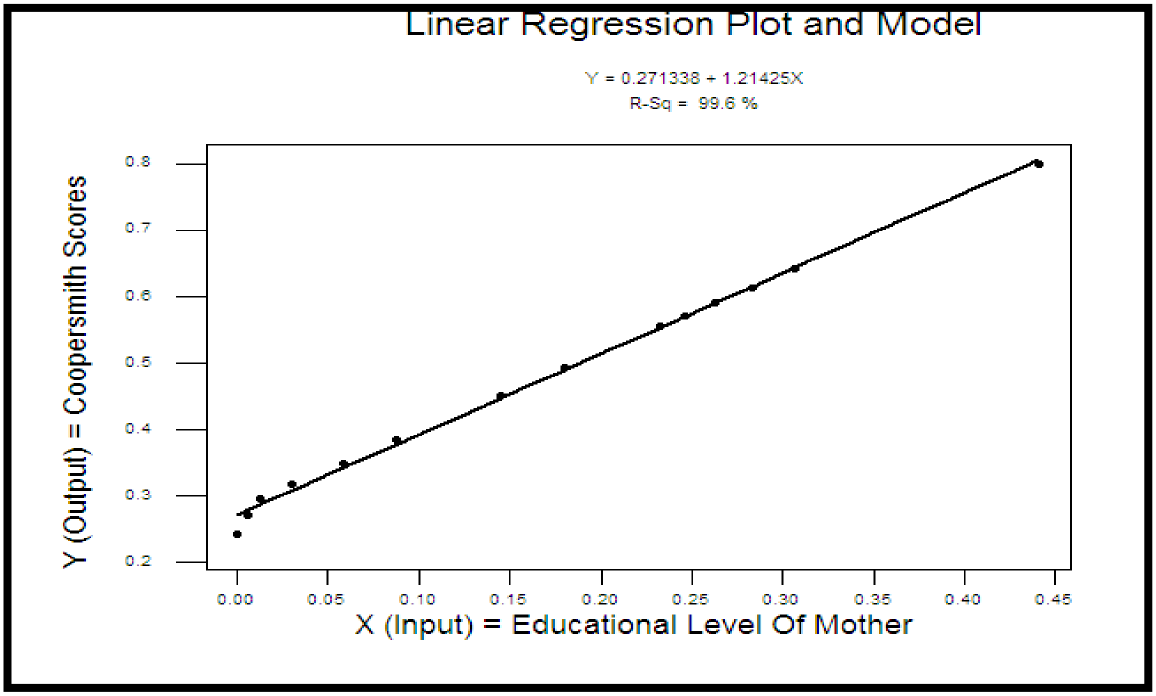


Figure 5.4 Linear Regression Plot and Model for Efficient 14 (fourteen) Training Data Pairs

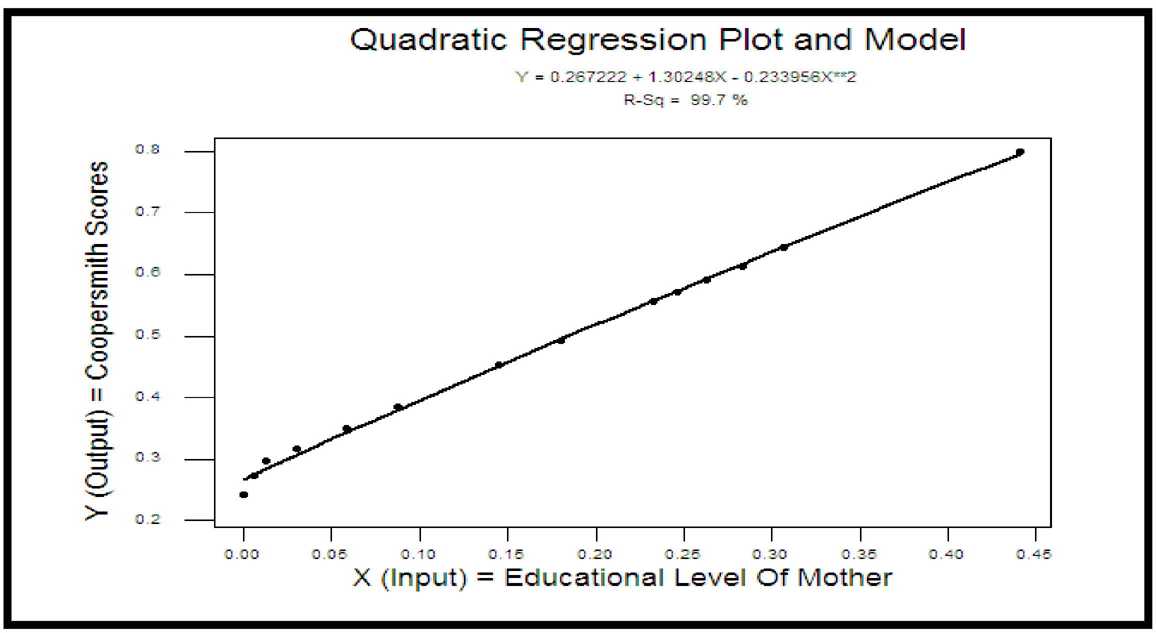


Figure 5.5 Quadratic Regression Plot and Model for Efficient 14 (fourteen) Training Data Pairs

The linear regression model had a R^2 = 99.6% and the quadratic regression model had a R^2 = 99.7% for the 14 (fourteen) training data pairs used for both neural network and support vector machine. Since, the R^2 value for the quadratic regression model was the highest of the two models, the quadratic model was determined to be the optimal model or function for the efficient frontier $F_{(Eff\text{-}OLS1)}(X)$ for model OLS1, and $F_{(Eff\text{-}OLS2)}(X)$ for model OLS2. Both functions are identical and exactly the same for both models (i.e. $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$). This quadratic regression model or function was saved and was given as:

Y = 0.267222 + 1.30248X - 0.233956X**2

Likewise, the 6 (six) randomly selected data pairs by SANN in Table 5.8 used for testing the efficient frontier function for neural network and support vector machine, were entered into the quadratic efficient frontier function for OLS1 and OLS2 to test the function.

The actual output values and the predicted output values given by the OLS1 efficient frontier function $F_{(Eff\text{-}OLS1}(X)$ and the OLS2 efficient frontier function $F_{(Eff\text{-}OLS2)}(X)$ on the training and test tests in Table 5.7 and Table 5.8 are given in Table 5.20 and 5.21 respectively. The actual values and the predicted values given by the OLS1 efficient frontier function $F_{(Eff\text{-}OLS1)}(X)$ and the OLS2 efficient frontier function $F_{(Eff\text{-}OLS2)}(X)$ for the original 49 (forty-nine) DMUs or data pairs may be viewed in Appendix H1.

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
DMU48	0.000	0.242	**0.267222**
$DMU1_{Eff\text{-}expanded}$	0.006	0.272	**0.275028**
DMU15	0.013	0.296	**0.284115**
$DMU2_{Eff\text{-}expanded}$	0.030	0.317	**0.306086**
$DMU4_{Eff\text{-}expanded}$	0.058	0.350	**0.341979**
$DMU5_{Eff\text{-}expanded}$	0.088	0.385	**0.380028**
$DMU7_{Eff\text{-}expanded}$	0.145	0.452	**0.451163**
$DMU8_{Eff\text{-}expanded}$	0.180	0.493	**0.494088**
$DMU10_{Eff\text{-}expanded}$	0.233	0.556	**0.557999**
$DMU11_{Eff\text{-}expanded}$	0.246	0.572	**0.573474**
$DMU12_{Eff\text{-}expanded}$	0.263	0.591	**0.593592**
$DMU13_{Eff\text{-}expanded}$	0.283	0.614	**0.617087**
$DMU14_{Eff\text{-}expanded}$	0.307	0.643	**0.645033**
DMU44	0.441	0.800	**0.796116**

Table 5.20 Predicted Outputs for OLS1 and OLS2 Efficient Frontier Function on Training Set

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
$DMU3_{Eff\text{-}expanded}$	0.046	0.337	**0.326641**
$DMU6_{Eff\text{-}expanded}$	0.107	0.409	**0.403909**
$DMU9_{Eff\text{-}expanded}$	0.206	0.523	**0.525605**
$DMU15_{Eff\text{-}expanded}$	0.332	0.672	**0.673858**
$DMU16_{Eff\text{-}expanded}$	0.363	0.709	**0.709194**
$DMU17_{Eff\text{-}expanded}$	0.407	0.761	**0.758577**

Table 5.21 Predicted Outputs for OLS1 and OLS2 Efficient Frontier Function on Test Set

The sum of the squared errors for the output values for the efficient frontier for the two OLS models' function, where $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$, on the training set was **0.001038.**

The sum of the squared errors for the output values for the efficient frontier for the two OLS models' function, where $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$, on the test set was **0.000149.**

The total sum of the squared errors for the output values for the efficient frontier for the two OLS models' function, where $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$, on both training and test set was **0.001188.**

5.5.2 Inefficient Frontier and Function for OLS Regression Model OLS1

Utilizing Minitab 12.23, as mentioned in Section 4.5, the 14 (fourteen) randomly selected data pairs by SANN in Table 5.13 used for training the inefficient frontier function for both neural network and support vector machine, were entered into a Minitab spreadsheet and a linear and quadratic regression model for the 14 (fourteen) data pairs created so as to determine the inefficient frontier function for OLS1 model. No inefficient frontier function was computed for the OLS2 model. The linear and quadratic models created are shown in Figure 5.6 and Figure 5.7 respectively.

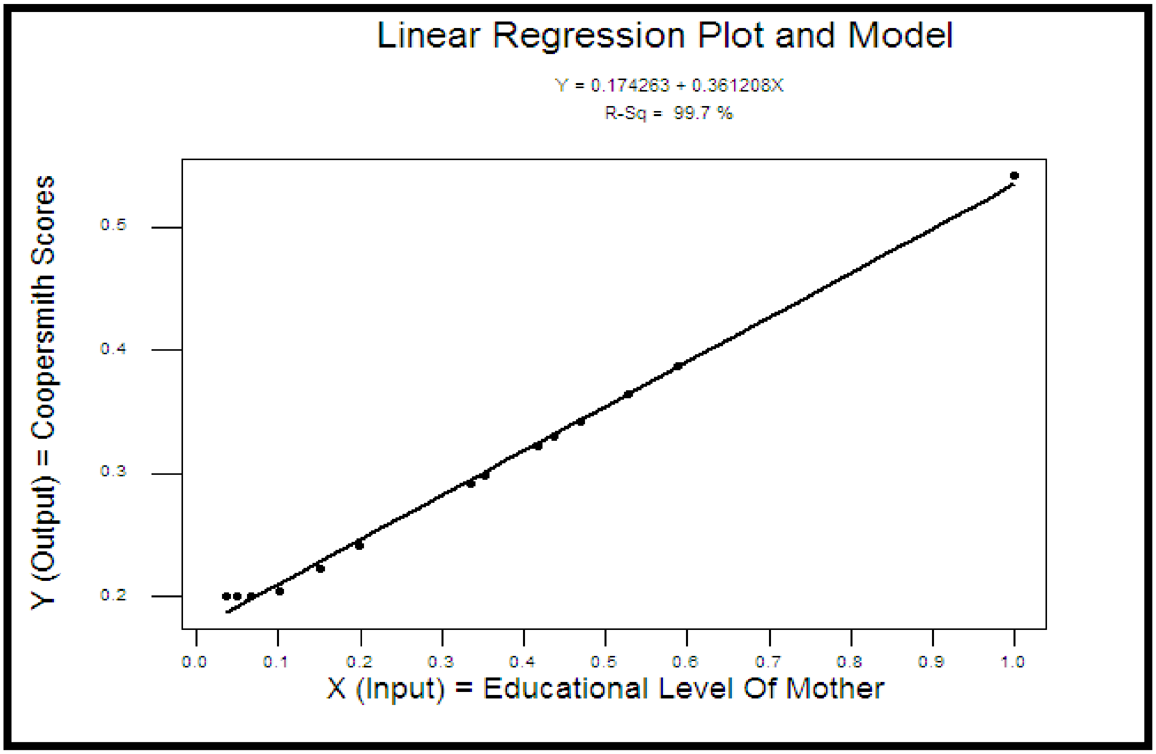


Figure 5.6 Linear Regression Plot and Model for Inefficient 14 (fourteen) Training Data Pairs

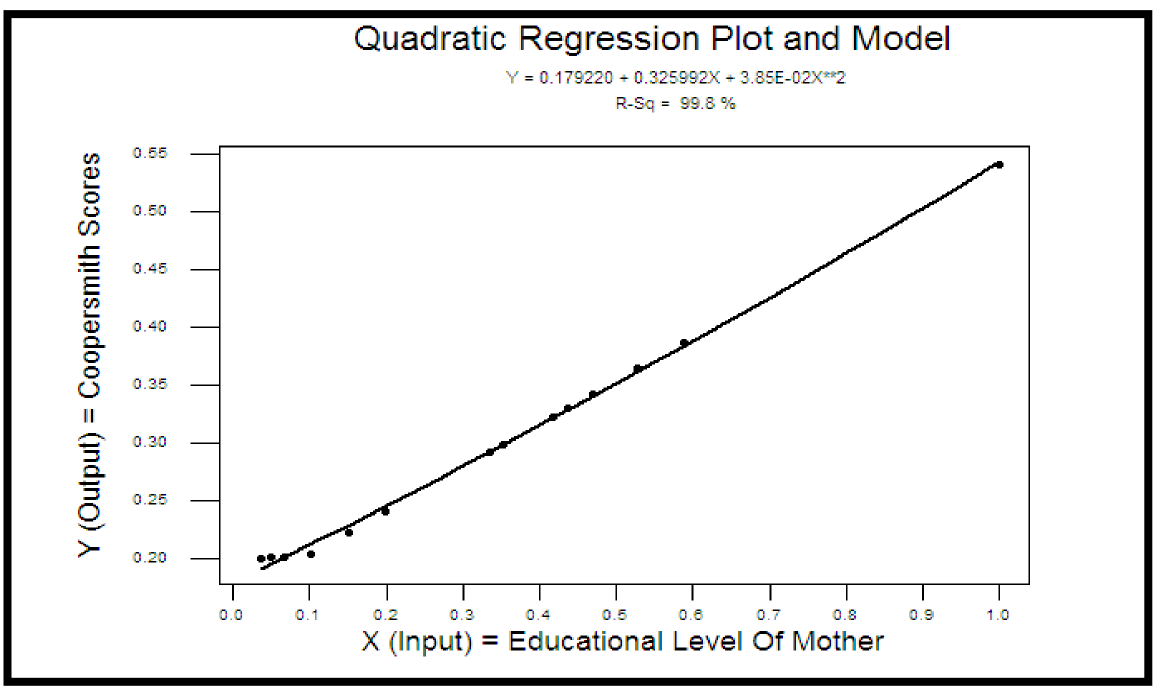


Figure 5.7 Quadratic Regression Plot and Model for Inefficient 14 (fourteen) Training Data Pairs

The linear regression model had a R^2 = 99.7% and the quadratic regression model had a R^2 = 99.8% for the 14 (fourteen) training data pairs used for both neural network and support vector machine. Since, the R^2 value for the quadratic regression model was the highest of the two models, the quadratic model was determined to be the optimal model or function for the inefficient frontier $F_{(\text{Ineff-OLS1})}(X)$ for model OLS1. This quadratic regression model or function was saved and was given as:

Y = 0.179220 + 0.325992X + 3.85E-02X**2

Likewise, the 6 (six) randomly selected data pairs by SANN in Table 5.14 used for testing the inefficient frontier function for neural network and support vector machine, were entered into the quadratic inefficient frontier function for OLS to test the function.

The actual output values and the predicted output values given by the OLS1 inefficient frontier function $F_{(\text{Ineff-OLS1}}(X)$ on the training and test tests in Table 5.13 and Table 5.14 are given in Table 5.22 and 5.23 respectively. The actual values and the predicted values given by the OLS1 inefficient frontier function $F_{(\text{Ineff-OLS1})}(X)$ for the original 49 (forty-nine) DMUs or data pairs may be viewed in Appendix H2.

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
DMU32	0.037	0.200	**0.191334**
$DMU1_{Ineff\text{-}expanded}$	0.050	0.201	**0.195616**
$DMU2_{Ineff\text{-}expanded}$	0.066	0.201	**0.200903**
DMU5	0.101	0.204	**0.212538**
$DMU4_{Ineff\text{-}expanded}$	0.151	0.223	**0.229323**
$DMU5_{Ineff\text{-}expanded}$	0.199	0.241	**0.245617**
DMU36	0.336	0.292	**0.293100**
$DMU7_{Ineff\text{-}expanded}$	0.353	0.299	**0.299093**
$DMU9_{Ineff\text{-}expanded}$	0.418	0.323	**0.322212**
$DMU10_{Ineff\text{-}expanded}$	0.437	0.330	**0.329031**
$DMU11_{Ineff\text{-}expanded}$	0.469	0.342	**0.340579**
$DMU12_{Ineff\text{-}expanded}$	0.529	0.365	**0.362444**
$DMU13_{Ineff\text{-}expanded}$	0.589	0.387	**0.384586**
DMU1	1.000	0.542	**0.543712**

Table 5.22 Predicted Outputs for OLS1 Inefficient Frontier Function on Training Set

DMU	Educational Level Of Mother (X-Input values)	Actual Coopersmith Scores (Y-Output values)	**Predicted Coopersmith Scores (Y-Output values)**
$DMU3_{Ineff\text{-}expanded}$	0.124	0.212	**0.220235**
$DMU6_{Ineff\text{-}expanded}$	0.257	0.262	**0.265543**
$DMU8_{Ineff\text{-}expanded}$	0.390	0.312	**0.312213**
$DMU14_{Ineff\text{-}expanded}$	0.680	0.422	**0.418697**
$DMU15_{Ineff\text{-}expanded}$	0.782	0.460	**0.457689**
$DMU16_{Ineff\text{-}expanded}$	0.909	0.508	**0.507359**

Table 5.23 Predicted Outputs for OLS1 Inefficient Frontier Function on Test Set

The sum of the squared errors for the output values for the inefficient frontier for the regression model OLS1 function $F_{(Ineff\text{-}OLS1)}(X)$ on the training set was **0.000258.**

The sum of the squared errors for the output values for the inefficient frontier for the regression model OLS1 function $F_{(Ineff\text{-}OLS1)}(X)$ on the test set was **0.000097.**

The total sum of the squared errors for the output values for the inefficient frontier for the regression model OLS1 function $F_{(Ineff-OLS1)}(X)$ on both training and test set was **0.000355.**

5.5.3 Analysis of Efficient/Inefficient Frontiers and Functions for OLS/NN/SVM

The summary of the sum of squared errors for the efficient and inefficient frontier functions for OLS1, NN, and SVM may be viewed in Table 5.24.

	OLS1 Train	OLS1 Test	**OLS1 Total**	NN Train	NN Test	**NN Total**	SVM Train	SVM Test	**SVM Total**
Efficient frontier	0.001038	0.000149	**0.001188**	0.001151	0.000147	**0.001298**	0.000555	0.000057	**0.000613**
Inefficient frontier	0.000258	0.000097	**0.000355**	0.000198	0.000055	**0.000254**	0.000047	0.000055	**0.000102**
Efficient frontier + Inefficient frontier	**0.001296**	**0.000246**	**0.001543**	**0.001349**	**0.000202**	**0.001552**	**0.000602**	**0.000112**	**0.000715**

Table 5.24 SOS Errors for Efficient/Inefficient Frontier for OLS, NN, and SVM

The sum of squared errors for the parametric ordinary least squares regression model OLS1 and the sum of squared errors for the non-parametric neural network model for efficient frontier, inefficient frontier, and efficient frontier + inefficient frontier are almost equal for both the training and test sets. The sum of squared errors however for the support vector machine is less than half that for the training and test set of both OLS1, as well as, the neural network. Support vector machine outperformed both ordinary least squares regression, as well as, neural network by more 2-to-1 in estimating a nonlinear function, and efficient and inefficient frontiers for the VRS case in this research.

5.6 Frontier Analysis

5.6.1 Determining Average Regression Function $F_{Reg}(X)$ Non-parametrically

5.6.1.1 Determining NN Average Regression Function $F_{Reg\text{-}NN}(X)$

As mentioned in Section 4.6.1, the neural network function for the non-parametric efficient frontier $F_{(Eff\text{-}NN)}(X)$ determined in Section 5.4.2.1 at $\omega_{NN} = 1$, and the neural network function for the non-parametric inefficient frontier $F_{(Inff\text{-}NN)}(X)$ determined in Section 5.4.3.1 at $\omega_{NN} = 0$ were combined as in Equation 4.5 to determine the average function $F_{Reg\text{-}NN}(X)$ non-parametrically for neural network.

Utilizing Equation 4.5 and starting at $\omega_{NN} = 0$, the ω_{NN} value was increased in steps of 0.1 or smaller as necessary until a normal or approximate normal distribution of the residuals was achieved, this being achieved when the sum of the left hand half and the right hand half equaled to zero or approximately zero. The average function for neural network was determined non-parametrically to be at $\omega_{NN} = 0.307$. At this optimal ω_{NN} for the neural network's average function $F_{Reg\text{-}NN}(X)$, the function was also set synonymously at $\lambda = 0$ so as to determine the optimal λ value at which the "true" DEA frontier is achieved later on in the research. The residual distributions for ω_{NN} values of 0.000, 0.100, 0.200, 0.300, 0.400, and the optimal ω_{NN} value of 0.307 are depicted in Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11, Figure 5.12, and Figure 5.13 respectively.

Sum of residuals = -4.857 at ω_{NN} = 0.000

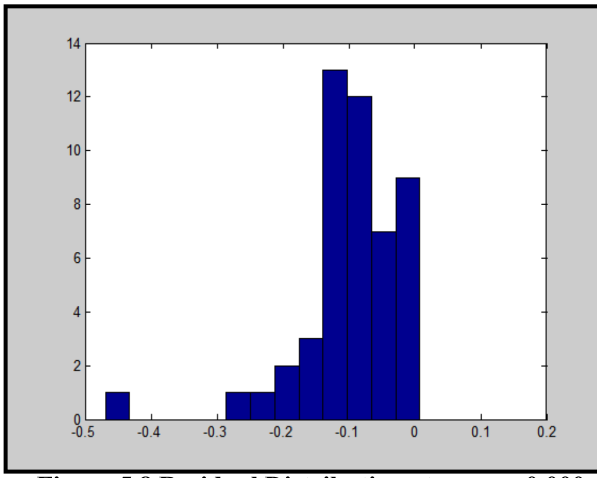


Figure 5.8 Residual Distribution at ω_{NN} = 0.000

Sum of residuals = -3.273 at ω_{NN} = 0.100

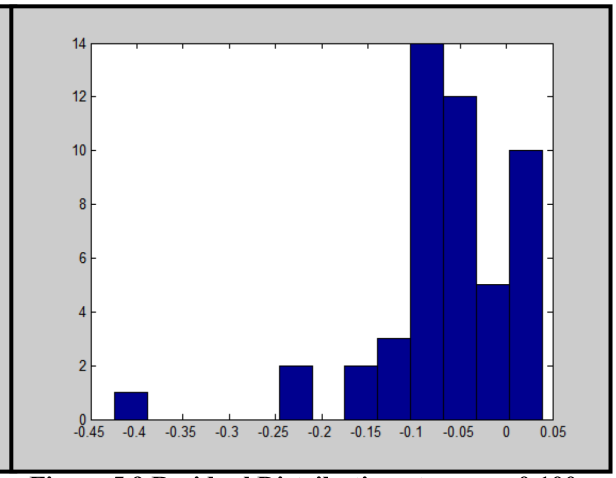


Figure 5.9 Residual Distribution at ω_{NN} = 0.100

Sum of residuals = -1.688 at ω_{NN} = 0.200

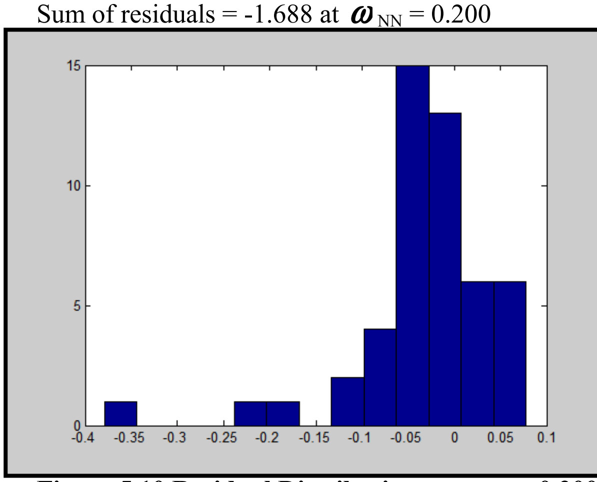


Figure 5.10 Residual Distribution at ω_{NN} = 0.200

Sum of residuals = -0.104 at ω_{NN} = 0.300

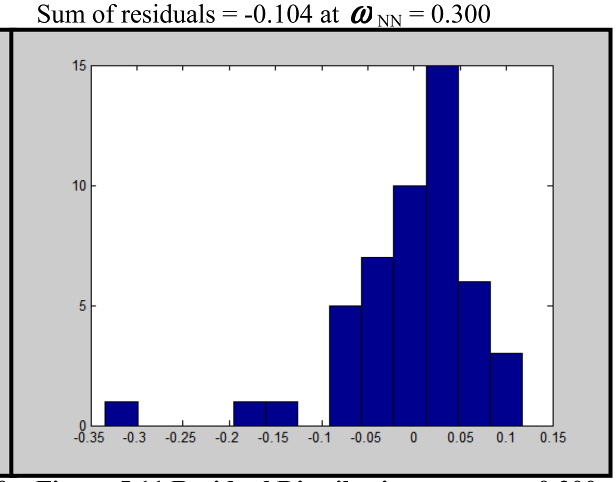


Figure 5.11 Residual Distribution at ω_{NN} = 0.300

Sum of residuals = 1.480 at ω_{NN} = 0.400

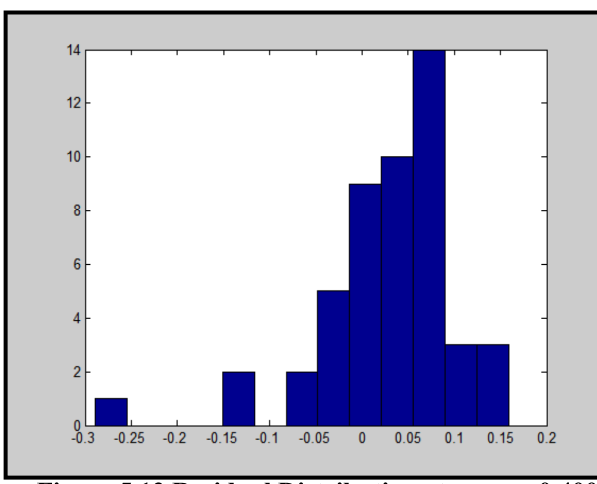


Figure 5.12 Residual Distribution at ω_{NN} = 0.400

Sum of residuals = 0.000 at ω_{NN} = 0.3066

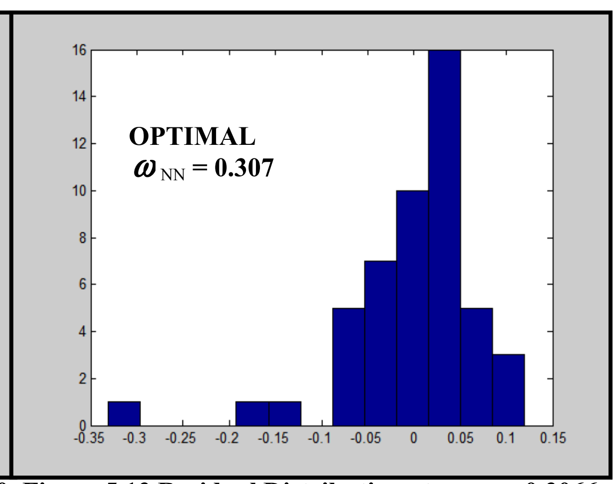


Figure 5.13 Residual Distribution at ω_{NN} = 0.3066

5.6.1.2 Determining SVM Average Regression Function $F_{Reg\text{-}SVM}(X)$

As mentioned in Section 4.6.1, the support vector machine function for the non-parametric efficient frontier $F_{(Eff\text{-}SVM)}(X)$ determined in Section 5.4.2.2 at $\omega_{SVM} = 1$, and the support vector machine function for the non-parametric inefficient frontier $F_{(Inff\text{-}SVM)}(X)$ determined in Section 5.4.3.2 at $\omega_{SVM} = 0$ were combined as in Equation 4.6 to determine the average function $F_{Reg\text{-}SVM}(X)$ non-parametrically for neural network.

