FUZZIFIED SCORING OF THE FUNCTIONAL ASSESSMENT INSTRUMENT

A Thesis presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

SHUANG WANG

Dr. Marjorie Skubic, Thesis Supervisor

MAY 2007

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

FUZZIFIED SCORING OF

THE FUNCTIONAL ASSESSMENT INSTRUMENT

presented by Shuang Wang,

a candidate for the degree of master of Electrical and Computer Engineering,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Marjorie Skubic

Professor James Keller

Professor Zhihai He

ACKNOWLEDGEMENTS

The author wishes to thank my supervisor Dr. Marjorie Skubic and Dr. James Keller. I could not have imagined having a better advisor and mentor, and without their knowledge, perceptiveness and guidance, I would never have finished. Thank-you to my graduate committee member Dr. Zhihai He. I would also like to thank all the members in Eldercare Research Group.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FUZZIFIED SCORING OF THE FUNCTIONAL ASSESSMENT INSTRUMENT

Shuang Wang

Dr. Marjorie Skubic, Thesis Supervisor

ABSTRACT

This thesis describes the application of fuzzy logic to the Short Physical Performance Battery (SPPB) test, a series of timed physical activities that have been created to evaluate, physical functional performance for both research and clinical purposes, primarily for physically impaired older adults. The original scoring system of the SPPB test uses crisp time boundaries to assign the subject to discrete classes of performance. The crisp nature of the crisp thresholds can easily produce anomalies. Fuzzy Logic theory allows the natural description, in linguistic terms, of input/output relationships rather than relying on precise numerical threshold values. This advantage, dealing with complicated systems in a simple way, is the main reason why fuzzy logic theory is widely applied. This thesis offers a new approach for scoring the SPPB test. It demonstrates that in the proposed system, the Fuzzy Short Physical Performance Battery (FSPPB), the sensitivity and data distribution of the scoring system for the SPPB test can be improved. This thesis presents the procedures of constructing a fuzzy inference system using fuzzy logic to score the SPPB test. It also presents the procedures of constructing a fuzzy inference system using Adaptive Neuro-Fuzzy Inference System (ANFIS) technology and a tuning system for the fuzzy SPPB test using Particle Swarm Optimization (PSO). As part of a large project in technology for Eldercare, the goal is to accurately measure trends in physical performance of seniors over time.

1 Introduction

1.1 Motivation

Elder adults are living longer and more fulfilled lives, and they desire to live as independently as possible in the home of their choice. However, independent lifestyles come with risks that are complicated by chronic illness and impairments in mobility, cognition, and the senses. In response to this trend, we have been investigating new approaches in caring for the elderly. One recent example of this research focus has resulted in TigerPlace, a 32-unit apartment complex for seniors that opened in Columbia, Missouri in 2004. A joint venture between MU's Sinclair School of Nursing and Americare Systems Inc., TigerPlace is one of four projects granted state approval to operate under the "aging in place" model of care giving [1]. Under that model, residents who would otherwise be required by state law to live in nursing homes may have health services brought to them in their apartments instead.

Technology that can help seniors "age in place" has been spotlighted in recent years, spurred by an aging population. One focus of our research is the creation of "intelligent software" that uses sensors to uncover patterns of activity helpful to caregivers [2], especially targeting mobility and cognitive impairment. Details can be found at http://eldertech.missouri.edu. A critical part of this effort is being able to sense, detect and assess changes in basic physical performance in elders. Toward this goal, investigated the Short Physical Performance Battery (SPPB) test [3] is investigated as an assessment instrument to evaluate, discriminate, and predict physical functional performance for older adults. The SPPB test was developed for use in the Established Populations for the Epidemiologic Studies of the Elderly – the study known as the EPESE [3]. This was a study funded by the National Institute on Aging and was a longitudinal epidemiologic study that involved several large populations across the United States. The SPPB test has been used to assess lower extremity function in more than 5,000 persons age 71 years and older [3].

There are three components to the SPPB test. First, it includes timed standing balance tests. These tests are done in a side-by-side position, a semi-tandem, and the tandem positions. The second component is a timed four meter/three meter or eight foot walk. Finally, the third component is a chair rise, both a single attempt, and if the person is successful at doing that, then asked he or she is to do five chair rises as quickly as possible.

