Simulation-Based Construction Productivity Improvement Using Neural-Network-Driven Fuzzy Reasoning System

Seyedfarid Mirahadi

A Thesis

in

The Department

of

Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Building Engineering) at

Concordia University

Montreal, Quebec, Canada

August 2013

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Seyedfarid Mirahadi

Entitled: Simulation-based Construction Productivity Improvement Using Neural-Network-Driven Fuzzy Reasoning System

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Civil Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. O. Moselhi Chair

Dr. Z. Zhu Examiner

Dr. Gopakumar, External-to-Program Examiner

Dr. T. Zayed Supervisor

Approved by _____
Chair of Department or Graduate Program Director

_____
Dean of Faculty

Date October 16, 2013

ABSTRACT

Simulation-Based Construction Productivity Improvement Using Neural-Network-Driven Fuzzy Reasoning System

Seyedfarid Mirahadi

Fuzzy-based models and Artificial Neural Network (ANN) based systems have provided effective tools for addressing uncertainties in decision-making. Uncertainty, as an ineradicable part of construction projects, justifies the utilization of such intelligent systems in the construction industry. In the past few years, these systems have been widely applied to develop forecasting models in the construction management area. The estimation of productivity of construction operations, as a basic element of project planning and control, has become a remarkable target for forecasting models. A glimpse into this interdisciplinary field of research exposes the need for a system, which 1) studies the effect of qualitative and quantitative variables on construction productivity, 2) improves the previous models in terms of accuracy of estimation, 3) is able to clearly illustrate the reasoning process, 4) considers the interdependence of input variables; and 5) has the capability of dealing with both crisp and linguistic input variables.

The main objective of this research is to develop a hybrid intelligent system for estimating productivity of construction operations based on several qualitative and quantitative factors. Among all models developed for productivity estimation, those established based on the functional relations and controlled by a specific number of control rules are more compatible with the human reasoning and logic. *Neural-Network-Driven Fuzzy Reasoning* (NNDFR) structure, as one of such models, displays a great

potential for modeling datasets among which clear clusters are recognizable. The lack of compatibility between conventional NNDFR and fuzzy clustering algorithms together with the insufficient attention paid to the optimization of number of clusters in this model, created a potential area for further research. Thus, the main contribution of the proposed model is to develop a modified NNDFR system for modeling construction data. It forms a nonlinear multi-dimensional membership function, which internally combines all fuzzy variables via Fuzzy C-Means (FCM) clustering. An ANN is then trained based on the clustering process to automate this step. While the clustering step constitutes "IF" parts of the rules, "THEN" parts are built by another set of ANNs. In addition, the parameters of the proposed system are optimized by Genetic Algorithm (GA) to fine-tune the system for the highest possible level of accuracy. The model is also capable of dealing with a combination of crisp and linguistic input variables through the use of a Hybrid Modeling Approach, which is based upon the application of alpha-cut technique.

The proposed model is further verified through simulating a construction operation considering qualitative and quantitative factors where a considerable improvement in the estimation accuracy is witnessed. Several models are developed using ANN, Adaptive Neuro-Fuzzy Inference System (ANFIS), conventional three-cluster NNDFR and the Genetically Optimized NNDFR. The proposed model showed 83%, 72% and 69% improvement over ANN, ANFIS and conventional NNDFR, respectively, in terms of Mean Squared Error (MSE). The developed model helps researchers and practitioners use historical data to forecast productivity of construction operations with a level of accuracy greater than what could be offered by traditional techniques.

ACKNOWLEDGEMENT

Foremost, I offer my most sincere gratitude to my supervisor, Dr Tarek Zayed, whose indelible support was my greatest asset throughout the present research. His immense patience, unfathomable knowledge and endless professionalism brought this thesis to a consummation. He maintained a perfect balance between mentoring and giving me the latitude to explore my own ideas. I have learned a lot from him and it was a great honor to work under his supervision. I could have not imagined a better advisor and mentor for my Master's study.

Secondly, I would like to thank my parents, Mansour Mirahadi and Nadia Pourshahi, for their great support and encouragement throughout my life. They have played an instrumental part in my development as a person and I owe them a debt of gratitude.

My sincere thanks to my friend Farid Vahdati for his detailed advices, and discussions, and brotherly help throughout my thesis work.

Last but not least, I have to offer my special thanks to my dearest friend, Babak Mohammadi, for his sincere help and support, and being there for me in ups and downs of life during last 10 years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
MSE	Mean Squared Error
NNDFR	Neural-Network-Driven Fuzzy Reasoning
EA	Evolutionary Algorithm
ANFIS	Adaptive Neuro-Fuzzy Inference System
GA	Genetic Algorithm
AI	Artificial Intelligence
ES	Expert System
FCM	Fuzzy C-Means
AVP	Average Validity Percentage
AIP	Average Invalidity Percentage
GUI	Graphical User Interface

CHAPTER 1: INTRODUCTION

1.1 CHAPTER OVERVIEW

Simulation is the imitation of the operation of a real-world process (Banks 2005). It can provide a probabilistic approach to handle the uncertainties of a problem. Simulation is a powerful tool that can be applied in different aspects of construction management, such as productivity estimation, risk management, scheduling, and resource planning. Most research in construction simulation has mainly focused on the simulation modeling with a little emphasis on the qualitative variables that affect the simulation process itself. However, over the past few years, a few researchers have employed different soft computing techniques to forecast productivity of construction operations based on several qualitative and quantitative factors. The productivity estimation of construction operations, as a decision criterion in project planning and control, has become an interesting target for forecasting models. Fuzzy-based models and Artificial Neural Network (ANN) based systems have evolved decision-making process by taking into account the uncertainty impact. Uncertainty, as an ineradicable part of construction projects, justifies the utilization of such intelligent systems in the construction industry. In the past few years, these systems have been widely applied to develop forecasting models in construction industry.

The integration of basic soft computing techniques, such as ANNs, fuzzy logic, Evolutionary Algorithms (EAs), etc., has empowered researchers to create more efficient

forecasting models. In such integrative models, the limitations of one method are compensated for by other methods. This trend led to the development of a variety of hybrid intelligent structures. The selection of an appropriate structure which offers high accuracy is based on the application area and the inherent features of the data at hand.

With the increasing volume of historical data provided to these kinds of models, the need for data analysis techniques has significantly grown. Data clustering can be regarded as the most well-known and prevalent technique in the exploratory data analysis. It provides a requisite data pre-processing step to identify homogeneous patterns in data, according to which consequent supervised models are built. Furthermore, the wide appeal and effectiveness of data clustering techniques have pushed researchers to combine them with other techniques, such as ANN and fuzzy reasoning. The recent trend has resulted in a diversity of cluster-aided models. Adaptive Neuro-Fuzzy Inference System (ANFIS) is a well-known example of such cluster-aided models, which have benefited from the intelligent partitioning of clustering algorithms.

The more comprehensively we study the variables that affect the final modeling outputs, the more number of variables must be considered. As a result, it is very likely to have a combination of linguistic terms (qualitatively described factors) and crisp values (quantitatively described factors) in any modeling process (Guyonnet et al. 2002). These cases are also very common in the construction simulation where the limited data provided for some factors can only be explained by linguistic terms (Sadeghi et al. 2010). Many researchers tried to find a way to estimate the output of these generalized models using both types of input values. Therefore, the flexibility of a model in dealing with both

types of input variables can be considered as a requirement for an efficient forecasting model.

1.2 PROBLEM STATEMENT

This research is mainly motivated by the little emphasis placed on the factors that affect the simulation processes. The shortcomings of the commonly used intelligent systems in the construction area have provided the impetus for investigating the more efficient forecasting models. The limitations and drawbacks of current methods can be encapsulated as follows:

1) Little emphasis is placed on the qualitative and quantitative factors that affect simulation process;
2) Common soft-computing models do not show a satisfactory accuracy in estimation;
3) ANN models do not explain the quality of input-output mapping process and act like a "black-box";
4) Conventional fuzzy reasoning is not able to consider the interdependence of variables during the process of membership function design;
5) Most of the forecasting models only accept crisp values as inputs and do not accept a combination of linguistic terms and crisp values;
6) There is a lack of comparison between different predictive models to show the strength of the proposed model over others.

1.3 RESEARCH OBJECTIVES

With respect to the mentioned problems, the overall objective of this research is to develop a hybrid intelligent system for estimating the productivity of construction operations considering several qualitative and quantitative factors. Based on the decomposition of the main objective to several sub-objectives, this research is inspired to:

1) Identify and study the shortcomings of fuzzy reasoning and ANN-based simulation;
2) Develop a modified Neural Network Driven Fuzzy Reasoning (NNDFR) model with optimized parameters;
3) Improve the developed model to deal with crisp values and linguistic terms simultaneously.

1.4 SUMMARY OF THE RESEARCH METHODOLOGY

The methodology of this research includes several steps as follows:

1) Literature review: The literature review encompasses subjects including the state of the art in construction simulation, explorative data mining and statistical data analysis, hybrid intelligent systems and genetic optimization;
2) Research Methodology and Model Development: A model will be proposed and developed to address the problems identified in the literature. The model development consists of three main sub-phases. The first phase comprises the concept and know-how about the implementation of the modified NNDFR. The model combines fuzzy models and ANN systems in such way that their integration will significantly improve the performance. In the second phase, the parameters of the assembled model are optimized using Genetic

Algorithm (GA). And finally, in the third phase, the Hybrid Approach of modeling is adopted in order to enable the model to work with both crisp and fuzzy variables;

3) Case Study and Data Collection: A case study will be introduced and its pertinent data will be gathered and analyzed to verify the outcomes generated by the developed model;
4) Implementation and Results: The case study's data will be implemented in the model and its results will be compared with the outcomes of other systems for the validation purpose.

1.5 THESIS ORGANIZATION

The thesis consists of six chapters. Chapter 1 includes the problem statement, research objectives, summary of the research methodology and the thesis organization. Chapter 2 presents an elaborated literature review on subjects regarding the state-of-the-art in construction simulation, explorative data mining and statistical data analysis, hybrid intelligent systems and genetic optimization. In Chapter 3, a comprehensive description of the proposed framework is provided. Chapter 4 introduces the case study and presents the data collection source, procedure and preparation. Chapter 5 reviews the results of the implementing the proposed model in the case study and highlights the merits of the proposed framework over other systems. At the end, Chapter 6 is about conclusions and recommendations.

CHAPTER 2: LITRATURE REVIEW

2.1 CHAPTER OVERVIEW

This chapter aims at providing a comprehensive literature review about the current state of the productivity estimation in the construction industry and the techniques used for this purpose. Figure 2-1 illustrates an overview of this chapter.

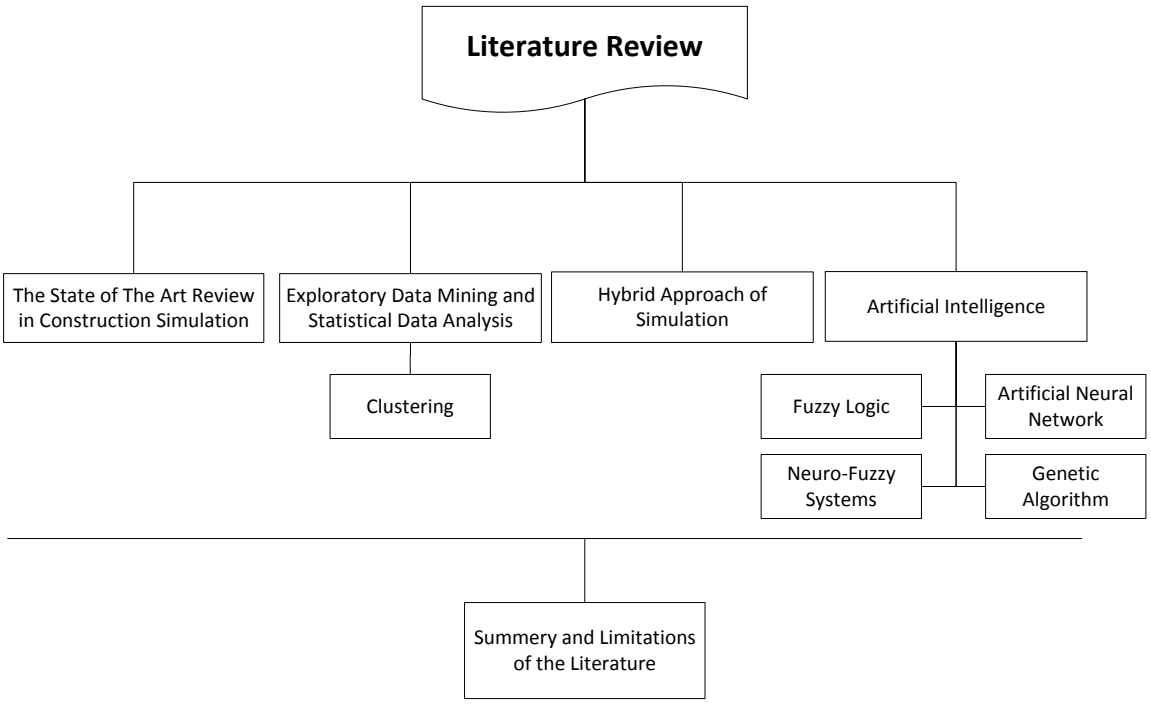


Figure 2-1: An Overview of the Literature Review

Section 2.2 reviews the literature related to the state of the simulation in construction management. Section 2.3 is related to the background and importance of the productivity assessment in the construction industry. Section 2.4 focuses on the literature related to

artificial intelligence including four subsections, namely, ANN, fuzzy reasoning, neuro-fuzzy systems and GA. The literature regarding explorative data mining and statistical data analysis is summarized in Section 2.5. Section 2.6 describes the previous research on the hybrid approach for the simulation that deals with two types of uncertainties (probability and possibility). And finally, identified the shortcomings of the reviewed literature will be presented in Section 2.7.

2.2 CONSTRUCTION SIMULATION

Many construction and engineering projects consist of repetitive activities, such as earthmoving projects. A considerable amount of time, money and effort can be saved in any project if proper decisions are made at the planning stage. Because of the stochastic nature of cyclic construction processes, historical data gathered from previous projects can assist planning engineers in making a better estimation about the upcoming productivity rates (Graham and Smith 2004). In the traditional estimation methods, planning engineers manually adjust productivity records to establish the expected values. The estimation and understanding of the production aspects of projects have been a crucial task for researchers and practitioners in construction engineering and management. By increasing the complexity and size of projects, planning and decision making in this area is likely to be either impossible or very inaccurate. As a remedy, the computer simulation has been proposed to build an abstract model of a particular system and estimate the performance of the system in a virtual environment.

Simulation is one of the most widely used techniques in the operational and managerial research (Law and Kelton 2000). Construction simulation is the process of developing computer-based models that represent real-world construction systems in order to investigate their underlying behavior (AbouRizk 2010). Construction simulation is a powerful tool that can be used by a construction company for several tasks, such as productivity measurement, risk analysis, resource planning, and the design and analysis of construction methods (Sawhneyet al. 1998). Among all these applications, the productivity measurement might be considered as the most important factor in the construction planning and control. Most research in the construction simulation has mainly focused on the simulation modeling with limited emphasis on study of the qualitative variables that affect the simulation process itself (Elwakil 2011). Although some studies have investigated the effect of qualitative and quantitative factors on different aspects of construction processes, there is still a lack of research in this area.

2.3 PRODUCTIVITY ASSESSMENT

The fierce competition in the construction industry propels all stakeholders to improve the productivity. That is why the productivity estimation has caught such a significant attention in both industry and academia. Today, the productivity management is recognized as a major project management concern in the construction industry (Park et al. 2005). The rate of the construction productivity varies from one project to another due to different environmental and managerial conditions. The considerable impact of these project-specific factors on the productivity rate makes it of a cardinal importance to

consider them in the productivity estimation using simulation (Park 2006). These project-specific factors can impact the productivity both positively and negatively.

The scheduled overtime, change orders, materials management, weather and human factors were identifed by Perk (2006) as the main factors that influence the productivity rate. The identification of these factors is a preliminary step in creating a model for the productivity estimation. The very common approach for productivity estimation is to use the historical data from previous projects of like nature as a baseline for the new projects.

For example, Moselhi et al. (1997) developed a decision support system called WEATHER to examine the impact of weather conditions on the productivity of construction operations. The developed model estimates the construction productivity, activity durations, and weather patterns in different modes to improve the accuracy of the planning and scheduling (Moselhi et al. 1997). Given that in this framework only weather factors are taken into account, other complementary models are needed for the consideration of other external factors.

Moselhi et al. (1991) investigated 57 different construction projects in order to study the impacts of change orders on the productivity of construction projects. They discovered a direct correlation between the labor component of change orders and the productivity loss in all types of projects (Moselhi et al. 1991).

Regression model is the most common statistical model for the productivity estimation when considering specific factors (Sanders and Thomas 1993; Smith 1999). Hanna et al. (1999) developed a regression model to investigate the effect of change orders on the

construction productivity. Koehn and Brown (1985) employed some non-linear equations to examine the effect of weather changes on the productivity rate. The learning curve is also an important theory in the productivity estimation. Based on the learning curve theory, the productivity of a repetitive process gradually increases as a result of the greater familiarity with the process, improved management, and more efficient application of tools and equipment (Oglesby et al. 1989). In most cases, there is no pre-identified functional relation between variables affecting the level of productivity and their outputs. Besides, there is no guarantee that simple models like linear regression can satisfy the expected accuracy of a forecasting model. In the wake of these limitations, researchers have considered the application of Artificial Intelligence (AI) systems that can be used to model complex relationships between a set of dependent and independent variables.

2.4 ARTIFICIAL INTELLIGENCE (AI)

There are several definitions of AI in the literature. For instance the below definitions are presented by some researchers:

- "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield 1993).
- "The study of the computations that make it possible to perceive, reason, and act" (Winston 1992).
- "The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil 1992).

Therefore, AI can be concisely explained as the field of understanding and building intelligent systems. There are an increasing number of AI applications. However, the most common AI applications are Game Playing, Speech Recognition, Understanding Natural Language, Computer Vision, Expert Systems (ESs) and Reasoning of Humans. Among all these applications, ES and Human Reasoning pay attention to the process of inferring new facts from knowledge and incoming data. Among the wide spectrum of AI applications and techniques, ANNs, Fuzzy Logic and EAs are the most well-known and frequently used ones. These techniques are elaborately explained in the following sections.

2.4.1 Artificial Neural Network (ANN)

ANN is a mathematical model used for finding patterns among the datasets where there are complex relationships between the inputs and outputs. ANN tries to simulate the structure and operation of human neural network system. An ANN structure comprises an interconnected set of artificial neurons that operate based on a connectionist approach of computation.

Each one of these artificial neurons is a mathematical function, which gets a weighted sum of several inputs and passes them through a "transfer function". In this structure, the output of each artificial neuron is an input for the others and collectively they build an interconnected net of ANN. Figure 2-2 shows a schematic feed-forward ANN, which can be considered the most frequently used, and yet the simplest, type of ANN. All the connections of a feed-forward network only move forward, directed from input, i.e.

independent, variables to output, i.e. dependent, variables. In a feed-forward network, artificial neurons are divided to three layers, namely, input layer, hidden layer and output layer. All rows of artificial neurons between the input, the first layer, and the output, the last layer, are deemed as the hidden layer. A crude ANN is constituted of a set of unknown weights and biases, which can be delivered via several proposed algorithms of network training, such as backward propagation of errors, or in short Back-propagation.

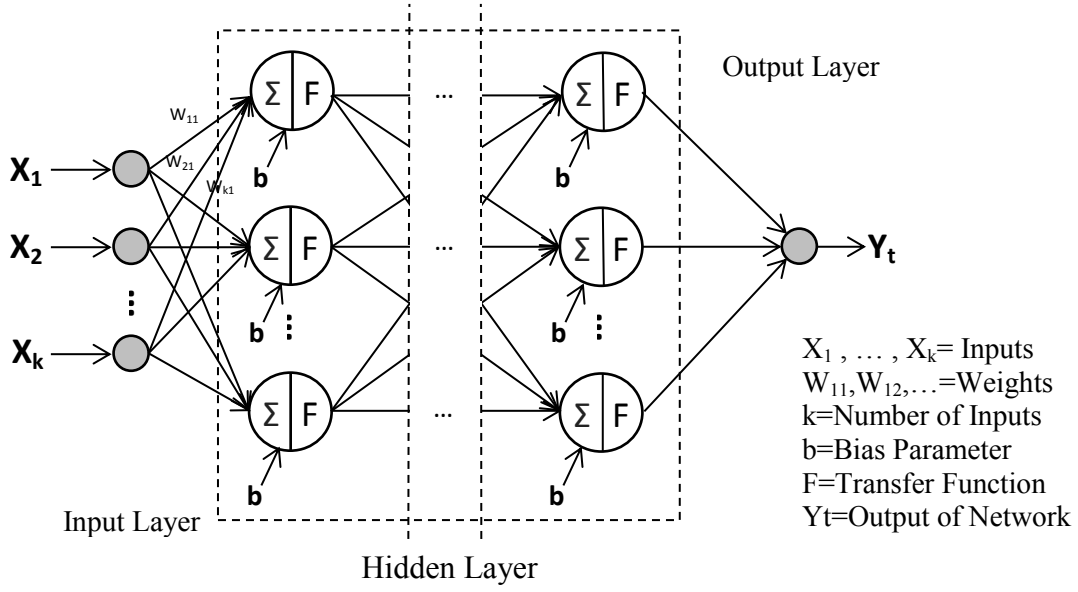


Figure 2-2: Schematic Diagram of a Multi-Layer Feed Forward ANN (Zhu Et Al. 2007)

Back-propagation, as the most common learning algorithm for ANN, iteratively attempts to extract the hidden relationships between inputs and outputs of a set of data. In this method, weights and biases initially adopt random values. By feeding the historical data to the network, in each iteration, the estimation errors are calculated and the weights and biases are accordingly updated. This process stops until a predefined termination

condition is reached. At this point, the trained ANN can be applied to any future set of data in which the same relationships between inputs and outputs exists (Bryson and Ho 1975; Werbos 1974; Alpaydin 2004; Rumelhart et al. 2002).

Thus, ANN performs two main tasks: (1) learning and (2) recalling (Hegazy and Moselhi 1994). Learning is the act of acquiring the suitable weights and biases of a row net to generate the nearest outputs according to defined targets (Zayed and Halpin 2005). There are two types of training. Supervised training refers to the method where outputs are provided to ANN. On the other hand, unsupervised training does not require any output to accomplish the learning process. Recalling is the process of catching an input vector and generating an output based on the network parameters that have been trained during the learning phase (Zayed and Halpin 2005). Then, the estimated outputs are compared with actual targets to represent an index for the performance error.

The ability to learn from examples made this technique a very useful tool in data modeling (Lawrence 1994). This technique can develop a predictive model where the relationships between inputs and outputs are not sufficiently known. Patterns and relationships in the historical data are recognized to help acquire the ‘‘knowledge’’ required to predict the unknown output values for a given set of input values (Sawhney et al. 2002). Since ANN acts like a “black-box” and cannot explain the process of reasoning, it is well-suited to the problems where the underlying reasons and the quality of input-output relations are not studied (Elwakil 2011).

In the past few decades, ANN was extensively applied in forecasting models in the construction industry. Researchers applied this technique in many aspects of construction management in order to benefit from its advantages over the conventional modeling methods. The projects' cash flow prediction, risk analysis, resource optimization, and the tendering outcomes prediction are only a few examples of ANN application in the construction industry (Boussabaine 1996; Li 1995).

Kim et al. (2004) compared the accuracy of estimating construction cost through three different models, namely, Multiple Linear Analysis, ANN and case-based reasoning. ANN showed the best performance in terms of the estimation accuracy. Moselhi et al. (1992) highlighted the potential application of ANN for estimating the productivity rate of construction trades based on several specific attributes. Zayed and Halpin (2005) utilized ANN technique to assess the productivity, cost and cycle time of the piling process. In their study, seven inputs and 10 outputs were included to comprehensively capture the different aspects of piling process. Khan (2005) analyzed and selected nine input parameters as the factors, which cause short-term variations in labor productivity rate. In the research, the effect of input variables was modeled by different structures of ANN and finally the result of the best structure was compared with the regression modeling result.

2.4.2 Fuzzy Reasoning

Fuzzy Reasoning is defined based on the theory of fuzzy sets and involves AI, information processing, logic to pure theories and applied mathematics, such as graph

theory, topology and optimization (Pappis and Siettos 2005). The basic definitions and concepts used in fuzzy reasoning are presented in the following sections.

2.4.2.1 Concept of Fuzzy Logic

Fuzzy logic (Zadeh 1977) was introduced as a response to the need for a systematic reasoning that better conforms to the human logic. The main goal of fuzzy logic is to connect an input space to an output space. This goal is achieved through the application of several if-then statements called fuzzy rules, as shown in Equation 1. Each fuzzy model has a rule base, which is the list of all fuzzy rules. The inference procedure is performed through the parallel evaluation of all defined fuzzy rules. In this approach, unlike ANN, interpretability of the inference procedure is at the center of attention. In contrast with the mathematical logic where variables only take numerical values, fuzzy logic often uses linguistic terms (fuzzy variables) to express rules and specifications (Zadeh 1996).

$$\text{IF x is A AND y is B, THEN z is C} \qquad \text{Eq. (1)}$$

Where x and y are linguistic variables and A & B are linguistic terms represented by fuzzy sets.

2.4.2.2 Fuzzy Sets

A fuzzy set is a class of objects without a clearly defined boundary. While in the classical set theory, elements have a one-to-one membership relation to sets, i.e. an element belongs to one set only, Fuzzy Set theory permits the partial memberships of the elements

to multiple sets. The degree of membership to each set can be measured using membership functions.

2.4.2.3 Fuzzification & Defuzzification

Fuzzification is the process of converting crisp values of input variables to fuzzy values through applying the fuzzy membership functions. A membership function is a curve that reflects the degree of membership of each point in the input space, as shown in Figure 2-3. Defuzzification is the process of converting membership values to crisp output values.

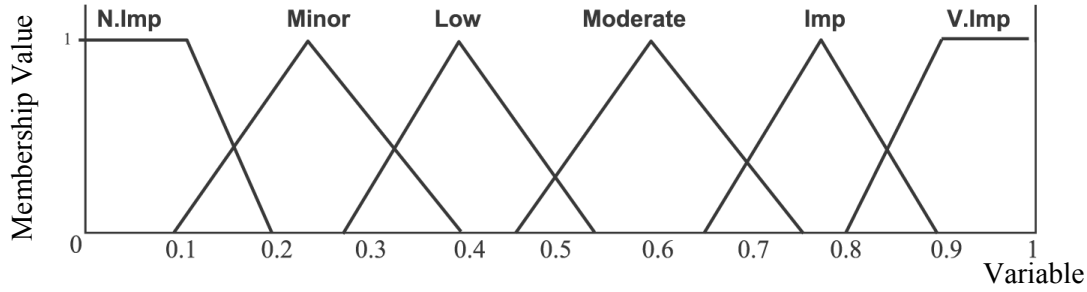


Figure 2-3: A Sample of Membership Functions

2.4.2.4 Possibility

Possibility is an alternative representation of uncertainty and was first introduced in the possibility theory of Zadeh (1978). Possibility theory was an extension of his previous fuzzy logic and fuzzy sets theories. Dubois and Prade (1988) further developed this idea and presented a method to report the available knowledge in the variable X. In this method, the available information is represented by fuzzy numbers which are, in turn, defined by membership functions. The calculated membership value, which is a value between 0 and 1, indicates the possibility of a certain value for the variable under study.

The term ‘‘possibility’’ relates to the idea of “lack of surprise” proposed by Shackle (1961). The assumption is that “the more possible a value is, the less surprising it would be” (Guyonnet et al. 2002).

$\prod(A)$ denotes the possibility of the specific event as shown in Equation 2 (Guyonnet et al. 2002).

$$\prod(\mathrm{A}) = \mathrm{Sup}_{\mathrm{x}\in\mathrm{A}}\mu(\mathrm{x}) \qquad \text{Eq. (2)}$$

Where $\mu(x)$ is the membership function for the variable X.

2.4.2.5 Alpha-Cut

The alpha-cut (α-cut) of a fuzzy set is the set of all crisp values with the membership values higher than or equal alpha (α). Figure 2-4 shows a fuzzy number representing the parameter P with the support of A_0. The crisp range that includes all the elements with the membership values greater than α is called the alpha-cut of this fuzzy number. At each level of α, the resultant fuzzy set has a support of A_α representing its alpha-cut (Abebe et al. 2000).

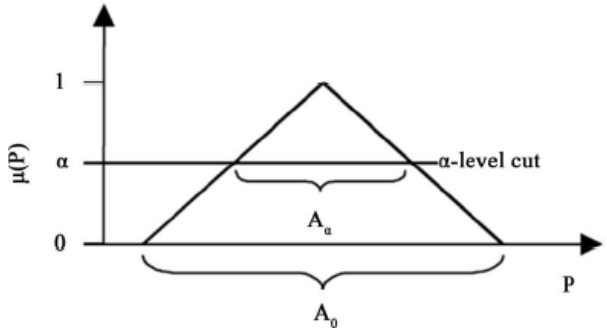


Figure 2-4: Fuzzy Number and Alpha-Cut (Abebe Et Al. 2000)

In fuzzy logic, alpha cut is used to decompose a fuzzy set into a nested form of the classical sets based on the resolution identity principle (Xexéo). This principle states that each fuzzy set can be expressed by a weighted set of alpha-cuts, as shown in Equation 3 (Kulik 2001).

$$A = \bigcup_{\alpha \in [0,1]} \{ \alpha A_{\alpha} \} \qquad \text{Eq. (3)}$$

Where A_{α} is the alpha-cut at the resolution level of α and A is the set of alpha-cuts. The resolution identity principle is the basis for the conversion of fuzzy sets to classical crisp sets. Because of this, it is considered as one of the most important principles in the fuzzy set theory.

2.4.2.6 Fuzzy Inference Process

There are five main steps for the interpretation of if-then rules as follows (MathWorks 2012):

1) **Fuzzification:** Fuzzifing all the inputs with the use of membership functions;
2) **Fuzzy Operations:** Converting the multiple sections of the IF (antecedent) part of the rules, if there are more than one, to one single value using fuzzy operators. This value is called degree of support for the rule. Degree of support for the rules with only one antecedent part equals the degree of membership of that fuzzy set;
3) **Implication:** Appling the degree of support to create the output fuzzy set. The consequent part of the rule presents a specific fuzzy set to the output. The output

membership function represents this output fuzzy set. If the antecedent is assigned a value less than one, then the fuzzy set is cut based on the implication method;

4) **Aggregation:** Combining the output fuzzy sets coming from different rules into a single fuzzy set so that decisions can be made based on the evaluation of all fuzzy rules. In the context of fuzzy reasoning, this process is called aggregation. The inputs of this process are all the truncated output fuzzy sets coming from different rules, and the output is one fuzzy set per each output variable. There are different methods of aggregation, such as Maximum or Sum;
5) **Defuzzification:** Defuzzifying all the aggregated fuzzy set. The input of this process is a fuzzy set and the output is a crisp output variable.

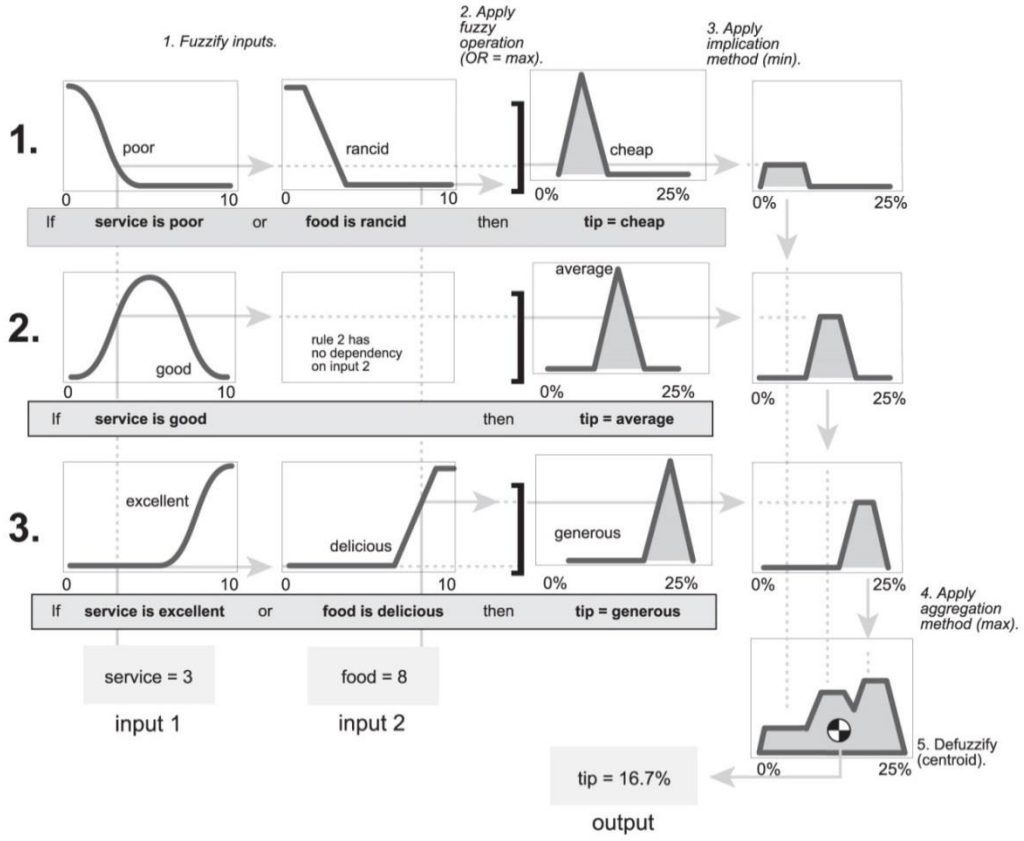
Figure 2-5 illustrates a simple example of the fuzzy rules interpretation.