Utilizing Equation 4.6 and starting at $\omega_{SVM} = 0$, the ω_{SVM} value was increased in steps of 0.1 or smaller as necessary until a normal or approximate normal distribution of the residuals was achieved, this being achieved when the sum of the left hand half and the right hand half equaled to zero or approximately zero. The average function for support vector machine was determined non-parametrically to be at $\omega_{SVM} = 0.328$. At this optimal ω_{SVM} for the support vector machine's average function $F_{Reg\text{-}SVM}(X)$, the function was also set synonymously at $\lambda = 0$ so as to determine the optimal λ value at which the "true" DEA frontier is achieved later on in the research. The residual distributions for ω_{SVM} values of 0.000, 0.100, 0.200, 0.300, 0.400, and the optimal ω_{SVM} value of 0.328 are depicted in Figure 5.14, Figure 5.15, Figure 5.16, Figure 5.17, Figure 5.18, and Figure 5.19 respectively.

Sum of residuals = -4.844 at ω_{SVM} = 0.000

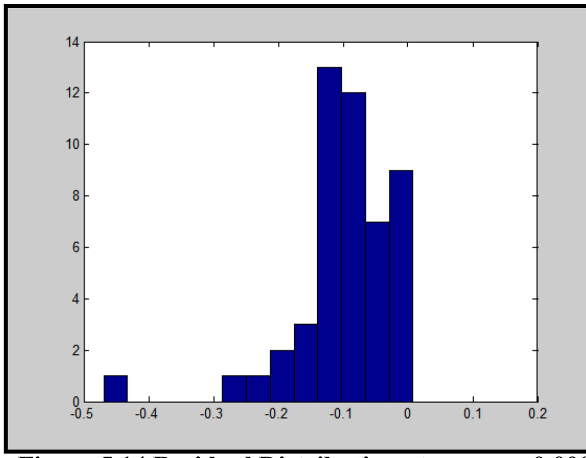


Figure 5.14 Residual Distribution at ω_{SVM} = 0.000

Sum of residuals = -3.367 at ω_{SVM} = 0.100

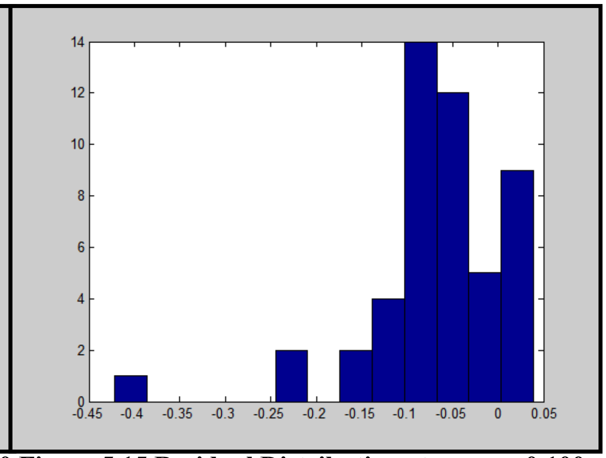


Figure 5.15 Residual Distribution at ω_{SVM} =0.100

Sum of residuals = -1.890 at ω_{SVM} = 0.200

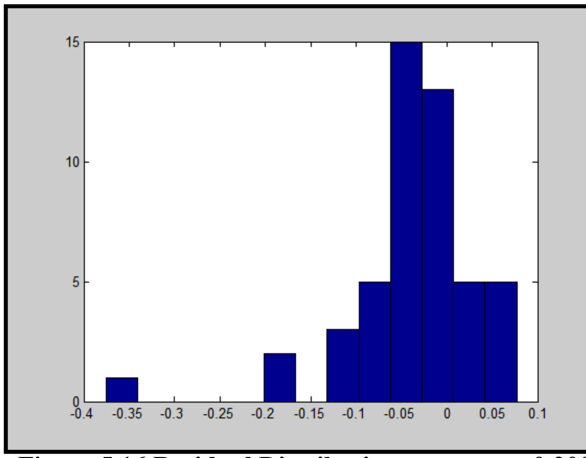


Figure 5.16 Residual Distribution at ω_{SVM} = 0.200

Sum of residuals = -0.413 at ω_{SVM} = 0.300

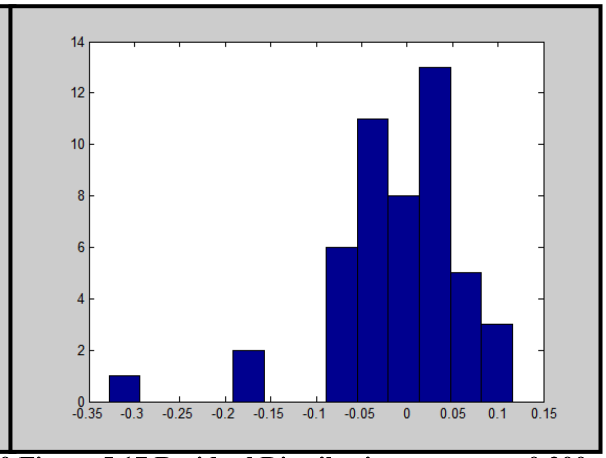


Figure 5.17 Residual Distribution at ω_{SVM} =0.300

Sum of residuals = 1.065 at ω_{SVM} = 0.400

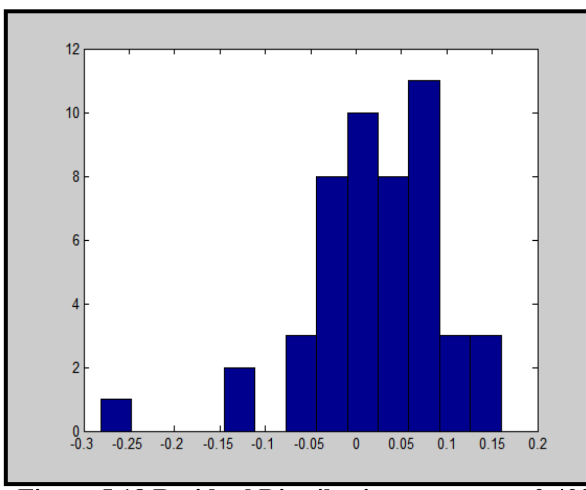


Figure 5.18 Residual Distribution at ω_{SVM} = 0.400

Sum of residuals = 0.000 at ω_{SVM} = 0.3279

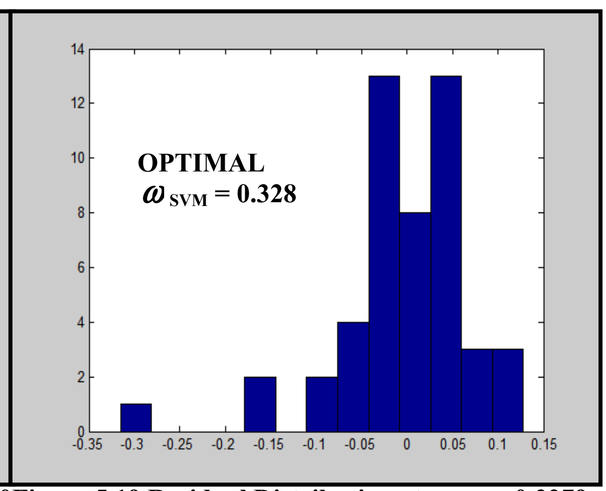


Figure 5.19 Residual Distribution at ω_{SVM}=0.3279

5.6.1.3 Analysis of Average Regression Function for NN and SVM

The summary of the optimal ω value for the average regression function found non-parametrically for both neural network and support vector machine from non-parametric efficient frontier function and the inefficient frontier functions are listed in Table 5.25.

	NN ω_{NN}	SVM ω_{SVM}
Optimal ω	**0.307**	**0.328**

Table 5.25 Summary of Optimal ω for Average Regression Function for NN and SVM

The summary of the residual distributions for the optimal ω value for the average regression function found non-parametrically for neural network $F_{Reg\text{-}NN}(X)$ at ω_{NN} = 0.307, and the optimal ω value for the average regression function found non-parametrically for support vector machine $F_{Reg\text{-}SVM}(X)$ at ω_{NN} = 0.328 are given in Figure 5.20(a) and Figure 5.20(b).

Sum of residuals = 0.000 at ω_{NN} = 0.3066
Residual Distribution at ω_{NN} = 0.3066

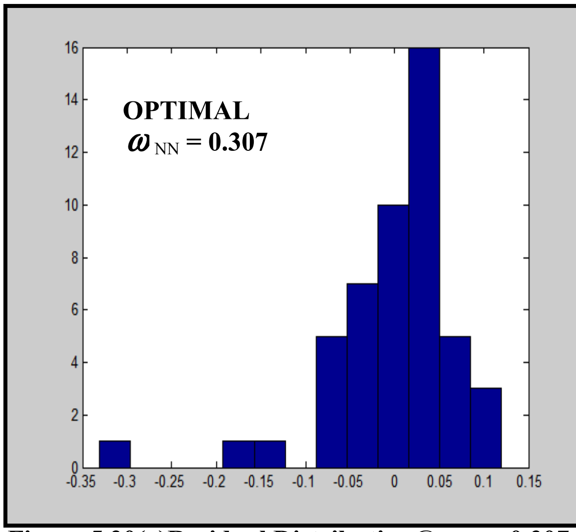


Figure 5.20(a)Residual Distribution@ ω_{NN}=0.307

Sum of residuals = 0.000 at ω_{SVM} = 0.3279
Residual Distribution at ω_{SVM} = 0.3279

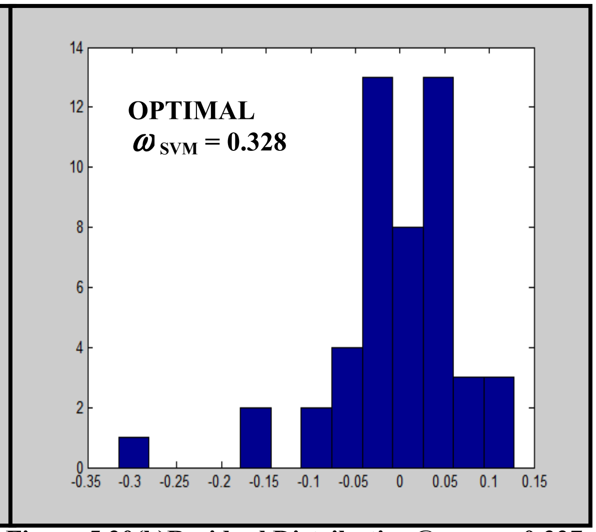


Figure 5.20(b)Residual Distribution@ ω_{SVM}=0.327

The optimal ω value of support vector machine, ω_{SVM} = 0.328 for the average regression function is a more accurate estimation of the average regression function non-parametrically than that of neural network ω_{NN} = 0.307 because as concluded in Section 5.4.3.3, support vector machine outperformed neural network by more than 2-to-1.

The lack of normality of the residual distributions for the average regression functions for both neural network and support vector machine are attributed possibly to errors incurred during the initial data collection for the study ‘Program Follow Through’ by Charnes, Cooper and Rhodes (1981). This is validated in some way by observing the scatter plot in Figure 5.1 for the 49 (forty-nine) original single-input, single-output data pairs. Viewing the scatter plot it can be observed that there are 4 (four) DMUs namely DMU1, DMU8, DMU16, and DMU44 which appear to be outliers considering the remaining 45 (forty-five) DMUs. Fitting to mention, the residuals of 3 (three) out of the

4 (four) of these DMUs, namely DMU8, DMU16, and DMU44 stand out in the residual distributions for the average regression function for both neural network and support vector machine. There is also a possibility that there is noise in the data set which also affected the normality of the residual distributions.

Finally, comparing both residual distributions in Figures 5.20(a) and 5.20(b), that for the support vector machine appears to be more symmetric than that for neural network.

5.6.2 Determining Average Regression Function $F_{Reg}(X)$ Parametrically

5.6.2.1 Determining OLS1 Average Regression Function $F_{Reg\text{-}OLS1}(X)$

As mentioned in Section 4.6.2, the ordinary least squares regression model OLS1 function for the parametric efficient frontier $F_{(Eff\text{-}OLS1)}(X)$ determined in Section 5.5.1 at $\omega_{OLS1} = 1$, and the OLS1 function for the parametric inefficient frontier $F_{(Inff\text{-}OLS1)}(X)$ determined in Section 5.5.2 at $\omega_{OLS1} = 0$ were combined as in Equation 4.7 to determine the average function $F_{Reg\text{-}OLS1}(X)$ parametrically for OLS1.

Utilizing Equation 4.7 and starting at $\omega_{OLS1} = 0$, the ω_{OLS1} value was increased in steps of 0.1 or smaller as necessary until a normal or approximate normal distribution of the residuals was achieved, this being achieved when the sum of the left hand half and the right hand half equaled to zero or approximately zero. The average function for OLS1 was determined parametrically to be at $\omega_{OLS1} = 0.273$. At this optimal ω_{OLS1} for the

OLS1's average function $F_{Reg\text{-}OLS1}(X)$, the function was also set synonymously at $\lambda = 0$ so as to determine the optimal λ value at which the "true" DEA frontier is achieved later on in the research. The residual distributions for ω_{OLS1} values of 0.000, 0.100, 0.200, 0.300, 0.250, and the optimal ω_{OLS1} value of 0.273 are depicted in Figure 5.21, Figure 5.22, Figure 5.23, Figure 5.24, Figure 5.25, and Figure 5.26 respectively.

Sum of residuals = -4.831 at ω_{OLS1} = 0.000

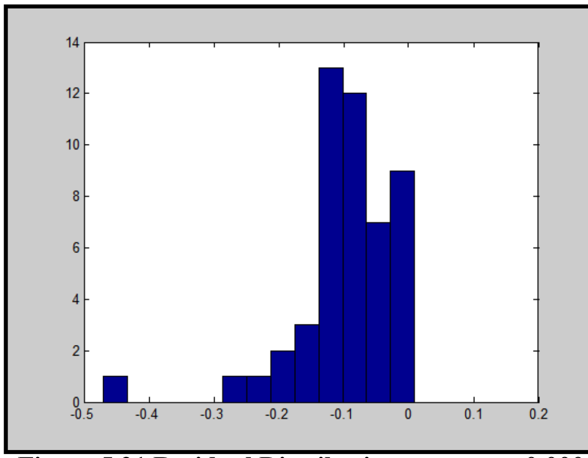


Figure 5.21 Residual Distribution at ω_{OLS1}=0.000

Sum of residuals = -3.062 at ω_{OLS1} = 0.100

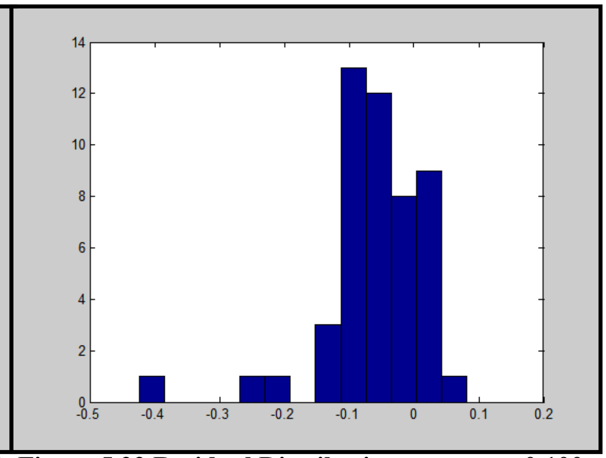


Figure 5.22 Residual Distribution at ω_{OLS1}=0.100

Sum of residuals = -1.294 at ω_{OLS1} = 0.200

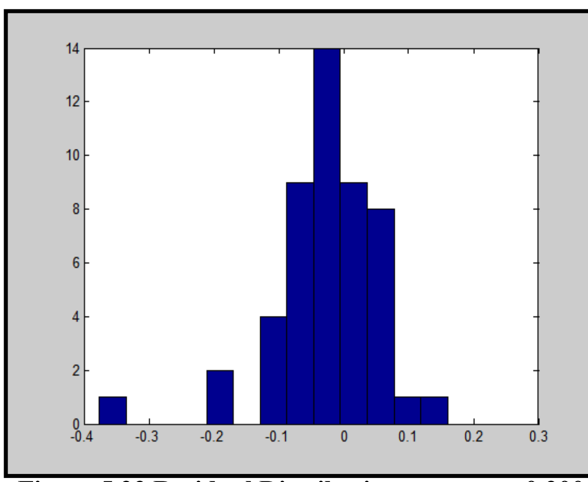


Figure 5.23 Residual Distribution at ω_{OLS1}=0.200

Sum of residuals = 0.474 at ω_{OLS1} = 0.300

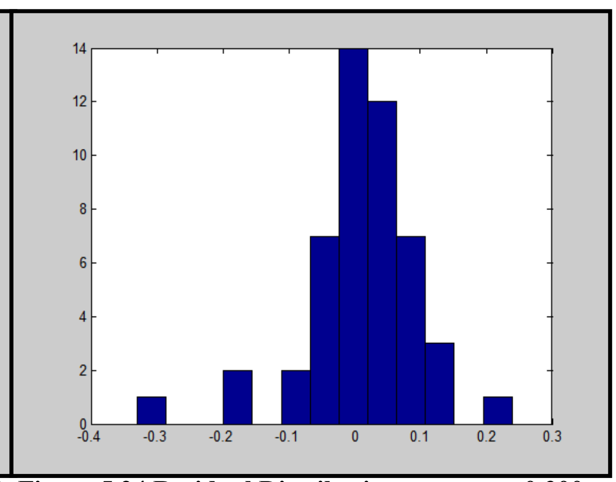
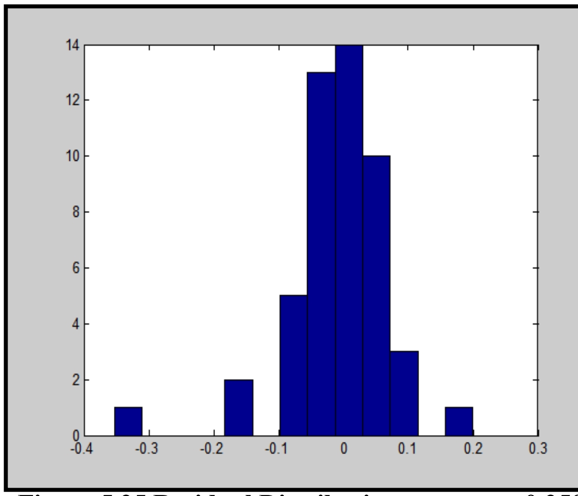


Figure 5.24 Residual Distribution at ω_{OLS1}=0.300

Sum of residuals = -0.410 at ω_{OLS1} = 0.250 Sum of residuals = 0.000 at ω_{OLS1} = 0.2732



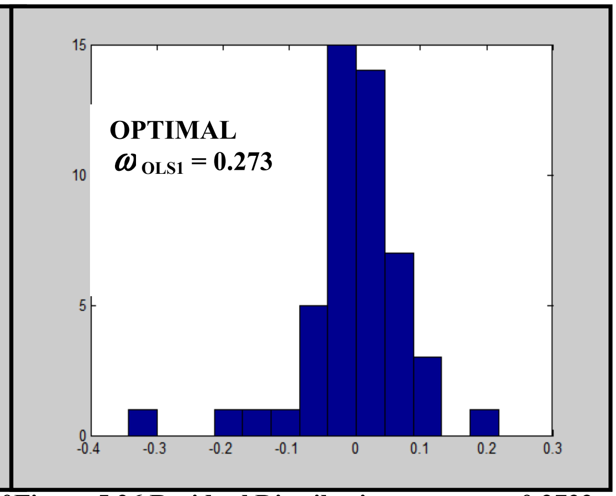


Figure 5.25 Residual Distribution at ω_{OLS1}=0.250 **Figure 5.26 Residual Distribution at ω_{OLS1}=0.2732**

5.6.2.2 Determining OLS2 Average Regression Function $F_{Reg-OLS2}$(X)

As mentioned in Section 4.6.2, the ordinary least squares regression model OLS2 function for the average regression function $F_{Reg-OLS2}$(X) was determined using Minitab 12.23.

For the original 49 (forty-nine) data pairs in Appendix A1, the x-input values were scaled between 0.000 and 1.000, and the y-output values scaled between 0.200 and 0.800 as in Appendix A2. Utilizing Minitab 12.23, the original 49 (forty-nine) scaled data pairs in Appendix A2, were entered into a Minitab spreadsheet and a linear and quadratic regression model for the 49 (forty-nine) scaled data pairs were created so as to determine the average regression function determined parametrically for the OLS2 model. The linear and quadratic models created are shown in Figure 5.27 and Figure 5.28 respectively.

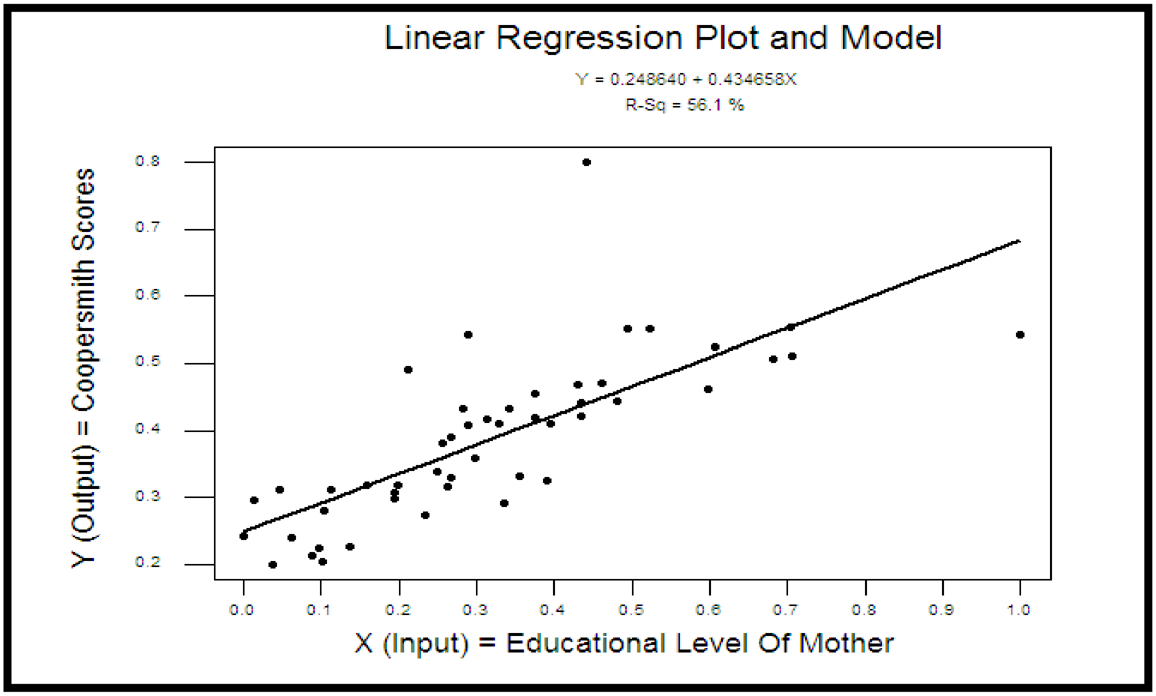


Figure 5.27 Linear Regression Plot and Model for Average Regression Function for OLS2

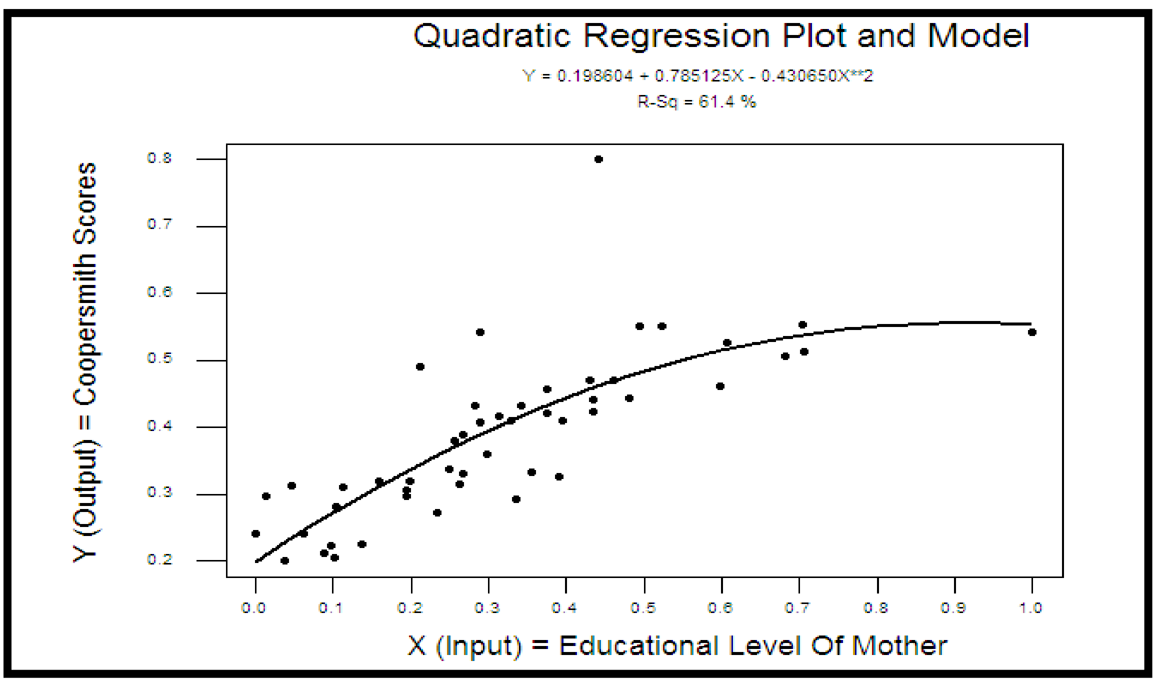


Figure 5.28 Quadratic Regression Plot and Model for Average Regression Function for OLS2

The linear regression model had a R^2 = 56.1% and the quadratic regression model had a R^2 = 61.4% for the original 49 (forty-nine) scaled data. Since, the R^2 value for the quadratic regression model was the highest of the two models, the quadratic model was determined to be the optimal model or average regression function $F_{Reg\text{-}OLS2}(X)$ for the OLS2 model. This quadratic regression model or function was saved and was given as:

Y = 0.198604 + 0.785125X - 0.430650X**2

This optimal OLS2's average regression function $F_{Reg\text{-}OLS2}(X)$, was also set synonymously at $\lambda = 0$ so as to determine the optimal λ value at which the "true" DEA frontier is achieved later on in the research for OLS2. The residual distribution for the OLS2 average regression function $F_{Reg\text{-}OLS2}(X)$ at this $\lambda = 0$ value, which is synonymous to the average regression functions found for NN, SVM, and OLS1 is depicted in Figure 5.29. The average function $F_{Reg\text{-}OLS2}(X)$ did not have an ω value as OLS1, NN, and SVM.