While the crisp scoring of the SPPB has been validated on good sized populations of elders, the crisp nature of the judgment leads to common anomalies near threshold boundaries and does not possess a fine enough granularity to be used within a frequent automated evaluation of the physical capabilities of a particular senior. In this thesis, a fuzzy logic rule-based system is introduced that preserves the original design of the SPPB but addresses these two shortcomings.

1.2 Overview

In the second chapter, background of the fuzzy rule based system and the ANFIS system is introduced. In the third chapter, the details of the SPPB test are described with the conversion between different distances in the walk test. In the next chapter, the fuzzified SPPB test is described along with results. The fifth chapter introduces learning the fuzzified SPPB with the ANFIS method and includes results of the learning process. The sixth chapter provides the details of a tuning system of the fuzzified SPPB test scoring system with results and a description of a new interface. The conclusion includes a summary of the contributions and suggestions for future work.

2 Background on Fuzzy Rule-based Systems

2.1 Fuzzy Rule-based System

The idea of fuzzy sets was originated by Zadeh [4]. The main concept of fuzzy sets is that many problems in the real world are imprecise rather than exact. It is believed that the effectiveness of the human brain is not only from precise cognition, but also from fuzzy concepts, fuzzy judgment, and fuzzy reasoning. Fuzzy systems reason with multi-valued sets or fuzzy sets (i.e., the sets of values between 0 and 1) instead of bi-valued sets or crisp sets (i.e., the sets of values of 0 and 1). An advantage of fuzzy classification techniques lies in the fact that they provide a soft decision, a value that describes the degree to which a pattern fits within a class, rather than only a hard decision, i.e., a pattern matches a class or not. The other advantage of this theory is that it allows the natural description of problem domains, in linguistic terms, rather than in terms of relationships between precise numerical values. Fuzzy systems have been used successfully to handle many applications in the real world problems such as control systems and pattern classification problems. Some well-known fuzzy systems are fuzzy-rule-base methods [5], fuzzy c-means [6], fuzzy k-nearest-neighbor [7][8], and fuzzy decision tree [9].

2.1.1 Fuzzy Logic System Components

A typical fuzzy logic inference system has four components: a fuzzifier, a fuzzy rule base, an inference engine, and a defuzzifier. The function of the fuzzifier is to determine the degree of membership of a crisp input in a fuzzy set. The fuzzy rule base is used to represent the fuzzy relationships between input-output fuzzy variables. The output of the fuzzy rule base is determined based on the degree of membership specified by the fuzzifier. The inference engine calculates the rule's conclusion based on its membership degree. A defuzzifier is used to convert outputs of the fuzzy rule base into crisp values. Figure 2.1 illustrates a block diagram of a fuzzy system.

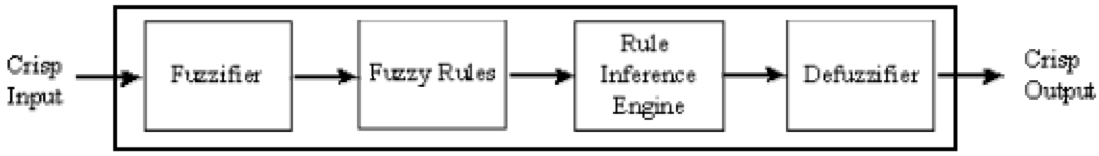


Figure 2.1 A Block Diagram of a Fuzzy Inference System

2.1.2 Membership Function

A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse. The simplest membership functions are formed using straight lines. Of these, the simplest is the triangular membership function. It is nothing more than a collection of three points forming a triangle. The trapezoidal membership function has a flat top and really is just a truncated triangle curve. These straight line membership functions have the advantage of simplicity. Gaussian and bell membership functions are popular methods for specifying fuzzy sets because of their smoothness and concise notation. Both of Gaussian and bell curves have the advantage of being smooth and nonzero at all points. There's a very wide selection to choose from when selecting membership function.