2.4.2.7 Fuzzy Inference Types

There are two main types of fuzzy inference systems. The most frequently used type, which has a greater conformity to the human's intuition, is called Mamdani. It was first proposed by Ebrahim Mamdani in 1975 (Yager and Filev 1994). This type of fuzzy inference is comprised of all the afore-mentioned steps of inference, as shown in Figure 2-5.

Takagi-Sugeno-Kang, or so-called Sugeno, is the other type of fuzzy inference systems that was first proposed in 1985. The main difference between Mamdani and Sugeno systems is that based on the Sugeno method the consequent parts of the rules are either single values (singletons) or linear functions. These linear functions transform the inputs

to a crisp value. The final output of the system is the weighted average of all the crisp outputs of the rule. The weights are the same as the degree of support for each rule in Mamdani systems. Figure 2-6 illustrates the inference procedure of a singleton-output-function Sugeno structure.



2.4.2.8 Figure 2-5: Fuzzy Inference Process Diagram–Mamdani Type (Mathworks 2012)Application of Fuzzy Reasoning in Construction

Fuzzy reasoning concept has been applied in different areas of the construction management. For instance, Carr and Tah (2001) used fuzzy concepts to address the project risk assessment and analysis. Zhang et al. (2002) showed the application of fuzzy logic in discrete-event simulation in order to address the uncertainties in the resource

demands and durations of the processes. Chang et al. (1990) applied the integration of fuzzy systems and expert systems for the project resource allocation. Paek et al. (1992) applied a multi-criterion decision-making methodology using a fuzzy-logic system for the selection of the successful Design/Build proposal.

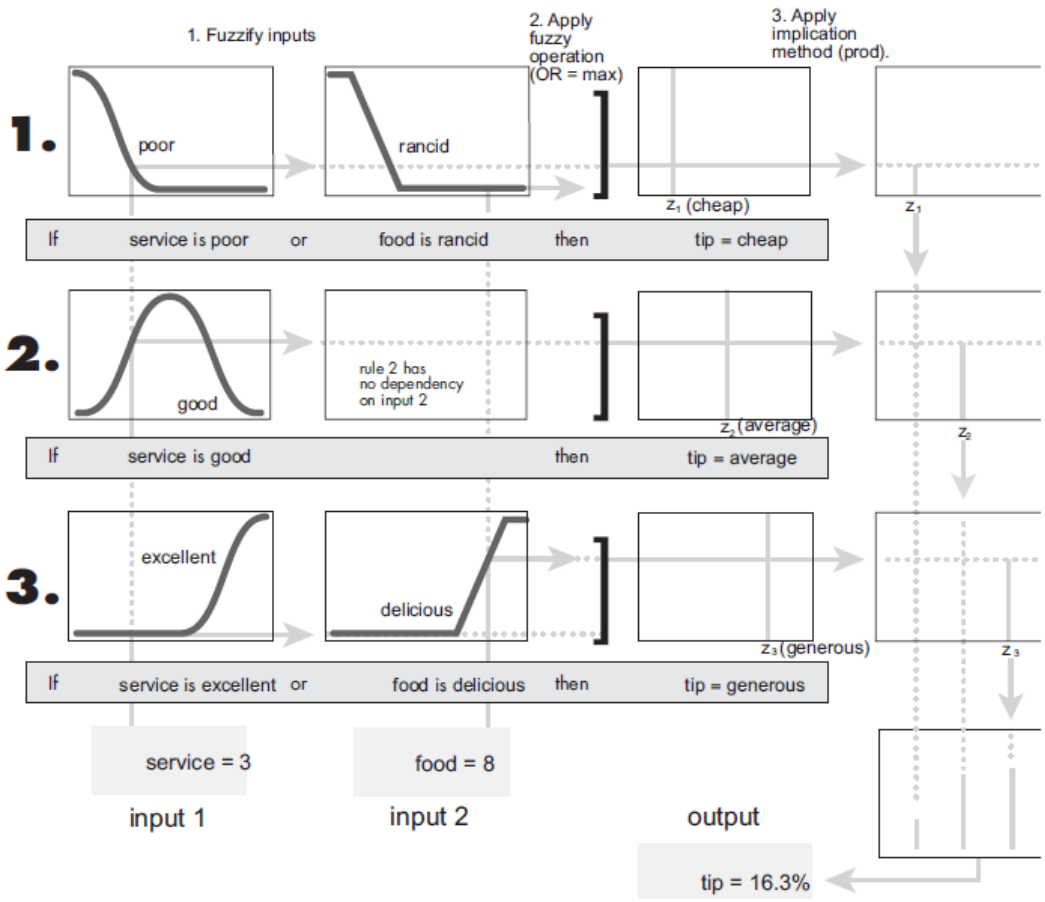


Figure 2-6: Singleton-Output-Function Sugeno Inference System (Mathworks 2012)

2.4.3 Neuro-Fuzzy Systems

ANN and fuzzy logic are two complimentary technologies. ANN is capable of learning from the data; however, it cannot explain the quality of input-output mapping process. On

the other hand, fuzzy reasoning provides a systematic reasoning method which is more compatible with the human logic and intuition. However, the learning process of fuzzy models needs to adopt self-regulating techniques from other areas. The limitations of these two techniques led to the development of neuro-fuzzy systems. In the field of hybrid intelligent systems, the term neuro-fuzzy refers to the fusion of ANN and fuzzy logic (Jang 1993). In these structures, the shortcomings of each technique are offset by the other. Neuro-fuzzy systems automate the tuning process of the membership functions using the learning capability of ANN.

2.4.3.1 Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is a structure of ANN that works based on the principles of Takagi–Sugeno fuzzy inference system (Jang 1993). This framework benefits from strengthens of both ANN and fuzzy logic by integrating the systematic reasoning of fuzzy logic with the learning ability of ANN. ANFIS is categorized as an adaptive network that function in the same way as fuzzy inference systems. This technology tries to embed the whole process of fuzzy reasoning in an ANN assembly.

Figure 2-7 shows a schematic structure of ANFIS. This model represents a two-input first-order Sugeno fuzzy model with the following two rules:

Assuming two inputs X and Y and one output Z:

Rule 1: IF x is A_1 and y is B_1, THEN $f_1 = p_1x + q_1y + r_1$

Rule 2: IF x is A_2 and y is B_2, THEN $f_2 = p_2x + q_2y + r_2$

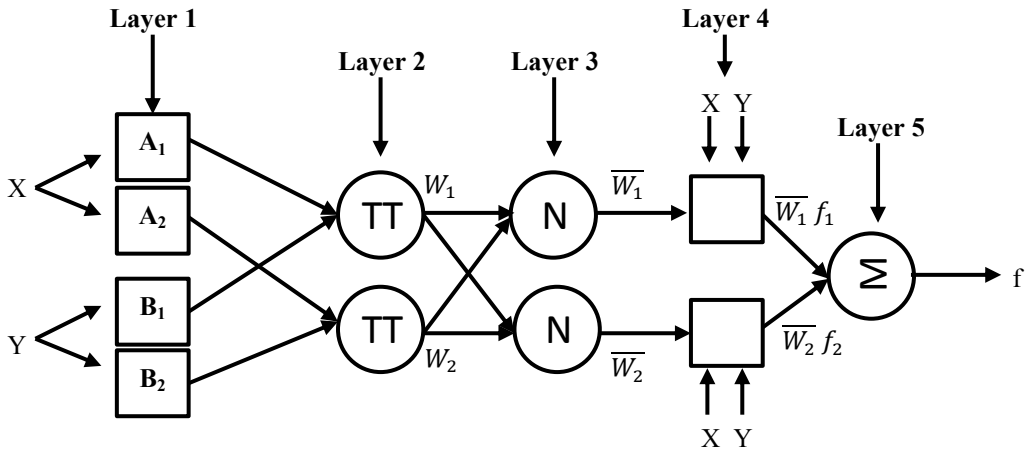


Figure 2-7: ANFIS Equivalent for Two-Input First-Order Sugeno Fuzzy Model with Two Rules (Jang 1993)

Input membership functions are also assumed as depicted in Figure 2-8.

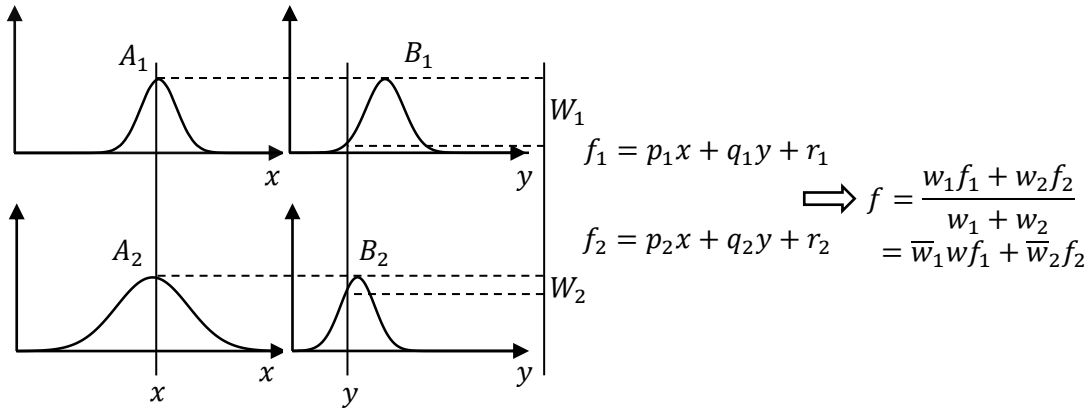


Figure 2-8: Inference Procedure of a Two-Input First-Order Sugeno Fuzzy Model with Two Rules (Jang 1993)

As shown below, each layer of the ANFIS structure mimics a specific phase of the Sugeno inference procedure.

Layer 1: Every node in this layer is an adaptive node with a function that represents the membership function of a linguistic term; where x (or y) is the input to the node and Ai (or Bi) is a linguistic term. The output of this layer is the degree of membership (μ_{Ai} or μ_{Bi}) of the input (x or y) that is associated with that specific linguistic term Ai (or Bi). Parameters of the node function in this layer are called *premise parameters*.

Layer 2: The output of this layer (W_i) is the product of all incoming signals, Equation 4.

$$W_i = \mu_{Ai} \times \mu_{Bi} \qquad \text{Eq. (4)}$$

Where μ_{Ai} (μ_{Bi}) is the degree of membership of the input x (or y) that is associated with that specific linguistic term Ai (or Bi). The output of each node states the firing strength of a rule.

Layer 3: The i^{th} node in this layer calculates the ratio of the firing strength of the i^{th} rule to the summation of all the firing strengths, as shown in Equation 5.

$$\overline{W_i} = \frac{W_i}{W_1 + W_2} \qquad \text{Eq. (5)}$$

Where W_i is the firing strength of the ith rule.

Layer 4: Every node in this layer is an adaptive node with a function shown in Equation 6.

$$\text{Node function:} \quad \overline{W_i} f_i = \overline{W_i}(p_i x + q_i y + r_i) \qquad \text{Eq. (1)}$$

Where x and y are given variables and f_i is the consequent relationship (function). Parameters of this node are referred to as *consequent parameters*.

Layer 5: This single node calculates the summation of all the incoming signals, as shown in Equation 7.

$$\text{Final Output} = \sum_i \overline{W_i} f_i \qquad \text{Eq. (7)}$$

ANFIS utilizes a hybrid learning algorithm to identify the premise and consequent parameters. The hybrid approach of learning applies an integration of least-squares method and back-propagation gradient descent method to train the membership function parameters.

2.4.3.2 Neural-Network-Driven Fuzzy Reasoning (NNDFR)

NNDFR was the first application of ANNs in self-regulating design of membership functions. NNDFR was proposed by Takagi and Hayashi in 1992. It is categorized as a neuro-fuzzy model that has an accurate performance in the estimation of the output of the naturally clustered data spaces. In this structure, fuzzy logic controls the selection of the best inference process and ANN builds the inference system and membership functions.

Figure 2-9 shows the schematic structure of NNDFR and the interaction of its constituent parts. The design procedure of NNDFR can be summarized in the following three steps: (1) Clustering the training dataset, (2) training the membership ANN (NNmem), and (3) training the consequent ANN (NN1-k) of each cluster.

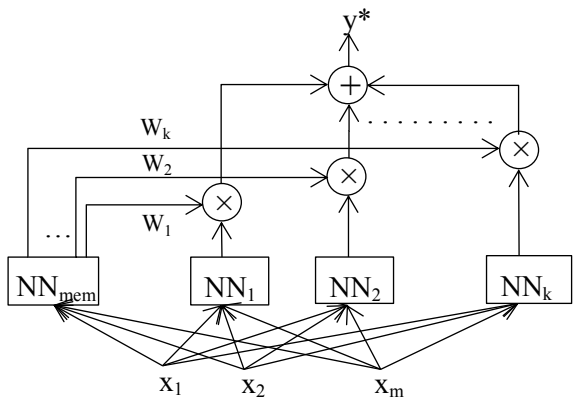


Figure 2-9: The Structure of NNDFR System (W Is The Membership Value And Y* Is Final Estimated Output)

In the first step, the input data space is partitioned into hard clusters through a clustering algorithm. In this neuro-fuzzy system, the number of rules equals the number of clusters.

In the second step, NNmem is trained between each input vector and its corresponding cluster assignment vector, as illustrated in Figure 2-10, where *m* is the number of input variables, *n* is the number of observations, and *k* is the number of clusters. For example, the supervised part of the learning process for a vector which belongs to the cluster 3 is (0,0,1).

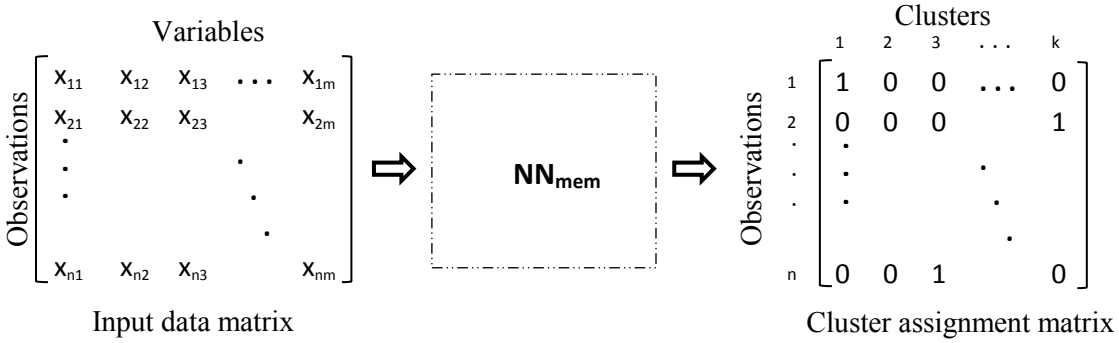


Figure 2-**10**: Second Step, Training the Membership **ANN**

In the third step, the consequent ANNs are trained between the members of each cluster, which were partitioned in the first step, and their corresponding outputs.

The NN_{mem} generates the membership functions of the premise, i.e. IF parts, of the rules and $NN_{1\text{-}k}$ prepare the consequent input-output relationships, i.e. THEN parts. This system calculates the final estimated output based on a weighted average of the output of THEN parts, such that the weights are the membership values produced by NN_{mem}. The resulting fuzzy model is expressed by the following rules (Takagi and Hayashi 1991):

$$\begin{gathered} Rule\ 1: IF\ x = (x_1, x_2, \dots, x_m)\ is\ C_1, THEN\ y_1 = NN_1(x_1, x_2, \dots, x_m) \\ Rule\ 2: IF\ x = (x_1, x_2, \dots, x_m)\ is\ C_2, THEN\ y_2 = NN_2(x_1, x_2, \dots, x_m) \\ . \\ . \\ . \\ Rule\ k: IF\ x = (x_1, x_2, \dots, x_m)\ is\ C_k, THEN\ y_k = NN_k(x_1, x_2, \dots, x_m) \end{gathered}$$

Where C_{1-k} denote the existing clusters. The final estimated values can be delivered through Equation 8 (Takagi and Hayashi 1991).

$$y_i^* = \frac{\sum_{s=1}^{k} W_s(x_i) \cdot y_s(x_i)}{\sum_{s=1}^{k} W_s(x_i)} \qquad \text{Eq. (8)}$$

Where k is the number of clusters, x_i is the input vector, W_s is the membership value, and y_s is the output of s^{th} consequent ANN.

This innovative fuzzification approach automatically considers the interdependence of the input variables, a capacity which has been missing in the conventional fuzzification process. This technique creates a (Neural-Network-Driven) NN-driven hyper-surface membership function, which combines all 2D membership functions of different

variables and results in a single multi-dimensional membership function. Membership functions dealing with the dependent variables must be curved hyper-surfaces with an axis representing the membership value of all variables. In this fashion, changes in one variable can alter the membership values of the others. The creation of these membership functions is possible only through the application of fuzzy clustering algorithms (Takagi and Hayashi 1991).

The main limitations of this model are:

1) Although the accuracy of its performance is highly sensitive to the number of clusters, and thus to the number of the consequent ANNs, no optimization has been performed to attain the optimum number of clusters;
2) The resulting NN_{mem} from the hard clustering algorithm provides a fuzziness that cannot be controlled or regulated. It is because the supervised part of the learning process is not flexible enough and only accepts the values 0 and 1. Consequently, the membership functions and the fuzziness of each point cannot be controlled. In other words, the system treats all the points in the same cluster indiscriminately and does not consider the distance from the centroids in the decision-making.

2.4.4 Genetic Algorithm (GA)

GA, developed by John Holland in 1975, is considered an AI-based optimization method that mimics the mechanism of natural evolution. In other words, GA solves problems with an evolutionary approach in which the best solutions are selected to produce the

potentially better solutions in future. Thus, the final solution can be named as the survivor (winner) over the whole evolution procedure.

In GA, an initial population of randomly generated individuals is produced and it evolves toward better generations by altering and mutating the properties of the population. The performance or suitability of each member of the population is ranked based on some fitness criteria. A fitness criterion provides a basis for the selection and migration to next generation. This algorithm, as the most well-known EA, involves the following main steps (Haupt and Haupt 2004):

1) Generate initial population (Initialization)
2) Evaluate fitness of population
3) Selection
4) Crossover
5) Mutation
6) Generate new population and evaluate fitness.

2.4.4.1 Coding

Coding is the act of assigning parameters and values to genes. Real coding and binary coding are two main types of genetic coding. Binary coding was the original type of coding used in GA. The chromosome is expressed by a row of 0 and 1 genes. These binary codes are representatives for the real form of the genes converted through a pre-identified mapping relationship. Thus, the search area turns into a binary domain and all genetic operators work with these binary values. Then, after the reproduction of each

generation, binary codes are decoded to their real form to evaluate the fitness function. On the other hand, real coding puts parameters as they are in the chromosomes without any conversion. Obviously, real coding is more efficient than binary approach in continuous optimization problems.

2.4.4.2 Operators

In a GA, a set of properties, i.e. genes, are assembled to make a candidate solution, i.e. chromosome. Based on the nature of the problem, firstly, a specific number of chromosomes, either randomly or biased over the more probable areas of optimal solution, is generated. After the selection of the fittest candidates in one generation, these better solutions will breed their offspring via selection, crossover and mutation operators.

2.4.4.3 Selection

Although there are different methods of selection between parent solutions, a generic selection procedure can be performed through following steps:

1) Each individual is evaluated with the fitness function and the fitness value is then normalized;
2) All individuals are sorted in descending order;
3) The cumulative fitness value of each individual is calculated, which is the fitness value of the individual plus all former fitness values of the individuals in the ranking;
4) A random number, between 0 to 1, is generated;

5) The first cumulative fitness value bigger than the generated random number is the selected parent.

This procedure is repeated until an agreed number of individuals are selected for the process of reproduction.

According to another type of selection called elitism, the best members of a generation are kept unchanged in the following generation. It usually works along with other selection techniques (Davis 1991).

2.4.4.4 Crossover

Crossover is the act of hiring more than one parent solution and reproducing a child based on a mixture of their properties. There are a plenty of crossover techniques, among which the one-point crossover can be pointed out as the most common and basic one. In this technique, a single point is located in both parent chromosomes. Then, all genes beyond this point are exchanged among the parents. The procedure is schematically illustrated in Figure 2-11.

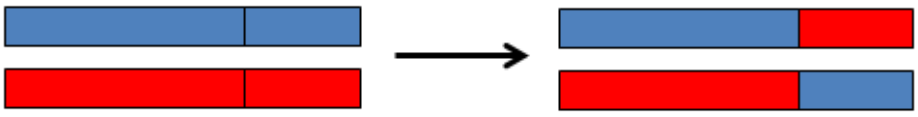


Figure 2-11: One-Point Crossover

2.4.4.5 Mutation

Mutation is the process of altering the values of one or more genes in one chromosome. This mechanism can change one solution to an entirely different one, and thus allow a more comprehensive and diverse search in the domain. Mutation enables a better genetic search through avoiding the stagnating at the local optima. Mutation is implemented according to a user-defined rate of mutation. If this rate is too high, the number of mutated chromosomes will be extensively high and, subsequently, a directed search will be degraded to a random search.

There are different types of mutation, each of which is suitable for a specific type of chromosome. The following mutation algorithms give us a clear understanding about the mutation process:

Flip Bit: This mutation selects a random gene and flips the bit. For example, (1 1 1 0 1 0 1) is reformed to (1 1 1 0 1 1 1). Bit string only applies to integer genes.

Boundary: This technique mutates the randomly selected gene of the string into either a lower or upper bound of the chosen gene. This can be used for integer and float genes.

Non-Uniform: The rate of mutation goes closer to 0 by advancing through the generations. In the early stages, it grants diversity to the population and thus prevents the search from stagnating. However as the algorithm moves toward the end, the rate of mutation goes down in order to enable a more delicate fine-tuning of the solution. This algorithm, also, can be used for both float and integer problems.

Uniform: First, the lower and upper bounds for each gene are defined. Then, a random gene is selected to adopt a random value between these user-defined bounds. The random value is the substitution for the original gene value. This algorithm is also suitable for both float and integer problems.

Gaussian: This mutation algorithm selects a random gene and substitute a unit Gaussian distributed random value with specified bounds instead.

2.5 EXPLORATORY DATA MINING AND STATISTICAL DATA ANALYSIS

Data analysis is the task of applying statistical or logical techniques for inspecting, cleaning and modeling data sets. This process provides a platform for retrieving useful information, making conclusions and facilitating decision making.

Data mining, as a data analysis technique, is mainly used for modeling and knowledge discovery purposes. From the statistical point of view, data analysis can be categorized into descriptive statistics, Exploratory Data Analysis (EDA), and Confirmatory Data Analysis (CDA). EDA tries to discover the new characteristics of data, while CDA evaluates the correctness of the existing hypotheses.

One major tasks of exploratory data mining and statistical data analysis is clustering which is used in the data processing stages of different areas such as, image processing, machine learning, information retrieval and computer simulation. The following section elaborates the literature related to clustering process.

2.5.1 Clustering

Data clustering can be regarded as the most well-known and prevalent EDA technique (Beringer and Hüllermeier 2006). It is the process of organizing a dataset into different groups, such that the members of the same cluster have more similar attributes compared to those of other clusters. K-Means (MacQueen 1967, Hugo Steinhaus 1957, Stuart Lloyd 1982) and Fuzzy C-Means (FCM) (Dunn 1973, Bezdek 1981) clustering can be considered as the dominant algorithms in both theoretical and practical applications of data mining.

K-Means partitions the data in such way that each point belongs only to one cluster. This method is a well-known member of a big category of clustering algorithms called Hard Clustering. Hard clusters are fully separated subsets of the data space that do not have any overlaps with each other.

On the contrary, FCM possesses a soft approach for reporting the memberships to different clusters. FCM is the dominant type of Fuzzy Clustering algorithms, which allows data points to be members of more than one cluster. Fuzzy clustering allows overlaps between clusters and signifies the extent to which data points are members of different clusters using an index known as “degree of membership”. Each data point holds a degree of membership to every cluster in the data set, with the summation of all its degrees of membership being 1. Fuzzy clustering accepts degrees of membership ranging from 0 to 1, while hard clustering only accepts crisp values of 0 and 1.

2.5.2 Optimal Number of Clusters

Many of the clustering algorithms, including K-Means and FCM, are based on the a priori knowledge of the number of clusters, and therefore a user defined number of clusters are required for the algorithm to run. This creates the challenge of determining the optimum number of clusters. Figure 2-12 shows how an improper determination of the number of clusters, Figure 2-12 (a), can noticeably distort the knowledge gained from the clustering procedure.

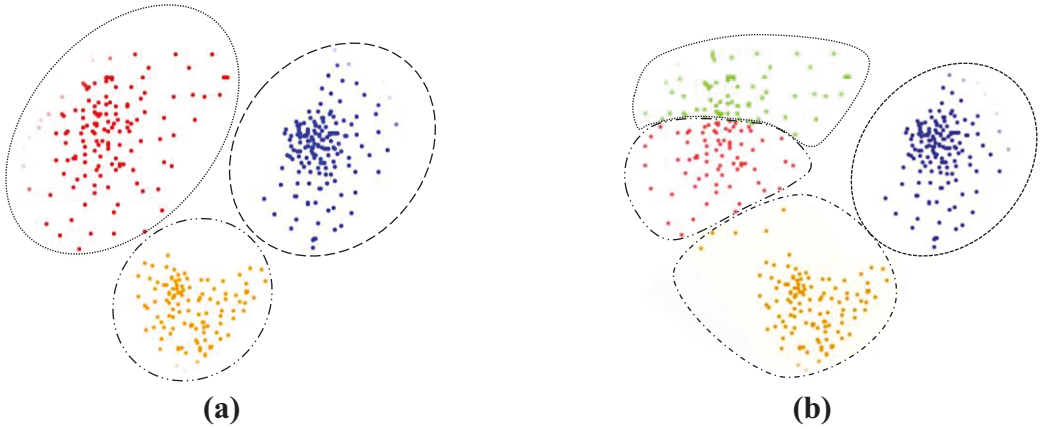


Figure 2-12: An Example of the (A) Improper and (B) Proper Determination of the Number of Clusters

Many methods were proposed to find the optimum number of clusters in a data set. However, most of them are solely based on the quantity of the data points and do not consider their distribution pattern. The rule of thumb, used in some of the existing methods, is that the number of clusters equals the square root of the number of data points divided by two (Mardia et al. 1979). Another method known as elbow method uses the number of clusters beyond which no considerable extra information can be attained.

In this method, the performance is plotted against the number of clusters and the answer is located at the elbow of the graph, where the slope of the plot obviously changes. However, this "elbow" is not always clearly visible.

Mountain clustering, proposed by Yager and Filev (1994), was an improved version of the earlier clustering methods. This heuristic technique is based on the density of data points. It applies a mountain function, i.e. density function, to the customized gridding of data space in order to find the grid point with the highest density value as the center of the first cluster, Figure 2-13 (a). This method continues by destructing the effect of each cluster mountain function to find the next greatest density value, Figure2-13 (b).While this approach is primarily considered as a stand-alone clustering technique, it can also function as a tool to obtain the initial number of clusters for other more complex techniques (Yager and Filev 1994). However, as the problem's dimension grows, so does the computations for evaluating all grid points, a problem known as the "curse of dimensionality" (Bellman 1961).

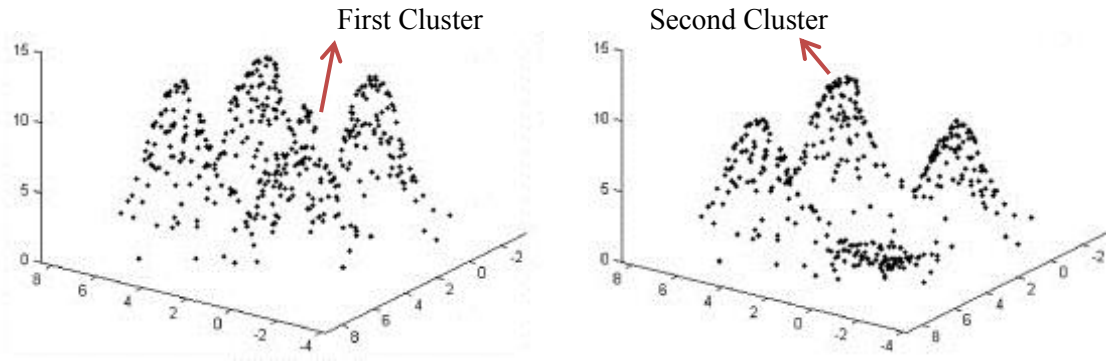


Figure 2-13: (A) Selecting the Highest Density Value as the First Cluster, And (B) Destructing the Effect of the First Cluster's Mountain Function

Chiu (1994) presented Subtractive Clustering to mitigate this problem. Subtractive Clustering only deems the data points as candidates for the center of clusters. In this way, computational complexity and effort grows proportionally to the size of the problem instead of its dimension (Hammouda and Karray 2000). However, this technique has some parameters, such as influence range, squash factor, accept ratio and reject ratio, which highly affects the results of the clustering. Thus, these parameters should be selected based on the inherent characteristics of the target data, which makes the analysis subjective. Table 2-1 compares different methods of clustering in terms of their strength and weakness.

Table 2-1: Clustering Methods

Method	Strength	Weakness
K-Means	• Robust partitioning • Accurate cluster centers	• Initial value must be provided
FCM	• Robust partitioning • Accurate cluster centers • Fuzzy approach	• Initial value must be provided
Subtractive	• No initial guess for number of clusters is needed • Density based method • Low computational complexity • Computational complexity proportional to problem dimension	• Inaccurate cluster centers • No basis for selecting the initial parameters, such as effective range and squash factor
Mountain	• No initial guess for number of clusters is needed • Density based method	• High computational complexity in multi- dimensional data spaces, Computational complexity proportional to data set dimension • Inaccurate cluster centers are proposed

2.5.3 Clustering Validation Index

It is of a great necessity to compare different sets of clusters and evaluate the "goodness" of the clustering for the validation purposes. Clustering validation is the act of determining how well an algorithm can recognize the underlying patterns of data. Usually in 2D and 3D data spaces, visualization is used to empirically validate the clustering. However, in case of the large multidimensional data spaces, inapplicability of an effective visualization leads to the application of more formal approaches (Kovács et al. 2005).

There are three main approaches to evaluate the quality of clustering as follows (Halkidi et al. 2001):

- **Internal Validation:** The clusters are assessed based on some data-oriented statistical metrics, which evaluates the inherent features of the data;
- **External Validation:** The clusters are assessed based on some intuition-oriented statistical metrics, which uses the user-defined intuitions;
- **Relative Validation:** The clusters are assessed based on the comparison between different clustering methods, which result from different clustering parameters.

2.5.3.1 Internal Validation

Internal validation is an approach to evaluate the clustered dataset based on the inherent features of the data itself (Halkidi et al. 2001). Different internal validity indices have been proposed as an assessment metrics for the compactness and separation among the

data distribution, such as Davies Bouldin index and Dunn index (Kovács et al. 2005). While the earlier examines the level of closeness of the members of each cluster to one another, the latter evaluates the extent to which the clusters are widely spaced (Berry and Linoff 1997). The main drawback of internal validation is that supreme values of an internal index do not necessarily lead us to the best information retrieval applications (Manning et al. 2008).

2.6 INTEGRATION OF LINGUISTIC TERMS AND CRISP VALUES

With the fuzzy and probabilistic approaches being both applicable to address uncertainties, it is usually the case that a combination of linguistic terms (fuzzy numbers) and crisp values needs to be considered for the modeling (Guyonnet et al. 2002). These cases are very likely in the construction simulation, where limited data is provided for many of the factors involved in the project (Sadeghi et al. 2010). Many researchers tried to find a way to estimate the output of these generalized models using both types of input values. Wonneberger et al. (1995) transformed all possibilities into probabilities in order to convert the problem to a pure probabilistic case (Figure 2-14)(Wonneberger et al. 1995). However, possibility and probability refer to uncertainty with different approaches that seem not to be interchangeable.