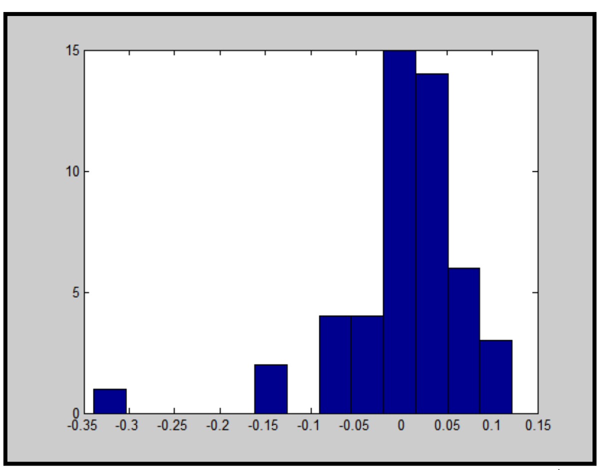


Figure 5.29 Residual Distribution for $F_{Reg\text{-}OLS2}(X)$

5.6.2.3 Analysis of Average Regression Function for OLS, NN, and SVM

The summary of the optimal ω value for the average regression function found non-parametrically for both neural network and support vector machine, and that found parametrically for OLS1 are listed in Table 5.26.

	OLS1 ω_{OLS1}	NN ω_{NN}	SVM ω_{SVM}
Optimal ω	**0.273**	**0.307**	**0.328**

Table 5.26 Summary of Optimal ω for Average Regression Function for OLS1, NN and SVM

The summary of the residual distributions for the optimal ω value for the average regression function found for neural network $F_{Reg-NN}(X)$ at ω_{NN} = 0.307, support vector machine $F_{Reg-SVM}(X)$ at ω_{NN} = 0.328, $F_{Reg-OLS1}(X)$ at ω_{NN} = 0.273, and $F_{Reg-OLS2}(X)$ are given in Figures 5.30(a), 5.30(b), 5.30(c), and 5.30(d).

Sum of residuals = 0.000 at ω_{NN} = 0.3066
Residual Distribution at ω_{NN} = 0.3066

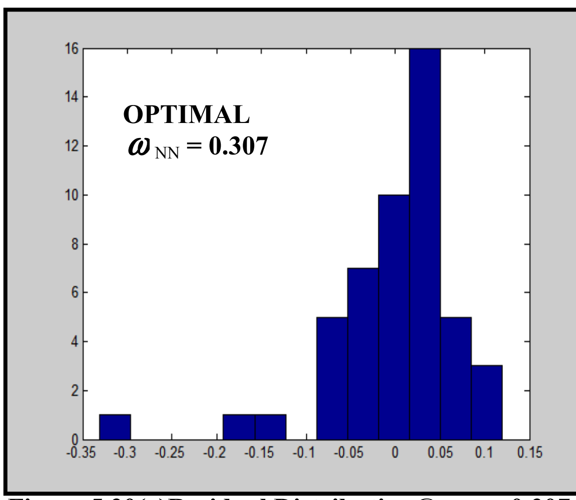


Figure 5.30(a)Residual Distribution@ ω_{NN}=0.307

Sum of residuals = 0.000 at ω_{SVM} = 0.3279
Residual Distribution at ω_{SVM} = 0.3279

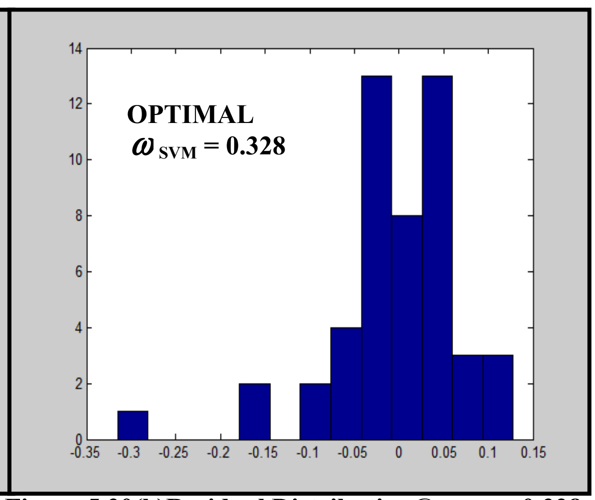


Figure 5.30(b)Residual Distribution@ ω_{SVM}=0.328

Sum of residuals = 0.000 at ω_{OLS1} = 0.2732
Residual Distribution at ω_{OLS1} = 0.2732

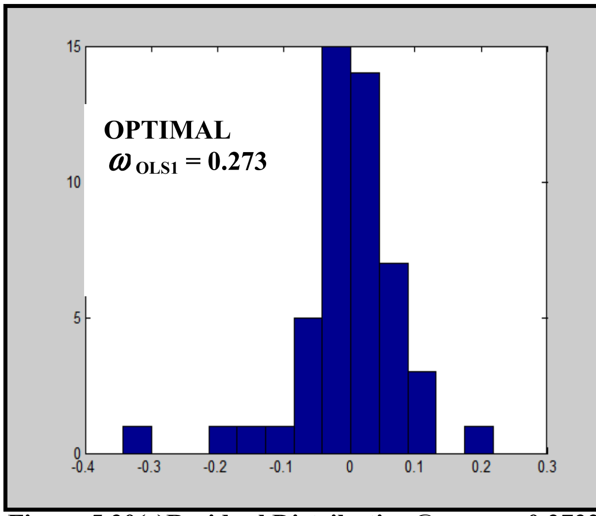


Figure 5.30(c)Residual Distribution@ω_{OLS1}=0.2732

Residual Distribution for $F_{Reg\text{-}OLS2}(X)$

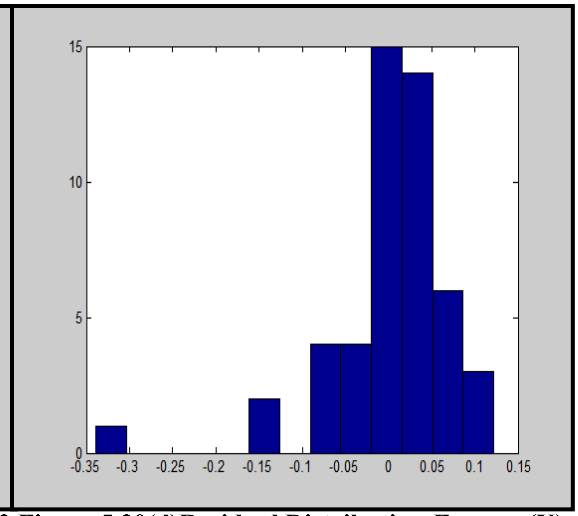


Figure 5.30(d)Residual Distribution-$F_{Reg\text{-}OLS2}$(X)

The optimal ω value of support vector machine, ω_{SVM} = 0.328 for the average regression function determined non-parametrically, is a more accurate estimation of the average function than that of neural network ω_{NN} = 0.307, as well as ω_{OLS1} = 0.273. This conclusion is made because as noted in Section 5.5.3, support vector machine outperformed both neural network and the parametric ordinary least squares regression models OLS1 and OLS2 by more than 2-to-1.

The lack of normality of the residual distributions for the average regression functions for OLS1, OLS2, neural network, and support vector machine are attributed possibly to errors incurred during the initial data collection for the study ‘Program Follow Through’ by Charnes, Cooper and Rhodes (1981). This is validated in some way by observing the scatter plot in Figure 5.1 for the 49 (forty-nine) original single-input, single-output data pairs. Viewing the scatter plot it can be observed that there are 4 (four) DMUs namely DMU1, DMU8, DMU16, and DMU44 which appear to be outliers considering the remaining 45 (forty-five) DMUs. Fitting to mention, the residuals of 3 (three) out of the 4 (four) of these DMUs, namely DMU8, DMU16, and DMU44 stand out in the residual distributions for the average regression function for OLS1, OLS2, neural network and support vector machine. There is also a possibility that there was noise in the data set which also affected the normality of the residual distributions.

Comparing all 4 (four) of the residual distributions in Figure 5.30(a), Figure 5.30(b), Figure 5.30(c), and Figure 5.30(d), if they were to be ranked for symmetry, the support vector machine would take the number 1(one) position, neural network number 2 (two) position, with OLS1 and OLS2 taking the number 3 (three) position.

5.6.3 Determining DEA's "true" Frontier Or Optimal Frontier at F_{λ}

5.6.3.1 Determining DEA's "true" Frontier Or Optimal $F_{\lambda\text{-}NN}(X)$ for NN

As mentioned in Section 4.6, the neural network function for efficient frontier $F_{(Eff\text{-}NN)}(X)$ at $\lambda_{NN} = 1$ determined in Section 5.4.2.1, and the average regression function for neural network $F_{Reg\text{-}NN}(X)$ at $\omega_{NN} = 0.307$ synonymously at $\lambda_{NN} = 0$, determined in Section 5.6.1.1 were combined together in Equation 4.8. Under the assumptions that the managerial errors (***u***) are a positive half-normal distribution, and the observational errors (***v***) are a normal distribution. The λ_{NN} value in Equation 4.8 for neural network was reduced in steps of 0.01 from $\lambda_{NN} = 1$ to $\lambda_{NN} = 0$ and the ratio based statistics for the residuals |A|/|B| as depicted in Figure 4.8 was calculated at each step. The results of the ratio statistic for each λ_{NN} value are given in Table 5.27.

λ_{NN} Value	Ratio Statistic	λ_{NN} Value	Ratio Statistic	λ_{NN} Value	Ratio Statistic
1.00	**0.00**	0.66	**0.03**	0.32	**0.13**
0.99	**0.00**	0.65	**0.03**	0.31	**0.14**
0.98	**0.00**	0.64	**0.03**	0.30	**0.14**
0.97	**0.00**	0.63	**0.03**	0.29	**0.15**
0.96	**0.00**	0.62	**0.03**	0.28	**0.16**
0.95	**0.00**	0.61	**0.04**	0.27	**0.16**
0.94	**0.01**	0.60	**0.04**	0.26	**0.17**
0.93	**0.01**	0.59	**0.04**	0.25	**0.18**
0.92	**0.01**	0.58	**0.04**	0.24	**0.19**
0.91	**0.01**	0.57	**0.04**	0.23	**0.20**
0.90	**0.01**	0.56	**0.05**	0.22	**0.21**
0.89	**0.01**	0.55	**0.05**	0.21	**0.22**
0.88	**0.01**	0.54	**0.05**	0.20	**0.23**
0.87	**0.01**	0.53	**0.05**	0.19	**0.25**
0.86	**0.01**	0.52	**0.06**	0.18	**0.26**
0.85	**0.01**	0.51	**0.06**	0.17	**0.28**
0.84	**0.01**	0.50	**0.06**	0.16	**0.30**
0.83	**0.01**	0.49	**0.06**	0.15	**0.31**
0.82	**0.01**	0.48	**0.07**	0.14	**0.34**
0.81	**0.01**	0.47	**0.07**	0.13	**0.36**
0.80	**0.01**	0.46	**0.07**	0.12	**0.38**
0.79	**0.01**	0.45	**0.08**	0.11	**0.41**
0.78	**0.01**	0.44	**0.08**	0.10	**0.44**
0.77	**0.01**	0.43	**0.08**	0.09	**0.47**
0.76	**0.02**	0.42	**0.09**	0.08	**0.51**
0.75	**0.02**	0.41	**0.09**	0.07	**0.55**
0.74	**0.02**	0.40	**0.09**	0.06	**0.59**
0.73	**0.02**	0.39	**0.10**	0.05	**0.65**
0.72	**0.02**	0.38	**0.10**	0.04	**0.70**
0.71	**0.02**	0.37	**0.11**	0.03	**0.77**
0.70	**0.02**	0.36	**0.11**	0.02	**0.84**
0.69	**0.02**	0.35	**0.11**	0.01	**0.91**
0.68	**0.02**	0.34	**0.12**	0.00	**1.00**
0.67	**0.03**	0.33	**0.13**		

Table 5.27 Ratio Statistic for λ_{NN} Values for Neural Network

The curve plot of the lambda values versus the ratio statistic for neural network is shown in Figure 5.31.

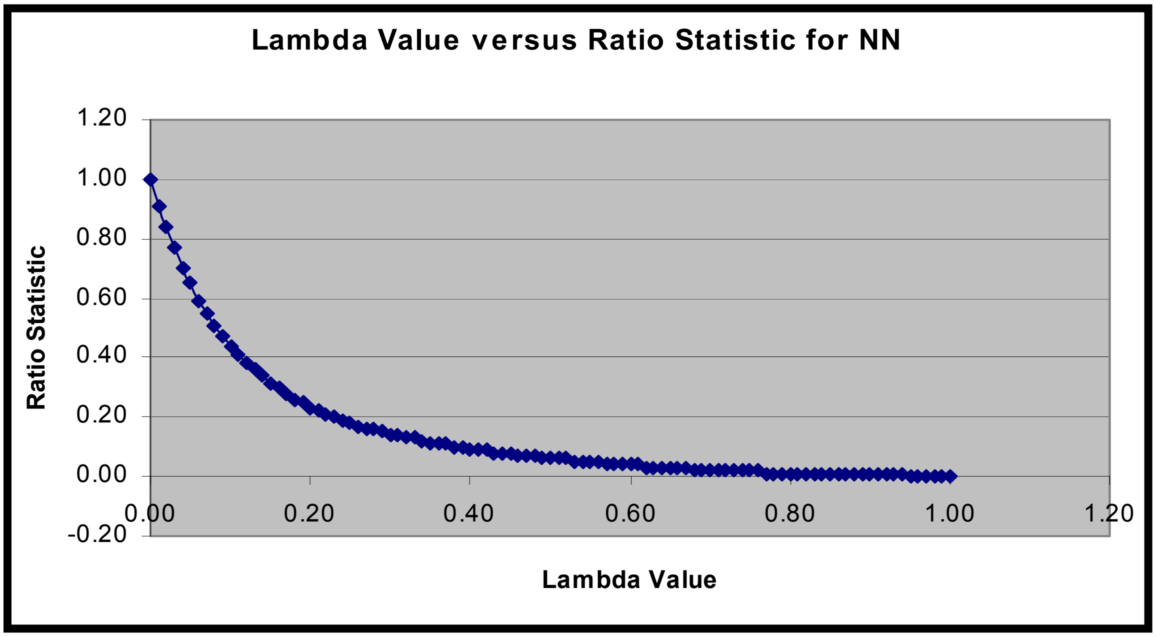


Figure 5.31 Plot of Lambda versus Ratio Statistic for NN

In intervals of 0.05, the slope from 1.00 to 0.00 for the curve plot for the lambda value λ_{NN} versus the ratio statistic for neural network in Figure 5.31 is given in Table 5.28.

Interval	Slope	Interval	Slope
1.00 – 0.95	0.00	0.50 – 0.45	0.40
0.95 – 0.90	0.20	0.45 – 0.40	0.20
0.90 – 0.85	0.00	0.40 – 0.35	0.40
0.85 – 0.80	0.00	0.35 – 0.30	0.60
0.80 – 0.75	0.20	0.30 – 0.25	0.80
0.75 – 0.70	0.00	0.25 – 0.20	1.00
0.70 – 0.65	0.20	0.20 – 0.15	1.60
0.65 – 0.60	0.20	0.15 – 0.10	2.60
0.60 – 0.55	0.20	0.10 – 0.05	4.20
0.55 – 0.50	0.20	0.05 – 0.00	7.00

Table 5.28 Slope in Intervals of 0.05 for Lambda versus Ratio Statistic for NN

Viewing the plot in Figure 5.31, as well as, the slopes for the various intervals in Table 5.28, moving from $\lambda_{NN} = 1$ towards $\lambda_{NN} = 0$, the slopes are almost constant, changing only by approximately 0.20 as the intervals change until interval 0.20 – 0.15, where the slope changes significantly by 0.60. Due to this significant change in slope the optimal lambda must have occurred before the interval 0.20 – 0.15. Observing the ratio statistics for the λ_{NN} values in Table 5.27, starting a $\lambda_{NN} = 1$, the ratio statistic is observed to be repetitive and fairly consistent up until $\lambda_{NN} = 0.27$ where the final repetitive characteristic is observed, thereafter the repetitiveness ceases. Hence, the optimal λ_{NN} value for DEA's "true" frontier for neural network is determined to be at $\lambda_{NN} = 0.26$, which is the first λ_{NN} value after the final repetitive ratio statistic characteristic ceases. Viewing the plot in Figure 5.31, the curve is observed to turn or shift distinctively while moving between interval 0.30 to 0.25 from the $\lambda_{NN} = 1$ direction, which concurs with the optimal λ_{NN} value determined.

5.6.3.2 Determining DEA's "true" Frontier Or Optimal $F_{\lambda\text{-SVM}}(X)$ for SVM

As mentioned in Section 4.6, the support vector machine function for efficient frontier $F_{(Eff\text{-}SVM)}(X)$ at $\lambda_{SVM} = 1$ determined in Section 5.4.2.2, and the average regression function for support vector machine network $F_{Reg\text{-}SVM}(X)$ at $\omega_{SVM} = 0.328$ synonymously at $\lambda_{SVM} = 0$, determined in Section 5.6.1.2 were combined together in Equation 4.9. Under the assumptions that the managerial errors ($\boldsymbol{u}$) are a positive half-normal distribution, and the observational errors ($\boldsymbol{v}$) are a normal distribution. The λ_{SVM} value in Equation 4.9 for support vector machine was reduced in steps of 0.01 from $\lambda_{SVM} = 1$ to $\lambda_{SVM} = 0$ and the ratio based statistics for the residuals $|\mathbf{A}|/|\mathbf{B}|$ as depicted in Figure 4.8 was calculated at each step. The results of the ratio statistic for each λ_{SVM} value are given in Table 5.29.

λ_{SVM} Value	Ratio Statistic	λ_{SVM} Value	Ratio Statistic	λ_{SVM} Value	Ratio Statistic
1.00	**0.01**	0.66	**0.04**	0.32	0.16
0.99	**0.01**	0.65	**0.04**	0.31	0.17
0.98	**0.01**	0.64	**0.04**	0.30	0.17
0.97	**0.01**	0.63	**0.04**	0.29	0.18
0.96	**0.01**	0.62	**0.05**	0.28	0.19
0.95	**0.01**	0.61	**0.05**	0.27	0.20
0.94	**0.01**	0.60	**0.05**	0.26	0.21
0.93	**0.01**	0.59	**0.05**	0.25	0.22
0.92	**0.01**	0.58	**0.05**	0.24	0.23
0.91	**0.01**	0.57	**0.06**	0.23	0.24
0.90	**0.01**	0.56	**0.06**	0.22	0.25
0.89	**0.01**	0.55	**0.06**	0.21	0.26
0.88	**0.02**	0.54	**0.06**	0.20	0.27
0.87	**0.02**	0.53	**0.07**	0.19	0.29
0.86	**0.02**	0.52	**0.07**	0.18	0.30
0.85	**0.02**	0.51	**0.07**	0.17	0.32
0.84	**0.02**	0.50	**0.08**	0.16	0.34
0.83	**0.02**	0.49	**0.08**	0.15	0.36
0.82	**0.02**	0.48	**0.08**	0.14	0.38
0.81	**0.02**	0.47	**0.09**	0.13	0.40
0.80	**0.02**	0.46	**0.09**	0.12	0.43
0.79	**0.02**	0.45	**0.09**	0.11	0.46
0.78	**0.02**	0.44	**0.10**	0.10	0.49
0.77	**0.02**	0.43	**0.10**	0.09	0.52
0.76	**0.02**	0.42	**0.11**	0.08	0.55
0.75	**0.03**	0.41	**0.11**	0.07	0.59
0.74	**0.03**	0.40	**0.11**	0.06	0.64
0.73	**0.03**	0.39	**0.12**	0.05	0.68
0.72	**0.03**	0.38	**0.12**	0.04	0.74
0.71	**0.03**	0.37	**0.13**	0.03	0.79
0.70	**0.03**	0.36	**0.13**	0.02	0.86
0.69	**0.03**	0.35	**0.14**	0.01	0.93
0.68	**0.03**	0.34	**0.15**	0.00	1.00
0.67	**0.04**	0.33	**0.15**		

Table 5.29 Ratio Statistic for λ_{SVM} Values for Support Vector Machine

The curve plot of the lambda values versus the ratio statistic for support vector machine is shown in Figure 5.32.

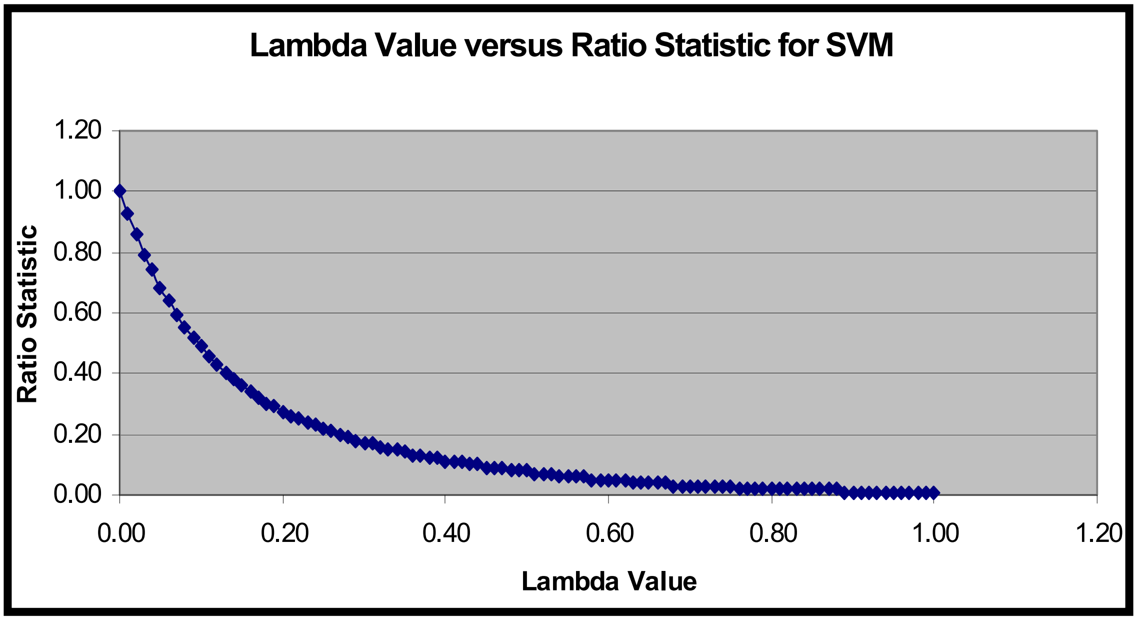


Figure 5.32 Plot of Lambda versus Ratio Statistic for SVM

In intervals of 0.05, the slope from 1.00 to 0.00 for the curve plot for the lambda value λ_{SVM} versus the ratio statistic for support vector machine in Figure 5.32 is given in Table 5.30.

Interval	Slope	Interval	Slope
1.00 – 0.95	**0.00**	0.50 – 0.45	**0.20**
0.95 – 0.90	**0.00**	0.45 – 0.40	**0.40**
0.90 – 0.85	**0.20**	0.40 – 0.35	**0.60**
0.85 – 0.80	**0.00**	0.35 – 0.30	**0.60**
0.80 – 0.75	**0.20**	0.30 – 0.25	**1.00**
0.75 – 0.70	**0.00**	0.25 – 0.20	**1.00**
0.70 – 0.65	**0.20**	0.20 – 0.15	**1.80**
0.65 – 0.60	**0.20**	0.15 – 0.10	**2.60**
0.60 – 0.55	**0.20**	0.10 – 0.05	**3.80**
0.55 – 0.50	**0.40**	0.05 – 0.00	**6.40**

Table 5.30 Slope in Intervals of 0.05 for Lambda versus Ratio Statistic for SVM

Viewing the plot in Figure 5.32, as well as, the slopes for the various intervals in Table 5.30, moving from $\lambda_{SVM} = 1$ towards $\lambda_{SVM} = 0$, the slopes are almost constant, changing by 0.20 at most as the intervals change until interval 0.40 – 0.35, then between interval 0.30 – 0.25 it changes by 0.40 then remained constant for interval 0.30 – 0.25 then the slope changes significantly by 0.80 at interval 0.20-0.15. Due to this significant change in slope the optimal lambda must have occurred before the interval 0.20 – 0.15. Observing the ratio statistics for the λ_{SVM} values in Table 5.30, starting a $\lambda_{SVM} = 1$, the ratio statistic is observed to be repetitive and fairly consistent up until $\lambda_{SVM} = 0.30$ where the final repetitive characteristic is observed, thereafter the repetitiveness ceases. Hence, the optimal λ_{SVM} value for DEA's "true" frontier for support vector machine is determined to be at $\lambda_{SVM} = 0.29$, which is the first λ_{SVM} value after the final repetitive ratio statistic characteristic ceases. Viewing the plot in Figure 5.32, the curve is observed to turn or shift distinctively while moving between interval 0.30 to 0.25 from the $\lambda_{SVM} = 1$ direction, which concurs with the optimal λ_{SVM} value determined.