2.1.3 Fuzzy Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. Usually the knowledge involved in fuzzy reasoning is expressed as rules in the form:

If x is A Then y is B

where x and y are fuzzy variables and A and B are fuzzy values defined by fuzzy sets. The if-part of the rule "x is A" is called the antecedent or premise, while the then-part of the rule "y is B" is called the consequent or conclusion. Statements in the antecedent (or consequent) parts of the rules may well involve fuzzy logical connectives such as 'AND' and 'OR'. In the if-then rule, the word "is" gets used in two entirely different ways depending on whether it appears in the antecedent or the consequent part.

2.1.4 Fuzzy Inference System

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The process of fuzzy inference involves membership functions, fuzzy logic operators and if-then rules. The most popular models of fuzzy inference systems are the Mamdani models [10] and the Takagi-Sugeno-Kang (TSK) models [11]. The main difference between them is the consequent part of fuzzy rules. The Mamdani models describe the consequent part using linguistic variables, while the Takagi-Sugeno-Kang models use the linear combination of the input variables. Both models use linguistic variables to describe the antecedent part of fuzzy rules.

a) Mamdani Models

Mamdani fuzzy-rule based systems [10] consist of a linguistic description in both the antecedent parts and the consequent parts. Each rule is a description of a condition-action statement that may be clearly interpreted by the users. To describe a mapping from input $U_1 \times U_2 \times \ldots \times U_n$ (where $\times$ is the Cartesian product) to output W, the linguistic rule structure of Mamdani models is as follows:

$$R_i\text{: IF } x_1 \text{ is } A_{i1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{in} \text{ THEN } y \text{ is } C_i,\ i = 1, \ldots, L$$

where L is the number of fuzzy rules, $x_j \in U_j$, $j = 1, 2, \ldots, n$, are the input variables, y is the output variable, and A_{ij} and C_i are linguistic variables or fuzzy sets for x_j and y respectively. A_{ij} and C_i are characterized by membership functions $\mu_{A_{ij}}(x_j)$ and $\mu_{C_i}(y)$, respectively. Inputs are of the form:

$$x_1 \text{ is } A'_1,\ x_2 \text{ is } A'_2,\ \ldots,\ x_r \text{ is } A'_n$$

where A'_1, A'_2, …, A'_n are fuzzy subsets of U_1, U_2, … U_n, which are the universe of disclose (or the domain of interest) of inputs.

A specific example is shown in Figure 2.2.