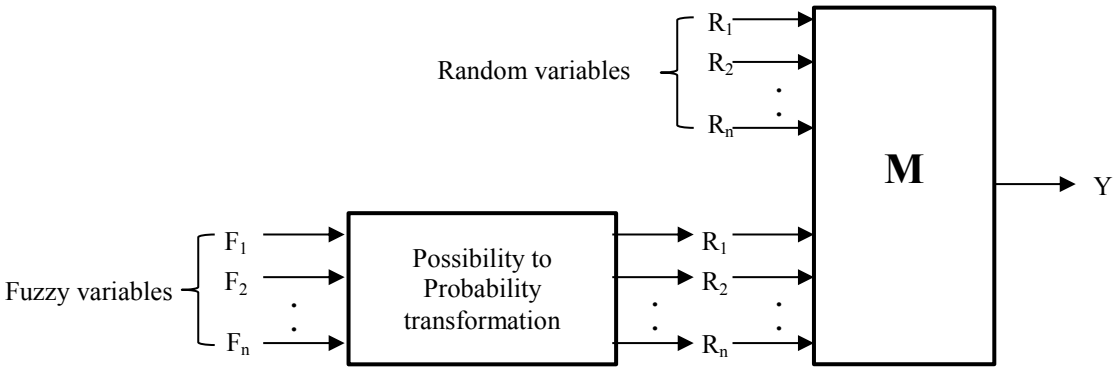


Figure 2-14: Transformation-Based Approach of Simulation (Sadeghi Et Al. 2010)

In 2003, Guyonnet et al. (2003) proposed a hybrid simulation approach as a solution for modeling both crisp inputs and linguistic terms. Hybrid method is able to handle the diversity of variables without any transformation processes from one to another. In this technique, each linguistic term is explained as a fuzzy number. Thus, the output can be expressed via an output membership function constructed by the alpha-cut technique.

The schematic hierarchy of Hybrid Approach is presented in Figure 2-15 and Figure 2-16. Figure 2-15 illustrates the model M that is fed by random numbers $(p_1, p_2, \dots, p_n)$, as crisp values, and fuzzy numbers $(F_1, F_2, \dots, F_m)$, as linguistic terms. The random numbers are generated by the probability distributions.

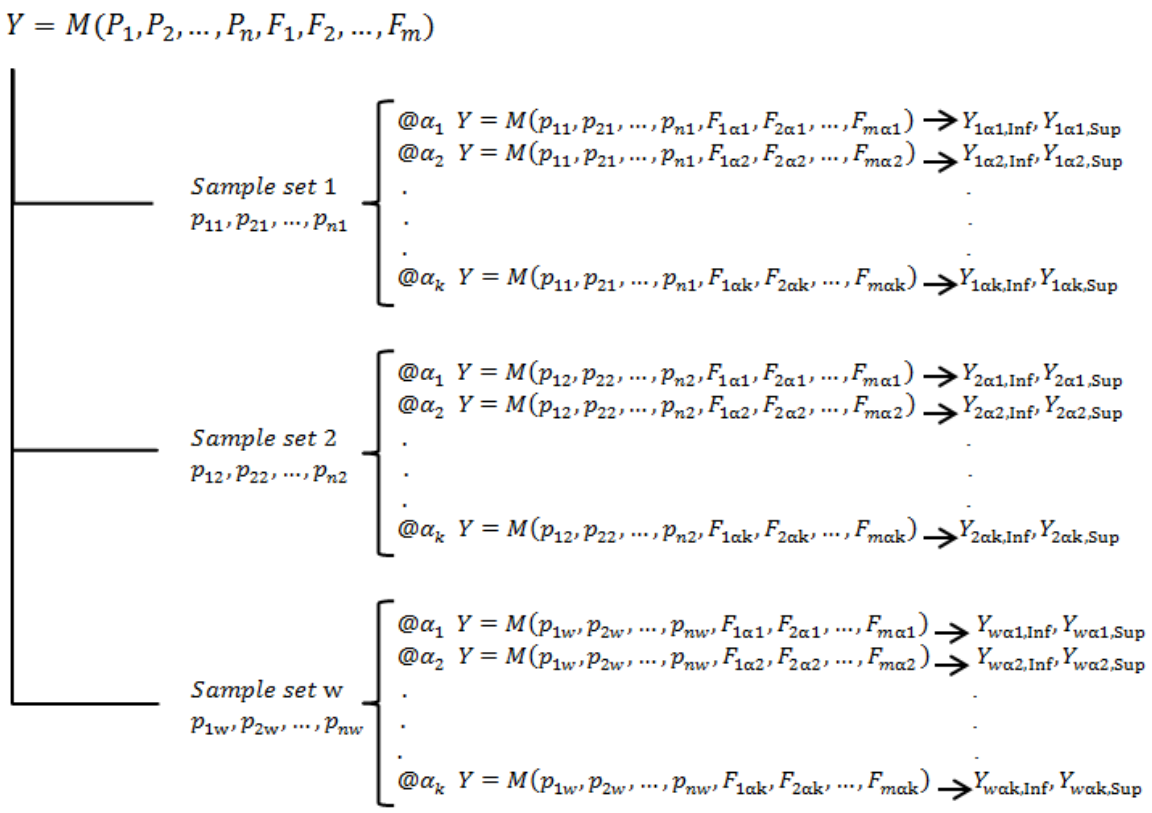


Figure 2-15: "Hybrid Approach" of Simulation (Adopted from Sadeghi Et Al. 2010)

To determine the output (Y) of the model, different sets of random numbers (w) are generated and assigned to crisp input variables, Figure 2-16 (a). In parallel, alpha-cut is performed for the fuzzy inputs at different levels of α, Figure 2-16 (b). In this manner, for each level of alpha and every set of random numbers, the model generates two outputs. $Y_{i\alpha j,INF}$ and $Y_{i\alpha j,SUP}$ denote the minimum and maximum outputs of the model, respectively. Consequently, by keeping the crisp values as fixed numbers and performing alpha cut at different levels of alpha, the output fuzzy set is gradually constructed by the minimum and maximum outputs of the model, Figure 2-16 (c).

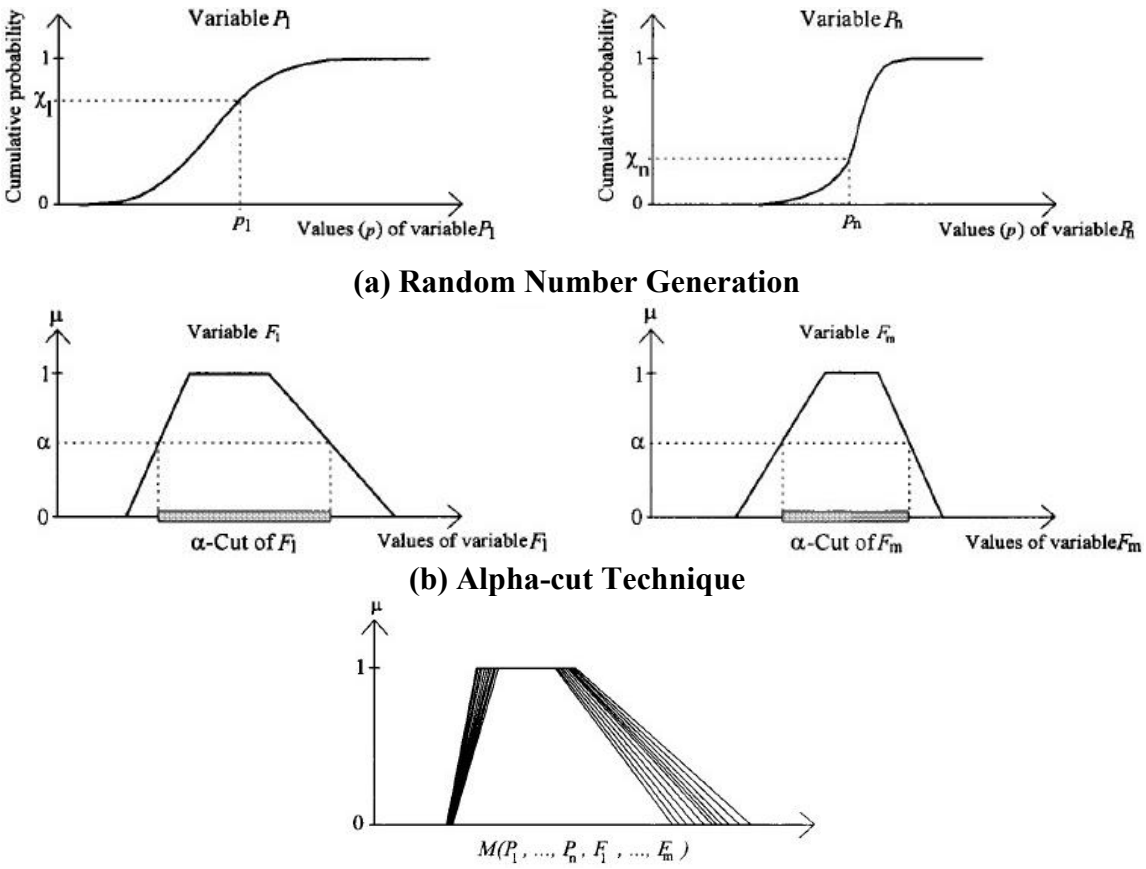


Figure 2-16: Illustration of the hybrid approach of simulation (Guyonnet et al. 2002)

The first model that applied this technique was a monotonic model which does not require any optimization algorithm to find the maximum and minimum outputs of the model. However, many of the developed models are not monotonic, and thus their extremum outputs cannot be easily retrieved without an appropriate optimization process.

This method is considerably important since it enables the model to use Monte Carlo simulation while it has some fuzzy inputs.

In case the possibility of an output less than a certain threshold is desired, it can be calculated, based on the possibility theory, for the fuzzy set F, membership function $\mu_F(x)$ and threshold T through Equation 9 (Sadeghi et al. 2010):

$$\prod(\mathrm{F} > \mathrm{T}) = \mathrm{Sup}_{\mathrm{x}>\mathrm{T}}\mu_F(\mathrm{x}) \qquad \text{Eq. (9)}$$

2.7 SUMMARY AND LIMITATIONS OF PREVIOUS LITERATURE

This chapter covered a wide continuum of topics to present an overview of the existing approaches to the application of soft computing techniques for the estimation of productivity in the construction industry. Since productivity rate is the best index for the progress assessment of construction projects, the productivity estimation has always been momentous for both academia and industry. As a result, practitioners and researchers are always trying to improve the productivity rate. This will not be materialized unless a comprehensive study of the factors affecting the construction productivity rate is conducted. As stated at the beginning of this chapter, most research in construction simulation has mainly focused on the simulation modeling without placing much emphasis on the study of the qualitative and quantitative factors that affect the features of each process, such as time and productivity. Weather conditions, managerial factors and work methods can be regarded as some of these factors. On the other hand, the limited research that has addressed these issues has solely concentrated on the study of the impact of some specific factors, without delving deep into the structure and evaluation of the model, and thus provides no generic solution or conceptual models.

Compared to other alternatives, AI is decisively dominant owing to its ability to mimic the intelligent behavior of human. AI facilitates the establishment of forecasting models that can predict the expected output of an activity through applying pattern recognition to historical data. Although the self-learning ability of ANN makes it an appropriate choice, its inability to explain the quality of input-output mapping or, in better words, reasoning process, renders it a black box. On the other hand, fuzzy reasoning, as another option, provides us with a systematic reasoning that is more tangible for human logic and intuition. However, the learning process of fuzzy models requires self-regulating techniques from other areas. Although neuro-fuzzy systems, where fuzzy reasoning and ANN are integrated and the shortcomings of one system are offset by the strengths of the other, seem to be an effective solution, there is no notable application of these types of models in the construction modeling. Moreover, there is a very strong bond between neuro-fuzzy systems and clustering algorithms, insofar as most of the neuro-fuzzy systems take advantage of clustering algorithms in their structure. Despite the fact that the main purpose of these cluster-aided systems is to provide an accurate and interpretable model, their performances are very sensitive to the proper definition of their constituent design parameters. It is shown that the improper determination of the number of clusters, as one of these parameters, can noticeably distort the fitness of such models.

At the other end of the pendulum, most of the proposed forecasting models only deal with the crisp input values. However, there are some cases where a combination of crisp values and linguistic terms are desired. These problems demand a system that works with both types of uncertainties (probability and possibility). It seems that because of the

dissimilar natures of probability and possibility, transformative methods, which transform one form to another, do not work properly. On the other hand, “Hybrid Approach” of modeling, which compared to other approaches is more successful in this area, cannot be easily incorporated with non-monotonic models without an appropriate optimization process.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 CHAPTER OVERVIEW

The generic flow diagram of the present research is presented in Figure 3-1. This research starts with a literature review on the area of soft computing and its application in the construction simulation. The review covered subjects such as the-state-of-the-art in construction simulation, explorative data mining and statistical data analysis, hybrid intelligent systems and genetic optimization. Following the literature review, the model development phase is dedicated to proposing a model that addresses all the shortcomings and weaknesses identified in the literature review and explained in the problem statement. Model development, in turn, consists of three main phases as follows: 1) the implementation of the modified NNDFR system, 2) the fine-tuning of the system, and 3) the implementation of a Hybrid Approach of modeling. Model development phase is then followed by a data collection phase, which is related to the data gathering and data analysis processes. In order to verify and validate the system, a case study will be conducted and, in this context, the performance of the proposed system will be compared to other existing systems. Finally, this research will be finalized with some conclusions and recommendations and also some proposed research areas for the future.

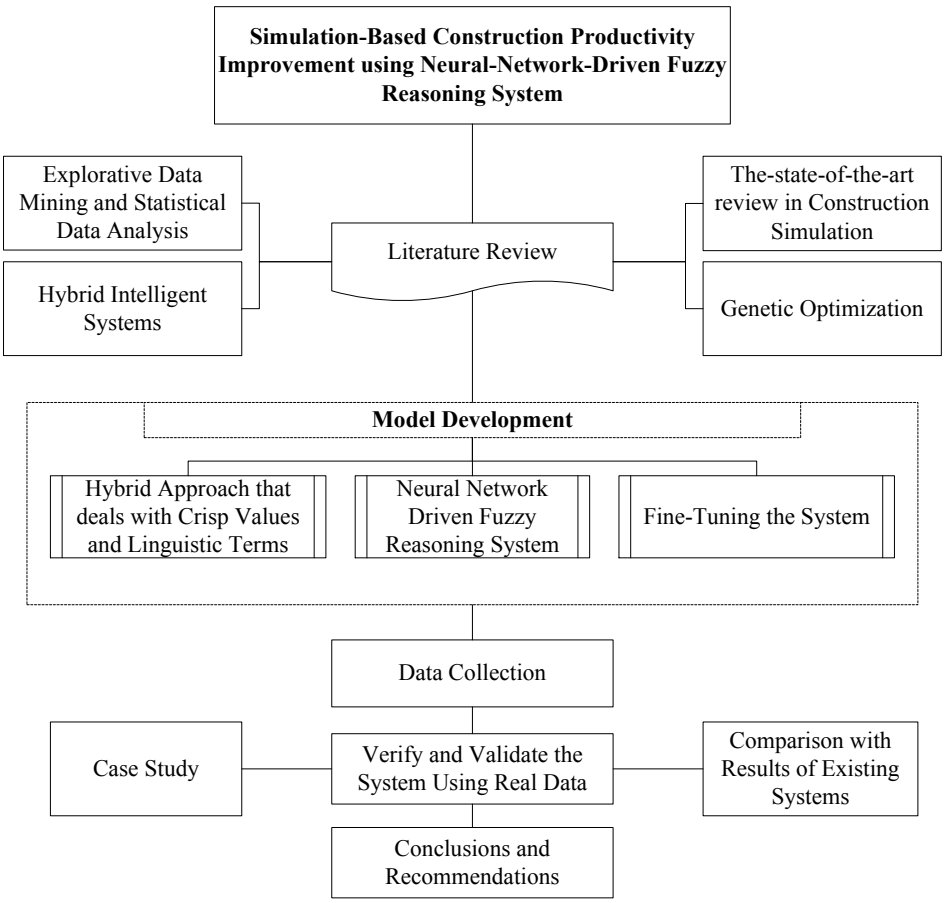


Figure 3-1: Flowchart of the Research Methodology

3.2 LITERATURE REVIEW

The literature review was presented in Chapter 2. It comprehensively covered the major research areas related to the application of soft computing in construction simulation. As shown in Figure 3-1, the literature review consists of four sub-sections as follows:

1) Explorative data mining and statistical data analysis;

2) Hybrid intelligent systems;

3) Review of the-state-of-the-art in construction simulation;

4) Genetic optimization.

The concepts, methods and applications of different approaches in each subject are elaborately discussed and the merits and shortcomings of each method as compared to its counterparts are presented.

3.3 MODEL DEVELOPMENT

This section aims at providing a detailed explanation of the model development process. The flowchart of techniques and actions that are required to implement the proposed framework is illustrated in Figures 3-2 and 3-3. This framework consists of three main phases: 1) the implementation of the modified NNDFR system, 2) the fine-tuning of the system, and 3) the implementation of a Hybrid Approach of modeling. The flow diagrams of the first and the second phases are provided in Figure 3-2. Section 3.3.1 presents a comprehensive description of the structure of the proposed model. Section 3.3.2 explores the fine-tuning of the model and elaborates the optimization process, which is performed to find the fittest model parameters. Third phase of the system development is elaborately discussed in Section 3.3.3 and its flow diagram is shown in Figure 3-3.

As shown in Figure 3-2, GA generates a combination of parameters, which controls the number of clusters and fuzziness of the membership functions. Then, the structure of the modified NNDFR is formed in the training part of the system based on the inherent features of the data along with the current values of parameters. The latter process will be elaborated later in this chapter. The NNDFR Testing section provides the pre-defined model with a testing sample from the data and computes the estimated outputs. It, then,

measures the accuracy of the estimation by comparing the result with the actual targets. Subsequently, this loop reiterates in order to check the model for other values of parameters. Finally, the parameters corresponding to the best results are considered to be the optimum choice.

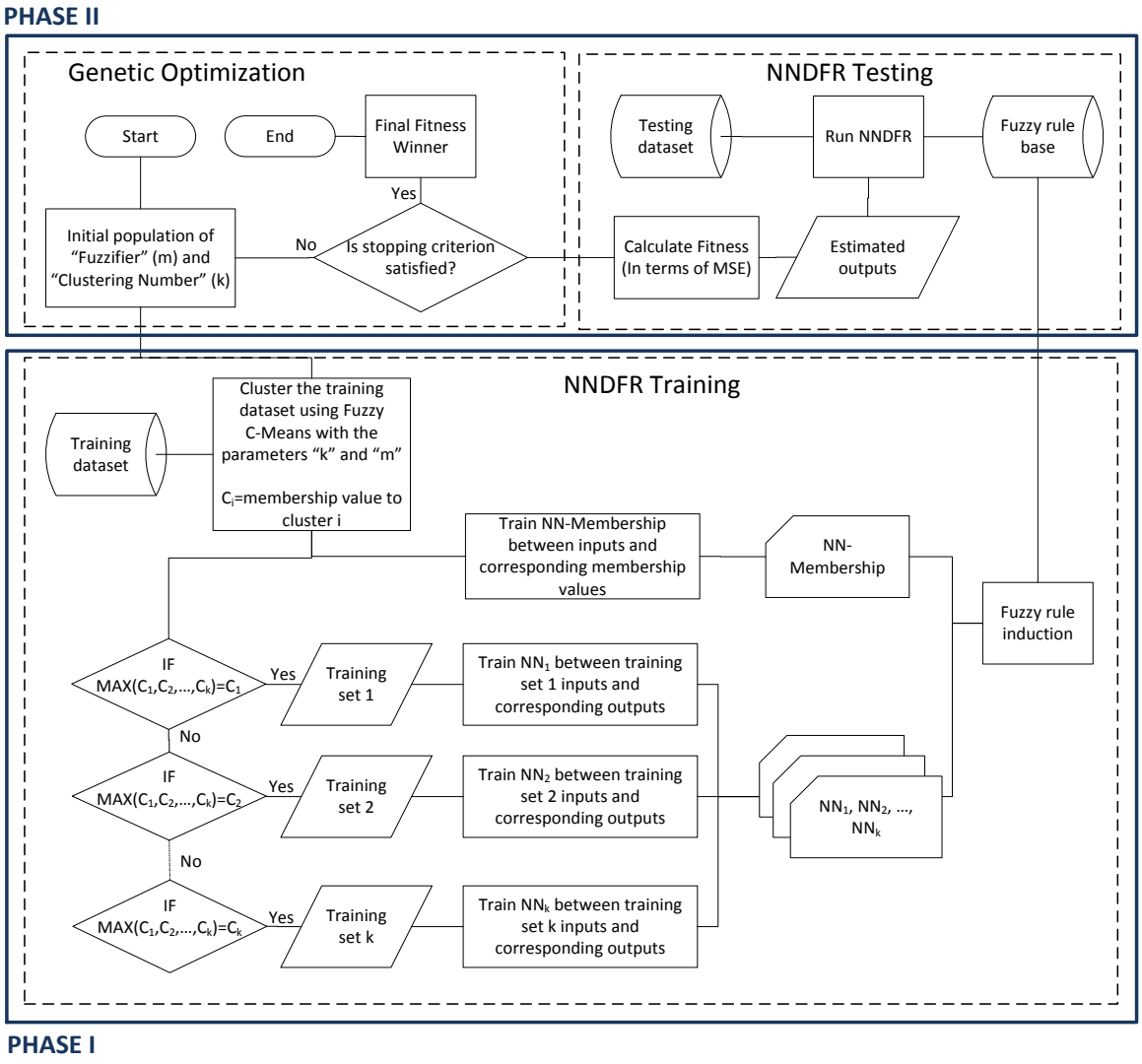


Figure 3-2: Model Development Flowchart (Phases I and II)

According to Figure 3-3, Hybrid Approach starts with distinguishing the crisp and fuzzy variables. Next, alpha-cut is performed at each level of alpha, while other variables are

given to model as fixed values. The model used in this part is the genetically optimized system from the previous section.

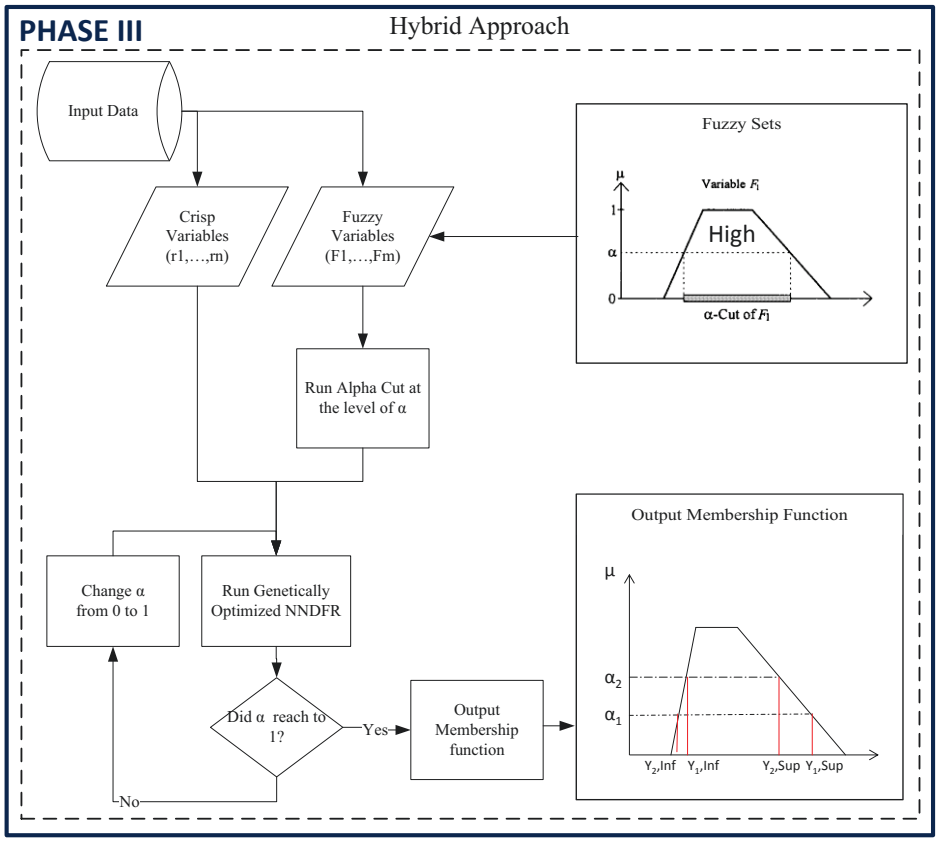


Figure 3-3: Model Development Flowchart (Phase III)

3.3.1 NNDFR Structure

3.3.1.1 K-Means Clustering

As explained elaborately in section 2.4.3.2, the conventional structure of NNDFR utilizes a hard clustering algorithm, such as K-Means, in order to determine and assign the membership values. This clustering mechanism controls the system to select and follow

one reasoning strategy. By presenting the K-Means algorithm, this section provides a better understanding of this phase.

K-means, as one of the simplest unsupervised clustering algorithms, partitions the data space in hard clusters. This iterative algorithm locates centroids via minimizing the objective function shown in Equation 10 (Matteucci 2006).

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \quad \text{Eq. (10)}$$

Where $x_i^{(j)}$ is the i^{th} measured data, c_j is the center of the j^{th} cluster, and ||*|| is a kind of distance between the d-dimensional vector $x_i^{(j)}$ and the d-dimensional vector c_j. The objective function can be minimized through the following steps (Matteucci 2006):

1) Randomly place K points representing initial centroids in the data space;
2) Assign each data point to the cluster that has closest centroid;
3) Calculate the revised position of each centroid;
4) If the positions of centroids didn't change go to the next step, else to the step 2;
5) End.

3.3.1.2 Hyper-Surface (Multi-dimensional) Membership Functions

The definite problem with the conventional fuzzification process is that it cannot deal with the problem of variables interdependency. Since each 2D membership function, that includes crisp and membership value, is designed separately, any changes in the value of one variable cannot alter the membership value of the other variable. For example,

humidity and temperature cannot be considered as completely independent factors. For such cases, a three dimensional membership function, e.g. two axes for the temperature and humidity and one for the degree of membership, is required. However, there is no way to tune this type of membership functions with intuition and experience.

The solution to above-mentioned problem is to create hyper-surface membership functions. As described in the second step of the NNDFR design procedure, a hyper-surface membership function can be produced by means of the pattern recognition ability of ANN. This innovative fuzzification approach automatically considers the interdependence of input variables, and thus mitigates the definite weakness of the conventional fuzzification process. In other words, this NN-Driven hyper-surface membership function combines all 2D membership functions of different variables and presents a single multi-dimensional membership function. The concept of hyper-surface membership function can be more clearly explained through the following example. Figure 3-4 plots a set of data points that can be visually categorized as two well-separated clusters. These data points are then clustered to two groups by K-Means algorithm. Table 3-1 tabulates a sample of coordinates and cluster assignment vectors corresponding to each input.

Next, an ANN with two inputs, i.e. X_1 and X_2, and two outputs, representing the associated cluster indices, is trained. This ANN simulates the clustering procedure through a pattern recognition mechanism. In other words, a surface fitting is done by the ANN. These surfaces are then visualized by feeding a large quantity of random numbers to the model in the desired domain and plotting the results. Figure 3-5 shows the

membership surfaces constructed by connecting all the ANN outputs against their input coordinates. The red surface represents the membership values related to the first cluster, i.e. the group of red points in Figure 3-4, and the blue surface represents the second cluster, i.e. blue points in Figure 3-4.

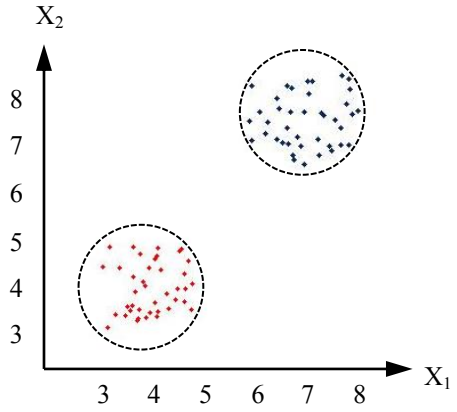


Figure 3-4: Two Well-Separated Clusters

Table 3-1: A Sample of Data Fed to Membership ANN

Input 1 (X1)	Input 2 (X2)	Output 1 (C1)	Output 2 (C2)
3.61	3.84	1	0
3.85	3.90	1	0
3.94	3.77	1	0
3.98	3.21	1	0
3.90	3.63	1	0
3.03	3.71	1	0
3.08	3.00	1	0
7.53	7.88	0	1
7.16	7.46	0	1
7.02	7.27	0	1
7.00	7.47	0	1
7.50	7.03	0	1
7.92	7.23	0	1
7.95	7.54	0	1

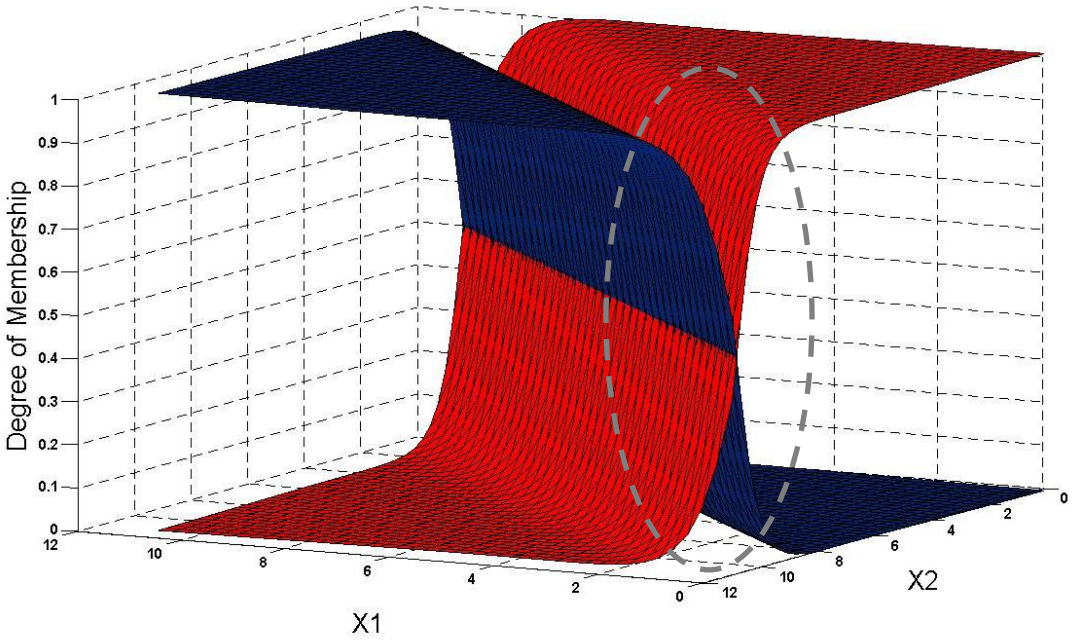


Figure 3-5: 3D Membership Function Generated Via K-Means Algorithm

3.3.1.3 Substituting FCM for K-Means

The membership ANN, obtained from the results of the hard clustering algorithm, creates a degree of fuzziness in the overlapping area of the clusters. This transition zone is depicted by the gray ellipse in Figure 3-5. In fact, the transition zone is the result of the interpolation done by ANN between the two hard clusters. However, the resulted fuzziness in this part is not under control and cannot be regulated. It is only created because the supervised part of the learning process does not have enough flexibility and it only accepts the values 0 and 1. Thus, we cannot decide about the shape of membership functions and modify the fuzziness assigned to each point. Additionally, such a hard membership ANN does not allow the system respond to the deviations within a cluster. In other words, the system treats all the points in the same cluster similarly and does not

consider the distance from the centroids in the decision-making. Needless to say, this can affect the performance of the models that use these hyper-surface membership functions.

In order to overcome the limitations of a typical multi-dimensional membership function, a fuzzy clustering algorithm can be implemented in this structure. FCM, as a fuzzy version of K-Means, can serve towards this end. Applying FCM algorithm gives a fuzzy approach to the membership assignment procedure. As will be explained in the following section, FCM formula has an initial parameter, which regulates the relative weights of the membership values to all existing clusters and consequently controls the fuzziness of the subsequent membership functions. In this way, the model is able to compute the final estimated outcome based on an optimized contribution of all the consequent relationships**.** Next section describes FCM algorithm.

FCM Algorithm

FCM allows each data point to belong to more than one cluster. This iterative algorithm is performed through the minimization of the objective function shown in Equation 11 (Matteucci 2006):

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \qquad \text{Eq. (11)}$$

Where u_{ij} is the degree of membership of x_i to the cluster j, m is any real number greater than 1, x_i is the i^{th} measured data, c_j is the center of the j^{th} cluster, and ||*|| is a type of

distance, among many others, between the d-dimensional vector x_i and d-dimensional vector c_j .