5.6.3.3 Determining DEA's "true" Frontier Or Optimal $F_{\lambda\text{-}OLS1}(X)$ for Model OLS1

As mentioned in Section 4.6, the parametric ordinary least squares regression model OLS1 function for efficient frontier $F_{(Eff\text{-}OLS1)}(X)$ at $\lambda_{OLS1} = 1$ determined in Section 5.5.1, and the average regression function for model OLS1 $F_{Reg\text{-}OLS1}(X)$ at $\omega_{OLS1} = 0.273$ synonymously at $\lambda_{OLS1} = 0$, determined in Section 5.6.2.1 were combined together in Equation 4.10. Under the assumptions that the managerial errors ($\boldsymbol{u}$) are a positive half-normal distribution, and the observational errors ($\boldsymbol{v}$) are a normal distribution. The λ_{OLS1}

value in Equation 4.10 for model OLS1 was reduced in steps of 0.01 from $\lambda_{OLS1} = 1$ to $\lambda_{OLS1} = 0$ and the ratio based statistics for the residuals $|\mathbf{A}|/|\mathbf{B}|$ as depicted in Figure 4.8 was calculated at each step. The results of the ratio statistic for each λ_{OLS1} value are given in Table 5.31.

λ_{OLS1} Value	Ratio Statistic	λ_{OLS1} Value	Ratio Statistic	λ_{OLS1} Value	Ratio Statistic
1.00	**0.00**	0.66	**0.02**	0.32	**0.12**
0.99	**0.00**	0.65	**0.03**	0.31	**0.12**
0.98	**0.00**	0.64	**0.03**	0.30	**0.13**
0.97	**0.00**	0.63	**0.03**	0.29	**0.14**
0.96	**0.00**	0.62	**0.03**	0.28	**0.14**
0.95	**0.00**	0.61	**0.03**	0.27	**0.15**
0.94	**0.00**	0.60	**0.03**	0.26	**0.16**
0.93	**0.00**	0.59	**0.03**	0.25	**0.16**
0.92	**0.00**	0.58	**0.04**	0.24	**0.17**
0.91	**0.00**	0.57	**0.04**	0.23	**0.18**
0.90	**0.00**	0.56	**0.04**	0.22	**0.19**
0.89	**0.01**	0.55	**0.04**	0.21	**0.20**
0.88	**0.01**	0.54	**0.04**	0.20	**0.21**
0.87	**0.01**	0.53	**0.05**	0.19	**0.23**
0.86	**0.01**	0.52	**0.05**	0.18	**0.24**
0.85	**0.01**	0.51	**0.05**	0.17	**0.25**
0.84	**0.01**	0.50	**0.05**	0.16	**0.27**
0.83	**0.01**	0.49	**0.06**	0.15	**0.29**
0.82	**0.01**	0.48	**0.06**	0.14	**0.31**
0.81	**0.01**	0.47	**0.06**	0.13	**0.33**
0.80	**0.01**	0.46	**0.06**	0.12	**0.35**
0.79	**0.01**	0.45	**0.07**	0.11	**0.38**
0.78	**0.01**	0.44	**0.07**	0.10	**0.42**
0.77	**0.01**	0.43	**0.07**	0.09	**0.45**
0.76	**0.01**	0.42	**0.08**	0.08	**0.49**
0.75	**0.01**	0.41	**0.08**	0.07	**0.54**
0.74	**0.01**	0.40	**0.08**	0.06	**0.58**
0.73	**0.02**	0.39	**0.09**	0.05	**0.64**
0.72	**0.02**	0.38	**0.09**	0.04	**0.69**
0.71	**0.02**	0.37	**0.10**	0.03	**0.76**
0.70	**0.02**	0.36	**0.10**	0.02	**0.83**
0.69	**0.02**	0.35	**0.10**	0.01	**0.91**
0.68	**0.02**	0.34	**0.11**	0.00	**1.00**
0.67	**0.02**	0.33	**0.11**		

Table 5.31 Ratio Statistic for λ_{OLS1} Values for OLS Model OLS1

The curve plot of the lambda values versus the ratio statistic for ordinary least squares regression model OLS1 is shown in Figure 5.33.

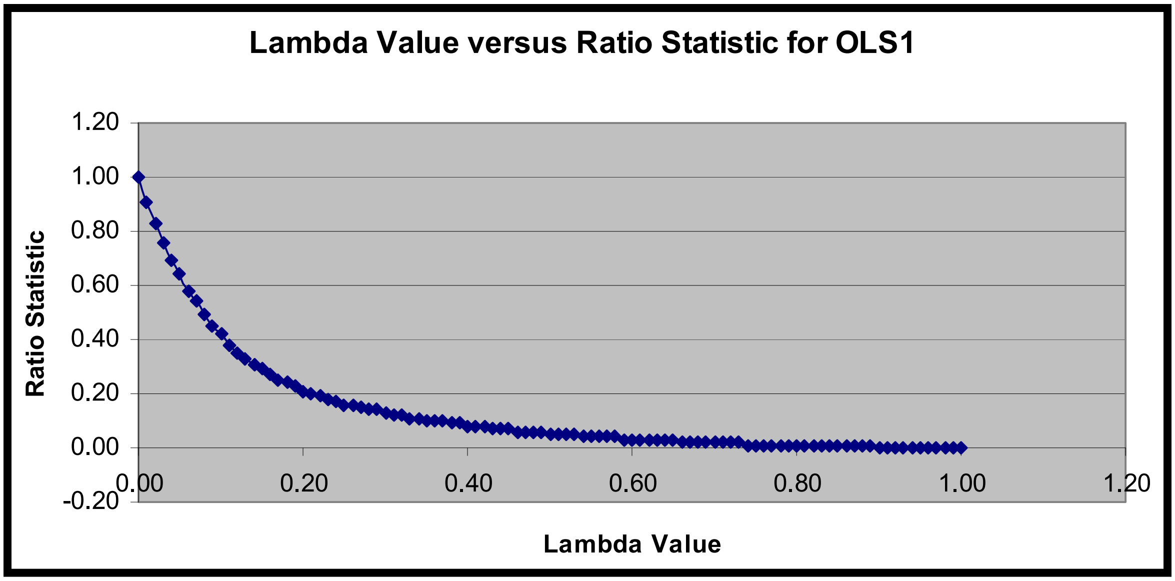


Figure 5.33 Plot of Lambda versus Ratio Statistic for OLS1

In intervals of 0.05, the slope from 1.00 to 0.00 for the curve plot for the lambda value λ_{OLS1} versus the ratio statistic for model OLS1 in Figure 5.33 is given in Table 5.32.

Interval	Slope	Interval	Slope
1.00 – 0.95	**0.00**	0.50 – 0.45	**0.40**
0.95 – 0.90	**0.00**	0.45 – 0.40	**0.20**
0.90 – 0.85	**0.20**	0.40 – 0.35	**0.40**
0.85 – 0.80	**0.00**	0.35 – 0.30	**0.60**
0.80 – 0.75	**0.00**	0.30 – 0.25	**0.60**
0.75 – 0.70	**0.20**	0.25 – 0.20	**1.00**
0.70 – 0.65	**0.20**	0.20 – 0.15	**1.60**
0.65 – 0.60	**0.00**	0.15 – 0.10	**2.60**
0.60 – 0.55	**0.20**	0.10 – 0.05	**4.40**
0.55 – 0.50	**0.20**	0.05 – 0.00	**7.20**

Table 5.32 Slope in Intervals of 0.05 for Lambda versus Ratio Statistic for OLS1

Viewing the plot in Figure 5.33, as well as, the slopes for the various intervals in Table 5.32, moving from $\lambda_{OLS1} = 1$ towards $\lambda_{OLS1} = 0$, the slopes are almost constant, changing by 0.20 at most as the intervals change until interval 0.30 – 0.25, then between interval 0.25 – 0.20 it changes by 0.40 then the slope changes significantly by 0.60 at interval 0.20-0.15. Due to this significant change in slope the optimal lambda must have occurred before the interval 0.20 – 0.15. Observing the ratio statistics for the λ_{OLS1} values in Table 5.31, starting a $\lambda_{OLS1} = 1$, the ratio statistic is observed to be repetitive and fairly consistent up until $\lambda_{OLS1} = 0.25$ where the final repetitive characteristic is observed, thereafter the repetitiveness ceases. Hence, the optimal λ_{OLS1} value for DEA's "true" frontier for model OLS1 is determined to be at $\lambda_{OLS1} = 0.24$, which is the first λ_{OLS1} value after the final repetitive ratio statistic characteristic ceases. Viewing the plot in Figure 5.33, the curve is observed to turn or shift distinctively while moving between interval 0.25 to 0.20 from the $\lambda_{OLS1} = 1$ direction, which concurs with the optimal λ_{OLS1} value determined.

5.6.3.4 Determining DEA's "true" Frontier Or Optimal $F_{\lambda\text{-OLS2}}(X)$ for Model OLS2

As mentioned in Section 4.6, the parametric ordinary least squares regression model OLS2 function for efficient frontier $F_{(Eff\text{-}OLS2)}(X)$ at $\lambda_{OLS2} = 1$ determined in Section 5.5.1, and the average regression function for model OLS1 $F_{Reg\text{-}OLS2}(X)$ at $\lambda_{OLS2} = 0$, determined in Section 5.6.2.2 were combined together in Equation 4.11. Under the assumptions that the managerial errors ($\boldsymbol{u}$) are a positive half-normal distribution, and the observational errors ($\boldsymbol{v}$) are a normal distribution. The λ_{OLS2} value in Equation 4.11 for model OLS2 was reduced in steps of 0.01 from $\lambda_{OLS2} = 1$ to $\lambda_{OLS2} = 0$ and the ratio based statistics for the residuals |A|/|B| as depicted in Figure 4.8 was calculated at each step. The results of the ratio statistic for each λ_{OLS2} value are given in Table 5.33.

λ_{OLS2} Value	Ratio Statistic	λ_{OLS2} Value	Ratio Statistic	λ_{OLS2} Value	Ratio Statistic
1.00	**0.00**	0.66	**0.02**	0.32	**0.11**
0.99	**0.00**	0.65	**0.02**	0.31	**0.12**
0.98	**0.00**	0.64	**0.02**	0.30	**0.12**
0.97	**0.00**	0.63	**0.03**	0.29	**0.13**
0.96	**0.00**	0.62	**0.03**	0.28	**0.14**
0.95	**0.00**	0.61	**0.03**	0.27	**0.14**
0.94	**0.00**	0.60	**0.03**	0.26	**0.15**
0.93	**0.00**	0.59	**0.03**	0.25	**0.16**
0.92	**0.00**	0.58	**0.03**	0.24	**0.16**
0.91	**0.00**	0.57	**0.04**	0.23	**0.17**
0.90	**0.01**	0.56	**0.04**	0.22	**0.18**
0.89	**0.01**	0.55	**0.04**	0.21	**0.19**
0.88	**0.01**	0.54	**0.04**	0.20	**0.20**
0.87	**0.01**	0.53	**0.04**	0.19	**0.21**
0.86	**0.01**	0.52	**0.05**	0.18	**0.23**
0.85	**0.01**	0.51	**0.05**	0.17	**0.24**
0.84	**0.01**	0.50	**0.05**	0.16	**0.26**
0.83	**0.01**	0.49	**0.05**	0.15	**0.27**
0.82	**0.01**	0.48	**0.06**	0.14	**0.29**
0.81	**0.01**	0.47	**0.06**	0.13	**0.32**
0.80	**0.01**	0.46	**0.06**	0.12	**0.34**
0.79	**0.01**	0.45	**0.06**	0.11	**0.37**
0.78	**0.01**	0.44	**0.07**	0.10	**0.39**
0.77	**0.01**	0.43	**0.07**	0.09	**0.43**
0.76	**0.01**	0.42	**0.07**	0.08	**0.46**
0.75	**0.01**	0.41	**0.08**	0.07	**0.50**
0.74	**0.01**	0.40	**0.08**	0.06	**0.55**
0.73	**0.02**	0.39	**0.08**	0.05	**0.60**
0.72	**0.02**	0.38	**0.09**	0.04	**0.66**
0.71	**0.02**	0.37	**0.09**	0.03	**0.73**
0.70	**0.02**	0.36	**0.10**	0.02	**0.81**
0.69	**0.02**	0.35	**0.10**	0.01	**0.89**
0.68	**0.02**	0.34	**0.10**	0.00	**1.00**
0.67	**0.02**	0.33	**0.11**		

Table 5.33 Ratio Statistic for λ_{OLS2} Values for OLS Model OLS2

The curve plot of the lambda values versus the ratio statistic for ordinary least squares regression model OLS2 is shown in Figure 5.34.

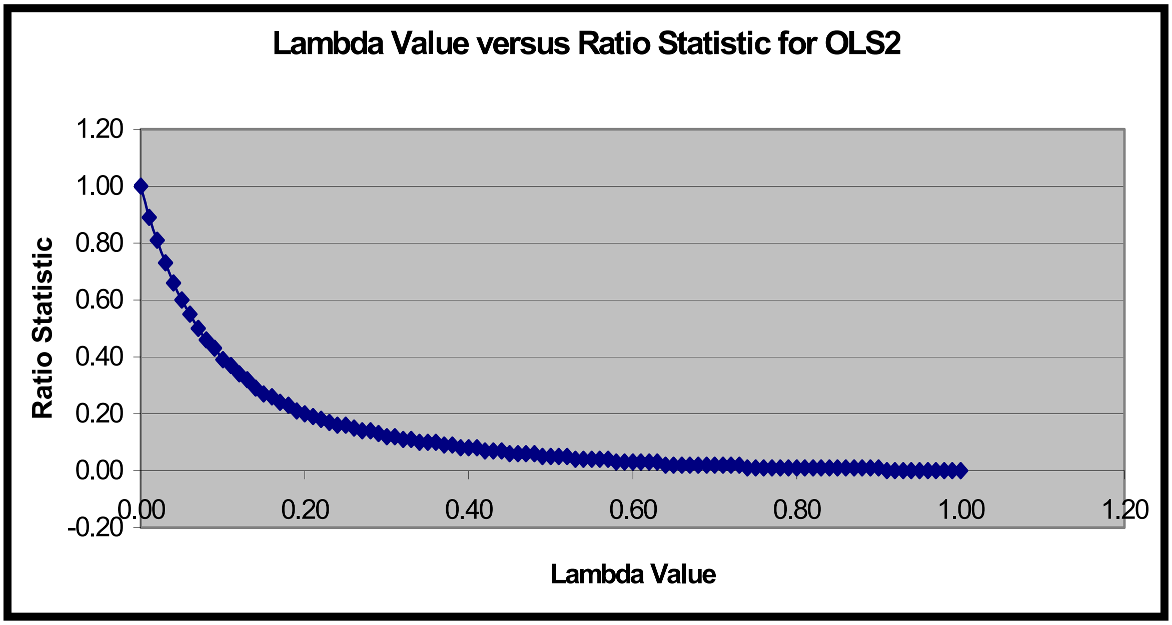


Figure 5.34 Plot of Lambda versus Ratio Statistic for OLS2

In intervals of 0.05, the slope from 1.00 to 0.00 for the curve plot for the lambda value λ_{OLS2} versus the ratio statistic for model OLS2 in Figure 5.34 is given in Table 5.34.

Interval	Slope	Interval	Slope
1.00 – 0.95	**0.00**	0.50 – 0.45	**0.20**
0.95 – 0.90	**0.20**	0.45 – 0.40	**0.40**
0.90 – 0.85	**0.00**	0.40 – 0.35	**0.40**
0.85 – 0.80	**0.00**	0.35 – 0.30	**0.40**
0.80 – 0.75	**0.00**	0.30 – 0.25	**0.80**
0.75 – 0.70	**0.20**	0.25 – 0.20	**0.80**
0.70 – 0.65	**0.00**	0.20 – 0.15	**1.40**
0.65 – 0.60	**0.20**	0.15 – 0.10	**2.40**
0.60 – 0.55	**0.20**	0.10 – 0.05	**4.20**
0.55 – 0.50	**0.20**	0.05 – 0.00	**8.00**

Table 5.34 Slope in Intervals of 0.05 for Lambda versus Ratio Statistic for OLS2

Viewing the plot in Figure 5.34, as well as, the slopes for the various intervals in Table 5.34, moving from $\lambda_{OLS2} = 1$ towards $\lambda_{OLS2} = 0$, the slopes are almost constant, changing by 0.20 at most as the intervals change until interval 0.35 – 0.30, then between interval 0.30 – 0.25 it changes by 0.40 then remained constant for interval 0.25 – 0.20 then the slope changes significantly by 0.60 at interval 0.20-0.15. Due to this significant change in slope the optimal lambda must have occurred before the interval 0.20 – 0.15. Observing the ratio statistics for the λ_{OLS2} values in Table 4.33, starting a $\lambda_{OLS2} = 1$, the ratio statistic is observed to be repetitive and fairly consistent up until $\lambda_{OLS2} = 0.24$ where the final repetitive characteristic is observed, thereafter the repetitiveness ceases. Hence, the optimal λ_{OLS2} value for DEA's "true" frontier for model OLS2 is determined to be at $\lambda_{OLS2} = 0.23$, which is the first λ_{OLS2} value after the final repetitive ratio statistic characteristic ceases. Viewing the plot in Figure 5.34, the curve is observed to turn or shift distinctively while moving between interval 0.25 to 0.20 from the $\lambda_{OLS2} = 1$ direction, which concurs with the optimal λ_{OLS2} value determined.

5.6.3.5 Analysis of DEA's "true" Frontier for OLS, NN, and SVM

The summary of the optimal λ value for DEA's "true" frontier for the two ordinary least squares regression models OLS1 and OLS2, neural network and support vector machine are listed in Table 5.35.

	OLS1 λ_{OLS1}	**OLS2** λ_{OLS2}	**NN** λ_{NN}	**SVM** λ_{SVM}
Optimal λ	**0.24**	**0.23**	**0.26**	**0.29**

Table 5.35 Summary of Optimal λ for DEA's "true" Frontier Function F_λ for OLS1, NN and SVM

The summary of the curves for λ value versus the ratio statistic for OLS1, OLS2, neural network, and support vector machine are given in Figure 5.35.

The optimal λ values for DEA's "true" frontier determined for the two parametric ordinary least squares regression models $\lambda_{\mathbf{OLS1}}$ and $\lambda_{\mathbf{OLS2}}$ were observed to be almost equal, whereas that for neural network $\lambda_{\mathbf{NN}}$ was closer to $\lambda_{\mathbf{OLS1}}$ than it was to the optimal λ for support vector machine $\lambda_{\mathbf{SVM}}$.

These results conclude that considering managerial errors, as well as, observational errors, the latter which are often not taken under consideration for traditional DEA, the "true" frontier for DEA may be well below the traditional DEA frontier. SVM was observed to outperform both neural network and ordinary least squares regression models in this research, while the performance of neural network and ordinary least squares regression were almost the same.

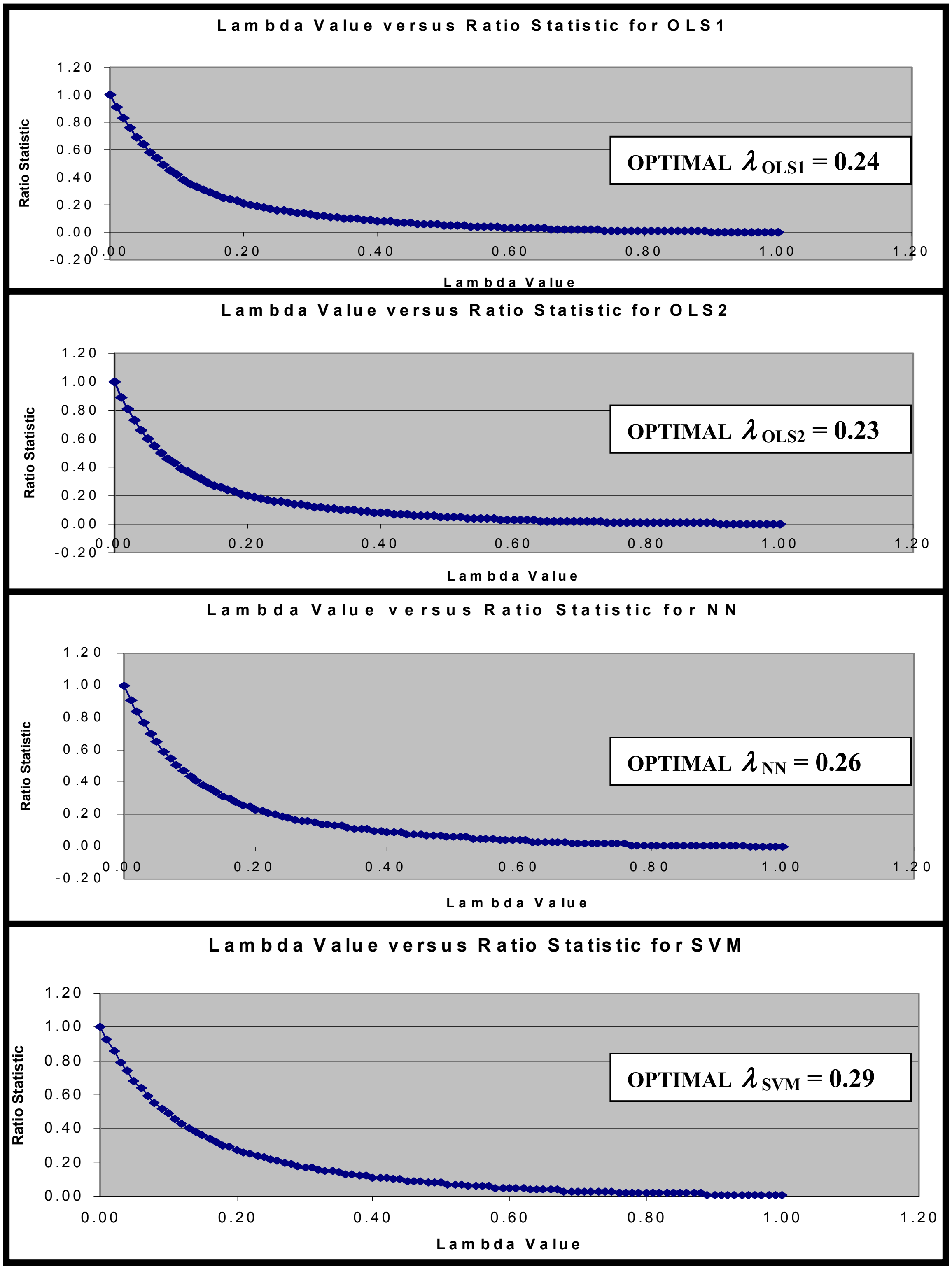


Figure 5.35 Summary of curves for λ value versus the ratio statistic for OLS1, OLS2, NN, and SVM

5.6.4 Benchmarking for Ratio Statistic Method

In order to assess the performance of the ratio statistic applied in Section 5.6.3 to determine DEA's "true" frontier for neural network, support vector machine, OLS1 and OLS2, some form of benchmarking is necessary. To achieve this benchmarking, the probability density function (pdf) and the area under the curve statistics are applied to the residuals obtained at the optimal λ for NN, SVM, OLS1, and OLS2. The residual distributions for the optimal λ values for NN ($\lambda_{NN} = 0.26$), SVM ($\lambda_{SVM} = 0.29$), OLS1 ($\lambda_{OLS1} = 0.24$), and OLS2 ($\lambda_{OLS2} = 0.23$) are depicted in Figure 5.36, Figure 5.37, Figure Figure 5.38, and Figure 5.39 respectively.

Residual Distribution for Optimal $\lambda_{NN} = 0.26$

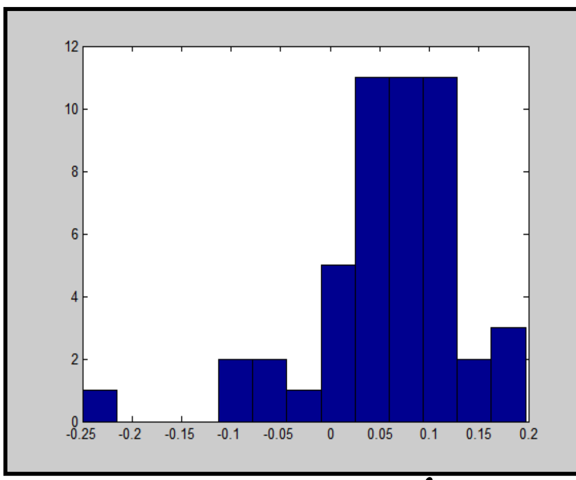


Figure 5.36 Residual Distribution@ $\lambda_{NN} = 0.26$

Residual Distribution for Optimal $\lambda_{SVM} = 0.29$

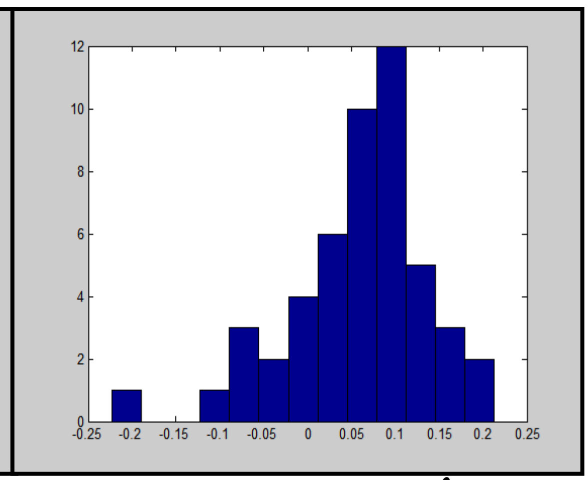


Figure 5.37 Residual Distribution@ $\lambda_{SVM} = 0.29$

Residual Distribution for Optimal $\lambda_{OLS1} = 0.24$

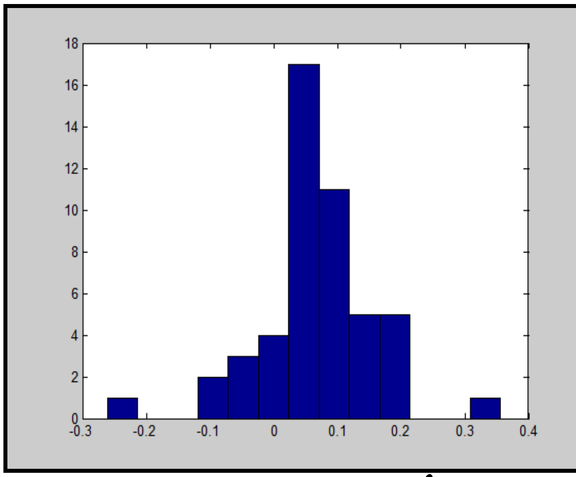


Figure 5.38 Residual Distribution@ $\lambda_{OLS1} = 0.24$

Residual Distribution for Optimal $\lambda_{OLS2} = 0.23$

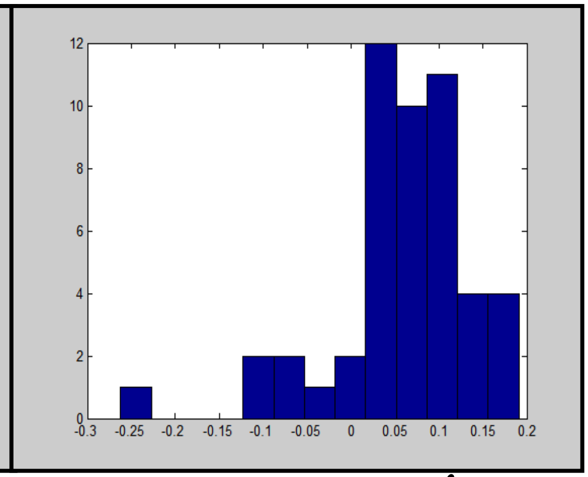


Figure 5.39 Residual Distribution@ $\lambda_{OLS2} = 0.23$

Applying the probability density function and the area under the curve to the residual distribution for neural network given in Figure 5.36, the ratio of the frequency count on the left hand side to that of the right hand side is:

11/49 * 49/38 = 11/38 = **0.29**

This benchmark value is approximately 11.54% (percent) above the optimal lambda value determined by the ratio statistic method used for the research which was concluded to be $\lambda_{NN} = 0.26$.