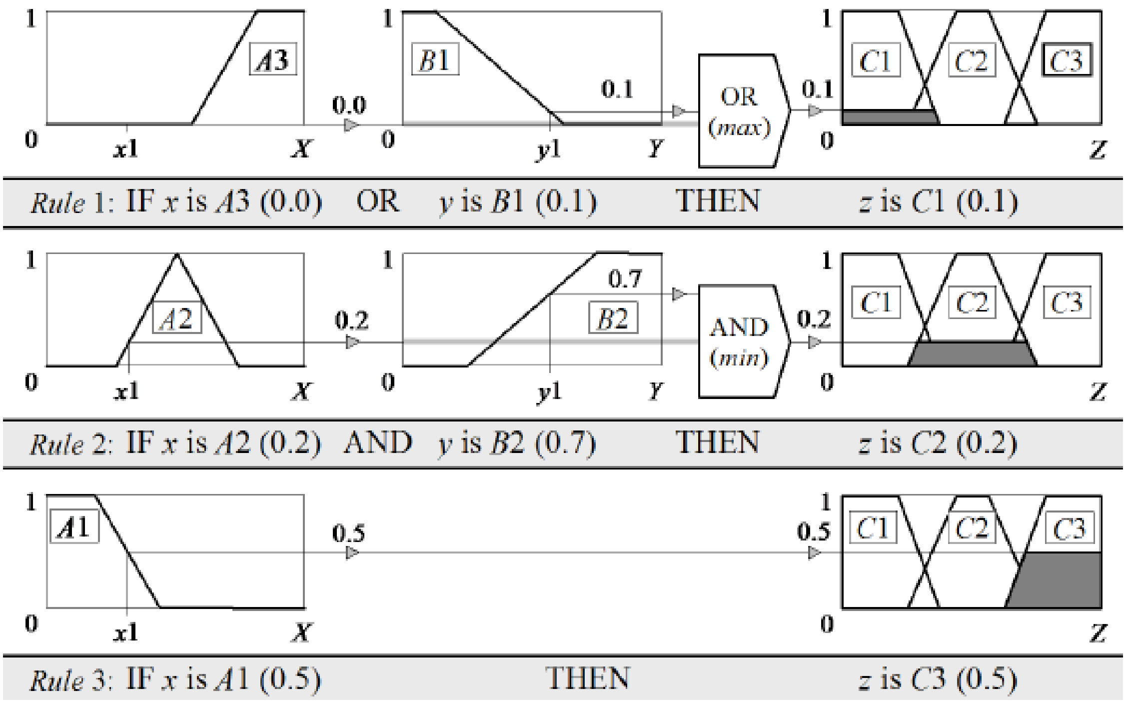


Figure 2.2 Example of Mamdani Model

b) Takagi-Sugeno-Kang (TSK) Models

Instead of working with linguistic rules as in Mamdani models [Mandani74], Takagi, Sugeno, and Kang [11], [13] proposed a new model based on rules where the antecedent was composed of linguistic variables and the consequent was represented by a function of the input variables. The most usual form of these kinds of rules is the one shown in the following, in which the consequent comprises a linear combination of the variables involved in the antecedent. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. A typical rule in a Sugeno fuzzy model has the form

If Input1 = x and Input2 = y, then Output is $z = ax + by + c$

where Input1 and Input2 are the system input variables, z is the output variable, a,b,c are the numerical constant parameters, and x and y are linguistic labels associated in the form of fuzzy sets.

For a zero-order Sugeno model, the output level z is a constant *(a=b =0).* The output level Z_i of each rule is weighted by the firing strength W_i of the rule. For example, for an AND rule with Input1 = *x* and Input2 = *y,* the firing strength is

$$W_i = AndMethod\ (F1(x), F2(y))$$

where F1, F2 are the membership functions for Input1 and Input2 respectively. The final output of the system is the weighted average of all rule outputs, computed as

$$\text{Final Output=A=}\frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \qquad (2.1)$$

The inference performed by the TSK model is an interpolation of the entire relevant linear model. The degree of relevance of a linear model is determined by the degree the input data belong to the fuzzy subspace associated with the linear model. These degrees of belongingness become the weight in the interpolation process.

Figure 2.3 shows the example as a Sugeno system.