The iterations of above-mentioned objective function optimization proceed through updating u_{ij} and c_j in each step using Equations 12 and 13 (Matteucci 2006):

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \qquad \text{Eq. (12)}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \qquad \text{Eq. (13)}$$

The iterations will stop when the following condition is satisfied:

$$\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon \quad (0 < \varepsilon < 1)$$

In the above inequality, k is the iteration step.

Hyper-surface membership functions generated by FCM

According to the FCM algorithm, the parameter *m*, which is called fuzzifier or weighting exponent, can greatly influence the performance of the system. When the fuzzifier is close to 1, the result of FCM is identical to that of k-means. When the fuzzifier approaches infinity, each cluster is only assigned to its centroid and the rest of the points

will have a membership value of 0. Therefore, FCM regulates the fuzziness of the clusters by adjusting this weighting exponent. Figure 3-6 illustrates the membership ANN that is trained by the FCM algorithm with different fuzzifiers. In this example, the data points presented in Table 3-1 are clustered via FCM for two different fuzzifiers, namely m=1.9 and m=2.9. Table 3-2 presents the results of this phase for both values of fuzzifier. Once FCM is applied to the generated data points, membership ANN is trained between the (X_1, X_2) coordinates and the corresponding FCM membership outputs.

Table 3-2: The Data Generated By FCM and Fed to Membership ANN

Input 1 (X1)	Input 2(X2)	m=1.9		m=2.9	
		Output 1 (C1)	Output 2 (C2)	Output 1 (C1)	Output 2 (C2)
3.61	3.84	0.99713	0.00287	0.94030	0.05970
3.85	3.90	0.99356	0.00644	0.91521	0.08479
3.94	3.78	0.99426	0.00574	0.91955	0.08045
3.98	3.22	0.99517	0.00483	0.92620	0.07380
3.90	3.63	0.99689	0.00311	0.93867	0.06133
3.03	3.71	0.99438	0.00562	0.92024	0.07976
3.09	3.00	0.99355	0.00645	0.91612	0.08388
7.53	7.88	0.00253	0.99747	0.05603	0.94397
7.16	7.46	0.00237	0.99763	0.05394	0.94606
7.02	7.27	0.00656	0.99344	0.08466	0.91534
7.00	7.47	0.00567	0.99433	0.07953	0.92047
7.50	7.03	0.00404	0.99596	0.06812	0.93188
7.92	7.23	0.00378	0.99622	0.06639	0.93361
7.95	7.54	0.00289	0.99711	0.05911	0.94089

By simulating a large number of input points within this area and plotting the outputs against corresponding (X1, X2) coordinates, the surfaces shown in Figure 3-6 will appear. As shown, when m is very close to 1, FCM acts as K-Means, and with the exception of a narrow overlapping area between clusters, all data points take the membership value of almost 1 to one cluster and 0 to the others. It means that, mostly,

only one consequent ANN is in effect. However, when any higher values of m are used, only the center of each cluster is assigned to that cluster and corresponding consequent ANN. In this case, any deviation from the center of cluster will result in the activation of other consequent relationships.

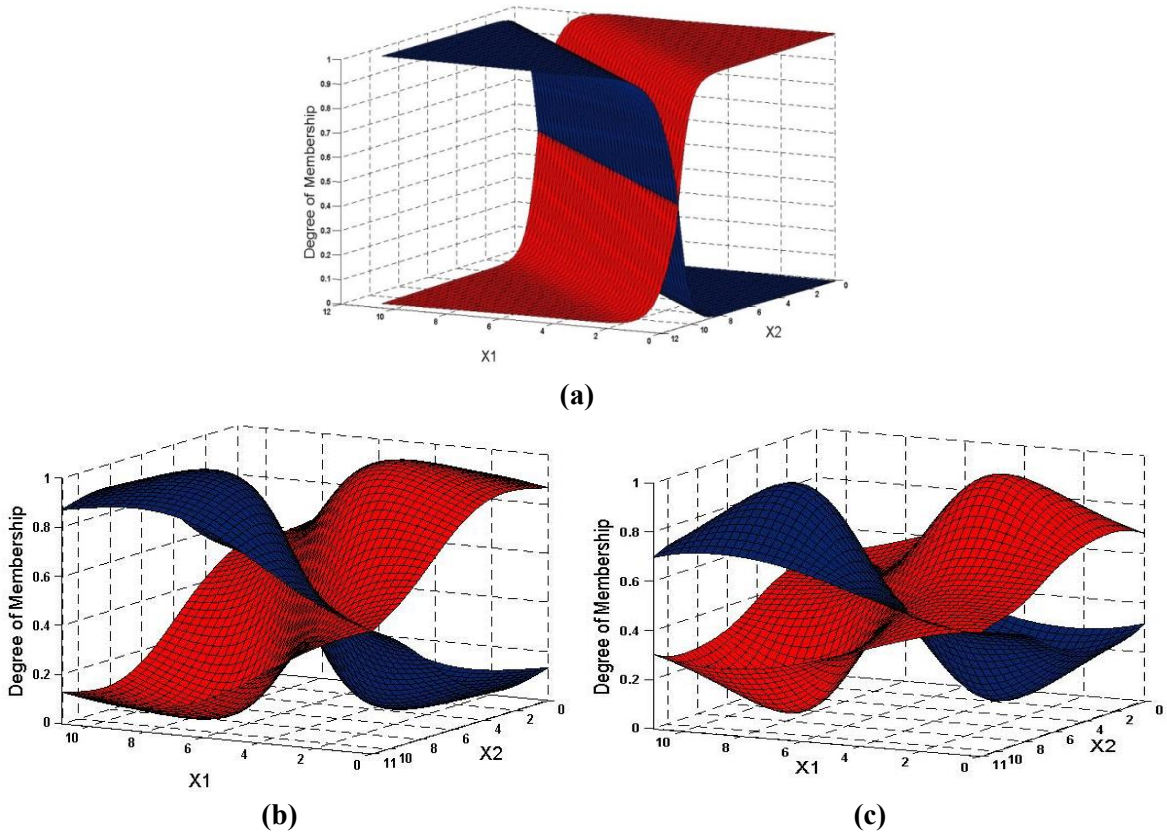


Figure 3-6: Illustration of Membership ANNs Trained by the FCM Algorithm for (A) M≈1, (B) M =1.9, And (C) M =2.9

3.3.1.4 Separate Training Sets

Due to the fuzzy nature of FCM, each data point belongs to all clusters, but with different degrees. However, consequent ANNs need separate training data sets. To mitigate this

problem, it is proposed to tag each data point to one cluster with the maximum membership value. For example, point A with the membership values of (0.15 0.2 0.65) will be separated as a member of NN_3 training data set.

3.3.1.5 Modified NNDFR

Figure 3-7 shows the schematic structure of the proposed modified NNDFR. There are two types of ANNs in this assembly. First, several clusters are defined using FCM, and a membership ANN is trained based on FCM result in order to automate this process. In parallel, consequent ANNs are connecting the input data space to target data space in different parts of the domain ($y_s = NN_s(x_i)$). The membership ANN generates some weights (W_s) for each consequent ANN. The final output is calculated through the weighted average of the output of the consequent ANNs.

3.3.1.6 Validation

The goal of this step is to evaluate the accuracy of the model in output estimation. There are many mathematical indices for measuring the error of prediction. This error appears as a result of deviation from the actual targets of the model.

Mean Square Error

Mean Square Error (MSE) measures the average of the squares of the errors. This index is calculated as shown in Equation 14.

$$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{E_i} - \mathrm{A_i})^2 \qquad \text{Eq. (14)}$$

Where *E* is the estimated output, *A* is the actual target and *n* is the number of testing data points. To put this index in perspective, the closer the MSE to 0, the more accurate the model.

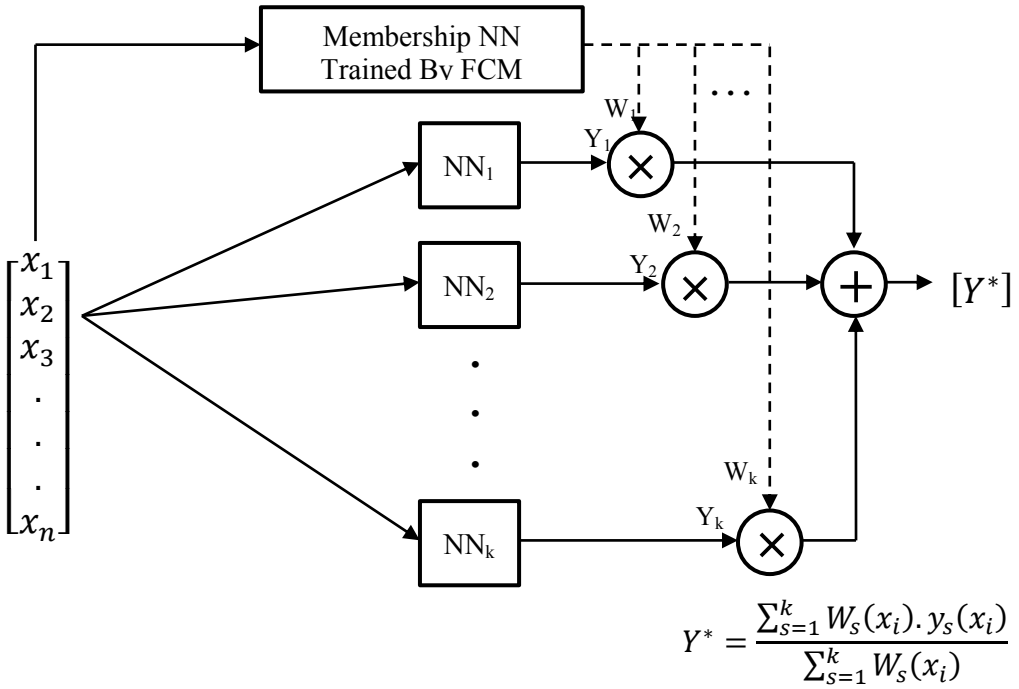


Figure 3-7: Schematic Structure of the Modified NNDFR

Average Invalidity Percentage (AIP)

This index was first proposed by Zayed and Halpin (2005). The closer AIP to 0, the fitter the model, and correspondingly the further it deviates from 0, the more inaccurate the model is. Average Validity Percentage (AVP) equals the subtraction of AIP from 100. Obviously, a model with an AVP of 100 is the fittest. Equations and present the formula of these two indices.

$$\mathrm{AIP} = \frac{\left(\sum_{i=1}^{n}\left|1-\left(\frac{E_i}{A_i}\right)\right|\right)\times 100}{n} \qquad \text{Eq. (15)}$$

$$\mathrm{AVP} = 100 - \mathrm{AIP} \qquad \text{Eq. (16)}$$

Where E is the estimated output, A is actual target and n is the number of testing data points.

3.3.1.7 Training Algorithm of ANN

One common problem during the ANN training is what is called over-fitting. It is when the calculated error of estimation for the training sample is very small but the error for a new set of testing data is too large. It could be stated that the network memorized the training data points and it is not able to generalize the prediction for new cases. One method to improve the quality of the generalization in a trained net is called regularization. Regularization involves applying another performance measurement other than the sum of the squared output errors, which is usually selected as the performance indicator. Bayesian Regularization algorithm, as a regularization method, combines squared errors and weights in a mathematical relationship and minimizes them in order to find the best combination that has the best generalization ability. Bayesian Regularization allows the network have smaller weights and biases, which will, in turn, result in less susceptibility to over-fitting (MacKay 1992).

A more detailed explanation of Bayesian Regularization algorithm is out of the scope of this research. All of the assembled ANNs in this model are trained based on this learning algorithm. This can highly improve the performance of model for future applications. The

command **`net.trainFcn = 'trainbr'`** changes the default learning function of the program to Bayesian Regularization.

3.3.2 Fine Tuning The model

3.3.1.8 Fuzzifier

Following the previous discussion on the impact of fuzzifier on membership functions, it is understood that the desirable shape of the membership function could be reached via fine-tuning the fuzzifier. However, there is still no robust theoretical way to find the optimal value for this exponent. It seems that the determination of the fittest value is highly dependent on the characteristics of the problem. Here, an optimization process in which the performance of NNDFR is set as the target function can deliver the optimal value for the fuzzifier. First, the accuracy of the estimation is calculated via an error index. Then, the selected optimization technique changes the values of fuzzifier within a pre-defined range and measures the error for each case. And finally, the scenario with the least performance error is considered the final solution.

3.3.2.1 Number of Clusters

The other parameter required for the clustering is an initial value for the number of clusters. Given that in NNDFR the number of clusters defines the number of rules, an improper determination of the initial value can significantly distort the fitness of the model. Thus, an optimization on this parameter can make a reasonable compromise between the complexity and efficiency of the model. Similar to the fuzzifier, this

parameter, too, needs to be subject to optimization, within a pre-defined range, in order to find the best possible structure for NNDFR.

3.3.2.2 Genetic Optimization

Conceivably, separate optimizations of these two parameters, i.e. the fuzzifier and the number of clusters, may not lead to a global optimum. Thus, the global optimization requires a concurrent consideration of both parameters, for which purpose the genetic algorithm can be used. In this optimization process, each individual, i.e. chromosome, consists of two parameters, i.e. genes, namely "fuzzifier" and "the number of clusters". Each member of the population assumes a combination of the parameters and accordingly forms the structure of NNDFR. All structures are trained with the same training data set and their finesses are evaluated comparatively, using the MSE of the NNDFR system as the evaluation metric. The GA iterates until the termination condition is satisfied. A specific number of generations can be used as the stopping criterion. In this case, once the maximum number of generations is reached, the highest ranked individual is adopted as the final solution. The final solution represents the best layout of the model based on the inherent features of the provided data. Figure 3-8 illustrates a schematic representation of a chromosome in this optimization process.

Individual (Chromosome)	
Number of Clusters	Fuzzifier

Figure 3-8: Schematic Representation of a Chromosome

The following sections describe the selection, crossover, and mutation techniques that are used in this framework.

Selection Algorithm

Before the next generation is reproduced, a method must be selected to choose suitable parents. The raw fitness values cannot be used by the selection function. Thus, a fitness scaling is required to transform these fitness values to some scaled values that signify the importance of each individual in the selection process. The selection function works in a way that chromosomes with the greater scaled values have a higher chance of selection.

(i) Rank Scaling

The range of the scaled values has an important role in the efficiency of the genetic optimization. If this range is too wide, chromosomes with the higher scaled values dominate the reproduction process and prevent the optimization from searching other areas of the solution domain. On the contrary, if the range is too narrow, the chance of selection is almost equal, and consequently the optimization will be time-consuming (Mathworks 2012). "Rank Scaling" is the most common fitness scaling function method that scales the individuals based on their position in the sorted list of individuals after the fitness evaluation. This method (1) scales an individual with the rank n to a value proportional to $\frac{1}{\sqrt{n}}$; and (2) scales the entire population's values such that the scaled values add up to the number of parents needed for reproduction

(ii) Stochastic Uniform

Stochastic uniform is a selection function that assumes a line with different sections, each of which holding a length proportional to the scaled value of that parent (Mathworks 2012). The algorithm, then, moves on the line with equal steps. At each step, the algorithm picks up the parent corresponding to the section it lands on. Figure 3-9 depicts this process clearly.

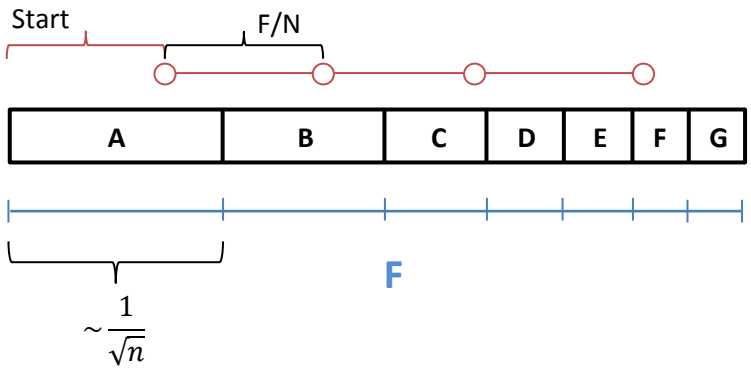


Figure 3-9: Stochastic Uniform Selection

F represents the total length of all scaled values, *n* represents the rank of each individual, and *N* represents the number of desired parents. Start is a random margin for the first selection.

(iii) Crossover Algorithm

The method employed for the crossover in this framework is called intermediate crossover. It reproduces children through a weighted combination of involved parents.

This method is only applicable to real populations. The offspring is produced based on the Equation 17.

$$\text{Offspring} = \text{parent}_1 + \alpha \times (\text{parent}_2 - \text{parent}_1) \qquad \text{Eq. (17)}$$

Where the ratio α is a random number chosen within the range [-i,1+i]. if i=0, then, all children lie between the parents. In better words, children are located inside a hypercube bounded by the parents on its corners. For α higher than 0, algorithm can go beyond the hypercube of parents and reproduces offspring in that area.

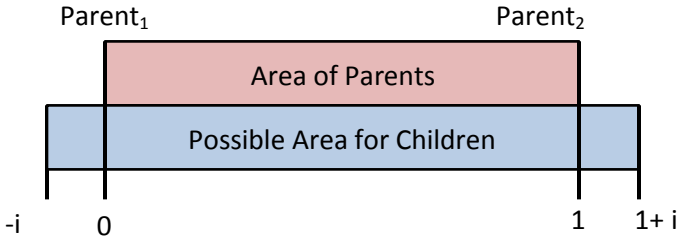


Figure 3-10: Intermediate Crossover

Mutation Algorithm

The mutation algorithm that is selected in this framework is called Adaptive Feasible. This algorithm examines the success of mutation in the last generation, and thus directs the algorithm towards success-prone mutations. In this method, genes are mutated with different probabilities. These probabilities are set on a line, and then randomly generated numbers go through this line and select an agreed number of genes for the mutation.

Elite Transfer

This option in genetic algorithm allows the best individuals in one generation migrate to the next generation without any changes to the structure of its chromosome. These individuals are called elite children. The parameter "elite count" defines the number of desired elite children in each generation. This value should be selected with discretion, as the high values of elite count lead to the dominancy of fittest individuals, which, in turn, makes the search less effective (Mathworks 2012).

3.3.3 Hybrid Approach of Modeling

In this phase, the model is complemented by the "Hybrid Approach" technique, which enables the system to accept both crisp values and linguistic terms. In this method, first a set of fuzzy variables is defined for each variable that can be explained linguistically. These fuzzy sets are then designed based on the experts' opinions. The developed model uses the given fuzzy sets whenever linguistic explanations are used for these variables.

When system runs, linguistic terms are transformed to a set of intervals. These intervals can be converted to a finite number of discrete points which properly represent the intervals. The resulting discrete points, along with the crisp inputs, are fed to the model to build the output membership function as the system's output. Figure 3-11 shows the required procedure for this section.

3.3.3.1 Computing Output Membership Functions

The implementation of the alpha-cut technique and the interval analysis of the fuzzy sets can be greatly simplified through the application of the vertex method (Dong and Shah 1987). This theory indicates that the vertices of the output fuzzy set at each level of α can be concluded only from the combinations of the input variables' vertices. In other words, the possible combinations of the maximums and minimums of the input intervals at each level of α can yield the minimum and maximum values of the output fuzzy set. However, vertex method is only applicable to monotonic models, where minimum and maximum values certainly occur at the boundary points of the input intervals. Many of the developed models are not monotonic and the exact extremum outputs of the unknown or non-monotonic models cannot be easily retrieved without an appropriate optimization process.

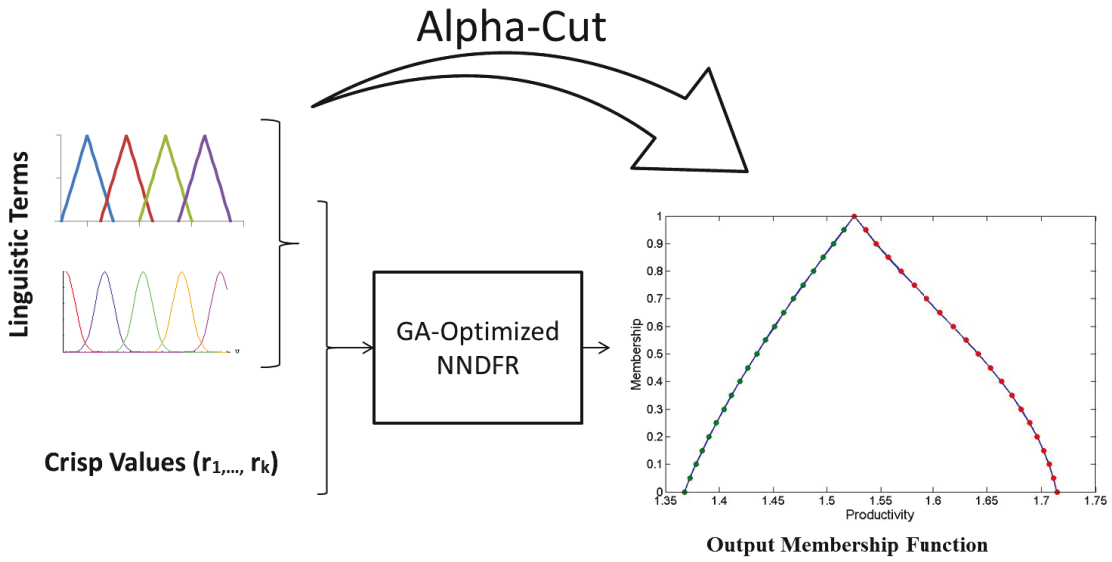


Figure 3-11: Hybrid Approach of Modeling

In this framework, each parameter's intervals at each level of α are decomposed into a specific number of discrete points and then the model is run for all the possible combinations of the inputs. In this way, all the possibilities are checked and the output fuzzy set is approximated. Figure 3-12 shows this procedure schematically.

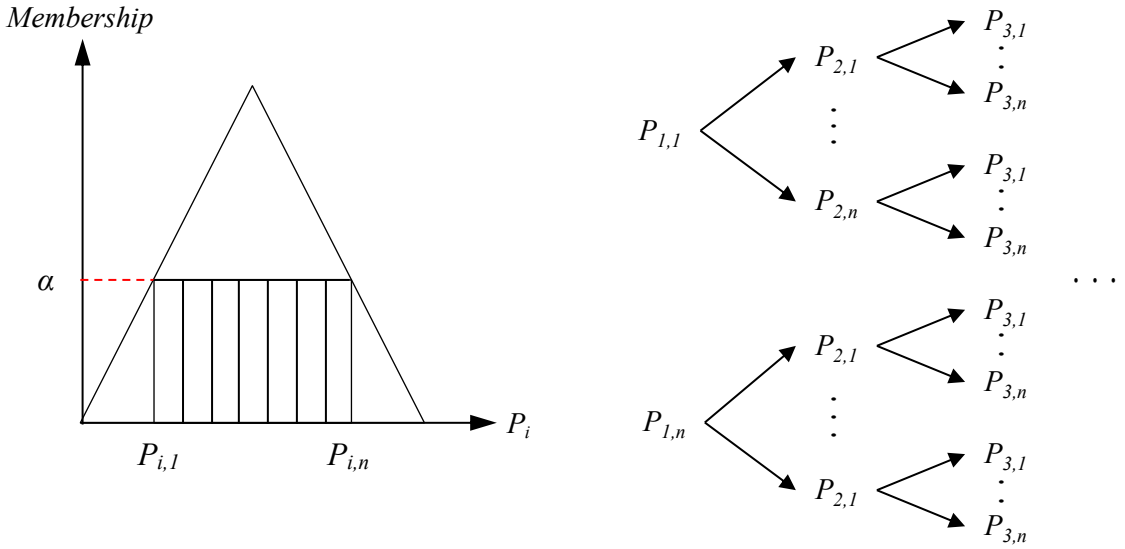


Figure 3-12: Hierarchy Of Alpha-Cut Technique, Where $P_{i,j}$ Is the J^{th} Discrete Point of the I^{th} Fuzzy Input Variable

3.3.3.2 Defuzzification

It is necessary to interpret the output fuzzy set with a single crisp value that represents the result of the modeling. To this end, it is required to defuzzify this fuzzy set. The most common approach for the difuzzification is centroid. However, because of the nonlinear nature of most of the construction models, output fuzzy sets usually do not have a symmetric or known shape. This means that the final output cannot be acquired through a straightforward mathematical equation. The only way is to resort to the original concept

of finding the center of a generic shape. The centroid of a plane shape *S* can be calculated through (1) dividing the shape into smaller and easier-to-compute sections, e.g. S_1, S_2, …, Sn; (2) finding their respective centroids, e.g. C_1, C_2, ..., Cn; and then (3) finding the final centroid using Equation 20.

$$C_x = \frac{\sum C_{ix} S_i}{\sum S_i}, C_y = \frac{\sum C_{iy} S_i}{\sum S_i} \qquad \text{Eq. (20)}$$

Where C_x and C_y denote the x-coordinate and y-coordinate, respectively. In the same manner, the trace produced by the alpha-cut technique can be divided into a finite number of trapezoids and its centroid can be measured using the aggregation. Figure 3-13 illustrates this procedure and its corresponding equation.

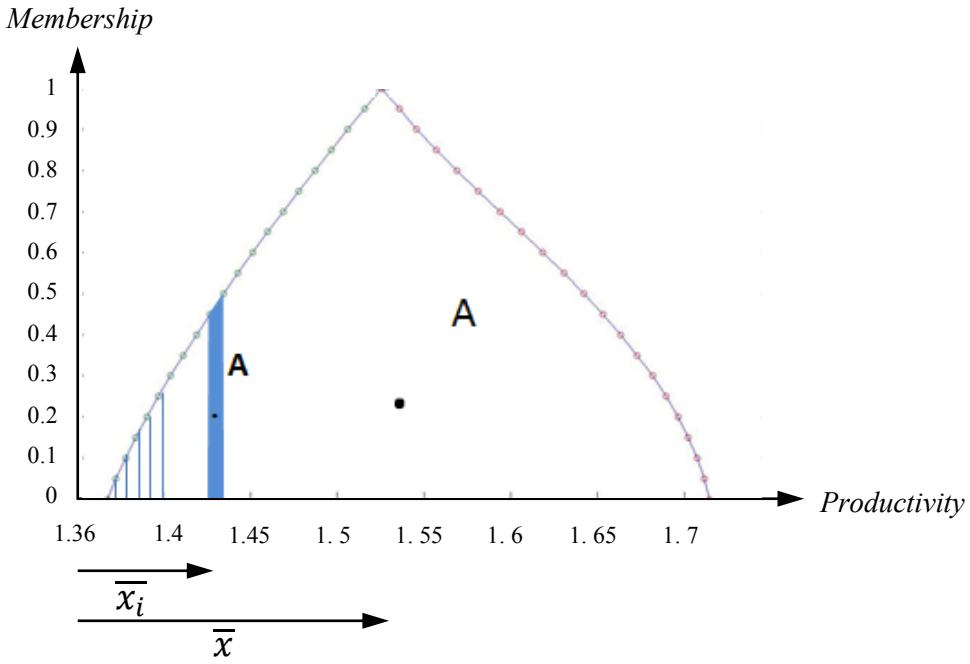


Figure 3-13: Defuzzification Process

3.3.3.3 Output Possibility Calculation

As explained in the literature review, based on the possibility theory, $\prod(A)$ denotes the possibility of a specific event A, where $\prod(A)$ is defined as shown in Equation 21 (Guyonnet et al. 2002).

$$\prod(\mathrm{A}) = \mathrm{Sup}_{\mathrm{x}\in\mathrm{A}}\mu(\mathrm{x}) \qquad \text{Eq. (21)}$$

Where $\mu(\mathrm{x})$ is the membership function of the variable *X* and *A* is a possible event. In this context, the possibility of having a productivity rate less than a specific threshold can be of interest. Regarding the previous equation, the possibility of an output (F) being less than the threshold T is calculated using Equation 22.

$$\mathrm{P}(\mathrm{F} < \mathrm{T}) = \mathrm{Sup}_{\mathrm{u}<\mathrm{T}}\mu_{\mathrm{F}}(\mathrm{u}) \qquad \text{Eq. (22)}$$

Where $\mu_{\mathrm{F}}(\mathrm{u})$ is the membership function of the variable *u* and *T* is an upper bound for variable u. Equation 22 indicates that the possibility of an output (F) being less than the threshold T, equals the maximum of the membership values for all productivities less than threshold T.

3.4 VERIFY AND VALIDATE THE SYSTEM USING REAL DATA

The developed model must be verified and validated through a real case study before put to the implementation. The accuracy of this innovative system can be assessed through comparing its outcomes with the other existing off-the-shelf models and techniques. For this purpose, a simple ANN, ANFIS, conventional NNDFR, and our Genetically

Optimized NNDFR will be fed with the same input data. The development procedure of the aforementioned systems will be briefly described and then, the reasons and the causes of any difference in the performance and the accuracy of their modeling will be analyzed and discussed. Furthermore, GA will check the efficiency and soundness of the model through testing dataset of the case study at the end of each fitness evaluation. This fitness evaluation will be executed by comparing the outcomes of the model and actual or expected outcomes.

3.5 COMPUTATIONS

The proposed model cannot be developed without the application of software that provides strong basis for both complex calculations and programming tasks. MATLAB, as a fourth-generation programming language, is a numerical computing platform that can satisfy these requirements. Appendix B presents s a detailed account of all the codes and programming done in MATLAB. The written codes, in accordance with the hierarchy of system development, have three main phases. Phase I is related to the main structure of NNDFR, which, in turn, includes two sub-phases for training the ANN and implementing the Fuzzy Inference System. Phase II addresses the optimization of the system parameters through the application of GA. The most crucial task in this part is to match the first and second phases. This involves the definition of NNDFR parameters and evaluation of NNDFR fitness by GA. Phase III is dedicated to implementing the hybrid approach of modeling that consists of the codes for alpha-cut and difuzzification processes.

CHAPTER 4: DATA COLLECTION

4.1 CHAPTER OVERVIEW

This chapter aims at introducing the case study project, the data collection process and the description of the processed data.

4.2 PROJECT DEFINITION

The eighteen-month long construction of Engineering, Computer Science and Visual Arts Complex of Concordia University was monitored using field observations and data collection. This 17-storey building, with a surface area of 86,000 square meters, is located at the heart of downtown in Montreal. For the case study of this research, the concrete pouring operations are scrutinized.

4.3 DATA COLLECTION PROCEDURE

Data collection is the act of gathering information about certain variables in order to examine hypotheses, answer research questions, and evaluate results. A part of data, used in this chapter, was used by Khan (2005) and Wang (2005) for studying the labor productivity. These researchers analyzed and selected the factors that affect the productivity. The rest of data are extracted from data records of World Wide Websites:

- Infrastructure Canada
- The Weather Networks

- Weather Spark

Figure 4-1 shows the selected variables and their classifications. The first group, i.e. weather, includes temperature, humidity, wind speed and precipitation. The second group is the crew variables and incorporates gang size and labor percentage. The project variables, which point out the operational details of construction process, are classified under the third group and include work type, floor level and work method. Several qualitative factors are also considered in each group so that their impacts on the productivity estimation can be studied along with the quantitative factors. The description and quantification method of all variables are presented in Table 4-1.

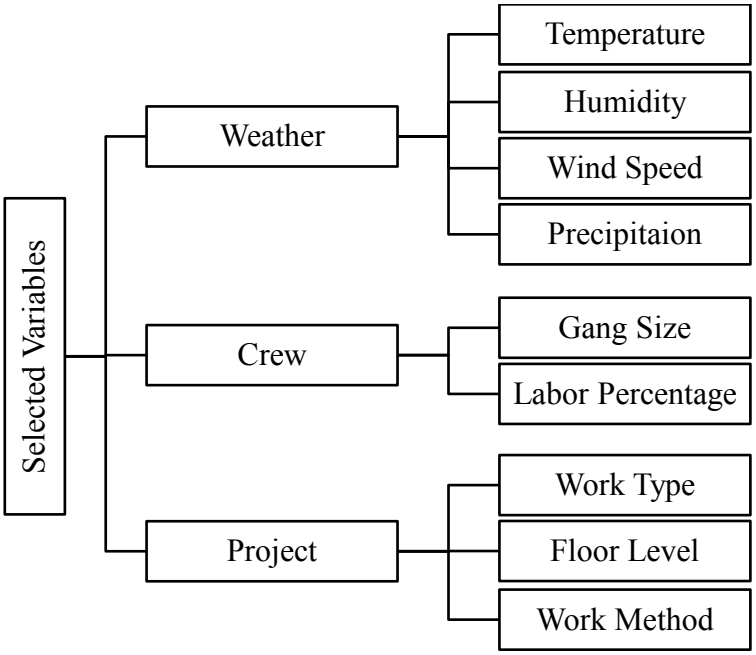


Figure 4-1: Hierarchy of the Considered Variables

Table 4-1: Variable Descriptions

	Variables	Description
1	Temperature (ºC)	Daily average of eight working hours
2	Humidity (%)	Daily average of eight working hours
3	Precipitation	Reported in terms of four numerical values: No precipitation = 0, Light rain = 1, Rain = 2, and Snow = 3
4	Wind Speed (km/h)	Daily average of eight working hours
5	Gang Size (workers)	Number of people in the gang
6	Labor Percentage (%)	The percentage of the labor (non-skilled workers) in the gang
7	Work Type	Reported in terms of activity type: Slabs= 1 and Walls = 2
8	Floor Level	The floor number
9	Work Method	Crane and bucket arrangement=1 and Pumping=2
10	Daily Productivity (m^3/man-hour)	Total cubic meters done during the day divided by the gang size and working hours

4.4 CONCRETE POURING PROCESS

According to Table 4-1, there are six quantitative variables and three qualitative variables that affect the level of productivity in the concrete pouring process. The qualitative factors, i.e. precipitation, work type and work method, were converted to numbers based on the provided descriptions in Table 4-1. The quantification of the variables is based on the expert opinion. The quantitative variables, including temperature, humidity, wind speed, gang size, labor percentage and floor level, are used as measured at the site or gathered from referenced sources, such as the weather network. There are 131 data points that are used in the modeling and validation phases. Table 4-2 shows a sample of concrete pouring data points.