Applying the probability density function and the area under the curve to the residual distribution for support vector machine given in Figure 5.37, the ratio of the frequency count on the left hand side to that of the right hand side is:

11/49 * 49/38 = 11/38 = **0.29**

This benchmark value is equal to the optimal lambda value determined by the ratio statistic method used for the research which was concluded to be $\lambda_{SVM} = 0.29$.

Applying the probability density function and the area under the curve to the residual distribution for OLS1 given in Figure 5.38, the ratio of the frequency count on the left hand side to that of the right hand side is:

10/49 * 49/39 = 10/39 = **0.26**

This benchmark value is approximately 8.33% (percent) above the optimal lambda value determined by the ratio statistic method used for the research which was concluded to be $\lambda_{OLS1} = 0.24$.

Applying the probability density function and the area under the curve to the residual distribution for OLS2 given in Figure 5.39, the ratio of the frequency count on the left hand side to that of the right hand side is:

8/49 * 49/41 = 8/41 = **0.20**

This benchmark value is approximately 13.04% (percent) below the optimal lambda value determined by the ratio statistic method used for the research which was concluded to be $\lambda_{OLS2} = 0.23$.

5.6.4.1 Analysis of Benchmarking Versus Ratio Statistic Results

The summary of the optimal λ value for DEA's "true" frontier for the two ordinary least squares regression models OLS1 and OLS2, neural network and support vector machine determined by the ratio statistic method employed in this research, and those determined by the benchmarking method are listed in Table 5.36.

	OLS1 λ_{OLS1}	OLS2 λ_{OLS2}	NN λ_{NN}	SVM λ_{SVM}
Ratio Statistic Optimal λ	**0.24**	**0.23**	**0.26**	**0.29**
Benchmarking Optimal λ	**0.26**	**0.20**	**0.29**	**0.29**
PERCENTAGE ERROR	**+8.33%**	**-13.04%**	**+11.54%**	**0.00%**

Table 5.36 Summary of Ratio Statistic and Benchmarking Optimal λs for DEA's "true" Frontier Function F_{λ} for OLS1, NN and SVM

These results conclude that the application of the ratio statistic compared to the area under the curve benchmarking method used for this research were equal in determining the optimal λ value for DEA's "true" frontier for support vector machine. The comparisons also showed the OLS1 model outperforming NN which came in third, as well as, the OLS2 model which placed last.

Once again, the results are consistent with the initial findings in this research in Section 5.5.3, which concluded that SVM outperformed both ordinary least squares regression, as well as, neural network in estimating a nonlinear function, and efficient and inefficient frontiers for the VRS case in this research. However, more importantly in this section, the results showed that the method utilized, those being OLS1, OLS2, NN, or SVM also affected the accuracy of the ratio statistic utilized in this research in order to determine DEA's "true" or optimal frontier.

5.7 Summary of Chapter Five

In this chapter, we observed the proposed methodology developed in Chapter Four to determine DEA's "true" or optimal frontier applied on a real data set, from the original study 'Program Follow Through' by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students and as may be viewed in Appendix A1. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), with a total of 49 (forty-nine) data pairs in its entirety were used under DEA's assumption of variable return to scale. A summary of the main results and analysis are as follows:

1. Three DMUs, namely DMU15, DMU44, and DMU48 were determined to exist on empirical or traditional DEA efficient frontier $F_{(Eff)}$.
2. Four DMUs, namely DMU1, DMU5, DMU32, and DMU36 were determined to exist on the newly introduced DEA inefficient frontier $F_{(Ineff)}$ in this research.
3. Utilizing the common heuristic to determine the neural network architecture in Section 4.4 for a single-input and single-output data pairs, the neural network architecture was determined to be a 1-3-1 (i.e. 1(one) input node, 3(three) hidden nodes, and 1(one) output node) for the efficient frontier for neural network $F_{(Eff\text{-}NN)}$. The optimal model and function along with its parameters given by Statistica 8.0 for the neural network function for efficient frontier $F_{(Eff\text{-}NN)}(X)$ were:
    - The neural network architecture was a three layer, 1-3-1 multi-layer perceptron model.

- The training algorithm which gave the optimal neural network was BFGS (i.e. Quasi-Newton Back Propagation by Broyden, Fletcher, Goldfarb, and Shanno (BFGS)) (Hagen, Demuth, and Beale, 1996; and Haykin, 2000).
    - The hidden activation was Tanh, and the output activation Tanh.
    - The error function was sum of squares (SOS).

4. The sum of the squared errors for the output values for the efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ on the training set was **0.001151.** The sum of the squared errors for the output values for the efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ on the test set was **0.000147.** The total sum of the squared errors for the output values for the efficient frontier for the neural network function $F_{(Eff\text{-}NN)}(X)$ on both training and test set was **0.001298.**
5. The optimal model and function along with its parameters given by Statistica 8.0 for the support vector machine function for efficient frontier $F_{(Eff\text{-}SVM)}(X)$ were:
    - The support vector type was Regression Type 1, also referred to as epsilon-svm.
    - The optimal kernel type was a radial basis function of gamma value 5 (five).
    - The number of support vectors was 13 (thirteen) with 2 (two) bounded.
    - The optimal C value was 68 (sixty-eight).
    - The optimal $\boldsymbol{\varepsilon}$ (epsilon) value was 0.0015.
6. The sum of the squared errors for the output values for the efficient frontier for the support vector machine function $F_{(Eff\text{-}SVM)}(X)$ on the training set was **0.000555.** The sum of the squared errors for the output values for the efficient frontier for the support vector machine function $F_{(Eff\text{-}SVM)}(X)$ on the test set was **0.000057.** The total

sum of the squared errors for the output values for the efficient frontier for the support vector machine function $F_{(Eff\text{-}SVM)}(X)$ on both training and test set was **0.000613.**

7. Utilizing the common heuristic to determine the neural network architecture in Section 4.4 for a single-input and single-output data pairs, the neural network architecture was determined to be a 1-3-1 (i.e. 1(one) input node, 3(three) hidden nodes, and 1(one) output node) for the inefficient frontier for neural network $F_{(Ineff\text{-}NN)}$. The optimal model and function along with its parameters given by Statistica 8.0 for the neural network function for inefficient frontier $F_{(Ineff\text{-}NN)}(X)$ were:
   - The neural network architecture was a three layer, 1-3-1 multi-layer perceptron model.
   - The training algorithm which gave the optimal neural network was BFGS (i.e. Quasi-Newton Back Propagation by Broyden, Fletcher, Goldfarb, and Shanno (BFGS)) (Hagen, Demuth, and Beale, 1996; and Haykin, 2000).
   - The hidden activation was Tanh, and the output activation Identity.
   - The error function was sum of squares (SOS).
8. The sum of the squared errors for the output values for the inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ on the training set was **0.000198.** The sum of the squared errors for the output values for the inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ on the test set was **0.000055.** The total sum of the squared errors for the output values for the inefficient frontier for the neural network function $F_{(Ineff\text{-}NN)}(X)$ on both training and test set was **0.000254.**

9. The optimal model and function along with its parameters given by Statistica 8.0 for the support vector machine function for inefficient frontier $F_{(Ineff\text{-}SVM)}(X)$ were:
    - The support vector type was Regression Type 1, also referred to as epsilon-svm.
    - The optimal kernel type was a radial basis function of gamma value 3 (three).
    - The number of support vectors was 12 (twelve) with 1 (one) bounded.
    - The optimal C value was 69 (sixty-nine).
    - The optimal ε (epsilon) value was 0.0000.
10. The sum of the squared errors for the output values for the inefficient frontier for the support vector machine function $F_{(Ineff\text{-}SVM)}(X)$ on the training set was **0.000047.** The sum of the squared errors for the output values for the inefficient frontier for the support vector machine function $F_{(Ineff\text{-}SVM)}(X)$ on the test set was **0.000055.** The total sum of the squared errors for the output values for the inefficient frontier for the support function $F_{(Ineff\text{-}SVM)}(X)$ on both training and test set was **0.000102.**
11. The optimal model and function given for the parametric ordinary least squares regression models OLS1 and OLS2 for efficient frontier $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$ was quadratic and given as:
    - Y = 0.267222 + 1.30248X - 0.233956X**2
12. The sum of the squared errors for the output values for the efficient frontier for the two OLS models' function, where $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$, on the training set was **0.001038.** The sum of the squared errors for the output values for the efficient frontier for the two OLS models' function, where $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$, on the test set was **0.000149.** The total sum of the squared errors for the output values for the

efficient frontier for the two OLS models' function, where $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$, on both training and test set was **0.001188.**

13. The optimal model and function given for the parametric ordinary least squares regression model OLS1 for inefficient frontier $F_{(Inff\text{-}OLS1)}(X)$ was quadratic and given as:
    - Y = 0.179220 + 0.325992X + 3.85E-02X**2
14. The sum of the squared errors for the output values for the inefficient frontier for the regression model OLS1 function $F_{(Ineff\text{-}OLS1)}(X)$ on the training set was **0.000258.** The sum of the squared errors for the output values for the inefficient frontier for the regression model OLS1 function $F_{(Ineff\text{-}OLS1)}(X)$ on the test set was **0.000097.** The total sum of the squared errors for the output values for the inefficient frontier for the regression model OLS1 function $F_{(Ineff\text{-}OLS1)}(X)$ on both training and test set was **0.000355.**
15. There was no inefficient frontier function for the parametric ordinary least squares regression model OLS2.
16. The sum of squared errors for the parametric ordinary least squares regression model OLS1, and the sum of squared errors for the non-parametric neural network model for efficient frontier, inefficient frontier, and efficient frontier + inefficient frontier were almost equal for both the training and test sets. The sum of squared errors however for the support vector machine was less than half that for the training and test set of both OLS1, as well as, the neural network. Support vector machine outperformed both ordinary least squares regression, as well as, neural network by more 2-to-1 in

estimating a nonlinear function, and efficient and inefficient frontiers for the VRS case in this research.

17. The optimal value of ω determined for the average regression functions for NN, SVM, and OLS1 were $\omega_{NN} = 0.307$, $\omega_{SVM} = 0.328$, and $\omega_{OLS1} = 0.273$ respectively.
18. The optimal model and function given for the parametric ordinary least squares regression model OLS2 for the average regression function $F_{Reg\text{-}OLS2}(X)$ was quadratic and given as:
    - Y = 0.198604 + 0.785125X - 0.430650X**2
19. The optimal ω value of support vector machine, $\omega_{SVM} = 0.328$ for the average regression function determined non-parametrically, was concluded to be a more accurate estimation of the average function than that of neural network $\omega_{NN} = 0.307$, as well as $\omega_{OLS1} = 0.273$. This conclusion was made because as noted in Section 5.5.3, support vector machine outperformed both neural network and the parametric ordinary least squares regression models OLS1 and OLS2 by more than 2-to-1.
20. The lack of normality of the residual distributions for the average regression functions for OLS1, OLS2, neural network, and support vector machine were attributed possibly to errors incurred during the initial data collection for the study 'Program Follow Through' by Charnes, Cooper and Rhodes (1981). This was validated in some way by observing the scatter plot in Figure 5.1 for the 49 (forty-nine) original single-input, single-output data pairs. Viewing the scatter plot it was observed where there were 4 (four) DMUs namely DMU1, DMU8, DMU16, and DMU44 which appear to be outliers considering the remaining 45 (forty-five) DMUs. Fitting to mention, the residuals of 3 (three) out of the 4 (four) of these DMUs, namely DMU8, DMU16, and

DMU44 stood out in the residual distributions for the average regression function for OLS1, OLS2, neural network and support vector machine. There was also a possibility that there was noise in the data set which also affected the normality of the residual distributions.

21. Comparing all 4 (four) of the residual distributions in Figure 5.30(a), Figure 5.30(b), Figure 5.30(c), and Figure 5.30(d), if they were to be ranked for symmetry, the support vector machine would take the number 1(one) position, neural network number 2 (two) position, with OLS1 and OLS2 taking the number 3 (three) position.
22. The ratio statistic optimal λ value determined for DEA's "true" or optimal frontier for NN, SVM, OLS1, and OLS2 were $\lambda_{NN} = 0.26$, $\lambda_{SVM} = 0.29$, $\lambda_{OLS1} = 0.24$, and $\lambda_{OLS2} = 0.23$ respectively.
23. The ratio optimal λ values for DEA's "true" frontier determined for the two parametric ordinary least squares regression models $\lambda_{\mathbf{OLS1}}$ and $\lambda_{\mathbf{OLS2}}$ were observed to be almost equal, whereas that for neural network $\lambda_{\mathbf{NN}}$ was closer to $\lambda_{\mathbf{OLS1}}$ than it was to the optimal λ for support vector machine $\lambda_{\mathbf{SVM}}$.
24. The benchmarking optimal λ value determined for DEA's "true" or optimal frontier for NN, SVM, OLS1, and OLS2 were $\lambda_{NN} = 0.29$, $\lambda_{SVM} = 0.29$, $\lambda_{OLS1} = 0.26$, and $\lambda_{OLS2} = 0.20$ respectively.
25. The application of the ratio statistic compared to the area under the curve benchmarking method used for this research were equal in determining the optimal λ value for DEA's "true" frontier for support vector machine. The comparisons also showed the OLS1 model outperforming NN which came in third, as well as, the OLS2 model which placed last.

26. The benchmarking results were consistent with the initial findings in Section 5.5.3, which concluded that SVM outperformed both ordinary least squares regression, as well as, neural network in estimating a nonlinear function, and efficient and inefficient frontiers for the VRS case in this research. However, more importantly, the benchmarking results showed that the method utilized, those being OLS1, OLS2, NN, or SVM also affected the accuracy of the ratio statistic utilized in this research in order to determine DEA's "true" or optimal frontier.

These results conclude that considering managerial errors, as well as, observational errors, the latter which are often not taken under consideration for traditional DEA, the "true" frontier for DEA may be well below the traditional DEA frontier. SVM was observed to outperform both neural network and ordinary least squares regression models in this research, while the performance of neural network and ordinary least squares regression were almost the same.

6. CHAPTER SIX: CONCLUSIONS AND FUTURE WORK

This chapter contains a general summary of the findings of the research on how support vector machine, neural network, and ordinary least squares regression performed relative to each other. Future work, for example, in the form of application of the proposed methodology developed to data sets consisting of multiple-inputs, multiple-outputs, larger number of decision making units, and other situations are also suggested in this chapter.

6.1 Conclusions

This research showed by the utilization of the data set from the original study ‘Program Follow Through’ by Charnes, Cooper and Rhodes (1981) for one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), under DEA’s VRS assumptions, that support vector machine outperformed both ordinary least squares regression and neural network models in predicting and estimating a non-linear function for efficient frontier, inefficient frontier, and efficient + inefficient frontiers. In fact, the performance of neural network was observed to be almost identical to that of ordinary least squares regression with neural network performance being only marginally better than ordinary least squares regression.

While researchers known to date, such as Wang (2003), have attempted to determine an average regression function for traditional DEA, how this may be determined with clarity has been addressed by this research. This research and it's results showed that by combining non-parametric efficient frontier and non-parametric inefficient frontier, this average regression function can be determined in a totally non-parametric environment with no assumptions whatsoever. The optimal ω value was determined for ordinary least squares regression, neural network, and support vector. Based upon the first experiments carried out and the results obtained in the research on the performance of these models, it is concluded that the optimal ω determined by support vector machine is a more accurate estimation than the value determined by OLS and NN for determining the optimal average regression function in this research. The lack of normality of the residual distributions for the average regression function for ordinary least squares regression, neural network, and support vector machine in this research is attributed to the possibility of errors which may have occurred in the data collection process by Charnes, Cooper and Rhodes (1981) as outliers for 4 (four) DMUs are salient in the scatter plot for the original 49 (forty-nine), single-input, single-output data pairs for the DMUs.

This research provided and tested a methodology for determining DEA's "true" or optimal frontier for OLS, NN, and SVM by combining the average regression function and efficient frontier function, then applying a ratio statistic based upon the residuals for the absolute values for observational errors and managerial errors while varying the λ value. The results showed that the optimal λ for neural network was closer to the optimal λ value for OLS than to the optimal λ obtained for SVM. Based upon the first experiments carried out and the results obtained in the research on the performance of these models, it is concluded that the optimal λ determined by support vector machine is a more accurate estimation than the value determined by OLS and NN for determining the “true” or optimal DEA frontier or function in this research.

The key finding in the research results, in the determination of an optimal λ value very much lower than that of the traditional DEA frontier at $\lambda = 1$, may benefit managers and decision-makers by assisting them in making better judgements and decisions relative to:

- How they allocate their company’s or organization’s resources.
- How they assess performance, productivity, and efficiency more accurately.
- How they may compete more profitably and effectively both in a national, as well as, an increasingly global environment.

6.2 Future Work

This research utilized the data set from the original study ‘Program Follow Through’ by Charnes, Cooper and Rhodes (1981) for assessing the educational programs for disadvantaged students. Low dimensional data, consisting of one input (i.e. educational level of mother = x), one output (i.e. coopersmith scores = y), with a total of 49 (forty-nine) data pairs in its entirety were used under DEA’s assumption of variable return to scale. As this research only considered a single-input and a single-output for 49 (forty nine) DMUs under the VRS case, future research on single-input and single-output, for a larger number of DMUs, such as possibly 100 (one-hundred) DMUs, 500 (five-hundred) DMUs, and 1000 (one-thousand) DMUs should be carried out to determine whether the methodology is applicable across both small and large numbers of DMUs. The CRS assumptions can be examined and investigated as well.

Further research utilizing multiple-inputs and single-output, multiple-outputs and single input, as well as, multiple-inputs and multiple-outputs would also be beneficial to determine if the methodology may be applicable in such cases as well. These may also be expanded to incorporate larger numbers of DMUs as mentioned in the previous paragraph, as well as, the CRS case.

While Statistica 8.0 was allowed to optimize the parameters for the neural network and support vector machine functions and models in this research, further research may explore the possibility of whether these optimal functions and models determined may be fine-tuned to give better or improved results. For the ordinary least squares regression models, instead of utilizing only a linear or quadratic regression function, other regression functions such as those included in Datafit 8.2 could be examined to determine whether they would improve the regression functions or models in this research.

Finally, the single-input, single-output data set used in this research was partially skewed as was observed in the scatter plot. Future research, may wish to apply the research methodology to a ‘highly’ normally distributed data set which is obtained by very careful collection, or by utilizing a data set which is known to be normally distributed, so as to see how the results compare to those in this research.

REFERENCES AND BIBLIOGRAPHY

Aigner D, Lovell C, Schmidt P. Formulation and estimation of stochastic frontier production models. *Journal of Econometrics*. 1977; 6:21-37.

Athanassopoulos AD, Curram SP. A comparison of data envelopment analysis and artificial neural networks as tool for assessing the efficiency of decision making units. *Journal of the Operational Research Society*. 1996; 47(8):1000-1016.

Banker RD, Charnes A, Cooper WW. Models for the estimation of technical and scale inefficiencies in data envelopment analysis. *Management Science.* 1984; 30:1078-1092.

Banxia Software Ltd. *Banxia DEA Software 3.0.* UK, 1995-2007. Available at: www.banxia.com. Accessed March 16, 2006.

Battese BE, Corra GS. Estimation of a production frontier model: With application to the pastoral zone off Eastern Australia. *Australian Journal of Agricultural Economics*. 1997; 21(3):169-179.

Bauer PW. Recent developments in the econometric estimation of frontiers. *Journal of Econometrics*. 1990; 46(1/2):39-56.

Bravo-Ureta BE, Pinheiro AE. Efficiency analysis of developing country agriculture: A review of the frontier function literature. *Agricultural and Resource Economic Review*. 1993; 22:88-101.

Cao L. Support vector machines experts for time series forecasting. *Neurocomputing*. 2003; 51:321-339.

Charnes A, Cooper WW, Lewin A, Seiford LW. *Data envelopment analysis: theory, methodology and applications.* Boston, MA: Kluwer Academic Publishers; 1994.

Charnes A, Cooper WW, Rhodes E. Evaluating program and managerial efficiency: An application of data envelopment analysis to 'Program Follow-Through'. *Management Science*. 1981; 27(6):668-697.

Charnes A, Cooper WW, Rhodes E. Measuring the efficiency of decision making Units. *European Journal of Operational Research*. 1978; 2:429-444.

Cherkassy V, Mulier F. *Learning from data: Concepts, theory, and methods.* New York, NY: John Wiley & Sons Inc.; 1998.

Chirikos TN, Sear AM. Measuring Hospital Efficiency: A Comparison of Two Approaches. *Health Services Research*. 2000; 34(6):1389-1408.

Clarke SM, Griebsch JH, Simpson TW. Analysis of support vector regression for approximation of complex engineering analyses. *Journal of Mechanical Design*. 2005; 127:1077-1087.

Coelli TJ. Recent developments in frontier modelling and efficiency measurement. *Australian Journal of Agricultural Economics.* 1995; 3:219-245.

Collobert R, Bengio S. SVM Torch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*. 2001; 1(1):143-160.

Cooper WW, Seiford LW, Tone K. *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software.* Boston, MA: Kluwer Academic Publishers; 2000.

Cooper WW, Seiford LW, Zhu J. *Handbook of Data Envelopment Analysis.* Boston, MA: Kluwer Academic Publishers; 2004.

Costa A, Markellos RN. Evaluating public transport efficiency with neural network models. *Transportation Research C*. 1997; 5(5):301-312.

Cristianni N, Shawe-Taylor J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge, UK: Cambridge University Press; 2000.

Demiriz A, Bennett K, Breneman C, Embrechts M. Support vector machine regression in chemometrics. *Computer Science and Statistics*. 2001.

Dibike YB, Velickov S, Solomatine S. Support vector machines: Review and applications in civil engineering. *In AI Methods in Civil Engineering Applications*. O. Schleider and A. Zijderveld, Eds, Cottbus, Germany, pp. 45-58, 2000.

Durbha SS, King RL, Younan NH. Support vector machines regression for retrieval of leaf area index from multiangle imaging spectroradiometer. *Elsevier:Remote Sensing of Environment*. 2007; 107:348-361.

Emrouznejad A. *"Ali Emrouznejad's DEA HomePage"*. Warwick Business School, Coventry CV4 7AL, UK, 1995-2001.

Flessig AR, Kastens T, Terrell D. Evaluating the semi-nonparametric fourier, aim, and neural networks cost functions. *Economics Letters*. 2000; 68(3):235-244.

Greene WH. A gamma-distributed stochastic frontier model. *Journal of Econometrics*. 1990; 46(1/2):141-163.

Gunn SR. Support vector machines for classification and regression. *Technical Report,* School of Electronics and Computer Science, University of Southhampton, UK, 1998.

Hagen M, Demuth H, Beale M. *Neural Networks Design*. Boston, MA : PWS Publishing; 1996.

Haykin S. *Neural Networks – A Comprehensive Foundation, 2nd Edition*. New York, USA: Prentice Hall; 1998.

Judd KL. *Numerical Methods in Economics.* Cambridge, MA: MIT Press; 1998.

Kebede TA. Farm Household Technical Efficiency: A Stochastic Frontier Analysis. *Masters Thesis*. Department of Economics and Social Sciences, Agricultural University of Norway. Norway, 2001.

Meeusen W, Van den Broeck J. Efficiency estimation from Cobb-Douglas production functions with composed error. *International Economic Review*. 1977; 8:435-444.

Muller KR, Smola AJ, Ratsch G, Scholkopf B, Kohlmorgen J, Vapnik V. Predicting time series with support vector machines. *In Proceedings ICANN 1997*. Berlin, Germany: Springer LNCS 1327; 1997; pp. 999-1004.

Musicant DR, Feinberg A. Active set support vector regression. *IEEE Transactions on Neural Networks*. 2004; 15(2):268-275.

PCW Consulting. TCNZ Efficiency Study based on Stochastic Frontier Analysis (SFA). *An Efficiency Study of Telecom New Zealand*. New Zealand, 2001.

Pendharkar PC, Rodger JA. Technical efficiency-based selection of learning cases to improve forecasting accuracy of neural networks under monotonicity assumption. *Decision Support Systems*. 2003; 36(1):117-136.

Poggio T, Torre V, Koch C. Computational and regularization theory. *Nature*. 1985; 317:314-319.

Pozdnoukhov A, Kanevski M. Multi-scale support vector regression for hot spot detection and modeling. *Research Report No. 06-007*. Institute of Geomatics and Analysis of Risk, University of Lausanne, Lausanne, Switzerland, Nov. 17, 2006.

Ruggiero J. *Data Envelopment Analysis with Stochastic Data*. Department of Economic and Finance, University of Dayton, Dayton, OH, 2003.

Sarafidis V. An assessment of comparative efficiency measurement techniques. *Europe Economics: Occasional Paper 2*. London, UK, Oct. 2002.

Schmidt P. Frontier production functions. *Econometrics Reviews*. 1985; 4(2):289-328.

Smola AJ, Scholkopf B, Muller KR. The connection between regularization operators and support vector kernels. *Neural Networks*. 1998; 11(4):637-649.

Sewell M. *SVMdark Software*. Department of Computer Science, University College London, London, UK, 2005.
Available at: http://www.cs.ucl.ac.uk/staff/M.Sewell/svmdark/

Sueyoshi T. Estimation of stochastic frontier cost function using data envelopment analysis: an application to the AT&T divestiture. *Journal of the Operational Research Society*. 1991; 42(6):463-477.

Thanassoulis E. Alternative methods for measuring efficiency and an application of DEA in education. *PowerPoint Presentation*. Aston Business School, University of Aston, Birmingham, UK, 2007.

van Lint JWC, Hoogendoorn SP, van Zuylen HJ. Robust and adaptive travel time prediction with neural networks. *Proceedings of the 6th annual TRAIL Congress (Part 2)*. 2000.

Vapnik V. *The nature of Statistical Learning Theory.* New York, NY: Springer; 1995.

Wang S. Adaptive non-parametric efficiency frontier analysis: a neural –network-based model. *Computer and Operations Research*. 2003; 30:279-295.