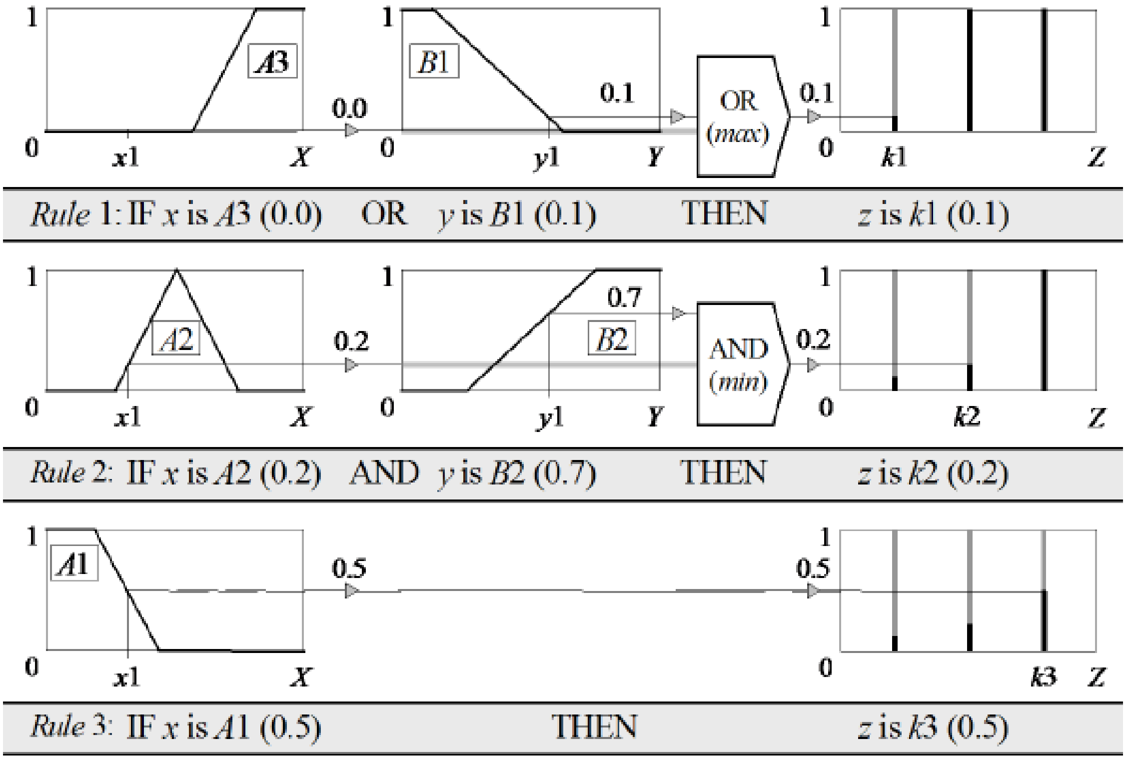


Figure 2.3 Example of Sugeno Inference System

The goal of this thesis is to produce and analyze a fuzzy logic rule-based system that preserves the original design of the SPPB but addresses shortcomings. Fuzzy logic has attracted the attention of several researchers in health care. Fuzzy logic provides a methodology that simulates human thinking by explicitly modeling and managing the imprecision and uncertainty inherent in health assessment. Grade of membership methods are employed to summarize functional status, and multinomial logic models provide information on the association between biological measures and function [14]. Wieland et al. used grade-of-membership analysis, to classify participants on the basis of their specific diseases, impairments, and disabilities [15]. Other studies in this field use grade of membership methods to evaluate health and mortality of the elderly [16], to analyze medical, behavioral, psychosocial, and characteristics of service use by nursing home residents [17], and to classify taste responses in the brain stem [18]. Still others use fuzzy logic in aging research to discuss the utility of the different grade of membership models for identifying the ageing phenomenon [19]. To our knowledge, fuzzy logic has not previously been applied to an assessment tool such as the SPPB.

2.2 ANFIS Learning System

Fuzzy systems are adopted as a natural framework for incorporating human knowledge. In a purely fuzzy system, the parameters do not appear in an analytical way; therefore, learning can not be applied for tuning fuzzy rules. On the other hand, neural networks can produce mapping rules from an empirical training set through learning. But the mapping rules in the network are not visible and are difficult to understand. Combining both aspects may be useful. The fundamental concept of such a hybrid system is to complement each other and thereby, create new approaches to solve problems. This gives rise to neuro-fuzzy systems, which have recently been investigated by many researchers; see, for instance, [20] and references therein. One of the main advantages of combining fuzzy systems and learning methods is that it is possible to use human knowledge to build an initial system, and then use measured input/output relations to optimize the system. By combining previous knowledge with measured input/output data, it is often possible to: make a system that interpolates and extrapolates better than a pure artificial neural network (ANN) system. Another advantage of these combined systems is that they often are more precise that pure Fuzzy systems because the learning algorithm may find a better solution. One of the best known of these combinations is the NeuroFuzzy system called ANFIS [21]. These combined systems, such as ANFIS have been used in many different cases, for instance [22] [23]or [24]. The usage of the ANFIS methods in disease, impairments and disabilities diagnosis has been increasing gradually. Diagnosis of lymph diseases which is a very common and important disease, was conducted with an ANFIS system [25]. Other studies in this field use the ANFIS method to evaluate fluctuations in Parkinson's disease [26], to predict respiratory motion in breast cancer patients [27]. These applications give clinicians a valuable tool to explore the importance of different variables and their relations in a disease and could aid treatment selection.