Table 4-2: A Sample of Concrete Pouring Data

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
-8	87	2	14.2	22	36	1	3	1	1.27
-8	87	2	14.2	23	30	2	3	1	1.14
-12.5	54	0	5.2	21	38	1	3	1	1.17
-12.5	54	0	5.2	20	30	2	3	1	1.04
-16	55	0	6	23	35	1	3	1	1.16
-15	51	2	18.7	17	29	2	4	1	1.99
-15	51	2	18.7	20	40	1	4	1	1.1
-8.5	58	0	26.5	18	33	2	4	1	1
-4	87	2	3.6	22	36	1	4	1	1.55
-14	42	0	10	23	35	2	4	1	1.26
-14.5	42	0	7.5	19	33	2	4	1	1.14
-14.5	42	0	7.5	16	37	1	4	1	1.27
1.5	85	0	9.4	21	33	1	5	1	1.45
-0.5	53	0	7.5	20	30	1	5	1	1.51
-0.5	53	0	7.5	22	36	2	5	1	1.37

- Precipitation: *No precipitation = 0, Light rain = 1, Rain = 2, and Snow = 3*
- Labor Percentage: *The percentage of the labor (non-skilled workers) in the gang*
- Work Type: *Reported in terms of activity type: Slabs= 1 and Walls = 2*
- Work Method: *Crane and bucket arrangement=1 and Pumping=2*

4.5 CLASSIFICATION

In this part, the distribution of the data points in the data space is scrutinized. The data space is examined to find clear and well-identifiable clusters. For this purpose, the dataset is partitioned to different numbers of clusters and a set of validity indices are used to compare the resulting options. The values of the highest ranked options represent a better separation and compactness in the clusters. In other words, the number of clusters in the highest ranked options indicates the most natural and the clearest classification. The following indices are selected to assess the resulting clusters based on the internal criteria:

4.5.1 Dunn Index

The Dunn index (Dunn 1974) is proposed to identify the compact and well-separated clusters. For each partitioning, the index can be defined according to Equation 23 (Dunn 1973).

$$D_{nc} = \min_{1\le i\le nc}\left\{\min_{1\le j\le nc, i\neq j}\left\{\frac{d(i,j)}{\max_{1\le k\le nc} d'(k)}\right\}\right\} \quad \text{Eq. (23)}$$

Where $d(i,j)$ represents any measures of distance between two clusters, such as the distance between the centroids of the clusters; and $d'(k)$ represents any measure of distance within a cluster, such as the distance between any pair of elements in the cluster k. Based on the definition of the Dunn's index, higher values of the index are more favorable.

4.5.2 Davies-Bouldin Index

The Davies-Bouldin index (Davies and Bouldin 1979) represents the average of similarity between each cluster and its most similar one. Equation 24 defines this index (Davies and Bouldin 1979).

$$DB_{nc} = \frac{1}{nc}\sum_{i=1}^{nc} \max_{1\le j\le nc, i\neq j}\left(\frac{s_i+s_j}{d(c_i,c_j)}\right) \quad \text{Eq. (24)}$$

Where nc is the number of clusters and c_x is the centroid of the cluster x. s_x represents the average distance between all the elements in the cluster and the centroid c_x, and $d(*,*)$ indicates the distance between different centroids. Contrary to the Dunn index, the lower values of Davies-Bouldin index are more favorable.

4.5.3 Internal Validation

Since Dunn and Davies-Bouldin validity indices are generally designed for hard clusters, the data set needs to be re-clustered by a k-means algorithm. The data space is divided to different number of clusters in a range between 2 to 10. Internal validity indices are calculated and then plotted as is shown in Figure 4-2. As shown in this Figure, the highest value for Davies-Bouldin and the lowest value for Dunn index are achieved when 3 clusters are used. In other words, the desired separation and compactness among the dataset is attained by 3 clusters.

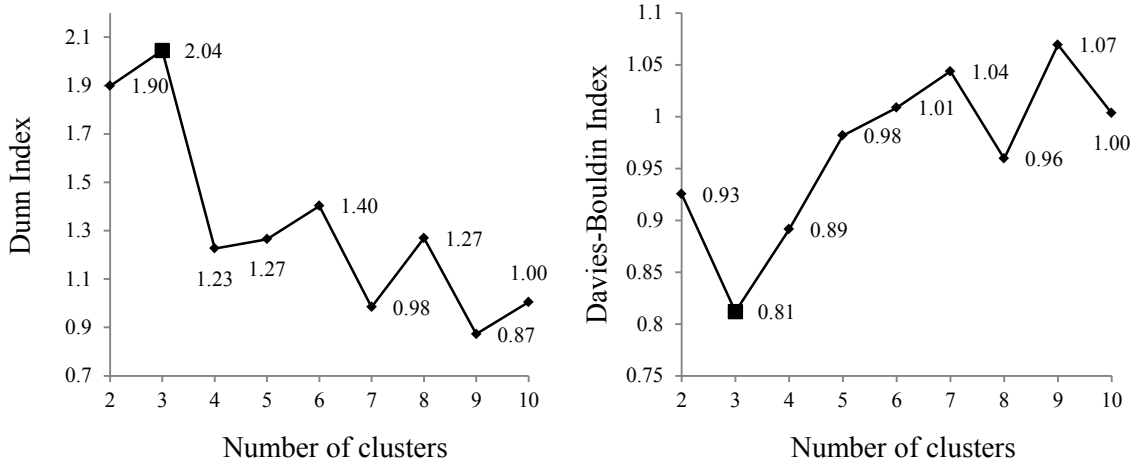


Figure 4-2: Calculated Values for Internal Validity Indices

4.6 DESCRIPTIVE DATA ANALYSIS

In this section, statistical measures are used to put the distribution characteristics of the variables' data into perspective. Table 4-3 presents a set of common statistical parameters, which help compare the variables' data in the case study. Additionally,

parameters such as Mean, Median and Standard Deviation will be further used for the calculation of the indices like Mean Squared Error.

Table 4-3: Statistical Measures of the Data

Statistical Index	Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Floor Level
Mean	5.16	66.37	13.35	17.16	34.85	10.60
Standard Deviation	11.19	16.99	5.90	4.75	3.53	3.82
Skewness	0.10	0.02	0.55	-0.60	0.48	-0.42
Kurtesis	2.13	1.97	2.81	1.99	2.81	2.22
Min	-17	36	3	8	29	3
1st Quartile	-4	53	8.7	12	33	8
Median	4.5	69	13	19	35	11
3rd Quartile	14.75	78.75	18	21	37	13
Max	25	97	29	24	44	17

CHAPTER 5: MODEL DEVELOPMENT AND IMPLEMENTATION

5.1 CHAPTER OVERVIEW

In this chapter, the methodology proposed in chapter 3 is implemented and applied to the case study in order to verify the developed model. As explained in the previous chapter, the dataset is gathered from the construction of Engineering, Computer Science and Visual Arts complex of Concordia University. The dataset consists of several quantitative and qualitative variables that affect the productivity of concrete pouring operations.

With respect to the methodology, the implementation of the framework involves three main phases. The first phase is to train the model and then analyze and process the data points. Next, in the second phase, the proposed structure of the model needs to be fine-tuned with the GA. The third phase is to implement the "hybrid approach" of modeling in the previously fine-tuned structure. Figure 5-1 illustrates these phases and how they are correlated to the defined objectives.

The collected data is divided into two sections to be used separately for the training and validation. The bigger set, which is used for the training, shapes the main structure of the model. The structure of the system is defined based on the inherent features of the data. The validation set is, then, used to verify the accuracy of the model.

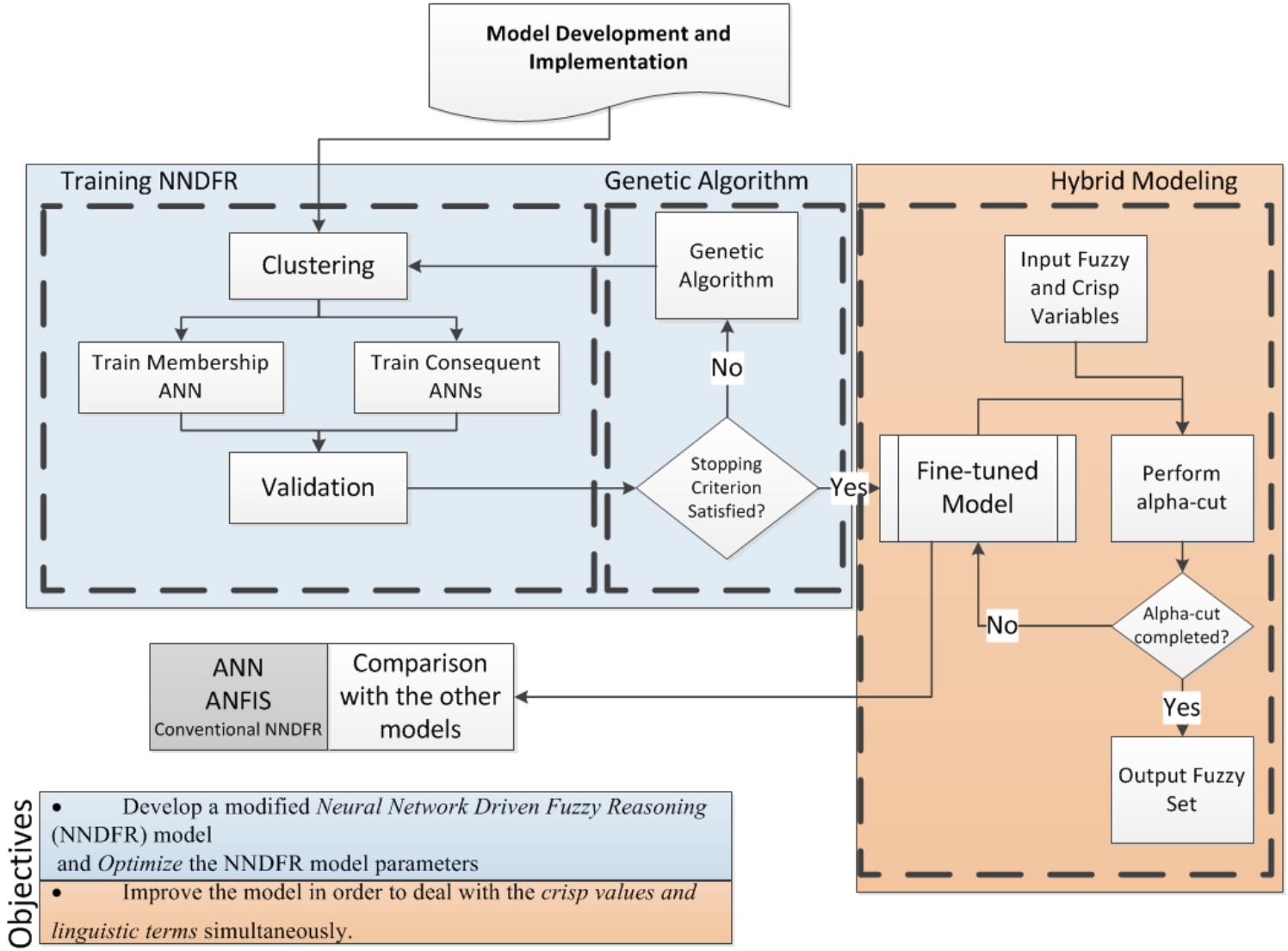


Figure 5-1: Model Development and Implementation Flowchart

5.2 TRAINING NNDFR

As explained in the chapter 3, the development and optimization stages are integrated in a single step so that the accuracy of the model is checked for each possible solution. Within this step, the optimization module generates several combinations of modeling parameters and for each one of these combinations a model is structured and trained. This section is dedicated to showing how the model is developed in a microcosm, i.e. using a small sample of parameters.

Table 5-1 presents a sample of data points that include the independent variables, e.g. Temperature, Humidity, Wind Speed, etc., and the dependent variables, e.g. Daily Productivity. Complete data set can be found in the APPENDIX A. It is assumed that the daily productivity of the concrete pouring process is affected by all these variables. Therefore, the main idea is to develop a model to connect the input data space to the output data space based on the logical correlations driven from the historical data.

5.2.1 Clustering

Training procedure starts with the clustering. The essence of this concept is to split a domain into several sub-domains, and build separate functional relationships between these sub-domains and their corresponding targets. The rationale behind this approach is that “the points located in the same cluster are likely to be governed by a similar rule”. First, the data set is divided to two parts for training and validation, containing 117 (90 %) and 14 (10 %) data points, respectively. This classification is shown in APPENDIX

A. Then, the training fragment is divided to three clusters using the FCM algorithm. The fuzzifier is set to "2.00".

Table 5-1: A Sample of Concrete Pouring Data

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
-8	87	2	14.2	22	36	1	3	1	1.27
-8	87	2	14.2	23	30	2	3	1	1.14
-12.5	54	0	5.2	21	38	1	3	1	1.17
-12.5	54	0	5.2	20	30	2	3	1	1.04
-16	55	0	6	23	35	1	3	1	1.16
-15	51	2	18.7	17	29	2	4	1	1.99
-15	51	2	18.7	20	40	1	4	1	1.1
-8.5	58	0	26.5	18	33	2	4	1	1
-4	87	2	3.6	22	36	1	4	1	1.55
-14	42	0	10	23	35	2	4	1	1.26
-14.5	42	0	7.5	19	33	2	4	1	1.14
-14.5	42	0	7.5	16	37	1	4	1	1.27
1.5	85	0	9.4	21	33	1	5	1	1.45
-0.5	53	0	7.5	20	30	1	5	1	1.51
-0.5	53	0	7.5	22	36	2	5	1	1.37

- Precipitation: *No precipitation = 0, Light rain = 1, Rain = 2, and Snow = 3*
- Labor Percentage: *The percentage of the labor (non-skilled workers) in the gang*
- Work Type: *Reported in terms of activity type: Slabs= 1 and Walls = 2*
- Work Method: *Crane and bucket arrangement=1 and Pumping=2*

Table 5-2 and Table 5-3 show the result of the clustering. Table 5-2 presents the coordinates of the cluster centroids and Table 5-3 lists the index of each data point in relation to its corresponding degree of membership to the defined clusters. It is worth reiterating that the summation of all degrees of membership for a data point equals one.

Table 5-2: Cluster Centers Generated by FCM

Centers	Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method
C1	3	79	0	13	11	37	1	12	2
C2	21	71	0	10	21	33	1	13	2
C3	5.5	46	0	12	19	33	2	10	1

Table 5-3: The Result of FCM

Index	C1	C2	C3	Index	C1	C2	C3
1	7.91E-04	9.99E-01	2.39E-04	60	9.99E-01	2.30E-04	1.07E-03
2	2.25E-03	9.97E-01	6.96E-04	61	9.92E-01	7.45E-03	3.73E-04
3	1.41E-03	1.79E-03	9.97E-01	62	1.00E+00	4.63E-06	7.08E-06
4	1.12E-03	1.26E-03	9.98E-01	63	9.98E-01	2.41E-03	4.12E-05
5	2.35E-03	4.01E-03	9.94E-01	64	1.00E+00	1.81E-05	1.20E-05
6	5.12E-04	4.91E-04	9.99E-01	65	1.00E+00	2.15E-04	1.52E-05
7	7.83E-04	8.32E-04	9.98E-01	66	1.00E+00	3.49E-04	7.28E-05
8	1.02E-02	9.66E-03	9.80E-01	67	1.00E+00	1.87E-05	1.73E-06
9	1.08E-03	9.99E-01	1.99E-04	68	1.00E+00	6.46E-05	1.86E-04
10	1.69E-04	8.29E-05	1.00E+00	69	1.00E+00	7.50E-05	3.00E-05
11	1.83E-04	1.01E-04	1.00E+00	70	1.00E+00	1.53E-04	7.07E-05
12	2.39E-04	1.47E-04	1.00E+00	71	9.99E-01	4.98E-04	3.76E-04
13	4.41E-04	6.32E-05	9.99E-01	72	1.00E+00	6.63E-05	6.82E-06
14	5.87E-04	8.91E-05	9.99E-01	73	1.00E+00	1.49E-04	1.92E-05
15	3.36E-05	5.82E-06	1.00E+00	74	1.00E+00	9.97E-06	3.85E-05
16	6.17E-04	9.99E-01	1.29E-04	75	1.00E+00	3.64E-05	2.14E-06
17	5.28E-05	1.00E+00	1.19E-05	76	1.00E+00	1.98E-04	1.58E-05
18	1.80E-03	9.98E-01	1.05E-04	77	9.94E-01	5.37E-03	2.79E-04
19	4.41E-04	1.00E+00	2.63E-05	78	9.98E-01	2.24E-04	1.70E-03
20	3.59E-04	1.00E+00	1.92E-05	79	1.00E+00	5.26E-06	1.38E-06
21	1.13E-03	9.99E-01	4.55E-05	80	9.99E-01	1.38E-03	4.03E-05
22	1.43E-03	9.99E-01	4.21E-05	81	9.99E-01	1.25E-03	1.00E-04
23	5.01E-04	9.99E-01	1.37E-05	82	9.99E-01	8.99E-04	4.59E-05
24	2.40E-09	4.59E-10	1.00E+00	83	1.00E+00	3.05E-04	7.82E-05
25	1.97E-07	3.55E-08	1.00E+00	84	1.00E+00	1.90E-05	4.47E-05
26	3.59E-04	1.84E-04	9.99E-01	85	2.79E-03	9.97E-01	2.13E-05
27	2.95E-04	1.56E-04	1.00E+00	86	9.62E-01	3.31E-02	4.38E-03
28	3.35E-05	4.32E-06	1.00E+00	87	1.00E+00	1.24E-04	1.92E-04
29	3.31E-05	4.26E-06	1.00E+00	88	9.94E-01	5.41E-03	2.63E-04
30	1.12E-04	2.02E-05	1.00E+00	89	9.91E-01	8.28E-03	4.75E-04
31	2.65E-05	5.86E-06	1.00E+00	90	1.44E-02	9.85E-01	4.62E-04
32	2.61E-05	5.78E-06	1.00E+00	91	4.97E-03	9.95E-01	1.83E-04
33	1.39E-05	3.34E-06	1.00E+00	92	6.49E-03	9.93E-01	4.64E-04
34	2.65E-03	7.91E-05	9.97E-01	93	1.97E-03	9.98E-01	1.02E-04
35	7.79E-04	2.06E-05	9.99E-01	94	1.89E-02	9.81E-01	2.61E-04
36	5.70E-04	1.73E-05	9.99E-01	95	9.99E-04	9.99E-01	4.02E-05
37	2.79E-04	8.00E-06	1.00E+00	96	9.81E-01	1.83E-02	7.44E-04
38	1.43E-03	9.99E-01	2.87E-05	97	2.89E-02	9.71E-01	5.34E-04
39	1.05E-04	1.00E+00	2.02E-06	98	3.51E-04	1.00E+00	1.20E-05
40	6.66E-04	2.58E-04	9.99E-01	99	1.24E-02	9.87E-01	2.03E-04
41	6.57E-04	2.55E-04	9.99E-01	100	1.61E-04	1.00E+00	5.03E-06
42	3.37E-04	2.51E-05	1.00E+00	101	1.51E-05	1.00E+00	3.78E-07
43	3.28E-04	2.49E-05	1.00E+00	102	4.14E-04	1.00E+00	1.75E-05
44	1.39E-03	9.99E-01	3.07E-05	103	2.38E-02	9.74E-01	1.73E-03
45	1.05E-02	3.97E-04	9.89E-01	104	2.47E-03	9.97E-01	1.40E-04
46	1.08E-02	3.84E-04	9.89E-01	105	1.09E-03	5.64E-04	9.98E-01
47	1.84E-02	2.70E-04	9.81E-01	106	8.70E-03	9.83E-01	7.93E-03
48	1.82E-02	2.83E-04	9.81E-01	107	5.45E-04	9.99E-01	1.96E-04
49	2.91E-03	1.12E-04	9.97E-01	108	2.14E-05	1.00E+00	2.45E-06
50	2.89E-03	1.12E-04	9.97E-01	109	1.09E-02	9.88E-01	7.32E-04

Index	C1	C2	C3	Index	C1	C2	C3
51	2.50E-02	4.66E-04	9.75E-01	110	6.74E-05	1.00E+00	5.05E-06
52	9.97E-01	3.52E-04	2.17E-03	111	2.78E-02	9.55E-01	1.69E-02
53	1.00E+00	1.66E-04	4.44E-06	112	4.71E-03	9.91E-01	3.97E-03
54	9.94E-01	1.46E-03	4.45E-03	113	7.72E-03	7.33E-03	9.85E-01
55	1.00E+00	4.90E-05	5.50E-05	114	2.07E-02	9.78E-01	1.57E-03
56	9.97E-01	4.64E-04	2.31E-03	115	4.92E-03	2.95E-03	9.92E-01
57	9.96E-01	3.98E-04	3.36E-03	116	2.64E-02	1.39E-03	9.72E-01
58	9.98E-01	1.70E-04	1.62E-03	117	2.28E-02	1.00E-03	9.76E-01
59	9.98E-01	1.86E-03	1.99E-05				

5.2.2 Multi-dimensional Membership Function

In this step, in order to automate the process of fuzzification, an ANN is trained between the input variables and the degrees of membership generated by the FCM. 80 percent of the data is used to train the network and the remaining 20 percent are used for testing, i.e. validation. The *Bayesian Regularization* is selected as the training algorithm. After several trial and errors, a network with three layers shows the best performance. The network consists of 9 neurons in the input layer, i.e. Hidden 1, 10 neurons in the hidden layer, i.e. Hidden 2, and 3 neurons in the output layer. Figure 5-2 schematically depicts this structure.

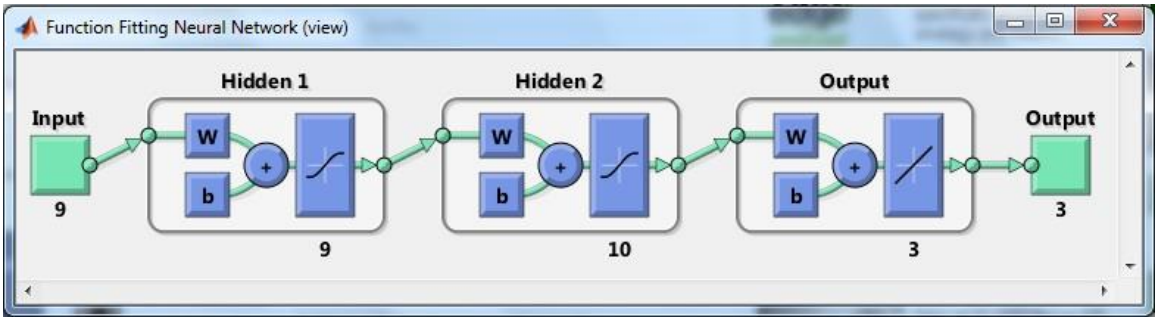


Figure 5-2: Structure of the Membership ANN

Figure 5-3 shows the information related to the training task. According to the results, the training stops at the MSE_{Train} of 1.8e-07 and the MSE_{Test} of 9.66e-04. The model is then saved to be later embedded in the fuzzification section of the NNDFR. So, each time used, this membership ANN generates some weights for the consequent ANNs and controls the accuracy of the estimation.

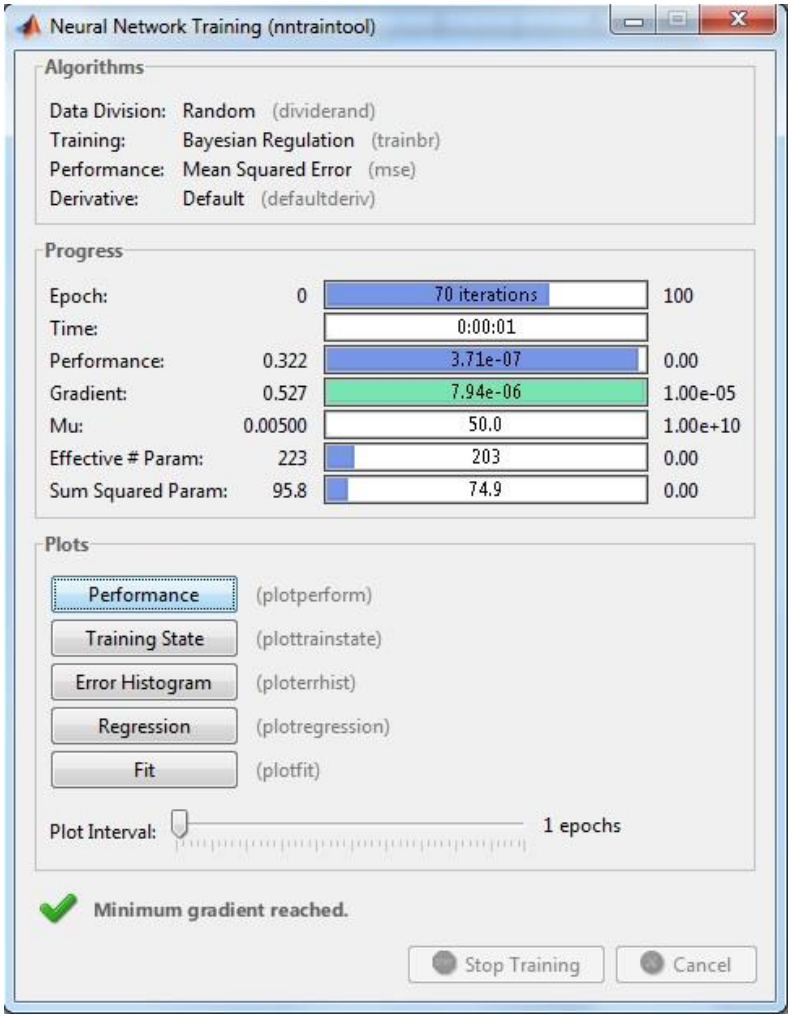


Figure 5-3: Reported Information of the Training Procedure

5.2.3 Consequent Neural Networks

A different set of ANN are assigned to transform the independent variables to dependent variables. These functional relationships are established using the learning ability of ANN. If the NNDFR is considered as a Takagi-Sugeno type fuzzy model, *"Multi-dimensional Membership Functions"* and "*Consequent Neural Networks"* constitute the IF part and the THEN part of the fuzzy model, respectively. Duo to the fuzzy nature of FCM, each data point belongs to all the clusters, but with different degrees of membership. However, the consequent ANNs need separate training data sets. To resolve this problem, each data point will be tagged to the cluster in which it has the highest degree of membership.

Table 5-4 separately tabulates the data provided to the different consequent ANNs. *Bayesian Regularization* is selected for the training of the consequent ANNs. After several trial and errors, a three-layer network shows the best performance. The network comprises 9 neurons in the input layer, i.e. Hidden 1, 10 neurons in the hidden layer, i.e. Hidden 2, and 1 neuron in the output layer. Figure 5-4 schematically shows this structure.

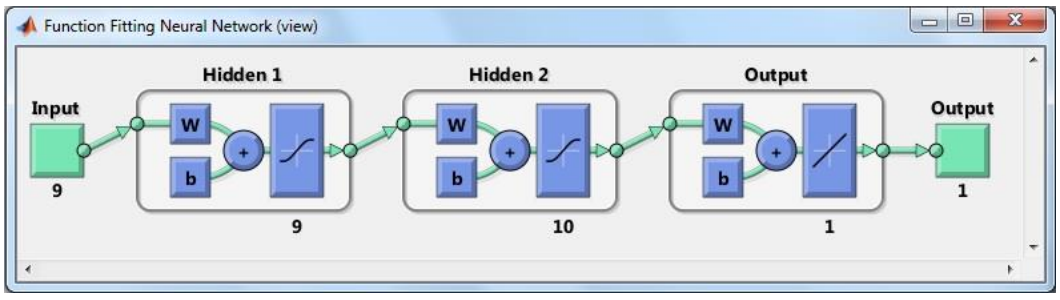


Figure 5-4 : Structure of Consequent Neural Networks

Table 5-4: Separate Sets of Training Data Fed To Consequent Neural Networks

Index	NN_1			Index	NN_2			Index	NN_3		
	C1	C2	C3		C1	C2	C3		C1	C2	C3
52	9.97E-01	3.52E-04	2.17E-03	**1**	7.91E-04	9.99E-01	2.39E-04	**3**	1.41E-03	1.79E-03	9.97E-01
53	1.00E+00	1.66E-04	4.44E-06	**2**	2.25E-03	9.97E-01	6.96E-04	**4**	1.12E-03	1.26E-03	9.98E-01
54	9.94E-01	1.46E-03	4.45E-03	**9**	1.08E-03	9.99E-01	1.99E-04	**5**	2.35E-03	4.01E-03	9.94E-01
55	1.00E+00	4.90E-05	5.50E-05	**16**	6.17E-04	9.99E-01	1.29E-04	**6**	5.12E-04	4.91E-04	9.99E-01
56	9.97E-01	4.64E-04	2.31E-03	**17**	5.28E-05	1.00E+00	1.19E-05	**7**	7.83E-04	8.32E-04	9.98E-01
57	9.96E-01	3.98E-04	3.36E-03	**18**	1.80E-03	9.98E-01	1.05E-04	**8**	1.02E-02	9.66E-03	9.80E-01
58	9.98E-01	1.70E-04	1.62E-03	**19**	4.41E-04	1.00E+00	2.63E-05	**10**	1.69E-04	8.29E-05	1.00E+00
59	9.98E-01	1.86E-03	1.99E-05	**20**	3.59E-04	1.00E+00	1.92E-05	**11**	1.83E-04	1.01E-04	1.00E+00
60	9.99E-01	2.30E-04	1.07E-03	**21**	1.13E-03	9.99E-01	4.55E-05	**12**	2.39E-04	1.47E-04	1.00E+00
61	9.92E-01	7.45E-03	3.73E-04	**22**	1.43E-03	9.99E-01	4.21E-05	**13**	4.41E-04	6.32E-05	9.99E-01
62	1.00E+00	4.63E-06	7.08E-06	**23**	5.01E-04	9.99E-01	1.37E-05	**14**	5.87E-04	8.91E-05	9.99E-01
63	9.98E-01	2.41E-03	4.12E-05	**38**	1.43E-03	9.99E-01	2.87E-05	**15**	3.36E-05	5.82E-06	1.00E+00
64	1.00E+00	1.81E-05	1.20E-05	**39**	1.05E-04	1.00E+00	2.02E-06	**24**	2.40E-09	4.59E-10	1.00E+00
65	1.00E+00	2.15E-04	1.52E-05	**44**	1.39E-03	9.99E-01	3.07E-05	**25**	1.97E-07	3.55E-08	1.00E+00
66	1.00E+00	3.49E-04	7.28E-05	**85**	2.79E-03	9.97E-01	2.13E-05	**26**	3.59E-04	1.84E-04	9.99E-01
67	1.00E+00	1.87E-05	1.73E-06	**90**	1.44E-02	9.85E-01	4.62E-04	**27**	2.95E-04	1.56E-04	1.00E+00
68	1.00E+00	6.46E-05	1.86E-04	**91**	4.97E-03	9.95E-01	1.83E-04	**28**	3.35E-05	4.32E-06	1.00E+00
69	1.00E+00	7.50E-05	3.00E-05	**92**	6.49E-03	9.93E-01	4.64E-04	**29**	3.31E-05	4.26E-06	1.00E+00
70	1.00E+00	1.53E-04	7.07E-05	**93**	1.97E-03	9.98E-01	1.02E-04	**30**	1.12E-04	2.02E-05	1.00E+00
71	9.99E-01	4.98E-04	3.76E-04	**94**	1.89E-02	9.81E-01	2.61E-04	**31**	2.65E-05	5.86E-06	1.00E+00
72	1.00E+00	6.63E-05	6.82E-06	**95**	9.99E-04	9.99E-01	4.02E-05	**32**	2.61E-05	5.78E-06	1.00E+00
73	1.00E+00	1.49E-04	1.92E-05	**97**	2.89E-02	9.71E-01	5.34E-04	**33**	1.39E-05	3.34E-06	1.00E+00
74	1.00E+00	9.97E-06	3.85E-05	**98**	3.51E-04	1.00E+00	1.20E-05	**34**	2.65E-03	7.91E-05	9.97E-01
75	1.00E+00	3.64E-05	2.14E-06	**99**	1.24E-02	9.87E-01	2.03E-04	**35**	7.79E-04	2.06E-05	9.99E-01
76	1.00E+00	1.98E-04	1.58E-05	**100**	1.61E-04	1.00E+00	5.03E-06	**36**	5.70E-04	1.73E-05	9.99E-01
77	9.94E-01	5.37E-03	2.79E-04	**101**	1.51E-05	1.00E+00	3.78E-07	**37**	2.79E-04	8.00E-06	1.00E+00
78	9.98E-01	2.24E-04	1.70E-03	**102**	4.14E-04	1.00E+00	1.75E-05	**40**	6.66E-04	2.58E-04	9.99E-01
79	1.00E+00	5.26E-06	1.38E-06	**103**	2.38E-02	9.74E-01	1.73E-03	**41**	6.57E-04	2.55E-04	9.99E-01
80	9.99E-01	1.38E-03	4.03E-05	**104**	2.47E-03	9.97E-01	1.40E-04	**42**	3.37E-04	2.51E-05	1.00E+00
81	9.99E-01	1.25E-03	1.00E-04	**106**	8.70E-03	9.83E-01	7.93E-03	**43**	3.28E-04	2.49E-05	1.00E+00

82	9.99E-01	8.99E-04	4.59E-05	**107**	5.45E-04	9.99E-01	1.96E-04	**45**	1.05E-02	3.97E-04	9.89E-01
83	1.00E+00	3.05E-04	7.82E-05	**108**	2.14E-05	1.00E+00	2.45E-06	**46**	1.08E-02	3.84E-04	9.89E-01
84	1.00E+00	1.90E-05	4.47E-05	**109**	1.09E-02	9.88E-01	7.32E-04	**47**	1.84E-02	2.70E-04	9.81E-01
86	9.62E-01	3.31E-02	4.38E-03	**110**	6.74E-05	1.00E+00	5.05E-06	**48**	1.82E-02	2.83E-04	9.81E-01
87	1.00E+00	1.24E-04	1.92E-04	**111**	2.78E-02	9.55E-01	1.69E-02	**49**	2.91E-03	1.12E-04	9.97E-01
88	9.94E-01	5.41E-03	2.63E-04	**112**	4.71E-03	9.91E-01	3.97E-03	**50**	2.89E-03	1.12E-04	9.97E-01
89	9.91E-01	8.28E-03	4.75E-04	**114**	2.07E-02	9.78E-01	1.57E-03	**51**	2.50E-02	4.66E-04	9.75E-01
96	9.81E-01	1.83E-02	7.44E-04					**105**	1.09E-03	5.64E-04	9.98E-01
								113	7.72E-03	7.33E-03	9.85E-01
								115	4.92E-03	2.95E-03	9.92E-01
								116	2.64E-02	1.39E-03	9.72E-01
								117	2.28E-02	1.00E-03	9.76E-01

Similar to the membership ANNs, 80 percent of the data is used to train the network and the remaining 20 percent are used the testing, i.e. validation. Table 5-5 shows the performance of the trained networks for both the training and testing sets in terms of MSE.