Wu CH, Wei CC, Su DC, Chang MH, Ho JM. Travel time prediction with support vector regression. *Proceedings of the 2003 IEEE Conference on Intelligent Transportation Systems*. Shanghai, China, 2003.

Wu D, Yang Z, Liang L. Using DEA-neural network approach to evaluate branch efficiency of a large Canadian bank. *Expert Systems with Applications*. 2006; 31(1):108-115.

APPENDIX A:
Raw Original 49(forty-nine) Single-Input, Single-Output Data Pairs and Original 49(forty-nine) Scaled Single-Input, Single-Output Data Pairs

Appendix A1:Original Raw 49(forty-nine) Single-Input, Single-Output Data Pairs (Source:Charnes, Cooper and Rhodes, 1981)

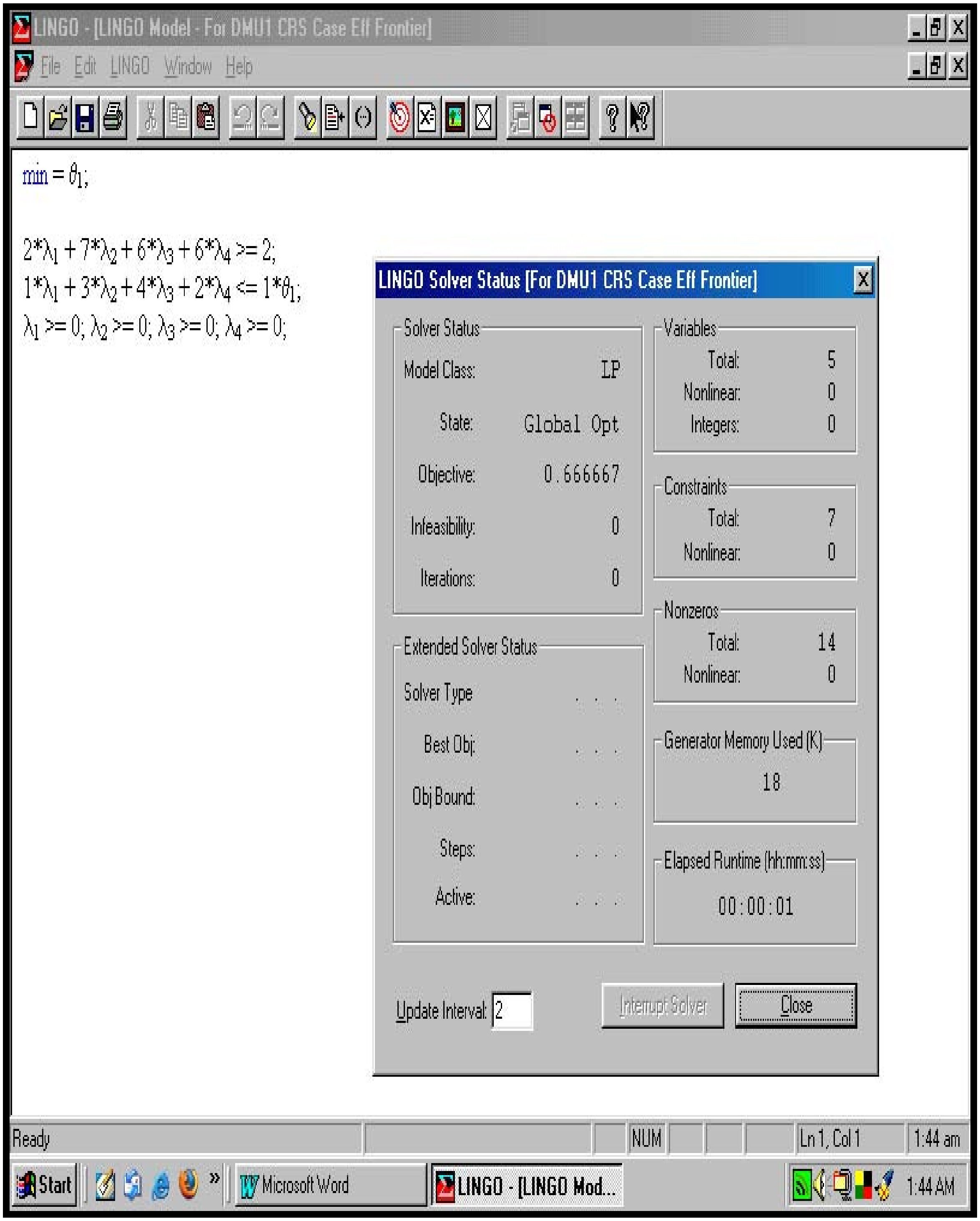
DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)
DMU1	86.13	38.16	**DMU26**	41.40	31.16
DMU2	29.26	26.02	**DMU27**	27.20	25.03
DMU3	43.12	28.51	**DMU28**	23.92	18.30
DMU4	24.96	16.19	**DMU29**	10.62	6.16
DMU5	11.62	5.37	**DMU30**	12.48	15.68
DMU6	11.88	12.84	**DMU31**	19.32	14.42
DMU7	32.64	17.82	**DMU32**	6.30	4.99
DMU8	20.79	33.16	**DMU33**	46.62	39.10
DMU9	34.40	26.29	**DMU34**	38.95	31.05
DMU10	61.74	35.20	**DMU35**	61.60	39.22
DMU11	52.92	30.29	**DMU36**	31.08	13.91
DMU12	36.00	25.35	**DMU37**	19.35	15.30
DMU13	39.20	26.54	**DMU38**	11.20	7.22
DMU14	14.60	7.47	**DMU39**	34.40	29.80
DMU15	4.29	14.33	**DMU40**	35.55	17.15
DMU16	27.25	38.19	**DMU41**	30.53	25.30
DMU17	22.63	12.07	**DMU42**	25.44	17.56
DMU18	28.00	20.44	**DMU43**	26.66	27.54
DMU19	53.56	36.54	**DMU44**	39.79	63.11
DMU20	25.42	23.34	**DMU45**	8.32	8.85
DMU21	31.57	27.44	**DMU46**	59.78	34.61
DMU22	16.34	16.52	**DMU47**	39.22	28.42
DMU23	44.28	38.97	**DMU48**	3.24	9.02
DMU24	19.74	16.54	**DMU49**	7.14	15.82
DMU25	24.40	22.43			

Appendix A2:Original 49(forty-nine) Scaled Single-Input, Single-Output Data Pairs

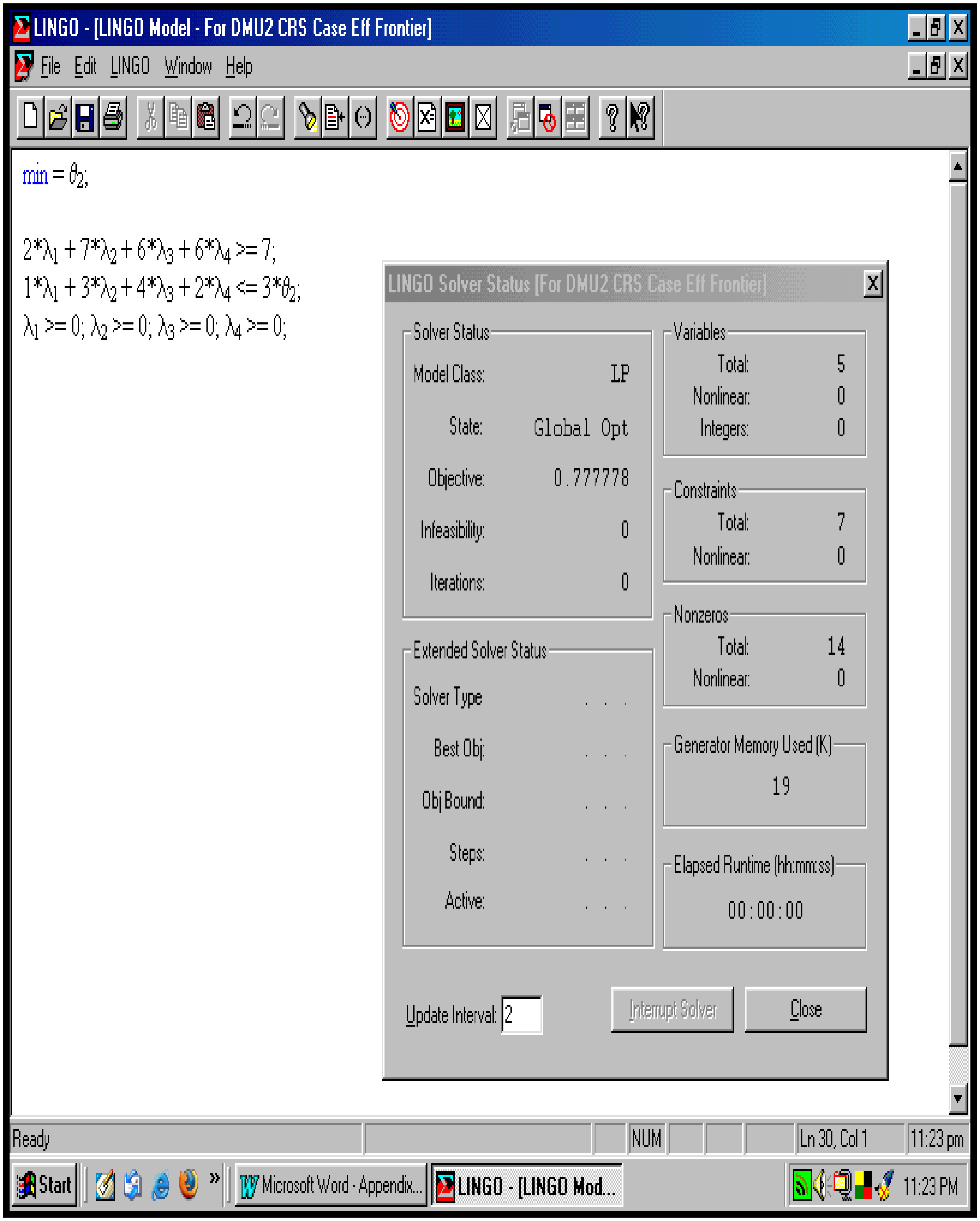
DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)	DMU	Educational Level Of Mother (X-Input values)	Coopersmith Scores (Y-Output values)
DMU1	1.000	0.542	**DMU26**	0.460	0.470
DMU2	0.314	0.417	**DMU27**	0.289	0.407
DMU3	0.481	0.443	**DMU28**	0.249	0.337
DMU4	0.262	0.316	**DMU29**	0.089	0.212
DMU5	0.101	0.204	**DMU30**	0.111	0.310
DMU6	0.104	0.281	**DMU31**	0.194	0.297
DMU7	0.355	0.332	**DMU32**	0.037	0.200
DMU8	0.212	0.491	**DMU33**	0.523	0.552
DMU9	0.376	0.420	**DMU34**	0.431	0.469
DMU10	0.706	0.512	**DMU35**	0.704	0.553
DMU11	0.599	0.461	**DMU36**	0.336	0.292
DMU12	0.395	0.410	**DMU37**	0.194	0.306
DMU13	0.434	0.422	**DMU38**	0.096	0.223
DMU14	0.137	0.226	**DMU39**	0.376	0.456
DMU15	0.013	0.296	**DMU40**	0.390	0.326
DMU16	0.290	0.543	**DMU41**	0.329	0.410
DMU17	0.234	0.273	**DMU42**	0.268	0.330
DMU18	0.299	0.359	**DMU43**	0.283	0.433
DMU19	0.607	0.526	**DMU44**	0.441	0.800
DMU20	0.268	0.389	**DMU45**	0.061	0.240
DMU21	0.342	0.432	**DMU46**	0.682	0.506
DMU22	0.158	0.319	**DMU47**	0.434	0.442
DMU23	0.495	0.551	**DMU48**	0.000	0.242
DMU24	0.199	0.319	**DMU49**	0.047	0.312
DMU25	0.255	0.380			

APPENDIX B:
Lingo 10.0 Codes and Printouts for Illustrative Example for CRS Case for Efficient Frontier

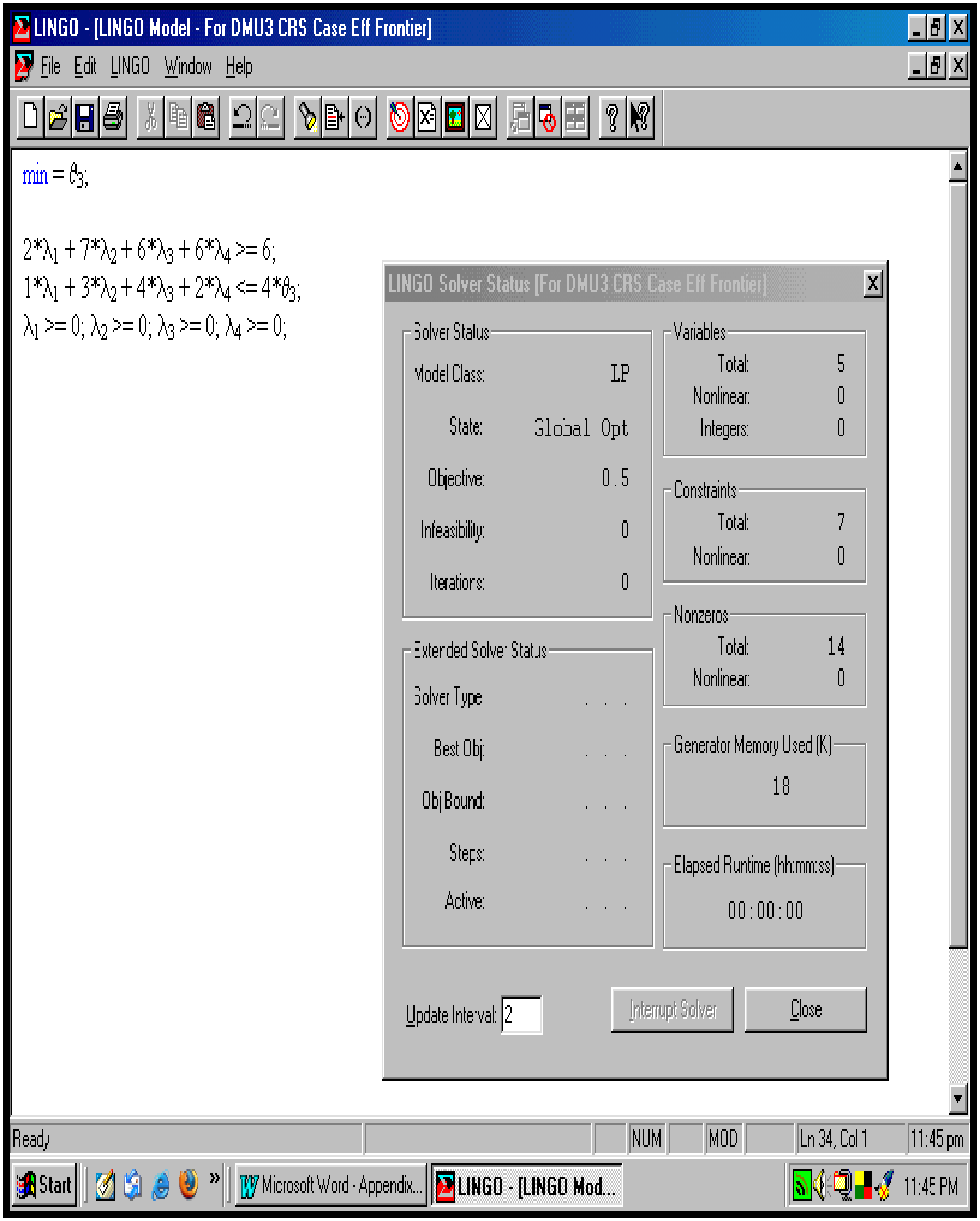
Appendix B1:Lingo 10.0 Code and Printout - DMU1 CRS Case Efficient Frontier



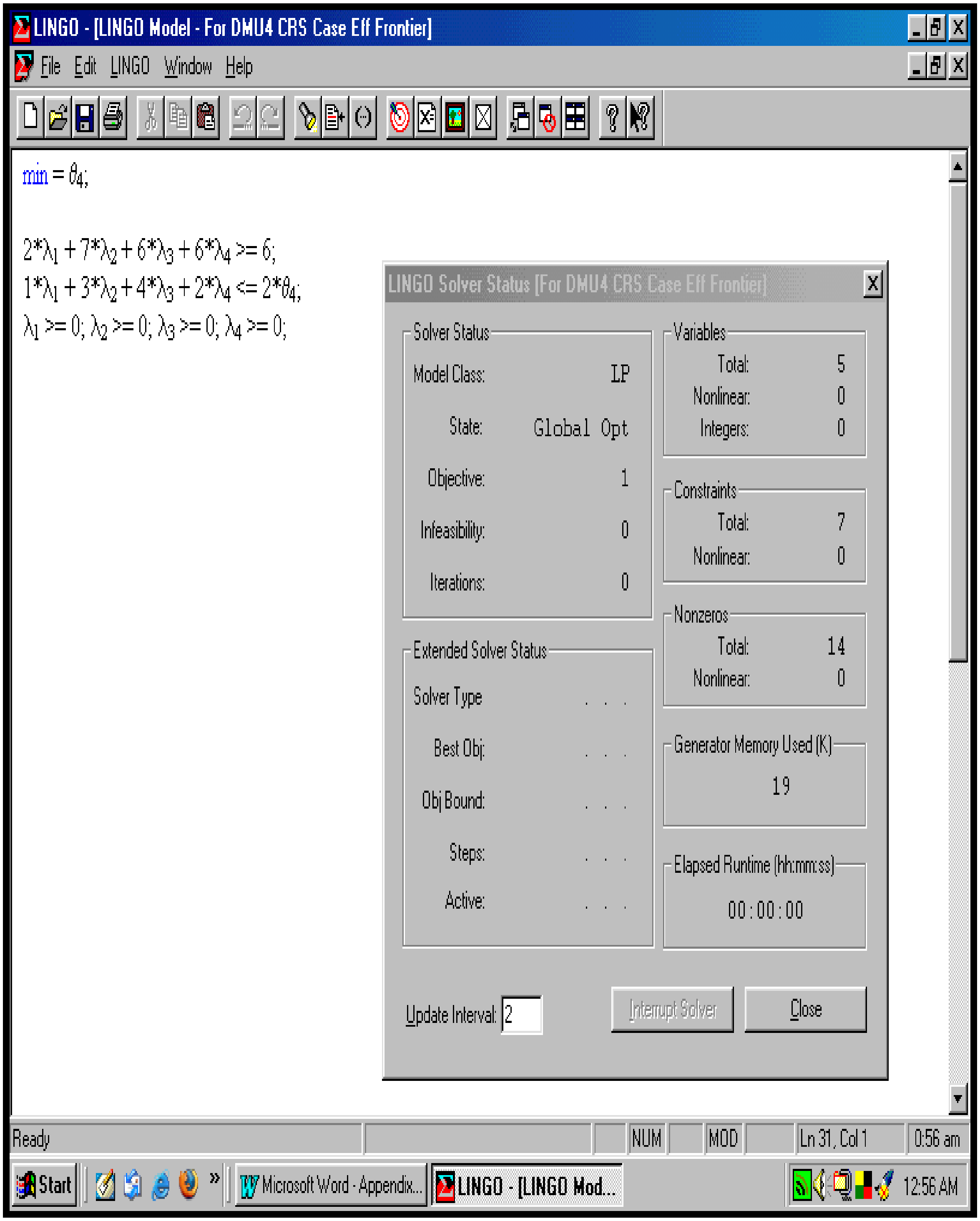
Appendix B2:Lingo 10.0 Code and Printout - DMU2 CRS Case Efficient Frontier



Appendix B3:Lingo 10.0 Code and Printout – DMU3 CRS Case Efficient Frontier

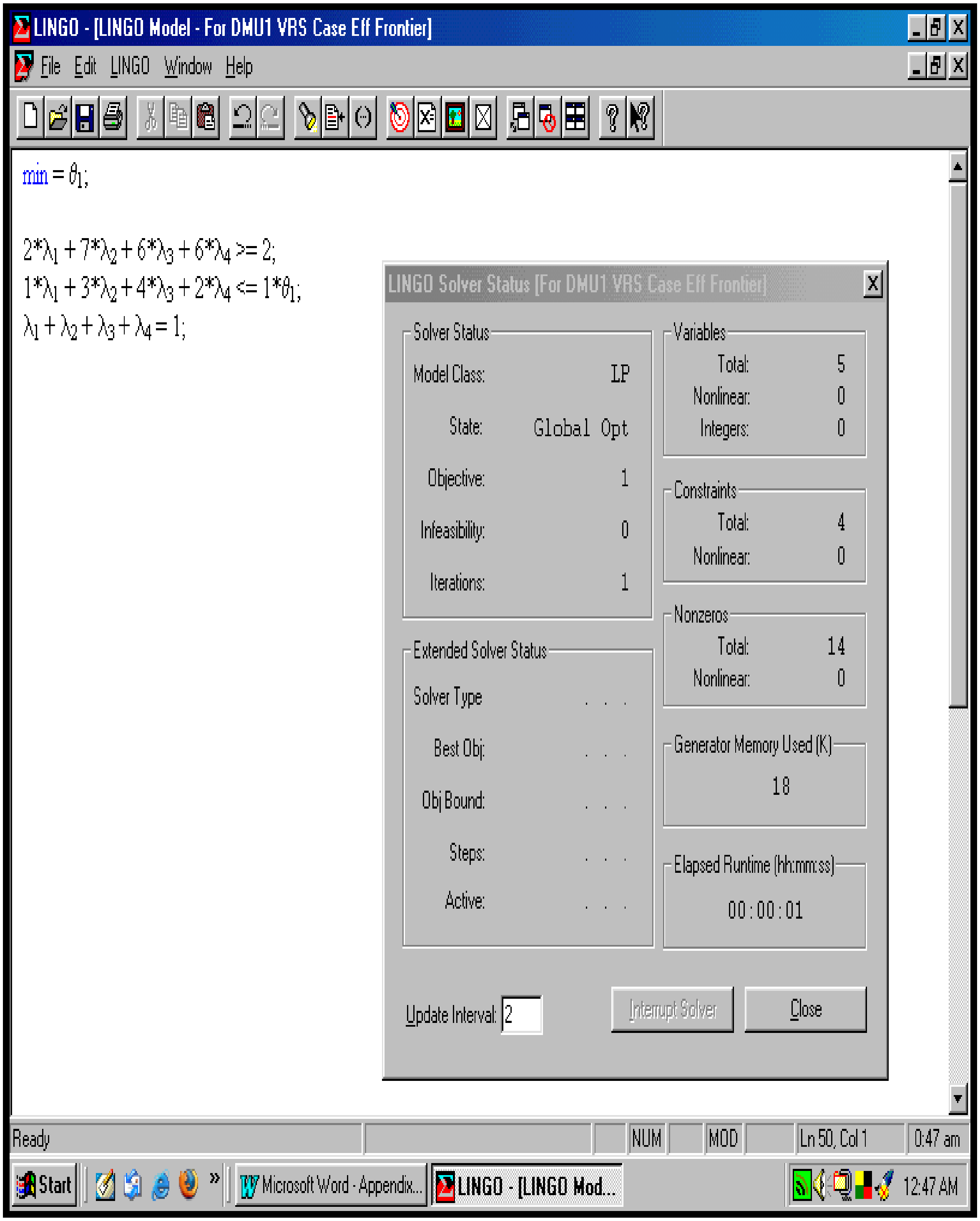


Appendix B4:Lingo 10.0 Code and Printout – DMU4 CRS Case Efficient Frontier

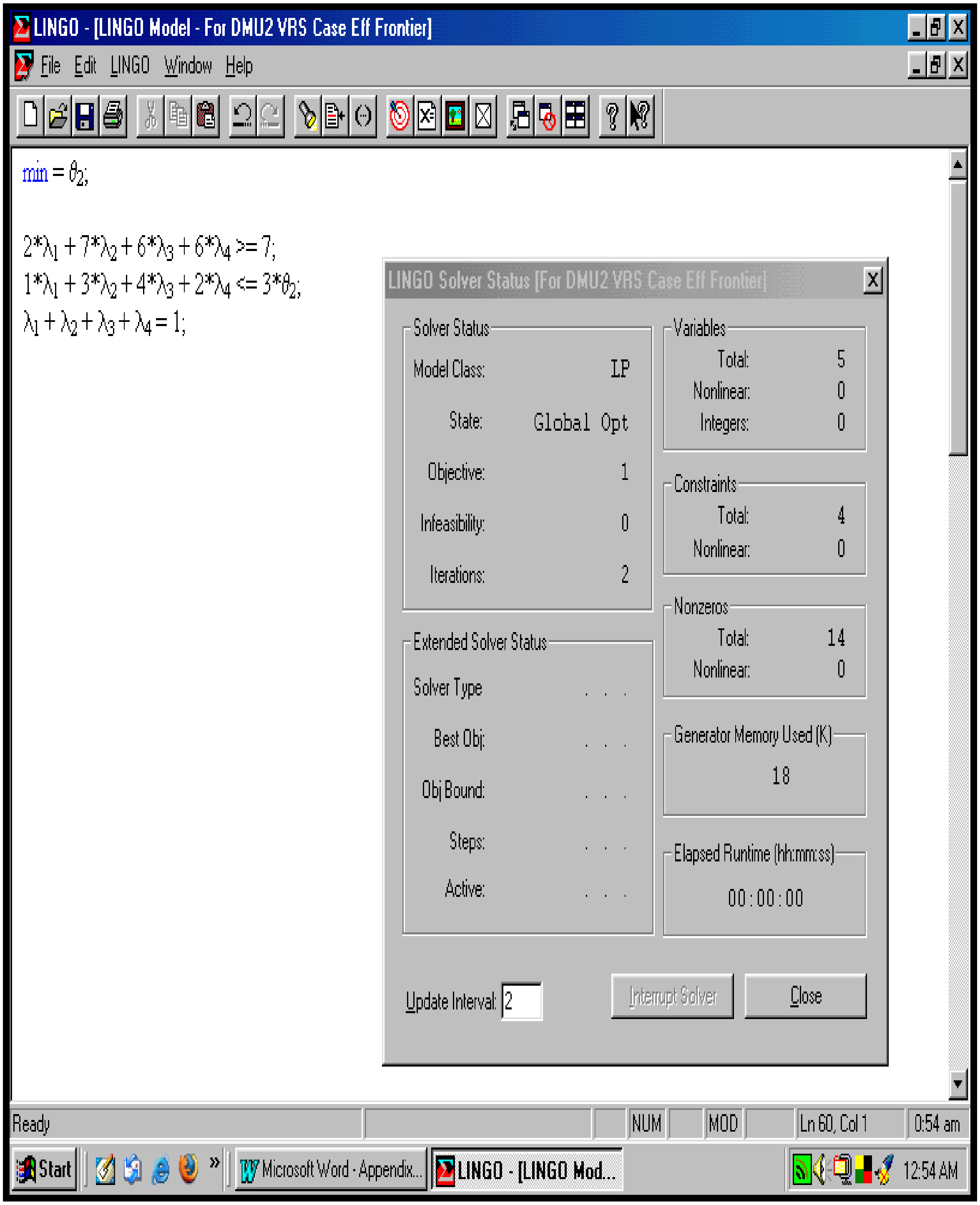


APPENDIX C:
Lingo 10.0 Codes and Printouts for Illustrative Example for VRS Case for Efficient Frontier