2.2.1 Architecture of ANFIS

The ANFIS is a fuzzy Sugeno model put in the framework of adaptive systems to facilitate learning and adaptation [21]. Such a framework makes the ANFIS modeling more systematic and less reliant on expert knowledge. To present the ANFIS architecture, two fuzzy if–then rules based on a first order Sugeno model are considered:

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 ($f_2=p_2x+q_2y+r_2$)

where x and y are the inputs, A_i and B_i are the membership functions of fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, p_i, q_i and r_i are the design parameters that are determined during the training process. The ANFIS architecture to implement these two rules is shown in Figure 2.4, in which a circle indicates a fixed node, whereas a square indicates an adaptive node.

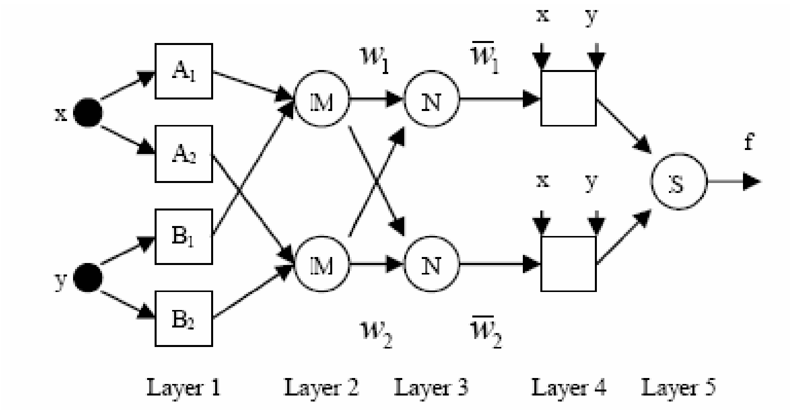


Figure 2.4 ANFIS architecture

In the first layer, all the nodes are adaptive nodes. The outputs of layer 1 are the fuzzy membership set of the inputs, which are given by:

$$O_i^1 = \mu_{Ai}(x) \quad i=1,2 \tag{2.2}$$

$$O_i^1 = \mu_{Bi-2}(y) \quad i=3,4 \tag{2.3}$$

where $\mu_{Ai}(x)$, $\mu_{Bi-2}(y)$ can adopt any fuzzy membership function. For example, if the bell-shaped membership function is employed, $\mu_{Ai}(x)$ is given by:

$$\mu_{A_i}(x) = \frac{1}{1+\left\{\left(\frac{x-c_i}{a_i}\right)^2\right\}^{b_i}} \tag{2.4}$$

where a_i, b_i and c_i are the parameters of the membership function, governing the bell-shaped functions accordingly.

In the second layer, the nodes are fixed nodes. They are labeled with M, indicating that they perform as a simple multiplier. The outputs of this layer can be represented as:

$$O_i^2 = w_i = \mu_{Ai}(x)\mu_{Bi}(y) \quad i=1,2 \tag{2.5}$$

which are the so-called firing strengths of the rules.

In the third layer, the nodes are also fixed nodes. They are labeled with *N*, indicating that they play a normalization role to the firing strengths from the previous layer.

The outputs of this layer can be represented as:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2} \qquad i = 1,2 \tag{2.6}$$

which are the so-called normalized firing strengths.

In the fourth layer, the nodes are adaptive nodes. The output of each node in this layer is simply the product of the normalized firing strength and a first order polynomial (for a first order Sugeno model). Thus, the outputs of this layer are given by:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \qquad i = 1,2 \tag{2.7}$$

In the fifth layer, there is only one single fixed node labeled with *S*. This node performs the summation of all incoming signals. Hence, the overall output of the model is given by:

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i = \frac{(\sum_{i-1}^{2} w_i f_i)}{w_1 + w_2} \tag{2.8}$$

It can be observed that there are two adaptive layers in this ANFIS architecture, namely the first layer and the fourth layer. In the first layer, there are three modifiable parameters $\{a_i, b_i, c_i\}$, which are related to the input membership functions. These parameters are the so-called premise parameters. In the fourth layer, there are also three modifiable parameters $\{p_i,q_i,r_i\}$, pertaining to the first order polynomial. These parameters are so-called consequent parameters [21].