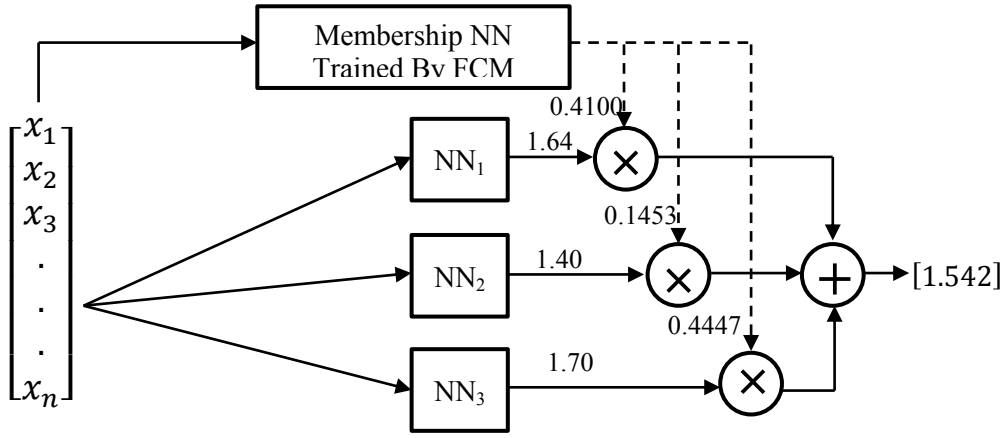
Table 5-5: Report of the Consequent ANN Training Algorithm

	NN_1	**NN_2**	**NN_3**
Training Algorithm		Bayesian Regularization	
MSE_{Train}	2.23e-08	2.22e-3	1.09e-05
MSE_{Test}	2.53e-2	5.16e-2	1.8e-2

5.2.4 Validation

At this point, the remaining 14 data points, which were reserved for the validation, are fed to the model. Each data point is separately injected to the consequent ANNs and the membership ANN as a vector. Each output of the membership ANN defines the effectiveness of the consequent ANN that is related to that cluster. Therefore, the output

of the model is a weighted average of all consequent ANNs. After forecasting the outputs based on the developed NNDFR model, the error of estimation should be calculated. The calculation method for the first data point of the testing sample is illustrated in the Figure 5-5. In addition, the outcomes of the model are tabulated against the actual values of the productivity in Table 5-6.



$$\Pr = \frac{0.41 \times 1.64 + 0.1453 \times 1.40 + 0.4447 \times 1.70}{0.4100 + 0.1453 + 0.4447} = 1.542(\frac{m^3}{man.hr})$$

Figure 5-5: The Calculation Method in the Modified NNDFR Model

5.3 GENETIC OPTIMIZATION

Section 5.2 presented the NNDFR training process for a defined set of parameters, i.e. clustering number, fuzzifier. However, the best combination of the parameters that result in the lowest MSE needs to be identified. For this purpose, the GA searches the solution domain and evaluates all the combinations of the parameters for the best answer. In each generation and for each member of the population the training process is repeated and a NNDFR structure is created. Finally, the created sets of NNDFR structure are tested

through comparing their estimated outputs with the actual targets using MSE as the measure of the accuracy of the model.

Table 5-6: Result of the Three-Cluster NNDFR Model

Index	Actual Daily Productivity (m^3/man-hour)	Estimated Daily Productivity (m^3/man-hour)
1	1.450	1.542
2	1.250	1.411
3	1.220	1.443
4	1.350	1.393
5	1.750	1.879
6	1.730	1.804
7	1.800	2.035
8	1.970	2.193
9	1.770	1.971
10	1.340	1.426
11	1.380	1.380
12	1.440	1.501
13	1.470	1.619
14	1.360	1.566

Within the optimization part, searching bounds for the number of clusters and fuzzifier are defined. The fuzzifier ranges from 1 to 3 and the number of clusters varies between 2 to 10. The parameters used in the GA are shown in Table 5-7.

Table 5-7: The Parameters of the GA

Parameter	Value/Type
Fitness function	MSE of NNDFR output
Population size	30
Number of generations	20
Selection function	Stochastic Uniform
Elite count	2
Crossover fraction	0.8
Crossover function	Intermediate Crossover (Ratio=1)
Mutation function	Adaptive Feasible
Search bounds	The number of clusters: [2,10]; Fuzzifier: (1,3]

Table 5-8 presents the winners of each generation based on the MSE, which was chosen as the fitness criterion. According to the table, from the fourth generation onward the algorithm has fixed the number of clusters and only modified the value of the fuzzifier. This indicates how the fuzziness of a hyper-surface membership function can influence the accuracy of the model. These optimized parameters result in the best fitness in terms of MSE, which is improved by 52 percent compared to the winner of first generation. The algorithm is terminated when maximum number of generations is reached, proposing 3 clusters and a fuzzifier equivalent to 1.2513. The final winner with m=1.2513 implies that a typical NNDFR structure made up of K-Means algorithm (m=1) cannot deliver the best possible solution.

Moreover, the optimum number of clusters attained by GA is confirmed by the best choice from the internal validation explained in Section 4.5.3. It can be concluded that the data points having more separate and compact clusters are more likely to follow the same functional relationship. In other words, the greater the resemblance of data points in a cluster, the better the system. Figure 5-6 plots the best and the mean penalty of different generations during the evolution process. This plot helps better comprehend the minimization trend. Table 5-9 encapsulates the detailed results of the Modified NNDFR model whose parameters are genetically optimized. According to the results, performance indices are calculated as shown in the set of Equations 25.

Table 5-8: The Results of GA

Generations	Individuals		MSE
	Clustering Number	Fuzzifier	
1	3	1.8516	0.01954
2	4	1.3957	0.01949
3	4	1.3957	0.01949
4	3	1.7288	0.01734
5	3	1.4151	0.01484
6	3	1.4151	0.01484
7	3	1.4151	0.01484
8	3	1.4151	0.01484
9	3	1.4151	0.01484
10	3	1.4687	0.01221
11	3	1.4687	0.01221
12	3	1.5054	0.00976
13	3	1.5054	0.00976
14	3	1.2513	0.00932
15	3	1.2513	0.00932
16	3	1.2513	0.00932
17	3	1.2513	0.00932
18	3	1.2513	0.00932
19	3	1.2513	0.00932
20	3	1.2513	0.00932

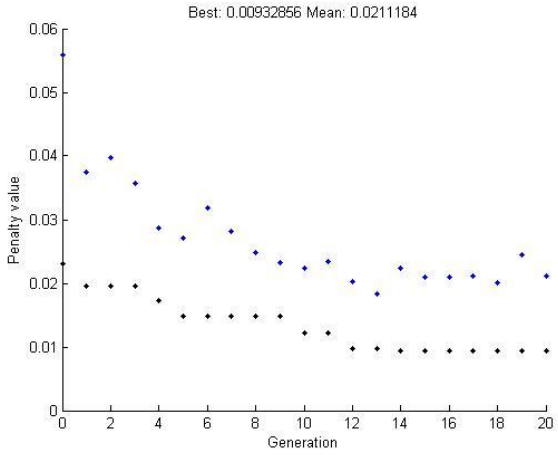


Figure 5-6: The Lowest and Mean Penalty Values (MSE) Over Different Generations

Table 5-9: Result of the Modified NNDFR Model

(Fuzzifier=1.2513, Clustering Number=3)

Index	Actual Daily Productivity (m^3/man-hour)	Estimated Daily Productivity (m^3/man-hour)
1	1.450	1.466
2	1.250	1.356
3	1.220	1.243
4	1.350	1.418
5	1.750	1.874
6	1.730	1.846
7	1.800	1.894
8	1.970	2.058
9	1.770	1.793
10	1.340	1.439
11	1.380	1.476
12	1.440	1.544
13	1.470	1.586
14	1.360	1.466

$$MSE_{Test} = \frac{\sum_{i=1}^{14}(AP_i - EP_i)^2}{14} = 0.009$$

$$MSE_{Train} = \frac{\sum_{i=1}^{117}(AP_i - EP_i)^2}{117} = 0.010$$

$$MSE_{Average} = \frac{0.009 \times 14 + 0.010 \times 117}{131} = 0.010 \qquad \text{Eq.(25)}$$

$$AIP = \left|1 - \frac{EP_i}{AP_i}\right| \times \frac{100}{14} = 5.44$$

$$AVP = 100 - AIP = 94.56$$

5.4 COMPARISON WITH OTHER METHODS

The merits and superiority of the proposed model is more visible when its results are compared with the results of other common forecasting models. To this end, separate models with different logics were developed. A plausible comparison can be expected only when all the models are trained and tested with the same data. Thus, the same set of data used in case study, with the similar training and testing partitions, was fed to all the alternative models. In this section, the alternative models used are a simple feed-forward ANN, ANFIS and a conventional NNDFR.

5.4.1 ANN

This model is a typical feed-forward net that catch all independent variables and directly generates the productivity. To have an acceptable bed for the comparison, the model must be developed with its best performance. After several trial and errors, the best structure was found to be a three-layer feed-forward net, which has 9 neurons in the input layer, 11 neurons in the hidden layer and 1 neuron in the output layer. *Bayesian Regularization* was selected as the learning algorithm. Since this algorithm increases the generalization ability of the network, a more accurate prediction is observed in the testing sample.

Table 5-10 illustrates all the information regarding the trained ANN. The most important criterion in the comparison is MSE_{Test}, which indicates the level of accuracy of the model. However, in order to make any decisions, all the indices representing the efficiency of the model in both testing and training phases must be considered.

Table 5-10: Result of Modeling with A Three-Layer Feed-Forward Net [9 11 1]

Index	Actual Daily Productivity (m^3/man-hour)	Estimated Daily Productivity (m^3/man-hour)
1	1.450	1.572
2	1.250	1.559
3	1.220	1.494
4	1.350	1.515
5	1.750	1.832
6	1.730	1.984
7	1.800	2.010
8	1.970	2.100
9	1.770	2.031
10	1.340	1.533
11	1.380	1.660
12	1.440	1.666
13	1.470	1.784
14	1.360	1.514

Index	Value
MSE_{Train}	0.032
MSE_{Test}	0.054
$MSE_{Average}$	0.034
AVP (%)	85.23
AIP (%)	14.77

5.4.2 ANFIS

In this section, ANFIS is employed to model the data. MATLAB ANFIS TOOLBOX is used for the implementation of the model. Figure 5-7 shows a screenshot of the main window of ANFIS TOOLBOX. After loading the data, Fuzzy Inference System (FIS) must be generated through either Grid Partitioning or Subtractive Clustering. Within this stage, the number of clusters, or better say the number of fuzzy rules, is defined. Subtractive Clustering is selected as the operator and its parameters are presented in Table 5-11. Based on the defined parameters, Subtractive Clustering detected 63 clusters as the most appropriate number of clusters. In this neuro-fuzzy system, the number of rules is equal to the number of clusters, i.e. 63. Supplementary plots including the ANFIS structure and generated fuzzy rules are presented in APPENDIX A. As is shown in Figure 5-7 the model reaches a training MSE of 0.021 after 1000 epochs. Table 5-12

shows the estimated daily productivities against their corresponding actual values and the significant performance indexes.

Table 5-11: The Parameters of the Subtractive Clustering

Parameter	Value
Influence Range	0.5
Squash Factor	1.25
Accept Ratio	0.5
Reject Ratio	0.15

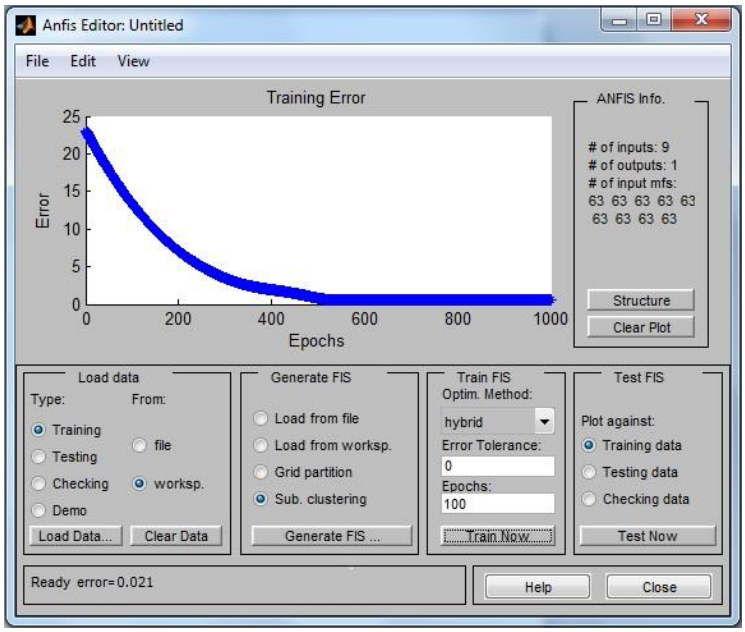


Figure 5-7: Screenshot from the ANFIS Editor of MATLAB during Training Procedure

Table 5-12: Results of Modeling with ANFIS

Index	Actual Daily Productivity (m^3/man-hour)	Estimated Daily Productivity (m^3/man-hour)
1	1.450	1.508
2	1.250	1.463
3	1.220	1.370
4	1.350	1.532
5	1.750	1.906
6	1.730	1.813
7	1.800	2.021
8	1.970	2.170
9	1.770	1.995
10	1.340	1.569
11	1.380	1.422
12	1.440	1.657
13	1.470	1.637
14	1.360	1.470

Index	Value
MSE_{Train}	0.021
MSE_{Test}	0.032
$MSE_{Average}$	0.022
AVP (%)	89.06
AIP (%)	10.94

5.4.3 Conventional NNDFR

This section presents the results of implementing a conventional NNDFR with three clusters. The idea behind the development of this model is to have another baseline for the comparison. This model has the same number of clusters as the proposed genetically optimized NNDFR but the fuzziness of its membership function is not controllable. The hyper-surface membership functions are trained based on K-Means clustering algorithm. In this structure, all the embedded ANNs have the same number of layers and neurons as the proposed modified NNDFR. In order to lay a fair ground for the comparison, the training algorithm is chosen to be Bayesian Regularization. Table 5-13 summarizes the results of this model.

Table 5-13: Result of Modeling with a Three-Cluster Conventional NNDFR

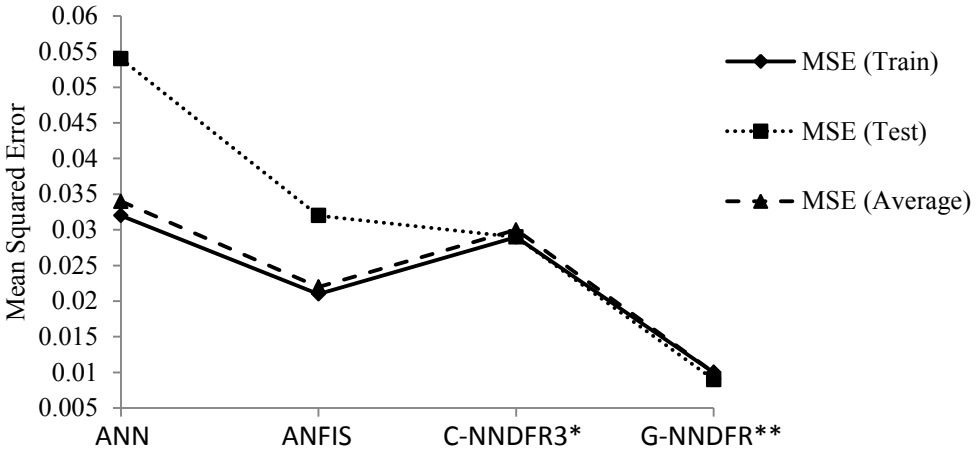
Index	Actual Daily Productivity (m^3/man-hour)	Estimated Daily Productivity (m^3/man-hour)
1	1.450	1.508
2	1.250	1.250
3	1.220	1.433
4	1.350	1.529
5	1.750	1.979
6	1.730	1.825
7	1.800	2.039
8	1.970	2.121
9	1.770	1.837
10	1.340	1.593
11	1.380	1.582
12	1.440	1.619
13	1.470	1.518
14	1.360	1.467

Index	Value
MSE_{Train}	0.029
MSE_{Test}	0.029
$MSE_{Average}$	0.030
AVP (%)	90.21
AIP (%)	9.79

5.4.4 Results and Comparison

Figure 5-8 and Figure 5-9 demonstrate the comparison of the results of all models. As can be seen, the genetically optimized NNDFR has a clear edge in terms of performance, both in training and testing phase. Another considerable observation is that the error of this model in the learning and validation are very close to each other. This indicates how successful this model is in the generalization. In other words, the model refrains from memorizing the input-output mapping process and is able to perform an accurate generalization based on the learned relations.

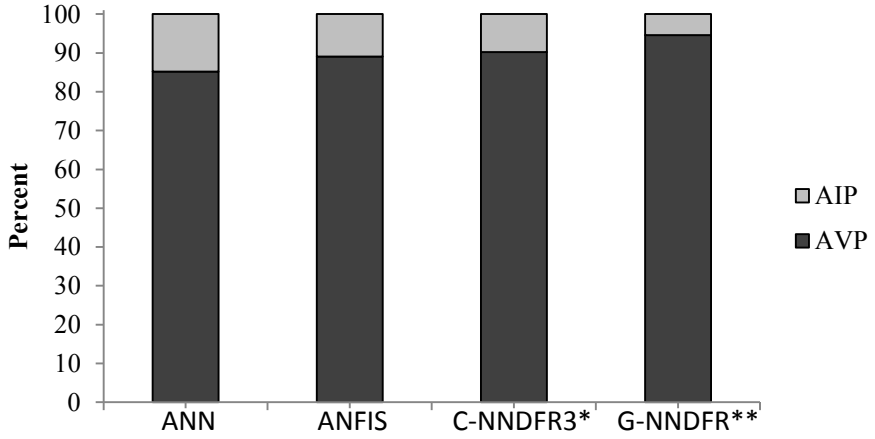
Although the three-clustered conventional NNDFR shows a satisfactory result in terms of generalization, it still has a lower accuracy compared to the modified NNDFR. It can be concluded that NNDFR models generally perform well in generalizing the recognized patterns.



*C-NNDFR3: Three-Clustered Conventional NNDFR
*G-NNDFR: Modified NNDFR With 3 Clusters And The Fuzzifier=1.2513

Figure 5-8: Comparison of Performances of Different Models In Terms Of MSE

Figure 5-9 compares different models with respect to their Average Validity Percentage and Average Invalidity Percentage. AVP results substantiate the verdict made based on the MSE comparison, indicating the better performance of the modified NNDFR.



*C-NNDFR3: Three-Clustered Conventional NNDFR
*G-NNDFR: Modified NNDFR With 3 Clusters And The Fuzzifier=1.2513

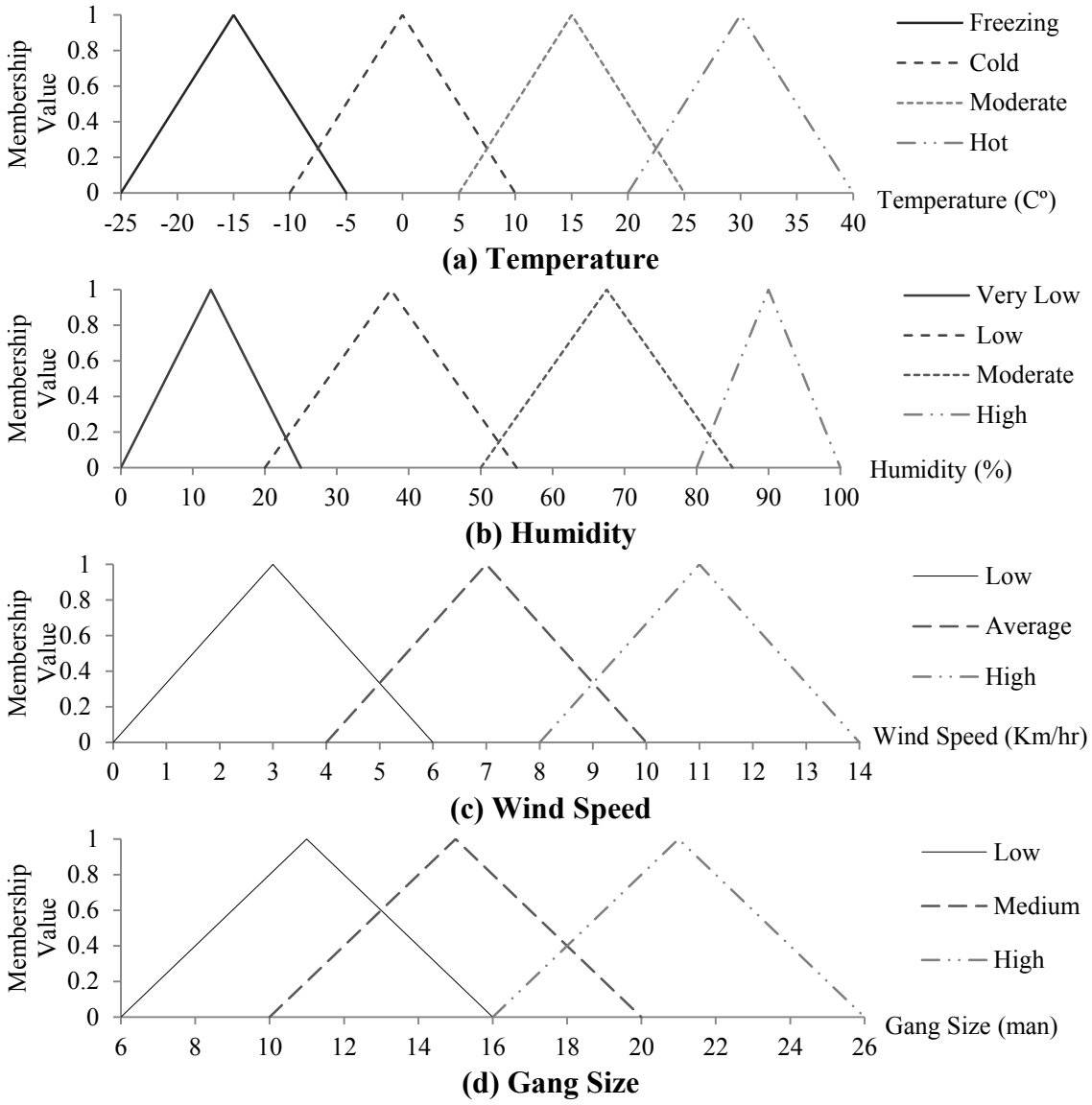
Figure 5-9: Comparison of Performances of Different Models in Terms of AVP & AIP

It is of importance to state that the same case study was used by Khan in 2005. In the research, the potential factors affecting the labor productivity of the concrete pouring operation were analyzed, ranked and selected. Different assemblies of ANN were established and tested to find the best structure with the highest performance. 221 data points were used in the research divided to two portions of testing (44 points) and training (77 points) sets. Based on the original manuscript, MSE of the best model for the whole dataset including training and testing sets is 0.0086. The modeling was also accomplished by regression method, which showed an MSE of 0.0642. Here, the main obstacle is that the testing sample error is not reported separately. As a result, comparison between the generalization ability of the aforementioned model and our Genetically-Optimized-NNDFR is not possible.

5.5 HYBRID APPROACH OF MODELING

Regarding the data gathered for the case study, there is a possibility to express some of the independent variables, e.g. Temperature, Humidity, Wind Speed, Gang Size, Labor Percentage and Floor Level, by linguistic terms. The only way to convert these linguistic terms to crisp values is to use fuzzy sets. Definitions of the fuzzy sets are subjective and case-dependent. In fact, different ranges must be defined for each variable in order for it to be presentable as a fuzzy set. The assumption made in this research is that all variables are following a triangular fuzzy set. As a result, any fuzzy variable can be defined by three values of minimum, maximum and most probable. With the consideration of the construction location, i.e. Montreal, and the involved activities, experts' opinions were used to develop the fuzzy sets. In this framework, climatic variables are defined based on

the common terms and ranges used in aerology. Variables related to the construction area are characterized based on the common construction terminology. Figure 5-10 (a) and (b) demonstrate the fuzzy sets for the temperature and humidity, respectively. The fuzzy sets related to the other factors are presented in Figure 5-10 (c) to (f).



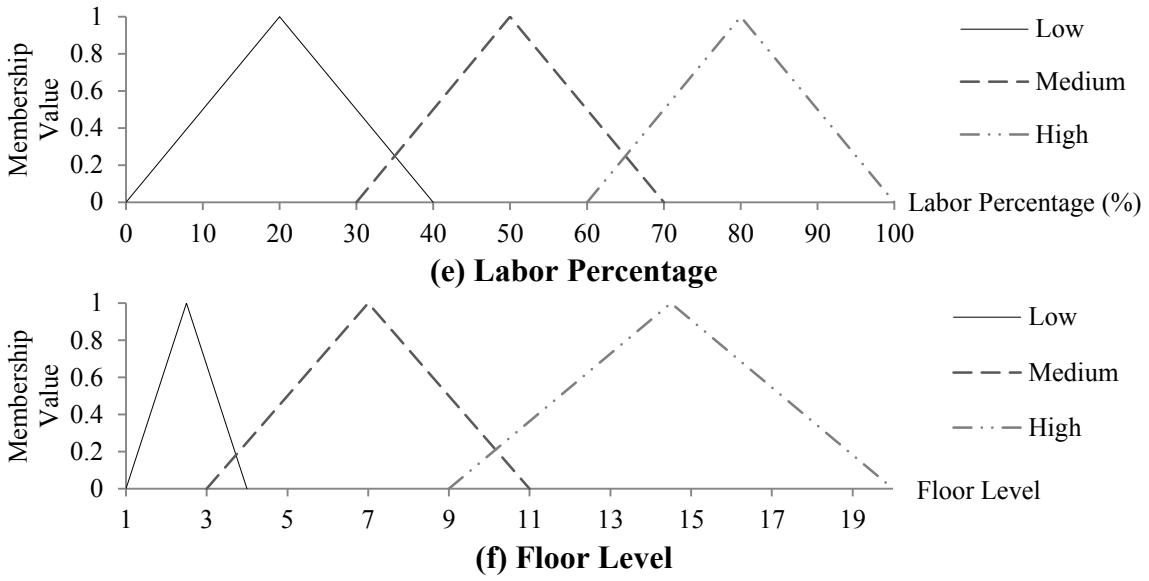


Figure 5-10: Fuzzy Sets of Different Variables

In order to show the functionality of the hybrid approach of modeling an example is presented. It is assumed that moderate temperature and high humidity are reported as fuzzy inputs, while other factors are given as crisp values. Such scenarios might happen when some variables are missing or they have not been supposed to be measured. In such cases, variables cannot be explained but in a range. Table 5-14 shows the data point that is selected to be modeled by hybrid approach.

Table 5-14: A Fuzzy-Crisp Data Point to Be Modeled By Hybrid Approach

Parameter	Value
Temperature	Fuzzy Set (Moderate)
Humidity	Fuzzy Set (High)
Precipitation	0
Wind Speed	16.6
Gang Size	18
Labor Percentage	33
Work Type	2
Floor Level	10
Work Method	1

To attain the output fuzzy set, alpha-cut is performed for different α-levels of fuzzy variables while other factors maintain their fixed values. In this example, alpha-cut technique is executed at intervals of 0.05. At each level of alpha, the temperature and humidity fuzzy sets are trimmed. Then, 100 equally spaced points are selected along each one of these two ranges. For instance, Figure 5-11 illustrates the trimmed fuzzy variables of the temperature and humidity at the α-level of 0.5. These α-level cuts the fuzzy variables such that the temperature and humidity alter within the ranges of [10, 20] and [85, 95], respectively. The model runs 100×100 times for any possible pair of the temperature and humidity from those 100 data points, while the other variables are fed to the model with their respective crisp values.

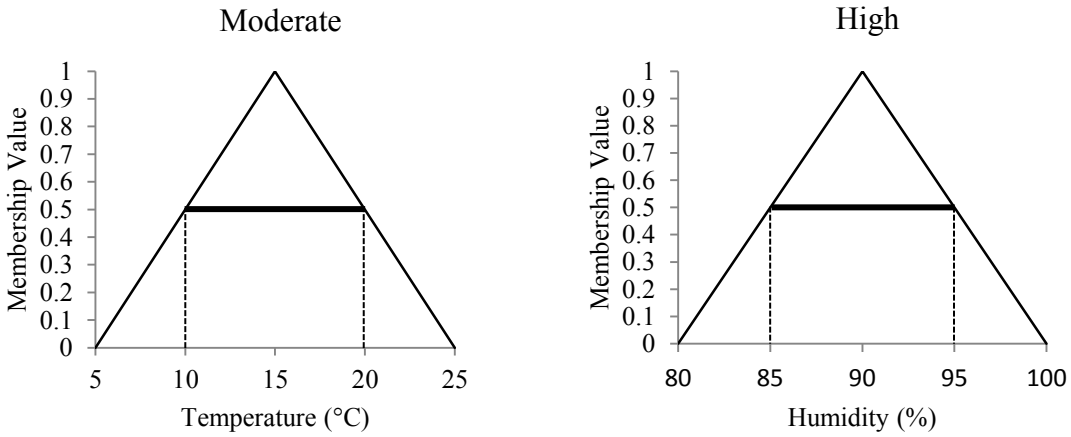


Figure 5-11: Alpha-Cut at the Level of A=0.5

Next, the maximum and minimum outcomes of the model are extracted at each level, using the following algorithm:

precip=0; wind=16.6; gang=18; labor=33; w_type=2; floor=10; w_method=1
for temp=10 to 20 stepping by 0.1

```
    for humd=85 to 95 stepping by 0.1
        input=[temp,humd,precip,wind,gang,labor,w_type,floor,w_method]
      Run output=NNDFR(input)
      If output>max then
                  max=output
       end if
      If output<min then
                  min=input
      end if

    end

end
```

The entire list of the maximum and minimum outcomes for different levels of α can be found in Table 5-15. Plotting and connecting those max and min values can reach us to a fuzzy set, which is shown in Table 5-12.