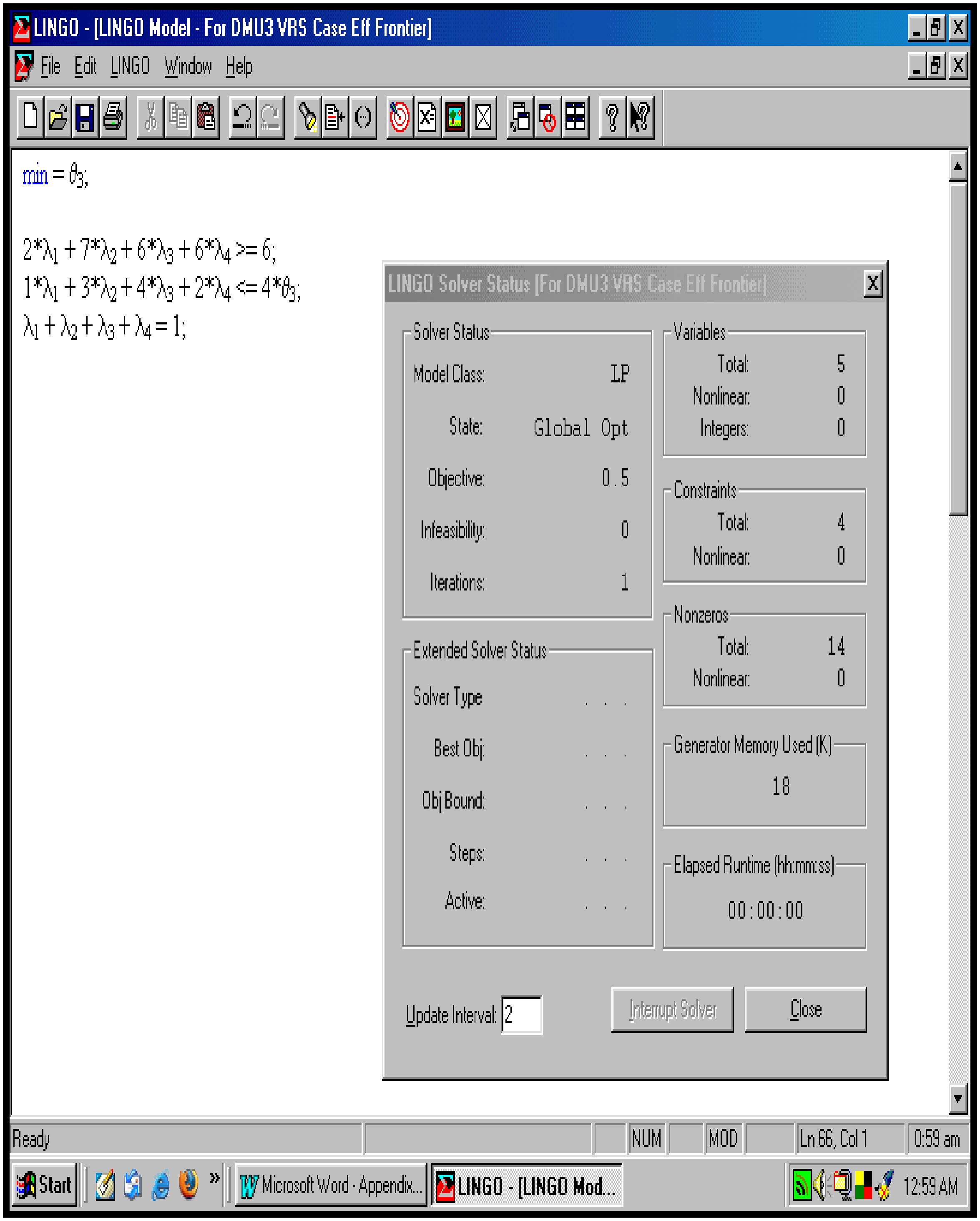
Appendix C1:Lingo 10.0 Code and Printout – DMU1 VRS Case Efficient Frontier



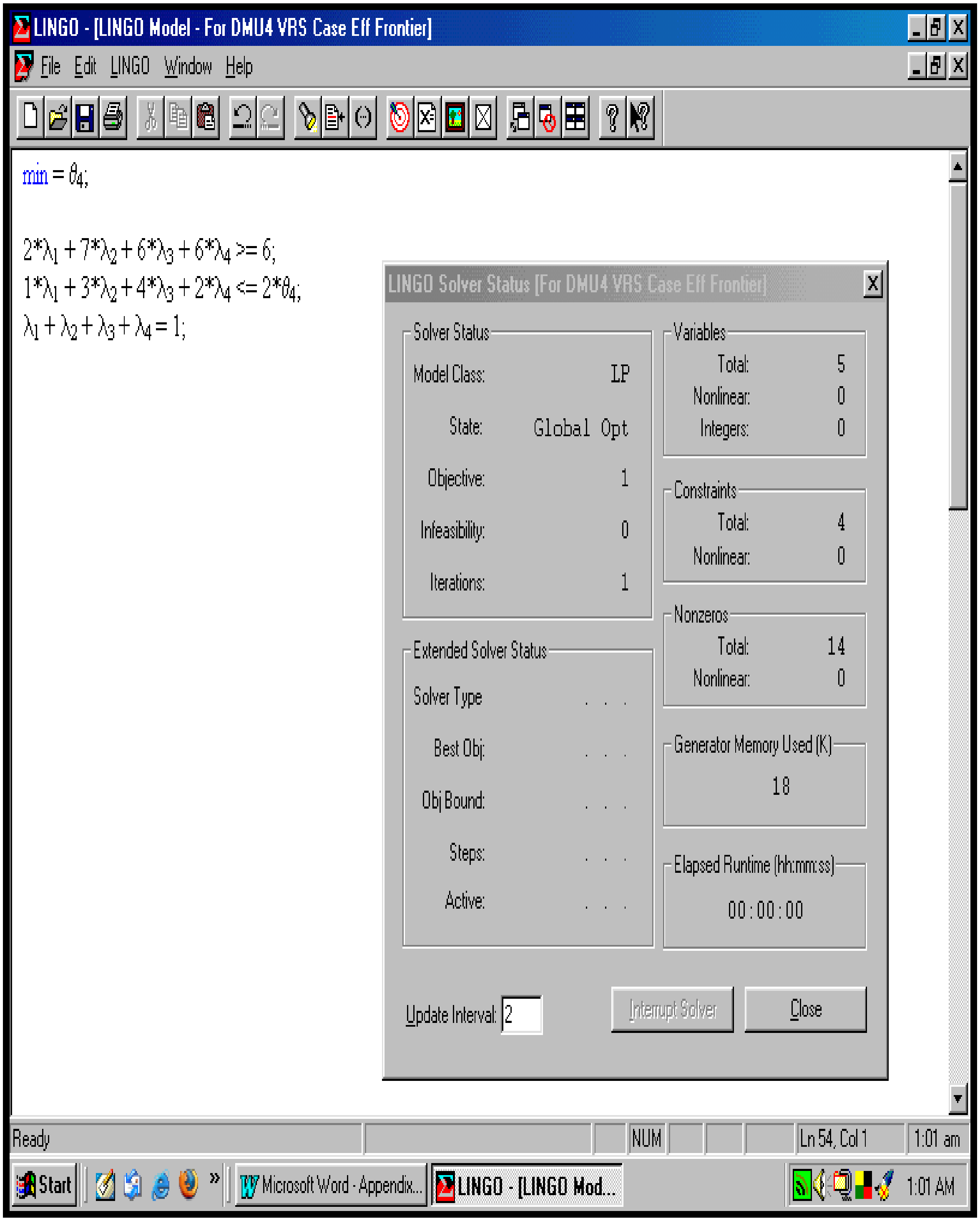
Appendix C2:Lingo 10.0 Code and Printout – DMU2 VRS Case Efficient Frontier



Appendix C3:Lingo 10.0 Code and Printout – DMU3 VRS Case Efficient Frontier

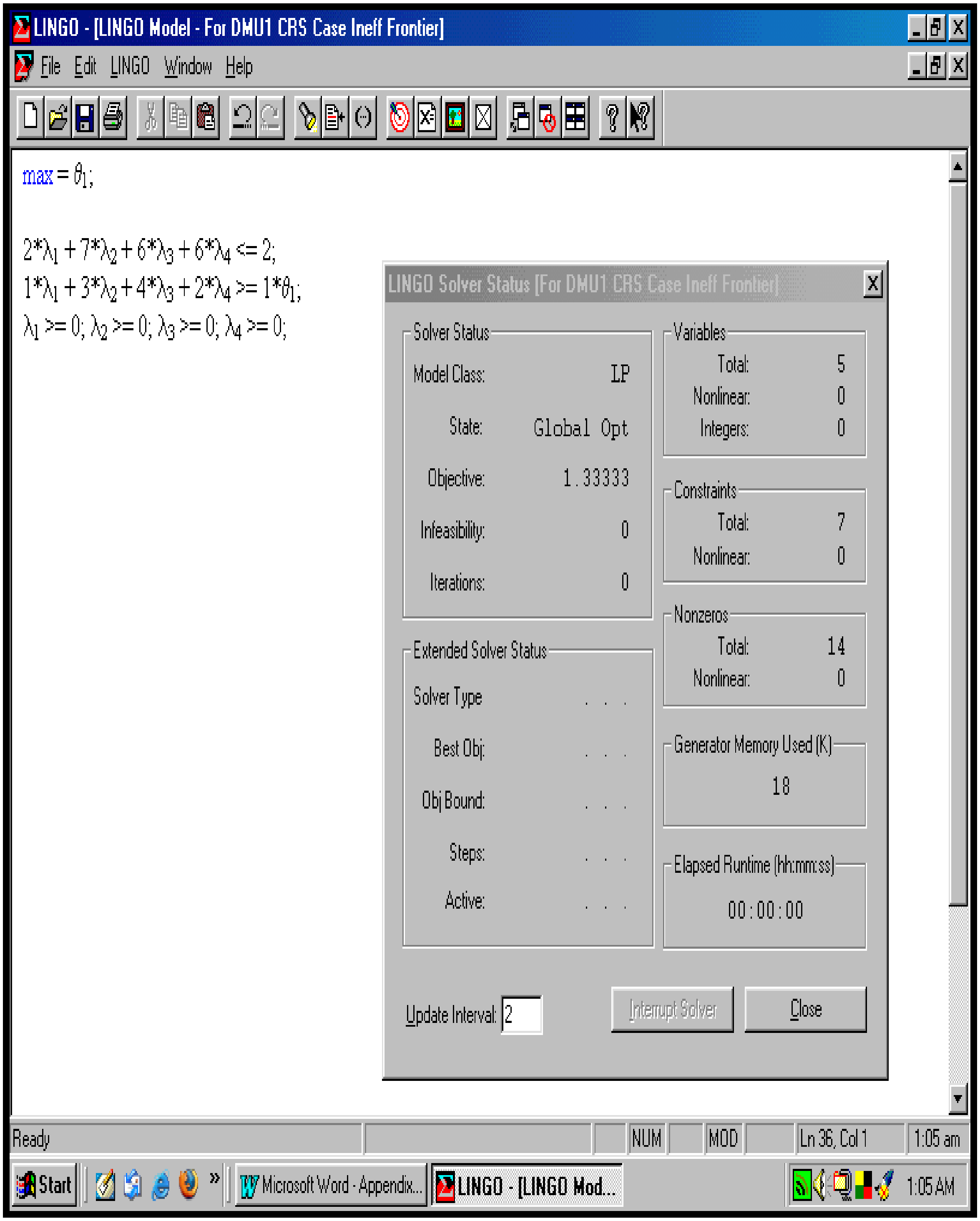


Appendix C4:Lingo 10.0 Code and Printout – DMU4 VRS Case Efficient Frontier

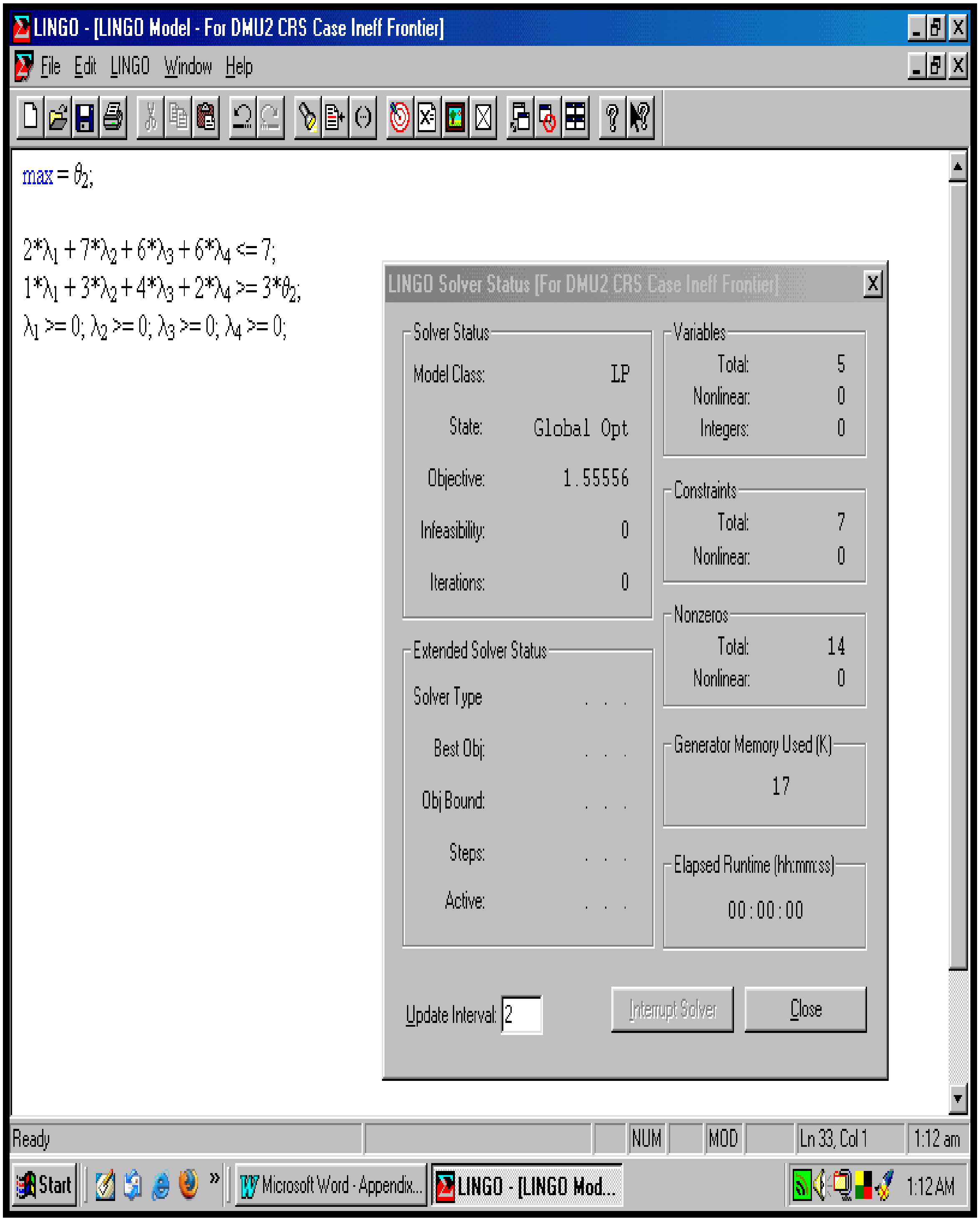


APPENDIX D:
Lingo 10.0 Codes and Printouts for Illustrative Example for CRS Case for Inefficient Frontier

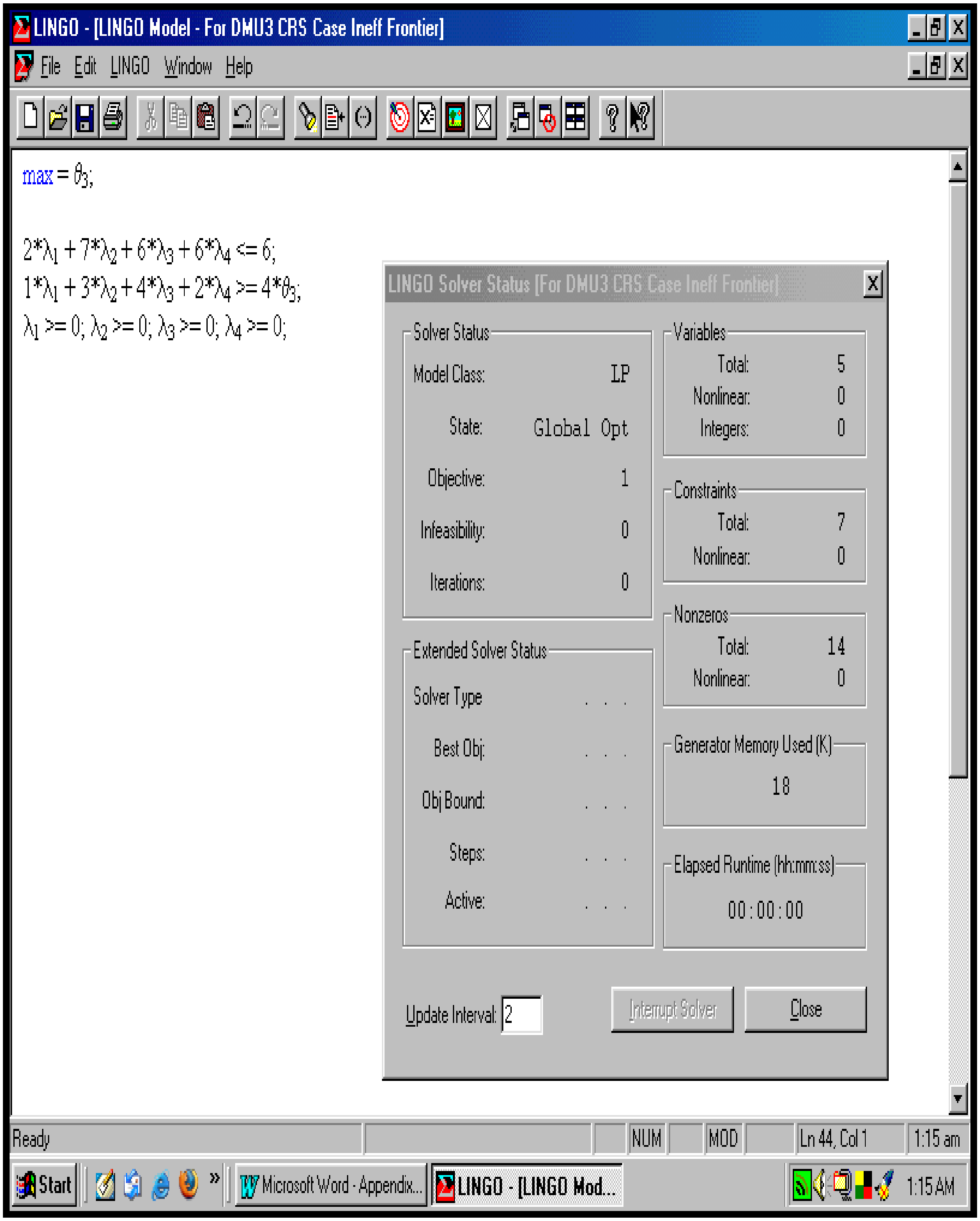
Appendix D1:Lingo 10.0 Code and Printout - DMU1 CRS Case Inefficient Frontier



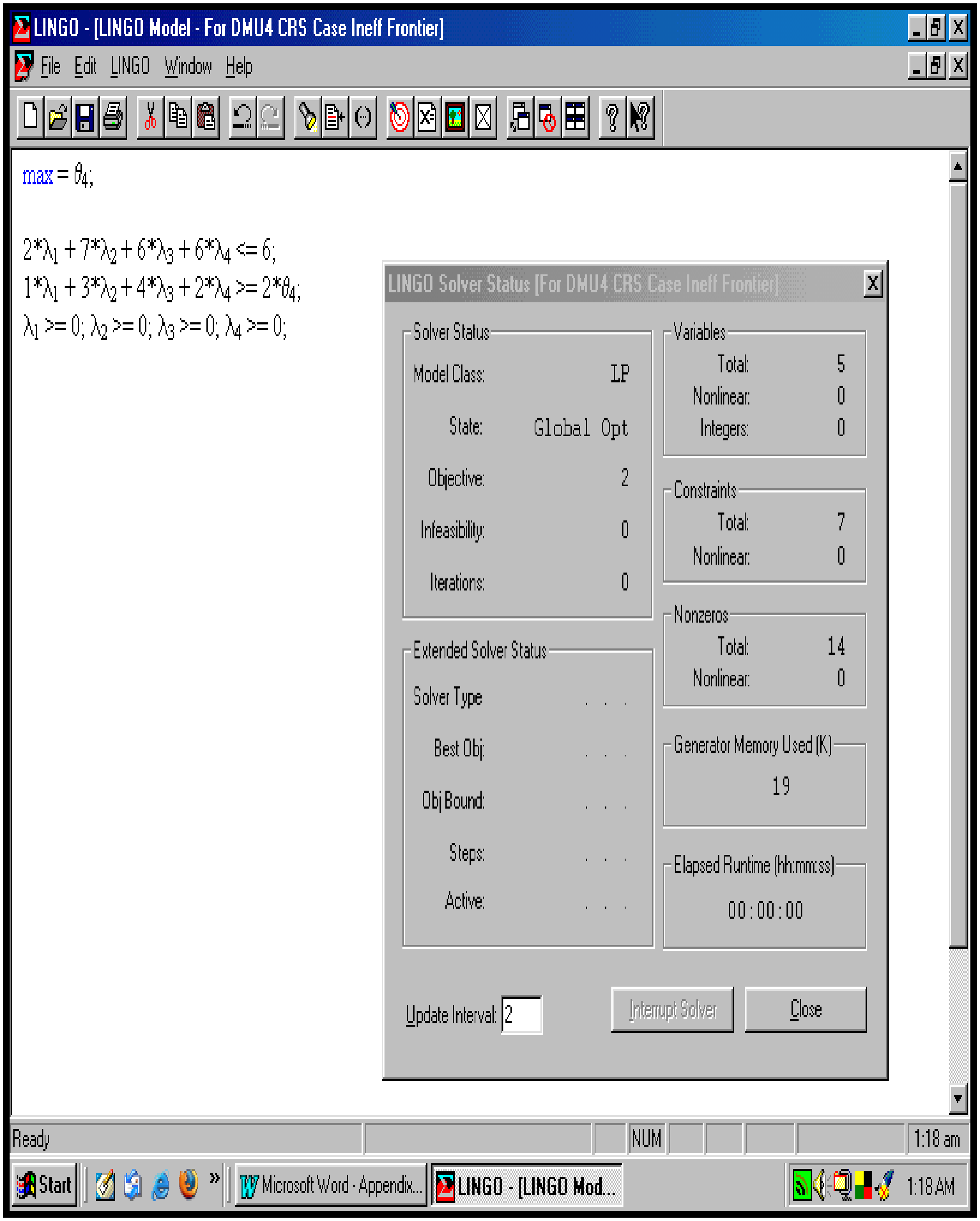
Appendix D2:Lingo 10.0 Code and Printout – DMU2 CRS Case Inefficient Frontier