2.2.2 Learning algorithm of ANFIS

The task of the learning algorithm for this architecture is to tune all the modifiable parameters, namely $\{a_i,b_i,c_i\}$ and $\{p_i,q_i,r_i\}$, to make the ANFIS output match the training data. When the premise parameters a_i, b_i and c_i of the membership function are fixed, the output of the ANFIS model can be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_1}{w_1 + w_2} f_2 \tag{2.9}$$

Substituting Eq. (2.5) into Eq. (2.8) yields:

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 \tag{2.10}$$

Substituting the fuzzy if–then rules into Eq. (2.10), it becomes:

$$f = \overline{w}_1(p_1 x + q_1 y + r_1) + \overline{w}_2(p_2 x + q_2 y + r_2) \tag{2.11}$$

After rearrangement, the output can be expressed as:

$$f = (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2 \tag{2.12}$$

which is a linear combination of the modifiable consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2. The least squares method can be used to identify the optimal values of these parameters easily. When the premise parameters are not fixed, the search space becomes larger and the convergence of the training becomes slower. A hybrid algorithm combining the least squares method and the gradient descent method is adopted to solve this problem. The hybrid algorithm is composed of a forward pass and a backward pass. The least squares method (forward pass) is used to optimize the consequent parameters with the premise parameters fixed. Once the optimal consequent parameters are found, the backward pass starts immediately. The gradient descent method (backward pass) is used to adjust optimally the premise parameters corresponding to the fuzzy sets in the input domain. The output of the ANFIS is calculated by employing the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard backpropagation algorithm. It has been shown that this hybrid algorithm is highly efficient in training the ANFIS [21].

3 Background on the Short Physical Performance Battery (SPPB) Test

The SPPB can be used to assess how well older subjects perform simple movements that represent the building blocks of daily activities that require good lower extremity function. The information concerning functional ability provided by these tests adds valuable insight to the assessment of the older subject. Aside from evaluating the current functional status of an individual, these tests also have been shown to be powerful predictors of future disability, hospitalization, and death even in participants who report no disability at initial testing. The prognostic characteristics of the SPPB make it very useful for identifying seemingly non-disabled persons who are at risk of disability. Early identification of these persons, who are presumably in a preclinical stage of disability, provides an opportunity for early intervention in the pathological processes underlying the progression to frank disability.

The basic SPPB consists of three types of physical maneuvers, the balance tests, the gait speed test, and the chair stand test. The tests are always performed in this order. Each of these three maneuvers is scored separately by the examiner. The SPPB is a timed performance test; each subtask score is an integer value in the range 0-4. A score of 0 indicates the inability to complete the task in a nominal time frame while categories 1-4 are assigned to the corresponding quartiles of time needed to perform the action. There are five subscales: semi-tandem standing (the heel of one foot on the side of the big toe of the other foot), side by side standing, full tandem position (the heel of one foot directly in front of the other foot), walking speed (eight feet on a smooth service with no obstructions), and a sit-to-stand test (rise from a chair five times with arms folded across the chest while sitting on a straight-backed chair next to a wall). It has been shown that such performance measures correlate well with the self-assessment of older persons across a broad spectrum of lower extremity function [28] [29] [30].

3.1. Balance Test

The tests of balance provide an assessment of the participant's ability to hold three basic standing positions with the eyes open. These positions are side-by-side stand, semi-tandem, and full tandem stand (or heel-to-toe) and are performed in this order. Participants taking this test must be able to stand unassisted without using a cane or a walker. The first position tested is the side-by-side stand. In this balance test, participants are requested to stand for 10 seconds with their feet together in a side-by-side position. Participants who are unable to hold the side-by-side stand for less than 10 seconds do not proceed further with the balance tests and are given a score of 0 for this portion of the battery. Participants who successfully complete the side-by-side test receive 1 point and proceed to the semi-tandem balance test. In the semi-tandem balance test, each participant starts with the heel of one foot placed to the side of the big toe of the other foot. Either foot can be placed in the forward position. Participants who successfully hold the semi-tandem position for 10 seconds are given 1 additional point and proceed to the final balance test. Those who fail to hold the position for 10 seconds receive no points and do not perform the tandem balance test. The final position evaluated in the balance tests is the tandem position. To assume the tandem position, the heel of one foot is placed directly in front of the toes of the other foot. Either foot can be placed in the forward position. Participants who hold this position for 10 seconds are awarded 2 additional points. Those who hold the position for 3 to 9.99 seconds are given 1 additional point, holding the position for less than 3 seconds results in no points.