Table 5-15: Maximum and Minimum Values of Alpha-Cut Technique

Alpha Level	Min (m^3/man-hour)	Max (m^3/man-hour)
0	1.429689	2.0087
0.05	1.452947	2.0039
0.1	1.480459	1.9988
0.15	1.501737	1.9935
0.2	1.523673	1.9878
0.25	1.546778	1.9820
0.3	1.56657	1.9763
0.35	1.607938	1.9700
0.4	1.636615	1.9633
0.45	1.656327	1.9561
0.5	1.675664	1.9489
0.55	1.694613	1.9412
0.6	1.713165	1.9331
0.65	1.731311	1.9247
0.7	1.749042	1.9159
0.75	1.766351	1.9066
0.8	1.783227	1.8969
0.85	1.799657	1.8861
0.9	1.81562	1.8737
0.95	1.831083	1.8603
1	1.8501	1.8501

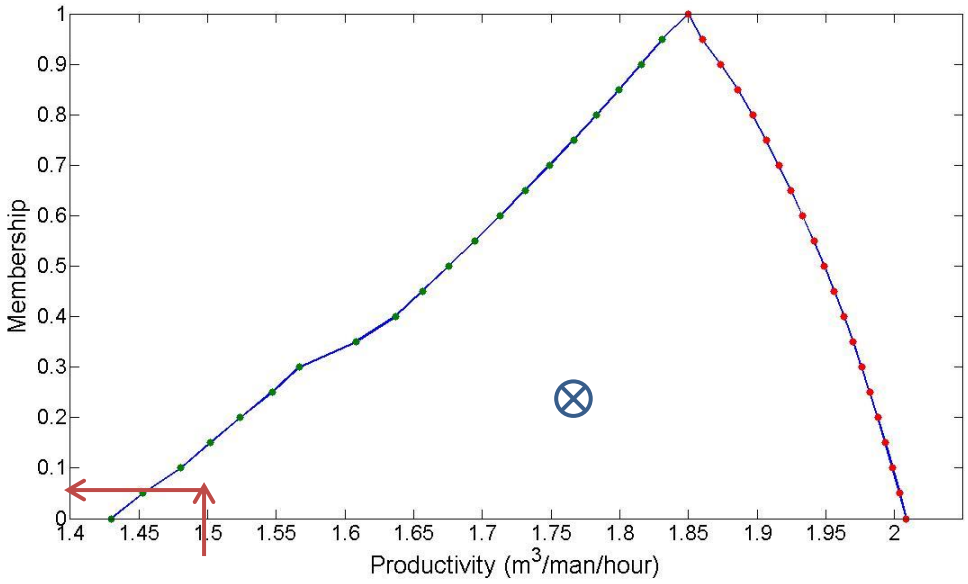


Figure 5-10: Output Fuzzy Set Traced By Alpha-Cut Technique

The output fuzzy set should be defuzzified to make it possible to represent the results with only one number. As expected, the output fuzzy set does not have a known geometric shape. Thus, the plot is divided to 40 back-to-back trapezoids in order to find the centroid based on the Equation 20. The detailed method of calculations is presented in Table 5-16. Based on these calculations, a productivity of 1.7602 is attained. Also, according to Equation 9, the possibility of having a productivity rate less than a specific value is equal to the maximum degree of membership in that range. For example, the possibility of a productivity rate less than 1.55 in this case study is:

$$P(Productivity < 1.55) = Sup_{prod<1.55}\mu_F(prod) = 0.14$$

Table 5-16: Centroid Calculation

Shape	S (area)	C (centroid)	= S * C
1	0.00058	1.41418	0.00082
2	0.00206	1.43766	0.00297
3	0.00266	1.46911	0.00391
4	0.00384	1.49025	0.00572
5	0.00520	1.51169	0.00786
6	0.00544	1.53658	0.00836
7	0.01344	1.54536	0.02078
8	0.01075	1.59328	0.01713
9	0.00838	1.62657	0.01363
10	0.00918	1.64649	0.01512
11	0.00995	1.66604	0.01657
12	0.01067	1.68520	0.01798
13	0.01134	1.70397	0.01932
14	0.01197	1.72234	0.02061
15	0.01255	1.74029	0.02184
16	0.01308	1.75782	0.02299
17	0.01355	1.77493	0.02406
18	0.01397	1.79160	0.02502
19	0.01430	1.80782	0.02586
20	0.01854	1.82149	0.03377
21	0.00991	1.84506	0.01828
22	0.01246	1.85359	0.02309
23	0.01084	1.86759	0.02025
24	0.00893	1.88076	0.01680
25	0.00749	1.89216	0.01418
26	0.00672	1.90203	0.01278
27	0.00596	1.91152	0.01139
28	0.00522	1.92058	0.01002
29	0.00471	1.92902	0.00909
30	0.00401	1.93749	0.00777
31	0.00343	1.94534	0.00667
32	0.00306	1.95257	0.00597
33	0.00252	1.96001	0.00494
34	0.00203	1.96697	0.00399
35	0.00158	1.97347	0.00313
36	0.00129	1.97926	0.00256
37	0.00100	1.98507	0.00198
38	0.00067	1.99096	0.00134
39	0.00038	1.99659	0.00076
40	0.00012	2.00233	0.00024
Sum	0.28379		0.49955
Centroid	**1.7602**		

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 SUMMARY AND CONCLUSION

The present research proposes a hybrid intelligent model to enhance the accuracy of the productivity estimation for construction operations. The current research is a response to the shortcomings of fuzzy reasoning and ANN based modeling in the construction management. The proposed framework models the effect of the qualitative as well as quantitative variables on the construction productivity and optimizes its dynamic structure according to the inherent characteristics of the data.

First, a thorough literature review was conducted to scrutinize the shortcomings in the current area of research. It was understood that the AI systems introduced to construction management do not display a satisfactory accuracy for the estimation. In addition, most of these forecasting models are only able to work with the crisp input variables. However, it is most likely to have a combination of crisp values and linguistic terms in a single modeling framework.

In light of the strengths and drawbacks of ANN and fuzzy systems, neuro-fuzzy structures appeared to be a promising solution domain. Investigation of the construction databases revealed that the data points have a tendency to be aggregated in particular areas. With respect to this fact, a Neural-Network-Driven Fuzzy Reasoning (NNDFR) structure with a high performance of estimation in the naturally clustered data spaces was

selected. It was then fine-tuned and further enhanced by a technique which allows the model deal with both types of input variables, i.e. crisp and fuzzy.

The methodology of current research encompassed three main phases, (1) modifying and training NNDFR; (2) optimizing the parameters; and (3) incorporating the hybrid approach of modeling. Given that in the conventional NNDFR membership functions are resulting from the hard clustering algorithms, the provided fuzziness is not under control and cannot be regulated. Thus, the first part of the methodology concentrated on the modification of NNDFR structure such that the fuzziness of the membership functions is adjustable. This was achieved through substituting the standard hard clustering algorithm with FCM. In the second part, model parameters that characterize the layout of model are optimized through GA. In the third part, a technique called hybrid approach is employed to simultaneously model the data sets that consist of both linguistic terms and crisp values. Hybrid approach is based on performing alpha-cut technique for the different α-levels of fuzzy variables and generating the output in form of fuzzy set.

The proposed methodology was further verified through the simulation of a construction operation in which several qualitative and quantitative factors affect the daily productivity of a concrete pouring process. The data was adopted from the construction of Engineering, Computer Science and Visual Arts complex of Concordia University. The data set, which included 131 data points, was partitioned into two sets of 117, i.e. 90% of the total, and 14, i.e. 10% of the total, data points for the training and validation/testing, respectively. Empirical results showed that the model performance in terms of testing MSE is improved by 52 percent as a result of the optimization. The same

data point were modeled by a simple ANN, ANFIS, conventional three-cluster NNDFR and the proposed Genetically Optimized NNDFR. The proposed model showed 83%, 72% and 69% improvement respectively over ANN, ANFIS and conventional NNDFR in terms of MSE. The present research helps researchers and practitioners to more effectively model construction operations using the inherent features of the data for fine-tuning the model.

6.2 RESEARCH CONTRIBUTIONS

The contributions of the present research to the AI modeling in construction are to:

1) Develop a modified NNDFR model to forecast the characteristics (productivity, time and, etc.) of construction operations;
2) Develop tunable hyper-surface membership functions enjoying both FCM and learning ability of ANN. These multi-dimensional membership functions consider interdependence of input variables;
3) Optimize the fuzziness of membership functions via GA;
4) Determine the optimum number of clusters in a specific data space using GA;
5) Develop a framework which lets us model a combination of crisp values and linguistic terms simultaneously through NNDFR model.

6.3 RESEARCH LIMITATIONS

The developed framework has the following limitations:

1) The research does not perform a variable selection on the parameters to neglect neutral or ineffective variables;
2) In the hybrid approach of modeling, alpha cut technique must be performed for all the possible combinations of the fuzzy variables at all levels of alpha. This demands a huge computational time and effort, especially in higher dimensions;
3) Although GA shows a great capability to prevent local minima, there is still a possibility for the algorithm to terminate without reaching to the global optimum parameters;
4) Since the model is constituted from several independent ANNs, it needs more historical data points to feed each network with a portion of the data set.

6.4 FUTURE RECOMMENDATIONS AND WORKS

The present research can be further enhanced through the following steps:

1) More variables, such as managerial conditions and the exact time of the work during the daytime, should be incorporated into the model to improve the efficiency of prediction. The more comprehensive and thorough the set of variables are, the better a model can mimic the behavior of the system. Then, a variable selection algorithm is required to select the most relevant variables that can improve the efficiency of the model.
2) The forecasting model can be validated by a greater number of testing data points to verify its generalization ability. The current model is trained by a limited number of data points and obviously the model responds better to the testing data points that are

more similar to the training set. Thus, the testing data set must be rich enough to cover the entire problem domain and be an indicator of the generalization ability of the model.

3) The optimization of the model's parameters should be executed by another algorithm that can completely eliminate the chance of local minima. Although, GA can lower the chance of falling in local minima by increasing the number of generations and population, still it cannot be guaranteed.
4) Maximum and minimum outputs of the model for each level of alpha-cut can be found by a time-efficient EA instead of the provided exhaustive search. In case of having too many fuzzy variables, all possible combinations of the input variables must be tested. This process demands a considerable amount of time and computational efforts.

The present research can be also be further extended through the investigation of the following areas:

1) It is recommended to work on finding a method which is able to visualize the hyper-surface membership functions. In this way, the multi-dimensional membership functions are more tangible and practical for the users.
2) A Graphic User Interface (GUI) can be developed in order to automate and visualize the reasoning mechanism of the presented model. In addition, the GUI can be connected to an updating database, which lets the model work in real-time mode.
3) The developed productivity estimation model should be integrated with other cost and time forecasting models. This will establish an integrated framework where all of the

interactions between the main decision criteria of project planning and control can be studied.

REFERENCES

Abebe, A., Guinot, V., and Solomatine, D. (2000). "Fuzzy alpha-cut vs. Monte Carlo techniques in assessing uncertainty in model parameters." *Proceedings of the 4th International Conference on Hydroinformatics,* Iowa City, USA.

AbouRizk, S. (2010). "Role of simulation in construction engineering and management." *J.Constr.Eng.Manage.,* 136(10), 1140-1153.

Alpaydin, E. (2004). *Introduction to machine learning.* MIT press, Cambridge, Massachusetts, USA.

Banks, J. (2005). *Discrete Event System Simulation, 4/e.* Pearson Education India, NJ, USA.

Bellman, R. E. (1961). *Adaptive control processes: a guided tour.* Princeton University Press, Princeton, NJ, USA.

Beringer, J., and Hüllermeier, E. (2006). "Online clustering of parallel data streams." *Data Knowl.Eng.,* 58(2), 180-204.

Berry, M. J., and Linoff, G. (1997). *Data mining techniques: For marketing, sales, and customer support.* John Wiley & Sons, Inc., Hoboken, NJ, USA.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms.* Kluwer Academic Publishers, New York City, USA.

Boussabaine, A. H. (1996). "The use of artificial neural networks in construction management: a review." *Construction Management & Economics,* 14(5), 427-436.

Bowen, P., and Edwards, P. (1985). "Cost modelling and price forecasting: practice and theory in perspective." *Constr.Manage.Econ.,* 3(3), 199-215.

Bryson, A. E., and Ho, Y. (1975). *Applied optimal control: optimization, estimation, and control.* Taylor & Francis, UK.

Carr, V., and Tah, J. (2001). "A fuzzy approach to construction project risk assessment and analysis: construction project risk management system." *Adv.Eng.Software,* 32(10), 847-857.

Chan, A. P., Chan, D. W., and Yeung, J. F. (2009). "Overview of the application of "fuzzy techniques" in construction management research." *J.Constr.Eng.Manage.,* 135(11), 1241-1252.

Chang, T., Ibbs, C. W., and Crandall, K. C. (1990). "Network resource allocation with support of a fuzzy expert system." *J.Constr.Eng.Manage.,* 116(2), 239-260.

Cheng, M., and Ko, C. (2003). "Object-oriented evolutionary fuzzy neural inference system for construction management." *J.Constr.Eng.Manage.,* 129(4), 461-469.

Chiu, S. L. (1994). "Fuzzy model identification based on cluster estimation." *Journal of Intelligent and Fuzzy Systems,* 2(3), 267-278.

Davies, D. L., and Bouldin, D. W. (1979). "A cluster separation measure." *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* (2), 224-227.

Davis, L. (1991). *Handbook of genetic algorithms.* Van Nostrand Reinhold, New York, NY, USA.

Dong, W., and Shah, H. C. (1987). "Vertex method for computing functions of fuzzy variables." *Fuzzy Sets Syst.,* 24(1), 65-78.

Dubois, D., Prade, H. M., Farreny, H., Martin-Clouaire, R., and Testemale, C. (1988). *Possibility theory: an approach to computerized processing of uncertainty.* Plenum press, New York, USA.

Dunn, J. C. (1973). *A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters.* Taylor & Francis, UK, 32-57.

Dunn, J. C. (1974). "Well-separated clusters and optimal fuzzy partitions." *Journal of Cybernetics,* 4(1), 95-104.

El Wakil, E., and Zayed, T. (2012). "Data Management for Construction Processes Using Fuzzy Approach." *Construction Research Congress 2012@ sConstruction Challenges in a Flat World,* ASCE, West Lafayette, IN, USA, 1222-1231.

Elwakil, E. (2011). “Knowledge Discovery Based Simulation System in Construction, Doctoral Dissertation.” *Concordia University,* Montreal, Canada.

Graham, D., and Smith, S. D. (2004). "Estimating the productivity of cyclic construction operations using case-based reasoning." *Advanced Engineering Informatics,* 18(1), 17-28.

Guyonnet, D., Bourgine, B., Dubois, D., Fargier, H., Côme, B., and Chilès, J. (2002). "Hybrid approach for addressing uncertainty in risk assessments." *J.Environ.Eng.,* 129(1), 68-78.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). "On clustering validation techniques." *J Intell Inform Syst,* 17(2-3), 107-145.

Hammouda, K., and Karray, F. (2000). "A comparative study of data clustering techniques." *Tools of Intelligent Systems Design.Course Project SYDE* 625, University of Waterloo, ON, Canada.

Hanna, A. S., Russell, J. S., Nordheim, E. V., and Bruggink, M. J. (1999). "Impact of change orders on labor efficiency for electrical construction." *J.Constr.Eng.Manage.,* 125(4), 224-232.

Haupt, R. L., and Haupt, S. E. (2004). *Practical genetic algorithms.* John Wiley & Sons, Inc., Hoboken, NJ, USA.

Hegazy, T., and Moselhi, O. (1994). "Analogy-based solution to markup estimation problem." *J.Comput.Civ.Eng.,* 8(1), 72-87.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, Ann Arbor, MI, USA.

Jain, L. C., and Jain, R. K. (1997). *Hybrid intelligent engineering systems.* World Scientific, Singapore.

Jang, J. (1993). "ANFIS: adaptive-network-based fuzzy inference system." *Systems, Man and Cybernetics, IEEE Transactions on,* 23(3), 665-685.

Jantzen, J. (1998). "Neurofuzzy modelling." *Technical University of Denmark, Department of Automation Aerospace Corp., Tech. report* 98-H: 874, Los Angeles, CA, USA.

Kelton, W. D., and Law, A. M. (2000). *Simulation modeling and analysis.* McGraw Hill Boston, MA, USA.

Ketchen, D. J., and Shook, C. L. (1996). "The application of cluster analysis in strategic management research: an analysis and critique." *Strategic Manage.J.,* 17(6), 441-458.

Khan, Z. U. (2005). "Modeling and Parameter Ranking of Construction Labor Productivity." *M.S. Thesis, Concordia University,* Montreal, Canada.

Kim, G., An, S., and Kang, K. (2004). "Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning." *Build.Environ.,* 39(10), 1235-1242.

Koehn, E., and Brown, G. (1985). "Climatic effects on construction." *J.Constr.Eng.Manage.,* 111(2), 129-137.

Kovács, F., Legány, C., and Babos, A. (2005). "Cluster validity measurement techniques." *6th International Symposium of Hungarian Researchers on Computational Intelligence,* Hungary .

Kulik, L. (2001). "A geometric theory of vague boundaries based on supervaluation." *Spatial information theory,* Springer Berlin Heidelberg, 44-59.

Kurzweil, R., Schneider, M. L., and Schneider, M. L. (1990). *The age of intelligent machines.* MIT press, Cambridge, MA, USA.

Lawrence, J. (1994). *Introduction to neural networks: design, theory, and applications.* California Scientific Software, CA, USA.

Leu, S., Chen, A., and Yang, C. (2001). "A GA-based fuzzy optimal model for construction time–cost trade-off." *Int.J.Project Manage.,* 19(1), 47-58.

Li, H. (1995). "Neural networks for construction cost estimation." *Building Research & Information,* 23(5), 279-284.

Liu, W., Xiao, C., Wang, B., Shi, Y., and Fang, S. (2003). "Study on combining subtractive clustering with fuzzy c-means clustering." *Machine Learning and Cybernetics, 2003 International Conference on,* IEEE, Xi'an, China, 2659-2662.

Lloyd, S. (1982). "Least squares quantization in PCM." *Information Theory, IEEE Transactions on,* 28(2), 129-137.

Lotfi Asker Zadeh. (1996). *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh.* World Scientific, Singapore.

Luger, G. F. (2005). *Artificial intelligence: Structures and strategies for complex problem solving.* Pearson education, Upper Saddle River, NJ, USA.

MacKay. (1992). " Information-based objective functions for active data selection." *Neural Computation,* 4(3), 415–447.

MacQueen, J. (1967). "Some methods for classification and analysis of multivariate observations." *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* CA, USA, 14.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press, Cambridge, MA, USA.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1980). *Multivariate analysis*. Academic Press, Waltham, MA, USA.

Martin Skitmore, R., and Thomas Ng, S. (2003). "Forecast models for actual construction time and cost." *Build.Environ.,* 38(8), 1075-1083.

Mathworks. (2013). "*Bayesian regulation backpropagation*." http://www.mathworks.com/help/nnet/ref/trainbr.html (07/15, 2013).

Mathworks. (2012). "Matlab User's Guide." *Rep. No. 2013,* Mathworks, MA, USA.

Mathworks. (2012). "Product Help 2012a." Mathworks, MA, USA.

Mathworks. (2012). "*Subtractive Clustering*." *Product Help 2012a,* http://www.mathworks.com/help/fuzzy/subclust.html (07/15, 2013).

Matteucci, M. (2008). "A tutorial on clustering algorithms." *See at: Http://home.Dei.Polimi.it/matteucc/Clustering/tutorial/html/index.Html* (07/15, 2013).

Moselhi, O., Gong, D., and El-Rayes, K. (1997). "Estimating weather impact on the duration of construction activities." *Canadian Journal of Civil Engineering,* 24(3), 359-366.

Moselhi, O., Hegazy, T., and Fazio, P. (1992). "Potential applications of neural networks in construction." *Canadian Journal of Civil Engineering,* 19(3), 521-529.

Moselhi, O., Leonard, C., and Fazio, P. (1991). "Impact of change orders on construction productivity." *Canadian Journal of Civil Engineering,* 18(3), 484-492.

Oglesby, C. H., Parker, H. W., and Howell, G. A. (1989). *Productivity improvement in construction.* McGraw-Hill, New York, USA.

Paek, J. H., Lee, Y. W., and Napier, T. R. (1992). "Selection of design/build proposal using fuzzy-logic system." *J.Constr.Eng.Manage.,* 118(2), 303-317.

Pappis, C. P., and Siettos, C. I. (2005). "Fuzzy reasoning." *Search Methodologies,* Springer, 437-474.

Park, H. (2006). "Conceptual framework of construction productivity estimation." *KSCE Journal of Civil Engineering,* 10(5), 311-317.

Park, H., Thomas, S. R., and Tucker, R. L. (2005). "Benchmarking of construction productivity." *J.Constr.Eng.Manage.,* 131(7), 772-778.

Ren, Q., Baron, L., and Balazinski, M. (2012). "Fuzzy identification of cutting acoustic emission with extended subtractive cluster analysis." *Nonlinear Dyn.,* 67(4), 2599-2608.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (2002). "Learning representations by back-propagating errors." *Cognitive Modeling,* 1 213.

Sadeghi, N., Fayek, A. R., and Pedrycz, W. (2010). "Fuzzy Monte Carlo simulation and risk assessment in construction." *Computer-Aided Civil and Infrastructure Engineering,* 25(4), 238-252.

Sanders, S. R., and Thomas, H. R. (1993). "Masonry productivity forecasting model." *J.Constr.Eng.Manage.,* 119(1), 163-179.

Sawhney, A., AbouRizk, S. M., and Halpin, D. W. (1998). "Construction project simulation using CYCLONE." *Canadian Journal of Civil Engineering,* 25(1), 16-25.

Sawhney, A., and Mund, A. (2002). "Adaptive probabilistic neural network-based crane type selection system." *J.Constr.Eng.Manage.,* 128(3), 265-273.

Shackle, G. L. S. (2010). *Decision order and time in human affairs*. Cambridge University Press, Cambridge, England.

Shafer, G. (1976). *A mathematical theory of evidence.* Princeton University Press, Princeton, NJ, USA.

Smith, S. D. (1999). "Earthmoving productivity estimation using linear regression techniques." *J.Constr.Eng.Manage.,* 125(3), 133-141.

Steinhaus, H. (1957). "Sur la division des corps matériels en parties." *Bulletin L'Académie Polonaise Des Science,* 4(12), 801–804.

Tah, J., and Carr, V. (2000). "A proposal for construction project risk assessment using fuzzy logic." *Construction Management & Economics,* 18(4), 491-500.

Takagi, H., and Hayashi, I. (1991). "NN-driven fuzzy reasoning." *International Journal of Approximate Reasoning,* 5(3), 191-212.

Takagi, T., and Sugeno, M. (1985). "Fuzzy identification of systems and its applications to modeling and control." *Systems, Man and Cybernetics, IEEE Transactions on,* (1), 116-132.

Wang, F. (2005). "On-Site Labor Productivity Estimation using Neural Networks." *M.S. Thesis, Concordia University,* Montreal, Canada.

Werbos, P. (1974). "Beyond regression: New tools for prediction and analysis in the behavioral sciences." *Doctoral Dissertation, Harvard University*, MA, USA.

Wonneberger, S., Kistinger, S., and Deckert, A. (1995). "Unbiased guess, a concept to cope with fuzzy and random parameters?" *European Commission Rep.no.EUR 16199 EN*.

Xexéo, G. (2009). "Fuzzy Logic." *Federal University of Rio de Janeiro, See at:* http://mida.. edor. it/documentazioneVaria/mida8/pdf/FuzzyLogic (07/15, 2013).

Yager, R., and Filev, D. (1994). "Generation of fuzzy rules by mountain clustering." *Journal of Intelligent and Fuzzy Systems,* 2(3), 209-219.

Yager, R. R., and Filev, D. P. (1994). "Approximate clustering via the mountain method." *Systems, Man and Cybernetics, IEEE Transactions on,* 24(8), 1279-1284.

Zadeh, L. A. (1977). *Theory of Fuzzy Sets.* Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, USA.

Zadeh, L. A. (1999). "Fuzzy sets as a basis for a theory of possibility." *Fuzzy Sets Syst.,* 100 9-34.

Zayed, T. M., and Halpin, D. W. (2005). "Pile construction productivity assessment." *J.Constr.Eng.Manage.,* 131(6), 705-714.

Zhang, H., Tam, C., and Shi, J. J. (2002). "Application of fuzzy logic to simulation for construction operations." *J.Comput.Civ.Eng.,* 17(1), 38-45.

Zhu, S., Lee, S., Hargrove, S., and Chen, G. (2007). "Prediction of combustion efficiency of chicken litter using an artificial neural network approach." *Fuel,* 86(5), 877-886.

APPENDIX A

Data Points

Table A - 1: All Data Points

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
-8	87	2	14.2	22	36	1	3	1	1.27
-8	87	2	14.2	23	30	2	3	1	1.14
-12.5	54	0	5.2	21	38	1	3	1	1.17
-12.5	54	0	5.2	20	30	2	3	1	1.04
-16	55	0	6	23	35	1	3	1	1.16
-15	51	2	18.7	17	29	2	4	1	1.99
-15	51	2	18.7	20	40	1	4	1	1.1
-8.5	58	0	26.5	18	33	2	4	1	1
-4	87	2	3.6	22	36	1	4	1	1.55
-14	42	0	10	23	35	2	4	1	1.26
-14.5	42	0	7.5	19	33	2	4	1	1.14
-14.5	42	0	7.5	16	37	1	4	1	1.27
1.5	85	0	9.4	21	33	1	5	1	1.45
-0.5	53	0	7.5	20	30	1	5	1	1.51
-0.5	53	0	7.5	22	36	2	5	1	1.37
-3.5	47	0	20	17	29	1	5	1	1.38
-3.5	47	0	20	22	36	2	5	1	1.25
-4	81	1	11.9	22	36	1	5	1	1.49
-4	81	1	11.9	16	38	2	5	1	1.34
3	97	0	8	22	36	1	5	1	1.36
3	97	0	8	15	40	2	5	1	1.22
2.5	92	0	6.2	19	42	1	6	2	1.34
2.5	92	0	6.2	18	33	2	6	1	1.2
3.5	88	1	7.6	24	38	1	6	2	1.39
4.5	86	1	9.1	24	38	1	6	2	1.41
4.5	86	1	9.1	22	36	2	6	1	1.26
-4.5	48	0	14.1	19	33	1	7	2	1.36
-4.5	48	0	14.1	20	30	2	7	1	1.21
-6.5	56	0	10.5	20	30	1	7	2	1.34
-6.5	56	0	10.5	21	33	2	7	1	1.09
-2.5	39	0	10	20	30	1	7	2	1.32

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
-2.5	39	0	10	20	30	2	7	1	1.37
-6	37	0	19.9	19	33	1	8	2	1.23
-7	41	0	7.9	20	30	1	8	2	1.47
-7	41	0	7.9	20	30	2	8	1	1.34
-4.5	53	2	13.1	21	33	1	8	2	1.49
-4.5	53	2	13.1	18	33	2	8	1	1.35
6.5	45	0	11.3	24	38	1	9	2	1.67
6.5	45	0	11.3	21	33	2	9	1	1.51
5.5	46	0	12	22	36	1	9	2	1.65
5.5	46	0	12	19	33	2	10	1	1.48
4.5	84	1	8.7	20	30	1	10	2	1.57
4.5	84	1	8.7	18	33	2	10	1	1.41
-5	57	0	15.8	19	33	1	10	2	1.56
-5	57	0	15.8	19	33	2	10	1	1.4
2	36	0	16.6	19	33	1	10	2	1.63
2	36	0	16.6	18	33	2	10	1	1.46
7	90	1	5.4	16	31	1	10	2	1.73
3	56	0	13.4	18	33	1	11	2	1.74
3	56	0	13.4	19	33	2	11	1	1.55
11	44	0	13.4	16	31	1	11	2	1.87
11	44	0	13.4	15	33	2	11	1	1.68
7.5	40	0	8	16	31	1	11	2	1.67
7.5	40	0	8	16	31	2	11	1	1.52
12	40	0	18	18	33	3	12	1	1.1
18	59	0	23	20	35	2	12	1	1.45
16	73	1	14	21	33	1	12	1	1.54
16	61	0	3	22	36	1	13	1	2.4
15	64	1	19	19	37	1	13	1	1.49
16	60	0	6	22	36	1	13	1	2.25
18	58	0	6	21	33	1	13	1	2.2
20	57	0	10	23	35	2	13	1	1.62
17	75	1	16	19	37	2	13	1	1.33
22	56	0	10	22	36	2	13	1	1.75
25	57	1	11	19	37	1	13	1	1.43
25	77	0	24	20	30	1	14	1	1.65
21	63	0	16	20	30	1	12	2	1.55
23	77	0	13	18	33	2	12	1	1.49
24	65	0	19	18	33	2	12	1	1.52
24	73	0	11	23	35	1	13	2	1.76

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
25	69	0	6	22	36	1	13	2	1.75
25	71	0	21	21	33	1	13	2	1.73
21	71	0	10	21	33	1	13	2	1.91
23	60	0	19	22	36	2	13	1	1.79
25	66	0	18	16	38	2	13	1	1.77
25	65	0	13	15	40	2	13	1	1.8
25	65	0	24	17	29	2	13	1	1.42
18	71	0	19	20	30	1	14	2	2
14	70	0	14	23	30	2	14	1	1.78
17.61	61	0	16	22	32	1	14	2	2.42
17	72	0	16	22	32	1	14	2	2.31
21	72	1	21	20	35	1	14	2	2.09
17	73	0	13	20	35	2	15	1	1.8
14	71	0	5	20	35	2	15	1	1.85
13	60	0	13	19	37	2	15	1	1.88
15	67	0	14	19	37	2	15	1	1.78
21	75	0	8	21	33	1	15	2	2.33
20	73	0	23	20	30	1	15	2	2.09
16	72	0	8	20	30	1	15	2	2.32
17	68	0	6	20	30	1	15	2	2.34
21	61	0	18	18	33	2	16	1	1.88
6	82	1	13	19	37	2	16	1	1.65
7	69	0	18	22	37	1	16	2	2.33
13	64	0	19	21	33	1	16	2	2.38
14	70	0	11	12	33	1	10	2	2.53
13	70	0	18	12	33	1	10	2	2.5
5	75	0	10	10	40	1	10	2	2.34
4	76	0	14	10	40	1	10	2	1.98
3	96	1	27	8	38	1	10	2	1.74
6	96	1	6	9	44	2	10	1	1.5
6	76	0	16	11	37	2	10	1	1.74
6	94	1	14	9	44	1	11	2	1.82
12	70	0	10	12	33	1	11	2	2.1
5	75	0	10	12	33	1	11	2	2.02
6	96	0	14	12	33	1	11	2	1.97
3	78	0	13	11	37	1	11	2	2
3	77	0	21	10	40	2	11	1	1.77
8	79	0	13	9	44	2	11	1	1.83
3	79	0	13	11	37	1	12	2	1.96

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
5	88	0	11	11	37	1	12	2	2.03
3	80	0	6	11	37	1	12	2	1.99
3	74	0	6	12	42	1	12	2	1.98
3	97	0	21	9	33	2	12	1	1.64
-7	53	0	19	9	33	2	12	1	1.24
-7	74	0	18	8	37	2	12	1	1.22
-10	87	0	6	12	42	1	13	2	1.31
-3	84	0	16	11	37	1	13	2	1.73
3	97	1	29	11	37	1	13	2	1.29
-5	90	3	26	11	37	1	13	2	1.34
-8	90	0	11	12	42	1	13	2	1.38
-1	93	1	14	11	37	1	13	2	1.23
-6	75	0	26	8	37	2	13	1	1.28
-9	76	0	18	11	37	1	14	2	1.45
-17	48	0	29	8	37	2	14	1	1.11
2	76	0	5	9	33	2	16	1	1.51
3	79	0	14	8	37	2	16	1	1.44
-6	41	0	3	11	37	1	17	2	1.47
-1	71	0	16	8	37	2	17	1	1.36
-12	49	0	26	9	33	2	17	1	1.18
5	48	0	19	12	42	1	17	2	1.52
8	42	0	11	12	42	1	17	2	1.67