Appendix D3:Lingo 10.0 Code and Printout – DMU3 CRS Case Inefficient Frontier

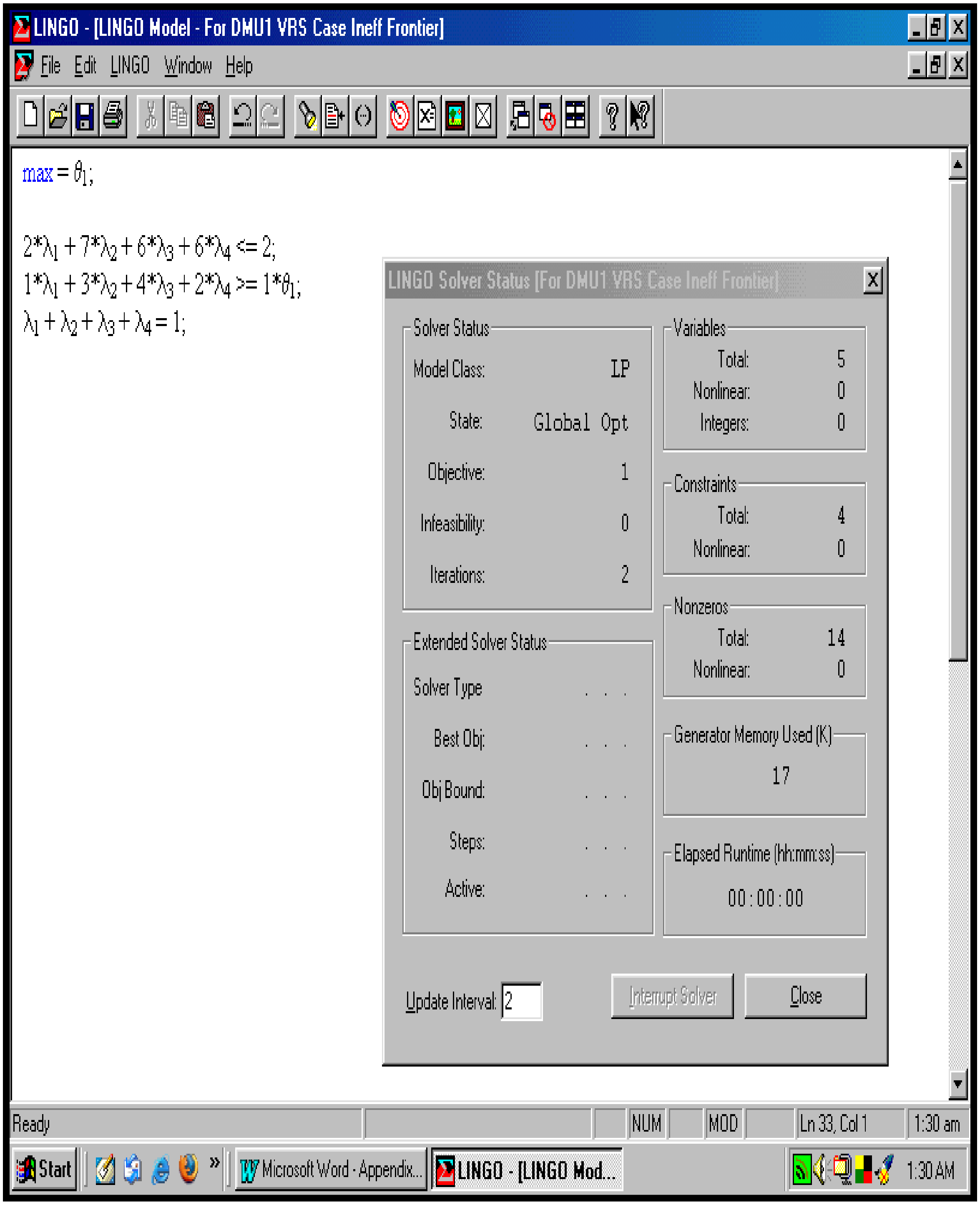


Appendix D4:Lingo 10.0 Code and Printout – DMU4 CRS Case Inefficient Frontier

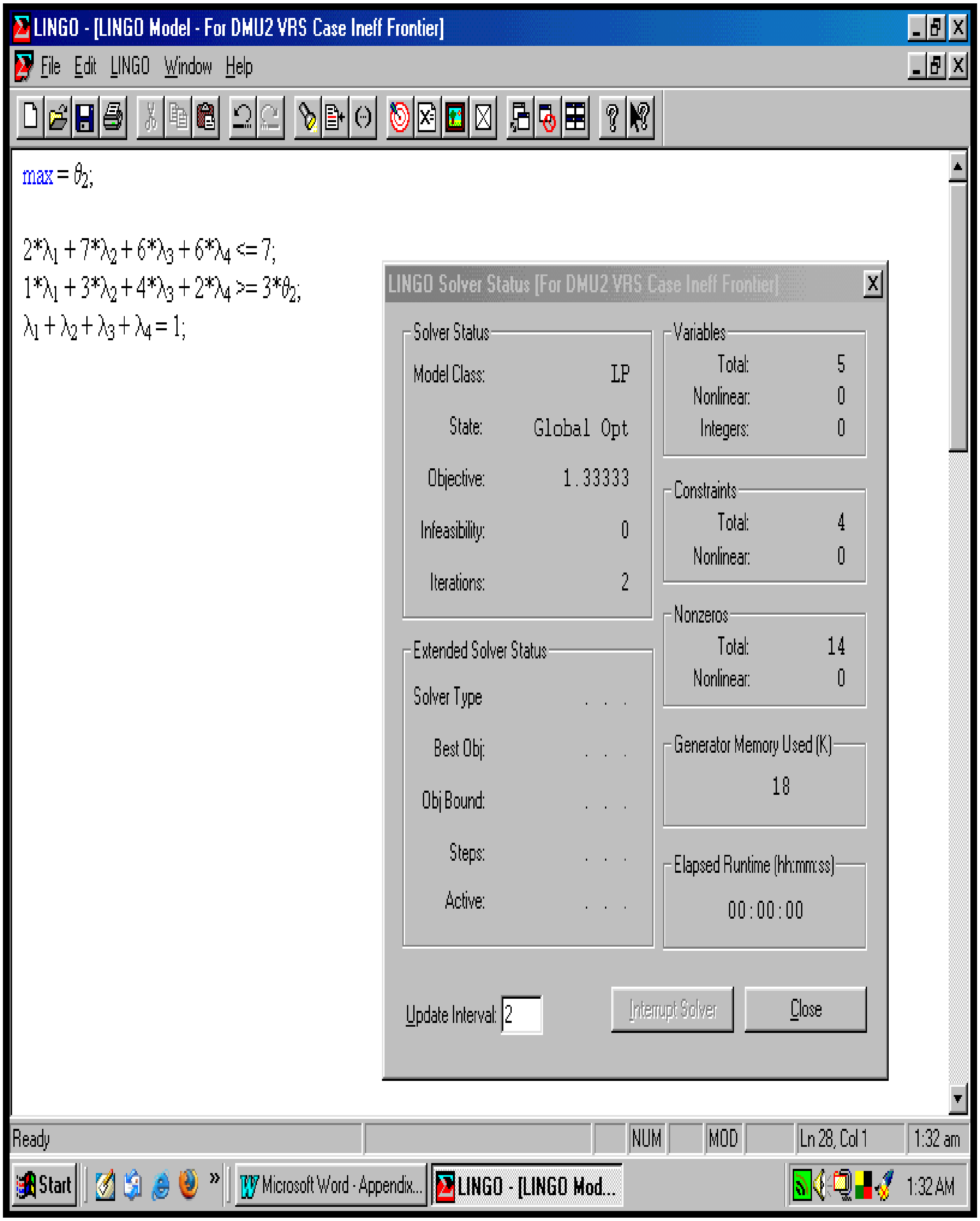


<u>APPENDIX E:</u>
Lingo 10.0 Codes and Printouts for Illustrative Example for VRS Case for Inefficient Frontier

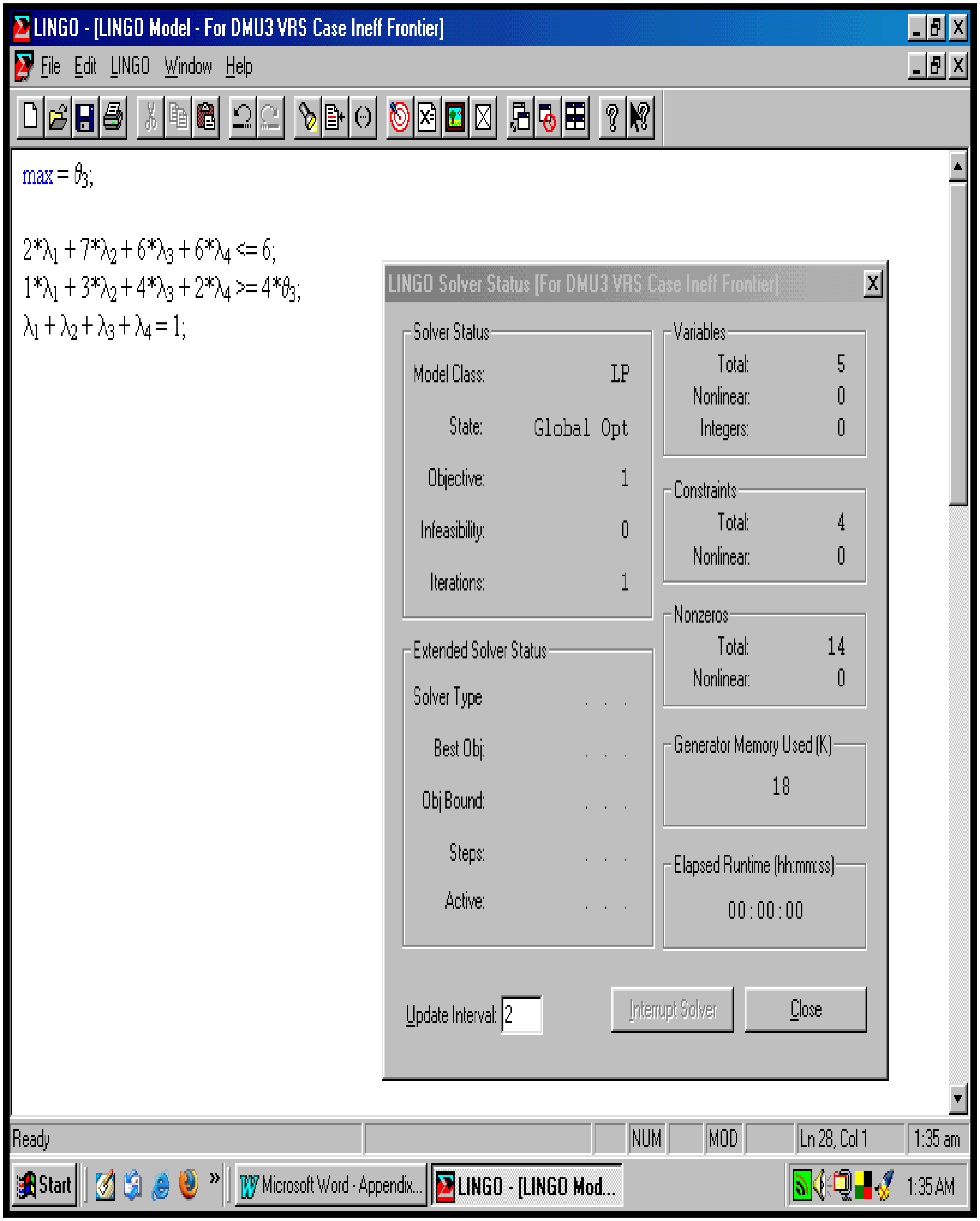
Appendix E1:Lingo 10.0 Code and Printout – DMU1 VRS Case Inefficient Frontier



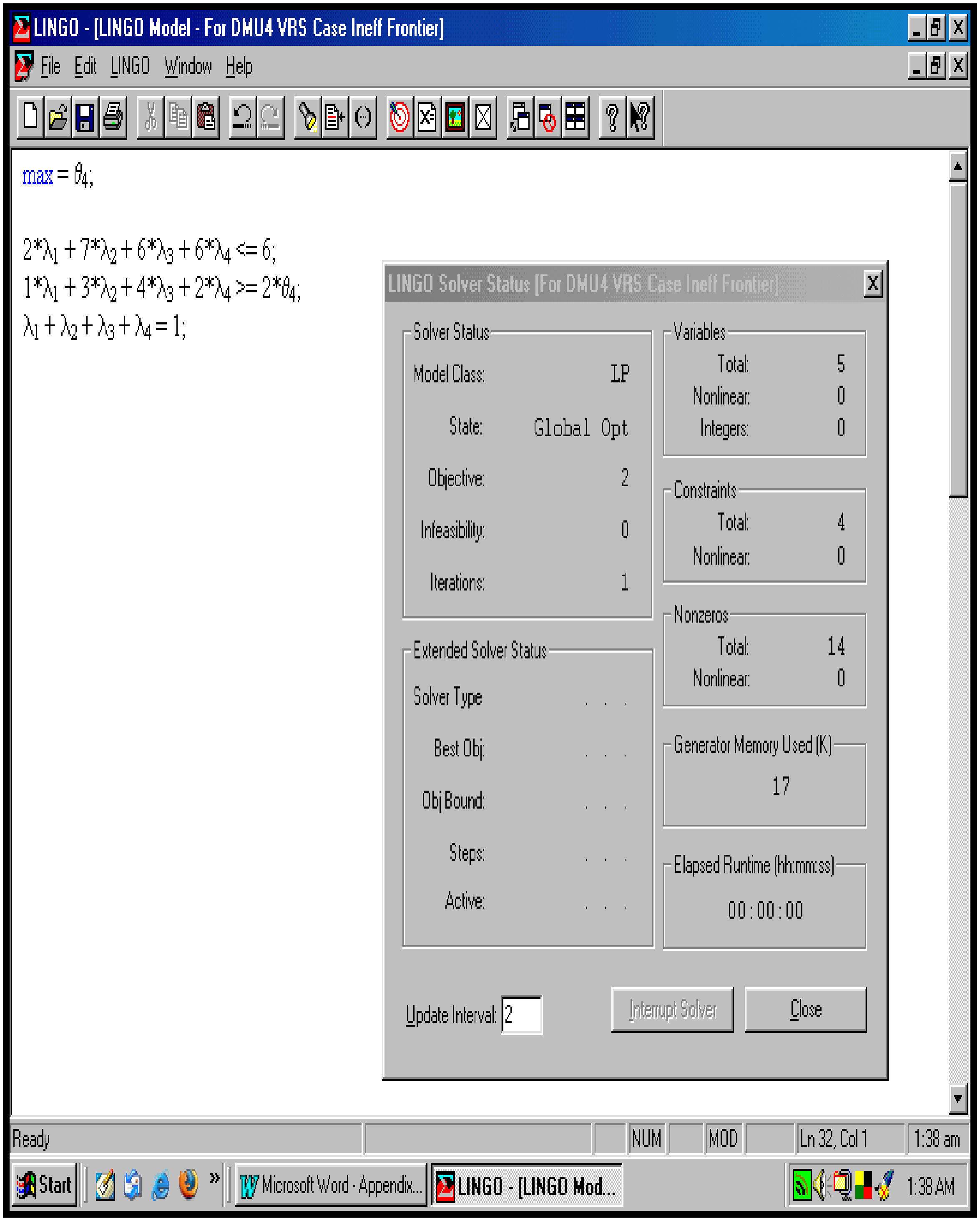
Appendix E2:Lingo 10.0 Code and Printout – DMU2 VRS Case Inefficient Frontier



Appendix E3:Lingo 10.0 Code and Printout – DMU3 VRS Case Inefficient Frontier



Appendix E4:Lingo 10.0 Code and Printout – DMU4 VRS Case Inefficient Frontier



<u>APPENDIX F:</u>
VRS Efficiency Scores for 49(forty-nine) DMUs for Efficient and Inefficient Frontiers

Appendix F1: VRS Efficiency Scores for 49(forty-nine) DMUs – Efficient Frontier

DMU	Efficiency Score	DMU	Efficiency Score
DMU1	0.25	**DMU26**	0.40
DMU2	0.44	**DMU27**	0.44
DMU3	0.34	**DMU28**	0.30
DMU4	0.23	**DMU29**	0.31
DMU5	0.28	**DMU30**	0.42
DMU6	0.34	**DMU31**	0.23
DMU7	0.21	**DMU32**	0.51
DMU8	0.87	**DMU33**	0.48
DMU9	0.38	**DMU34**	0.42
DMU10	0.32	**DMU35**	0.36
DMU11	0.30	**DMU36**	0.14
DMU12	0.34	**DMU37**	0.26
DMU13	0.34	**DMU38**	0.29
DMU14	0.22	**DMU39**	0.45
DMU15	**1.00**	**DMU40**	0.18
DMU16	0.79	**DMU41**	0.40
DMU17	0.17	**DMU42**	0.26
DMU18	0.31	**DMU43**	0.52
DMU19	0.38	**DMU44**	**1.00**
DMU20	0.43	**DMU45**	0.39
DMU21	0.44	**DMU46**	0.32
DMU22	0.36	**DMU47**	0.37
DMU23	0.50	**DMU48**	**1.00**
DMU24	0.30	**DMU49**	0.75
DMU25	0.42		

Appendix F2: VRS Efficiency Scores for 49(forty-nine) DMUs – Inefficient Frontier

DMU	Efficiency Score	DMU	Efficiency Score
DMU1	**1.00**	**DMU26**	1.70
DMU2	2.00	**DMU27**	2.07
DMU3	1.49	**DMU28**	1.72
DMU4	1.45	**DMU29**	1.26
DMU5	**1.00**	**DMU30**	2.81
DMU6	2.41	**DMU31**	1.67
DMU7	1.22	**DMU32**	**1.00**
DMU8	3.60	**DMU33**	1.85
DMU9	1.72	**DMU34**	1.80
DMU10	1.29	**DMU35**	1.40
DMU11	1.29	**DMU36**	**1.00**
DMU12	1.58	**DMU37**	1.77
DMU13	1.52	**DMU38**	1.41
DMU14	1.12	**DMU39**	1.95
DMU15	7.47	**DMU40**	1.08
DMU16	3.16	**DMU41**	1.86
DMU17	1.19	**DMU42**	1.55
DMU18	1.64	**DMU43**	2.33
DMU19	1.54	**DMU44**	2.16
DMU20	2.06	**DMU45**	2.35
DMU21	1.96	**DMU46**	1.31
DMU22	2.26	**DMU47**	2.52
DMU23	1.95	**DMU48**	6.15
DMU24	1.88	**DMU49**	4.96
DMU25	2.07		

APPENDIX G:
Actual and Predicted Y-Output Values for NN and SVM Efficient Frontier Function and Inefficient Frontier Function

Appendix G1:Actual and Predicted Output Values for 49(forty-nine) DMUs for NN Efficient Frontier Function $F_{(Eff\text{-}NN)}(X)$

DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)	DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)
DMU1	0.542	0.800	**DMU26**	0.470	0.789
DMU2	0.417	0.655	**DMU27**	0.407	0.623
DMU3	0.443	0.794	**DMU28**	0.337	0.573
DMU4	0.316	0.589	**DMU29**	0.212	0.386
DMU5	0.204	0.401	**DMU30**	0.310	0.413
DMU6	0.281	0.404	**DMU31**	0.297	0.509
DMU7	0.332	0.707	**DMU32**	0.200	0.318
DMU8	0.491	0.529	**DMU33**	0.552	0.798
DMU9	0.420	0.731	**DMU34**	0.469	0.776
DMU10	0.512	0.800	**DMU35**	0.553	0.800
DMU11	0.461	0.800	**DMU36**	0.292	0.684
DMU12	0.410	0.750	**DMU37**	0.306	0.509
DMU13	0.422	0.778	**DMU38**	0.223	0.395
DMU14	0.226	0.444	**DMU39**	0.456	0.731
DMU15	0.296	0.284	**DMU40**	0.326	0.745
DMU16	0.543	0.624	**DMU41**	0.410	0.675
DMU17	0.273	0.555	**DMU42**	0.330	0.596
DMU18	0.359	0.636	**DMU43**	0.433	0.615
DMU19	0.526	0.800	**DMU44**	0.800	0.781
DMU20	0.389	0.596	**DMU45**	0.240	0.350
DMU21	0.432	0.691	**DMU46**	0.506	0.800
DMU22	0.319	0.468	**DMU47**	0.442	0.778
DMU23	0.551	0.796	**DMU48**	0.242	0.273
DMU24	0.319	0.514	**DMU49**	0.312	0.331
DMU25	0.380	0.580			

Appendix G2:Actual and Predicted Output Values for 49(forty-nine) DMUs for SVM Efficient Frontier Function $F_{(Eff\text{-}SVM)}(X)$

DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)	DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)
DMU1	0.542	0.465	**DMU26**	0.470	0.815
DMU2	0.417	0.652	**DMU27**	0.407	0.623
DMU3	0.443	0.823	**DMU28**	0.337	0.576
DMU4	0.316	0.592	**DMU29**	0.212	0.388
DMU5	0.204	0.401	**DMU30**	0.310	0.411
DMU6	0.281	0.404	**DMU31**	0.297	0.508
DMU7	0.332	0.702	**DMU32**	0.200	0.323
DMU8	0.491	0.531	**DMU33**	0.552	0.811
DMU9	0.420	0.729	**DMU34**	0.469	0.793
DMU10	0.512	0.554	**DMU35**	0.553	0.556
DMU11	0.461	0.715	**DMU36**	0.292	0.678
DMU12	0.410	0.753	**DMU37**	0.306	0.508
DMU13	0.422	0.796	**DMU38**	0.223	0.395
DMU14	0.226	0.440	**DMU39**	0.456	0.729
DMU15	0.296	0.284	**DMU40**	0.326	0.746
DMU16	0.543	0.624	**DMU41**	0.410	0.670
DMU17	0.273	0.558	**DMU42**	0.330	0.599
DMU18	0.359	0.635	**DMU43**	0.433	0.616
DMU19	0.526	0.702	**DMU44**	0.800	0.802
DMU20	0.389	0.599	**DMU45**	0.240	0.355
DMU21	0.432	0.686	**DMU46**	0.506	0.584
DMU22	0.319	0.464	**DMU47**	0.442	0.796
DMU23	0.551	0.823	**DMU48**	0.242	0.270
DMU24	0.319	0.514	**DMU49**	0.312	0.337
DMU25	0.380	0.583			

Appendix G3:Actual and Predicted Output Values for 49(forty-nine) DMUs for NN Inefficient Frontier Function $F_{(Ineff\text{-}NN)}(X)$

DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)	DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)
DMU1	0.542	0.541	**DMU26**	0.470	0.338
DMU2	0.417	0.284	**DMU27**	0.407	0.275
DMU3	0.443	0.346	**DMU28**	0.337	0.261
DMU4	0.316	0.266	**DMU29**	0.212	0.208
DMU5	0.204	0.212	**DMU30**	0.310	0.215
DMU6	0.281	0.213	**DMU31**	0.297	0.242
DMU7	0.332	0.299	**DMU32**	0.200	0.192
DMU8	0.491	0.248	**DMU33**	0.552	0.362
DMU9	0.420	0.307	**DMU34**	0.469	0.327
DMU10	0.512	0.432	**DMU35**	0.553	0.431
DMU11	0.461	0.391	**DMU36**	0.292	0.292
DMU12	0.410	0.314	**DMU37**	0.306	0.242
DMU13	0.422	0.328	**DMU38**	0.223	0.210
DMU14	0.226	0.223	**DMU39**	0.456	0.307
DMU15	0.296	0.184	**DMU40**	0.326	0.312
DMU16	0.543	0.276	**DMU41**	0.410	0.290
DMU17	0.273	0.256	**DMU42**	0.330	0.268
DMU18	0.359	0.279	**DMU43**	0.433	0.273
DMU19	0.526	0.394	**DMU44**	0.800	0.331
DMU20	0.389	0.268	**DMU45**	0.240	0.199
DMU21	0.432	0.294	**DMU46**	0.506	0.422
DMU22	0.319	0.230	**DMU47**	0.442	0.328
DMU23	0.551	0.351	**DMU48**	0.242	0.182
DMU24	0.319	0.244	**DMU49**	0.312	0.195
DMU25	0.380	0.263			

Appendix G4:Actual and Predicted Output Values for 49(forty-nine) DMUs for SVM Inefficient Frontier Function $F_{(Ineff\text{-}SVM)}(X)$

DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)	DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)
DMU1	0.542	0.543	**DMU26**	0.470	0.340
DMU2	0.417	0.283	**DMU27**	0.407	0.273
DMU3	0.443	0.348	**DMU28**	0.337	0.257
DMU4	0.316	0.262	**DMU29**	0.212	0.205
DMU5	0.204	0.208	**DMU30**	0.310	0.210
DMU6	0.281	0.209	**DMU31**	0.297	0.236
DMU7	0.332	0.300	**DMU32**	0.200	0.197
DMU8	0.491	0.243	**DMU33**	0.552	0.363
DMU9	0.420	0.308	**DMU34**	0.469	0.329
DMU10	0.512	0.432	**DMU35**	0.553	0.431
DMU11	0.461	0.391	**DMU36**	0.292	0.292
DMU12	0.410	0.316	**DMU37**	0.306	0.236
DMU13	0.422	0.331	**DMU38**	0.223	0.207
DMU14	0.226	0.218	**DMU39**	0.456	0.308
DMU15	0.296	0.196	**DMU40**	0.326	0.314
DMU16	0.543	0.274	**DMU41**	0.410	0.290
DMU17	0.273	0.251	**DMU42**	0.330	0.265
DMU18	0.359	0.277	**DMU43**	0.433	0.271
DMU19	0.526	0.394	**DMU44**	0.800	0.333
DMU20	0.389	0.265	**DMU45**	0.240	0.200
DMU21	0.432	0.295	**DMU46**	0.506	0.422
DMU22	0.319	0.224	**DMU47**	0.442	0.331
DMU23	0.551	0.353	**DMU48**	0.242	0.195
DMU24	0.319	0.238	**DMU49**	0.312	0.198
DMU25	0.380	0.260			

APPENDIX H:

Actual and Predicted Y-Output Values for Ordinary Least Squares regression Models OLS1 and OLS2 Efficient Frontier Function and Inefficient Frontier Function

Appendix H1:Actual and Predicted Output Values for 49(forty-nine) DMUs for OLS1 and OLS2 Efficient Frontier Function $F_{(Eff\text{-}OLS1)}(X) = F_{(Eff\text{-}OLS2)}(X)$

DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)	DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)
DMU1	0.542	1.336	**DMU26**	0.470	0.817
DMU2	0.417	0.653	**DMU27**	0.407	0.624
DMU3	0.443	0.840	**DMU28**	0.337	0.577
DMU4	0.316	0.592	**DMU29**	0.212	0.381
DMU5	0.204	0.396	**DMU30**	0.310	0.409
DMU6	0.281	0.400	**DMU31**	0.297	0.511
DMU7	0.332	0.700	**DMU32**	0.200	0.315
DMU8	0.491	0.533	**DMU33**	0.552	0.884
DMU9	0.420	0.724	**DMU34**	0.469	0.785
DMU10	0.512	1.070	**DMU35**	0.553	1.068
DMU11	0.461	0.963	**DMU36**	0.292	0.678
DMU12	0.410	0.745	**DMU37**	0.306	0.511
DMU13	0.422	0.788	**DMU38**	0.223	0.390
DMU14	0.226	0.441	**DMU39**	0.456	0.724
DMU15	0.296	0.284	**DMU40**	0.326	0.740
DMU16	0.543	0.625	**DMU41**	0.410	0.670
DMU17	0.273	0.559	**DMU42**	0.330	0.599
DMU18	0.359	0.636	**DMU43**	0.433	0.617
DMU19	0.526	0.972	**DMU44**	0.800	0.796
DMU20	0.389	0.599	**DMU45**	0.240	0.346
DMU21	0.432	0.685	**DMU46**	0.506	1.047
DMU22	0.319	0.467	**DMU47**	0.442	0.788
DMU23	0.551	0.855	**DMU48**	0.242	0.267
DMU24	0.319	0.517	**DMU49**	0.312	0.328
DMU25	0.380	0.584			

Appendix H2:Actual and Predicted Output Values for 49(forty-nine) DMUs for OLS1 Inefficient Frontier Function $F_{(Ineff-OLS1)}(X)$

DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)	DMU	Coopersmith Scores Actual (Y-Output values)	Coopersmith Scores Predicted (Y-Output values)
DMU1	0.542	0.544	**DMU26**	0.470	0.337
DMU2	0.417	0.285	**DMU27**	0.407	0.277
DMU3	0.443	0.345	**DMU28**	0.337	0.263
DMU4	0.316	0.267	**DMU29**	0.212	0.209
DMU5	0.204	0.213	**DMU30**	0.310	0.216
DMU6	0.281	0.214	**DMU31**	0.297	0.244
DMU7	0.332	0.300	**DMU32**	0.200	0.191
DMU8	0.491	0.250	**DMU33**	0.552	0.360
DMU9	0.420	0.307	**DMU34**	0.469	0.327
DMU10	0.512	0.429	**DMU35**	0.553	0.428
DMU11	0.461	0.388	**DMU36**	0.292	0.293
DMU12	0.410	0.314	**DMU37**	0.306	0.244
DMU13	0.422	0.328	**DMU38**	0.223	0.211
DMU14	0.226	0.225	**DMU39**	0.456	0.307
DMU15	0.296	0.183	**DMU40**	0.326	0.312
DMU16	0.543	0.277	**DMU41**	0.410	0.291
DMU17	0.273	0.258	**DMU42**	0.330	0.269
DMU18	0.359	0.280	**DMU43**	0.433	0.275
DMU19	0.526	0.391	**DMU44**	0.800	0.330
DMU20	0.389	0.269	**DMU45**	0.240	0.199
DMU21	0.432	0.295	**DMU46**	0.506	0.419
DMU22	0.319	0.232	**DMU47**	0.442	0.328
DMU23	0.551	0.350	**DMU48**	0.242	0.179
DMU24	0.319	0.246	**DMU49**	0.312	0.195
DMU25	0.380	0.265			