Table A - 2: Training Data Set

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
-8	87	2	14.2	22	36	1	3	1	1.27
-8	87	2	14.2	23	30	2	3	1	1.14
-12.5	54	0	5.2	21	38	1	3	1	1.17
-12.5	54	0	5.2	20	30	2	3	1	1.04
-16	55	0	6	23	35	1	3	1	1.16
-15	51	2	18.7	17	29	2	4	1	1.99
-15	51	2	18.7	20	40	1	4	1	1.1
-8.5	58	0	26.5	18	33	2	4	1	1
-4	87	2	3.6	22	36	1	4	1	1.55

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
-14	42	0	10	23	35	2	4	1	1.26
-14.5	42	0	7.5	19	33	2	4	1	1.14
-14.5	42	0	7.5	16	37	1	4	1	1.27
-0.5	53	0	7.5	20	30	1	5	1	1.51
-0.5	53	0	7.5	22	36	2	5	1	1.37
-3.5	47	0	20	17	29	1	5	1	1.38
-4	81	1	11.9	22	36	1	5	1	1.49
-4	81	1	11.9	16	38	2	5	1	1.34
3	97	0	8	22	36	1	5	1	1.36
2.5	92	0	6.2	19	42	1	6	2	1.34
2.5	92	0	6.2	18	33	2	6	1	1.2
3.5	88	1	7.6	24	38	1	6	2	1.39
4.5	86	1	9.1	24	38	1	6	2	1.41
4.5	86	1	9.1	22	36	2	6	1	1.26
-4.5	48	0	14.1	19	33	1	7	2	1.36
-4.5	48	0	14.1	20	30	2	7	1	1.21
-6.5	56	0	10.5	20	30	1	7	2	1.34
-6.5	56	0	10.5	21	33	2	7	1	1.09
-2.5	39	0	10	20	30	1	7	2	1.32
-2.5	39	0	10	20	30	2	7	1	1.37
-6	37	0	19.9	19	33	1	8	2	1.23
-7	41	0	7.9	20	30	1	8	2	1.47
-7	41	0	7.9	20	30	2	8	1	1.34
-4.5	53	2	13.1	21	33	1	8	2	1.49
6.5	45	0	11.3	24	38	1	9	2	1.67
6.5	45	0	11.3	21	33	2	9	1	1.51
5.5	46	0	12	22	36	1	9	2	1.65
5.5	46	0	12	19	33	2	10	1	1.48
4.5	84	1	8.7	20	30	1	10	2	1.57
4.5	84	1	8.7	18	33	2	10	1	1.41
-5	57	0	15.8	19	33	1	10	2	1.56
-5	57	0	15.8	19	33	2	10	1	1.4
2	36	0	16.6	19	33	1	10	2	1.63
2	36	0	16.6	18	33	2	10	1	1.46
7	90	1	5.4	16	31	1	10	2	1.73
3	56	0	13.4	18	33	1	11	2	1.74
3	56	0	13.4	19	33	2	11	1	1.55
11	44	0	13.4	16	31	1	11	2	1.87
11	44	0	13.4	15	33	2	11	1	1.68

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
7.5	40	0	8	16	31	1	11	2	1.67
7.5	40	0	8	16	31	2	11	1	1.52
12	40	0	18	18	33	3	12	1	1.1
18	59	0	23	20	35	2	12	1	1.45
16	73	1	14	21	33	1	12	1	1.54
16	61	0	3	22	36	1	13	1	2.4
15	64	1	19	19	37	1	13	1	1.49
16	60	0	6	22	36	1	13	1	2.25
18	58	0	6	21	33	1	13	1	2.2
20	57	0	10	23	35	2	13	1	1.62
17	75	1	16	19	37	2	13	1	1.33
25	57	1	11	19	37	1	13	1	1.43
25	77	0	24	20	30	1	14	1	1.65
21	63	0	16	20	30	1	12	2	1.55
23	77	0	13	18	33	2	12	1	1.49
24	65	0	19	18	33	2	12	1	1.52
24	73	0	11	23	35	1	13	2	1.76
25	69	0	6	22	36	1	13	2	1.75
21	71	0	10	21	33	1	13	2	1.91
23	60	0	19	22	36	2	13	1	1.79
25	66	0	18	16	38	2	13	1	1.77
25	65	0	13	15	40	2	13	1	1.8
25	65	0	24	17	29	2	13	1	1.42
18	71	0	19	20	30	1	14	2	2
14	70	0	14	23	30	2	14	1	1.78
17.61	61	0	16	22	32	1	14	2	2.42
17	72	0	16	22	32	1	14	2	2.31
21	72	1	21	20	35	1	14	2	2.09
14	71	0	5	20	35	2	15	1	1.85
13	60	0	13	19	37	2	15	1	1.88
15	67	0	14	19	37	2	15	1	1.78
21	75	0	8	21	33	1	15	2	2.33
20	73	0	23	20	30	1	15	2	2.09
16	72	0	8	20	30	1	15	2	2.32
17	68	0	6	20	30	1	15	2	2.34
21	61	0	18	18	33	2	16	1	1.88
6	82	1	13	19	37	2	16	1	1.65
7	69	0	18	22	37	1	16	2	2.33
13	64	0	19	21	33	1	16	2	2.38

Temperature (ºC)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
14	70	0	11	12	33	1	10	2	2.53
13	70	0	18	12	33	1	10	2	2.5
5	75	0	10	10	40	1	10	2	2.34
4	76	0	14	10	40	1	10	2	1.98
3	96	1	27	8	38	1	10	2	1.74
6	96	1	6	9	44	2	10	1	1.5
6	76	0	16	11	37	2	10	1	1.74
6	94	1	14	9	44	1	11	2	1.82
12	70	0	10	12	33	1	11	2	2.1
5	75	0	10	12	33	1	11	2	2.02
3	78	0	13	11	37	1	11	2	2
8	79	0	13	9	44	2	11	1	1.83
3	79	0	13	11	37	1	12	2	1.96
5	88	0	11	11	37	1	12	2	2.03
3	80	0	6	11	37	1	12	2	1.99
3	74	0	6	12	42	1	12	2	1.98
3	97	0	21	9	33	2	12	1	1.64
-7	53	0	19	9	33	2	12	1	1.24
-7	74	0	18	8	37	2	12	1	1.22
-10	87	0	6	12	42	1	13	2	1.31
-3	84	0	16	11	37	1	13	2	1.73
3	97	1	29	11	37	1	13	2	1.29
-1	93	1	14	11	37	1	13	2	1.23
-6	75	0	26	8	37	2	13	1	1.28
-9	76	0	18	11	37	1	14	2	1.45
-17	48	0	29	8	37	2	14	1	1.11
2	76	0	5	9	33	2	16	1	1.51
-12	49	0	26	9	33	2	17	1	1.18
5	48	0	19	12	42	1	17	2	1.52
8	42	0	11	12	42	1	17	2	1.67

Table A - 3: Testing Data Set

Temperature (°C)	Humidity (%)	Precipitation	Wind Speed (km/h)	Gang Size (workers)	Labor Percentage (%)	Work Type	Floor Level	Work Method	Daily Productivity (m^3/man-hour)
1.5	85	0	9.4	21	33	1	5	1	1.45
-3.5	47	0	20	22	36	2	5	1	1.25
3	97	0	8	15	40	2	5	1	1.22
-4.5	53	2	13.1	18	33	2	8	1	1.35
22	56	0	10	22	36	2	13	1	1.75
25	71	0	21	21	33	1	13	2	1.73
17	73	0	13	20	35	2	15	1	1.8
6	96	0	14	12	33	1	11	2	1.97
3	77	0	21	10	40	2	11	1	1.77
-5	90	3	26	11	37	1	13	2	1.34
-8	90	0	11	12	42	1	13	2	1.38
3	79	0	14	8	37	2	16	1	1.44
-6	41	0	3	11	37	1	17	2	1.47
-1	71	0	16	8	37	2	17	1	1.36

GA Charts

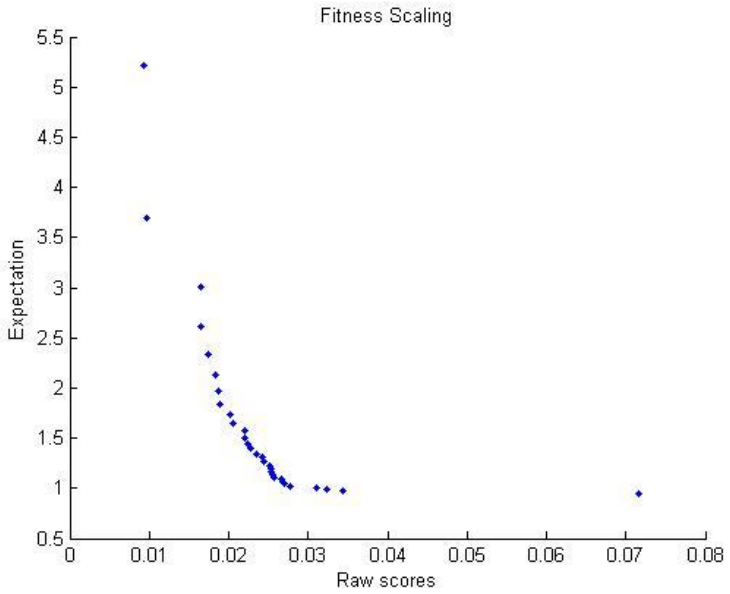


Figure A - 1: GA Fitness Scaling Chart

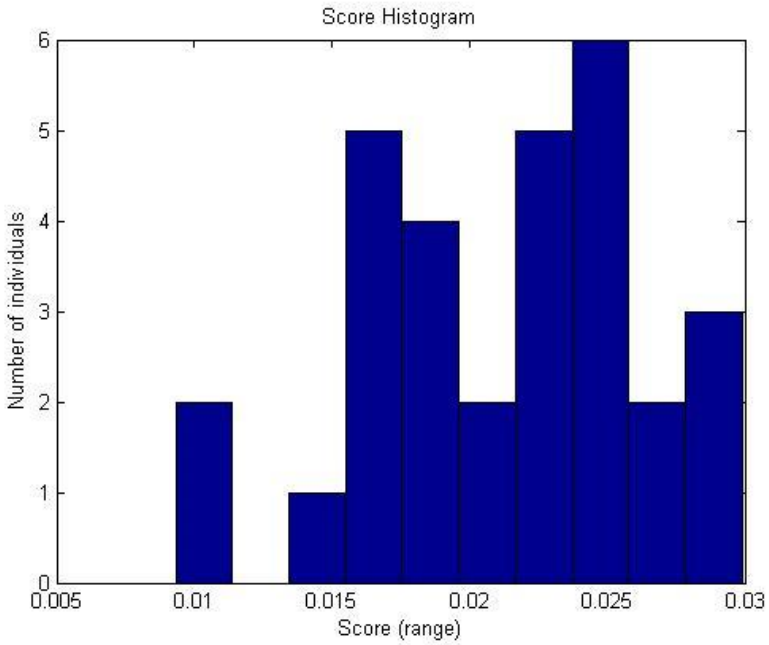


Figure A - 2: GA Score Histogram

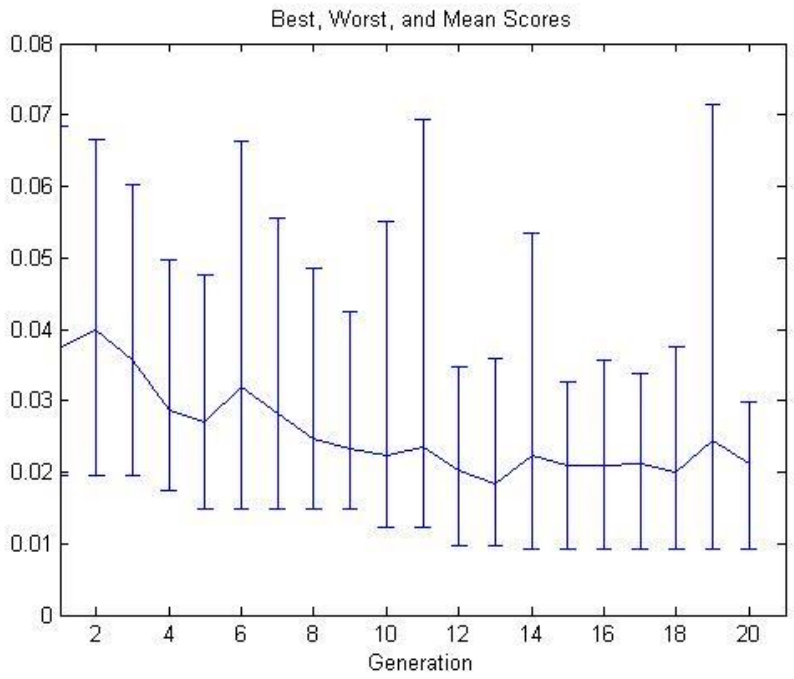


Figure A - 3: GA Best, Worst And Mean Scores

ANFIS Screenshots

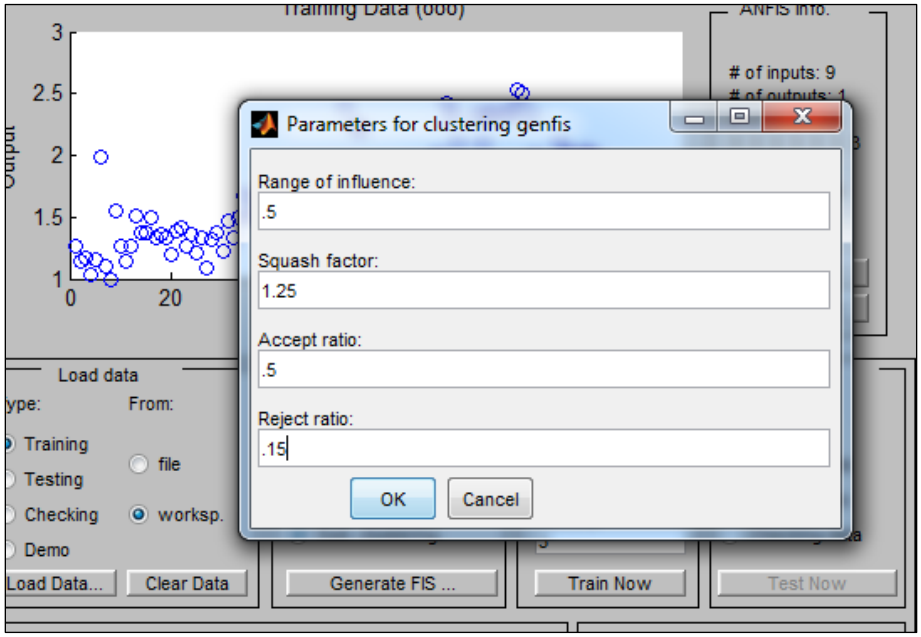


Figure A - 4: Setting Subtractive Clustering Parameters In ANFIS

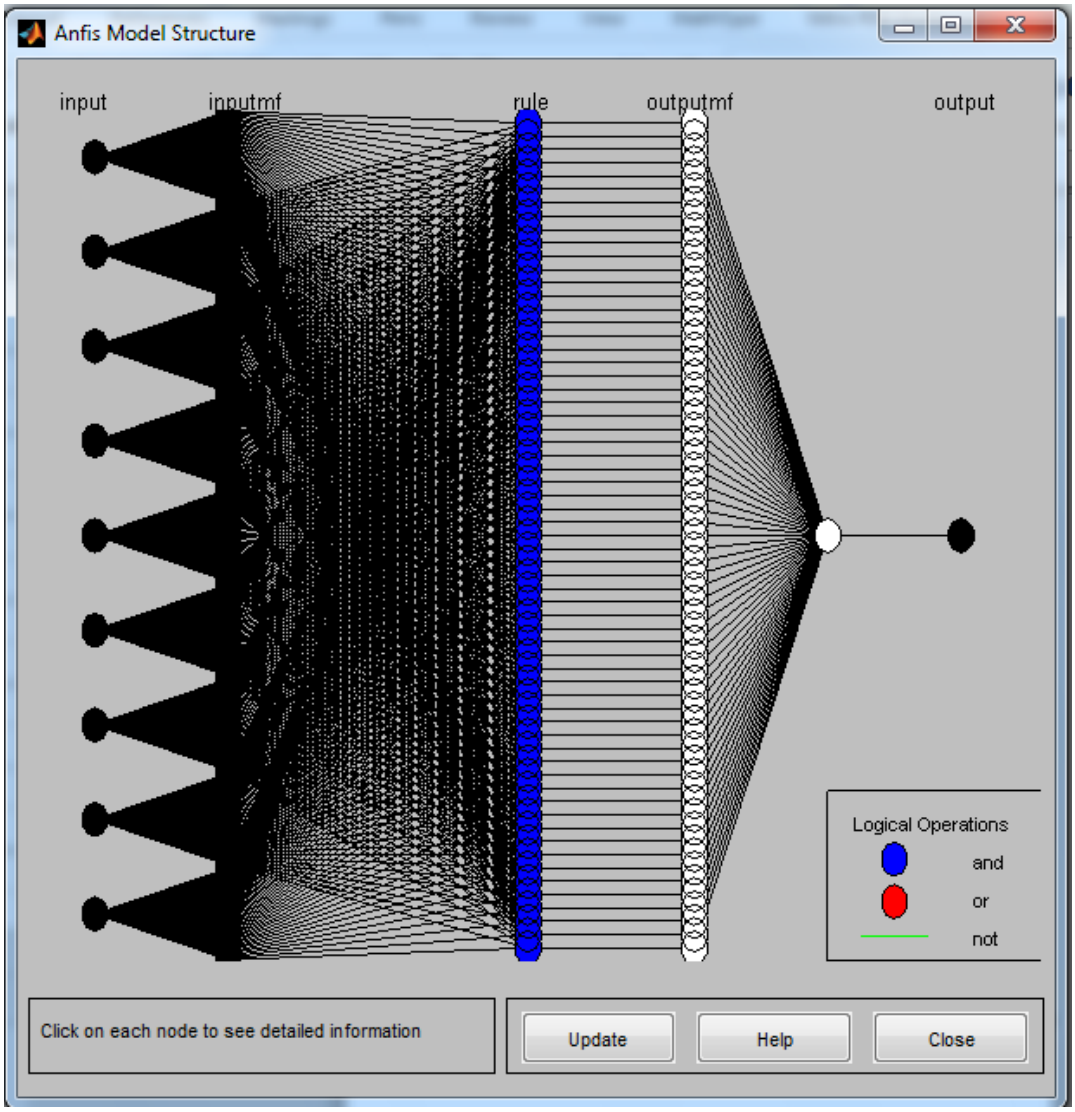


Figure A - 5: ANFIS Structure

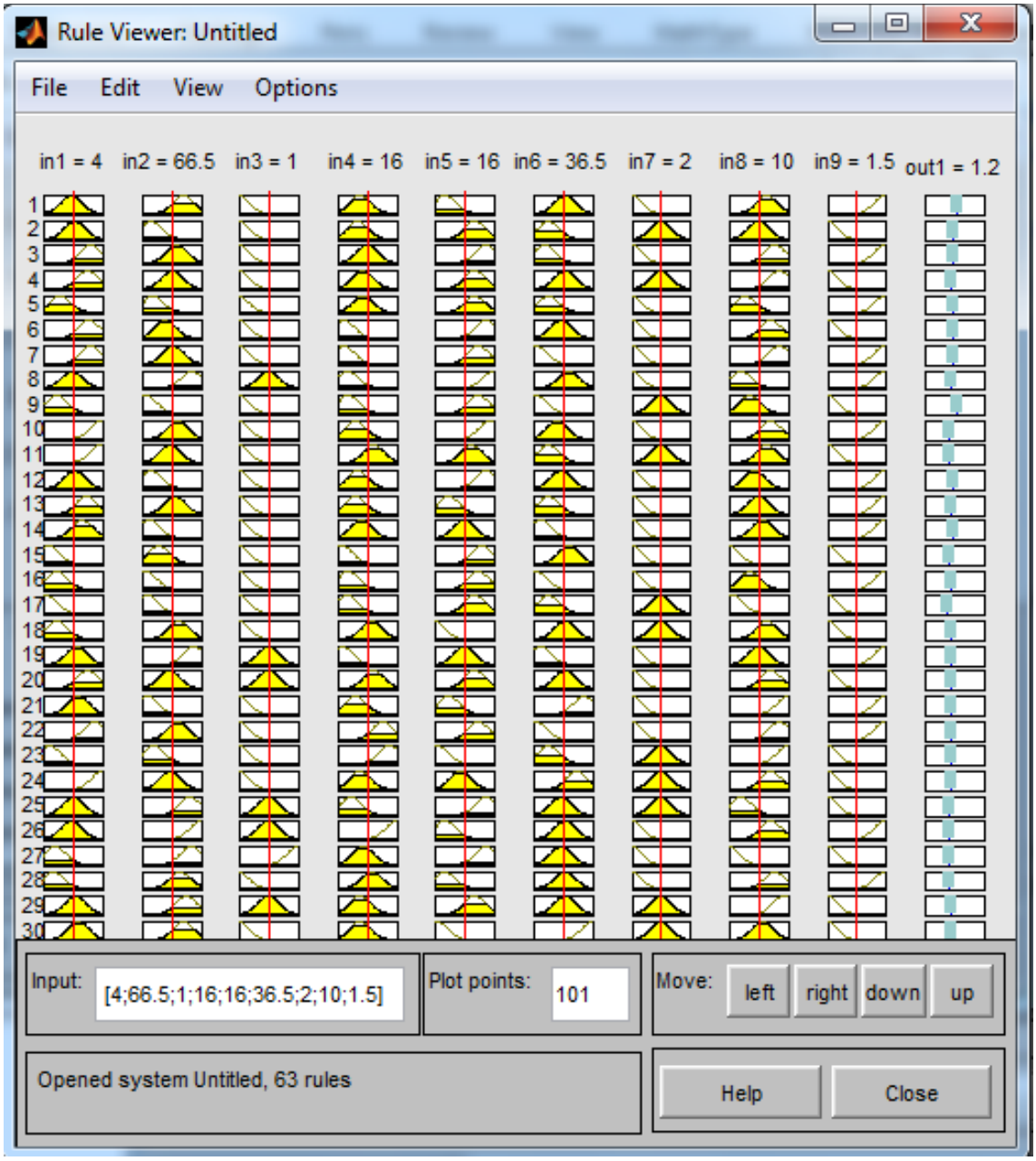


Figure A - 6: ANFIS Rule Viewer

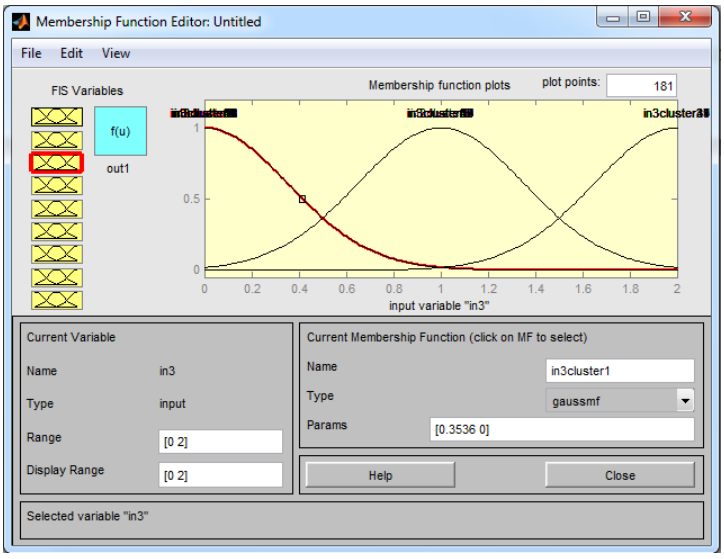


Figure A - 7: ANFIS Membership Functions

APPENDIX B

Main Script: Genetic Algorithm

```
clc
clear
clearvars -global
close all

%% Parameters

global TestResult
global TrainResult
global Pop_Cost
global PopCounter

Pop_Cost=zeros(1,6);
PopCounter=0;

%% Run GA
HandleFunction=@(x) NNDFR(x) % define fitness function fo GA
opt=gaoptimset('Generations',20,'PopulationSize',30,'Display','iter');
% set options for GA
[FinalClusExp,fval,exitflag,output,population,scores]=ga(HandleFunction
,2,[],[],[],[],[2 1.001],[10 3],[],[1],opt) % run GA

%% Result
Pop_Cost=sortrows(Pop_Cost,3);
```

Function: NNDFR

```
function MSE=NNDFR(ClusExp)
      %% Reset Random Stream
      RandStream.setGlobalStream(RandStream('mt19937ar','Seed',1));

      %% Check availability of the individual in past generations
      global Pop_Cost % matrix of individuals together with the cost of
      the fitness function

      FindInd=find(ismember(Pop_Cost(:,1:2),ClusExp,'rows'));
      if FindInd~=0
          MSE=Pop_Cost(FindInd,3)
      else
      %% Data
      % Insert data here

      %% Parameters

      global TestResult
      global TrainResult
      global PopCounter % number of populations

      IterK=1; %Number of iterations for training consequent neural
      networks
      IterMem=1; %Number of iterations for training membership neural
      networks
      TestPercentage=10;

      %
      ClusterNumber=ClusExp(1); %Number of clusters
      Exponent=ClusExp(2); %Fuzzifier

      [r c] = size(DATA); %r=number of data points, c=number of
      variables
      p=ceil(TestPercentage/100*r);

      TestIndices=sort(randsample(r,p));

      %% Train and test Partitioner

      TestSample=DATA(TestIndices,:);
      TrainIndices=setxor(1:r,TestIndices')';      %Complement Matrix
      of the TestIndices
      TrainSample=DATA(TrainIndices,:);

      %% Fuzzy Clustering
      [centerfcm,U,obj_fcn] = fcm(TrainSample(:,1:(c-
      1)),ClusterNumber,[Exponent 100 1e-5 0]); %FCM algorithm
      U=U'; %invert membership degrees
```

```
[TrainSize ~]=size(TrainSample);

%% Seperate training samples for different consequent neural
networks seved in "trains" cell
for k=1:ClusterNumber
c1=1;
for i=1:TrainSize

if U(i,k)==max(U(i,:))
trains{1,k}(c1,:)=TrainSample(i,:);
c1=c1+1;
end

end
end

%% Train Consequence Neural Networks

for i=1:ClusterNumber
[NNK(i).net TRK(i).tr PerfK(i)]=NeuralNetwork(trains{1,i}(:,1:(c-
1)),trains{1,i}(:,c),IterK);
end

%% Train Membership Neural Networks

[NNmem TRmem Perfmem]=NeuralNetwork(TrainSample(:,1:(c-
1)),U,IterMem);

%% Outcome Computation

Weights=[sim(NNmem,DATA(:,1:(c-1))')]'; % calculate weights for
different consequent neural networks

for i=1:r
    Est(i,:)=0; % Est: matrix of estimated productivities
  for l=1:ClusterNumber

    Est(i,:)=Est(i,:)+Weights(i,l)*sim(NNK(1,l).net,DATA(i,1:(c-
1))');

  end
  Est(i,:)=Est(i,:)/sum(Weights(i,:));
end

%% Error Computation

MSE=mse(DATA(:,c)-Est) % mse average (all)
mse_test=mse(TestSample(:,c)-Est(TestIndices,:));
mse_train=mse(TrainSample(:,c)-Est(TrainIndices,:));
aip=sum(abs(1-Est(TestIndices,:)./TestSample(:,c)))*100/p;
%average invalidity percentage
```

```
        %% Save Population

        PopCounter=PopCounter+1;
        Pop_Cost(PopCounter,:)=[ClusExp MSE mse_train mse_test aip];

        %% Save Samples

        TrainResult{PopCounter,1}=[TrainIndices TrainSample(:,c)
        Est(TrainIndices,:)];
        TestResult{PopCounter,1}=[TestIndices TestSample(:,c)
        Est(TestIndices,:)];
        end

end
```

Function: Training Code for Neural Network

```
function [Net,Tr,Performance]=NeuralNetwork(x,y,it)
      %% Iterations

      for i=1:it

      inputs = x';
      targets = y';

      % Create a Fitting Network
      hiddenLayerSize = [9 10];
      net = fitnet(hiddenLayerSize);

      % Choose Input and Output Pre/Post-Processing Functions
      net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
      net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};

      % Setup Division of Data for Training, Validation, Testing
      net.divideFcn = 'dividerand';  % Divide data randomly
      net.divideMode = 'sample';  % Divide up every sample
      net.divideParam.trainRatio = 80/100; % Train Sample
      net.divideParam.testRatio = 20/100; % Validation Sample

      net.trainFcn = 'trainbr';  % Bayesian Regularization

      % Choose a Performance Function
      net.performFcn = 'mse';  % Mean squared error

      % Choose Plot Functions
      net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
        'plotregression', 'plotfit'};

      net.trainParam.showWindow=0; % Not to show training precedure
      net.trainParam.epochs=100;  % Number of training epochs

      % Train the Network
      [net,tr] = train(net,inputs,targets);

      % Test the Network
      outputs = net(inputs);
      errors = gsubtract(targets,outputs);
      performance = perform(net,targets,outputs);

      % Recalculate Training, Validation and Test Performance
      trainTargets = targets .* tr.trainMask{1};
      valTargets = targets  .* tr.valMask{1};
      testTargets = targets  .* tr.testMask{1};
      trainPerformance = perform(net,trainTargets,outputs);
      valPerformance = perform(net,valTargets,outputs);
```

```
        testPerformance = perform(net,testTargets,outputs);

        % View the Network
        view(net)

        % Save all the Networks and Trs and Performances
        TR(i).net=tr;
        NN(i).net=net;
        PERFORMANCE(i)=performance;

        clearvars -except TR NN PERFORMANCE x y it i
        end

        %% Find the best Network among all iterations
        [Performance,idx]=min(PERFORMANCE);
        Net=NN(idx).net;
        Tr=TR(idx).net;

end
```

Script: Alpha-Cut

```
j=1;
for alpha=0:0.05:1 % alpha levels, interval=0.05

    % line equation of moderate temperature fuzzy variable
    x1=(alpha+0.5)/0.1;
    x2=(alpha-2.5)/-0.1;
    % line equation of high temperature fuzzy variable
    y1=(alpha+8)/0.1;
    y2=(alpha-10)/-0.1;

    i=1;
    for fuzz_input1=x1:(x2-x1)/10:x2 % devide the range of alpha cut to
descrete points
        for fuzz_input2=y1:(y2-y1)/10:y2 % devide the range of alpha
cut to descrete points

    data=[fuzz_input1 fuzz_input2 0 16.6    18   33   2    10   1]; %
combine with crisp inputs

    %% Model the data with Genetically Optimized NNDFR
    Weights=[sim(NNmem,data')]';
    Est(i,:)=0;
         for l=1:3
         Est(i,:)=Est(i,:)+Weights(1,l)*sim(NNK(1,l).net,data');
         end
    Est(i,:)=Est(i,:)/sum(Weights(1,:));
    i=i+1;
        end
    end

    %% Save max and min outputs
    infsup(j,:)=[min(Est) max(Est)];
    clearvars Est
    j=j+1;

end

%% Plot
plot(infsup(:,1),0:0.05:1);
hold on
plot(infsup(:,2),0:0.05:1);
hold on
scatter(infsup(:,1),0:0.05:1);
hold on
scatter(infsup(:,2),0:0.05:1);
hold off
```

Script: Centroid Defuzzification

```
[r c]=size(infsup);
alpha=linspace(0,1,r)';

%% for trapazoid before the peak (related to min outputs)
for i=1:(r-1)
    % coordinate of the center of the trapazoid
    dis(i,1)=infsup(i,1)+(infsup(i,1)-
infsup(i+1,1))/3*(alpha(i,1)+2*alpha(i+1,1))/(alpha(i,1)+alpha(i+1,1));
    % area of the trapazoid
    area(i,1)=(alpha(i,1)+alpha(i+1,1))*(infsup(i,1)-infsup(i+1,1))/2;
end

%% for trapazoid after the peak (related to max outputs)
flip_infsup=flipud(infsup(:,2));
flip_alpha=flipud(alpha(:,1));
for i=1:(r-1)
    % coordinate of the center of the trapazoid
    dis(r-1+i,1)=flip_infsup(i,1)+(flip_infsup(i,1)-
flip_infsup(i+1,1))/3*(flip_alpha(i,1)+2*flip_alpha(i+1,1))/(flip_alpha
(i,1)+flip_alpha(i+1,1));
    % area of the trapazoid
    area(r-
1+i,1)=(flip_alpha(i,1)+flip_alpha(i+1,1))*(flip_infsup(i,1)-
flip_infsup(i+1,1))/2;
end

%% centroid
defuzz=sum(dis.*area)/sum(area) % defuzzified value(productivity)
```