The scoring system for the balance test is as follow [3]:

Score	Side by side stand	Semi-tandem stand	Full tandem stand
0	t< 10 seconds	Not attempted	Not attempted
1	t=10 seconds	t< 10 seconds	Not attempted
2	t=10 seconds	t=10 seconds	t< 3 seconds
3	t=10 seconds	t=10 seconds	3seconds<=t<10seconds
4	t=10 seconds	t=10 seconds	t=10 seconds

Table 3.1 Scoring performance on tests of standing balance

3.2. Gait Speed Test and Conversion between 50-foot, 8-foot and 3-m/4-m Gait Speed Test

The second test in the Short Physical Performance Battery is the gait speed test. In this test, the participant's ability to walk 8 feet is assessed. The walking course should be unobstructed and include at least an extra one-half meter on each end. Participants are instructed to walk at their usual speed, and timing is stopped when the first foot completely crosses the 8-foot mark. The faster of two timed walks is used for scoring purposes.

The scoring system for the 8-foot gait speed test is as follow [3]:

Score	8-foot Walk Time
1	t>=5.7 seconds
2	4.1seconds<=t<=5.7seconds
3	3.2seconds<=t<4.1seconds
4	t<3.2seconds

Table 3.2 Scoring performance on test for 8-foot walk

An 8-ft (2.44 m) walk was used in EPESE because of concern about limited space in participants' homes [3]. However, for the assessment of gait speed over a short distance the 3-m/4-m walk is now the distance of choice because it has been demonstrated to be feasible in the home as well as in the clinical setting and its longer distance may improve measurement accuracy. In the Women's Health and Aging Study it was demonstrated that in small houses and apartments in Baltimore, a 4-m walk was possible for 90% of subjects, with 9% requiring a 3-m walk and 0.8% not having space for a 3-m walk [31]. Because of the advantage of using the 3-m/4-m walk for both home and clinic gait speed tests, the 8-ft walk was converted into estimated velocity for a 3-m/4-m walk [31]. Figure 3.1 shows the relationship of gait speed between the 8-foot walk and the 3-m/4-m walk. The following equations were used to convert an 8-ft gait speed to a 3-m/4-m gait speed:

For 8-ft gait speed<=1.0m/s 3-m/4-m speed=0.01+(8-ft speed)*1.052
For 8-ft gait speed>=1.0m/s 3-m/4-m speed=0.481+(8-ft speed)*0.581

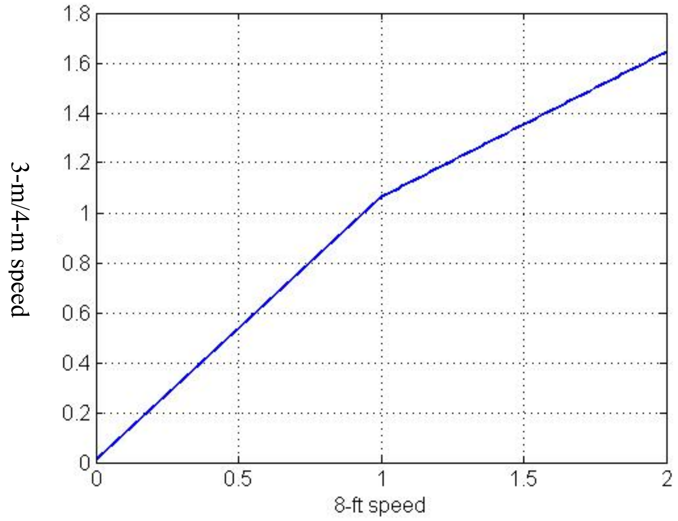


Figure 3.1 Gait speed relationship between 8-ft and 3-m/4-m walk test

The scoring system for the 3-m/4-m gait speed test is as follow:

Score	3-m Walk Time (Seconds)	4-m Walk Time (Seconds)
0	unable	unable
1	t>6.52	t>8.70
2	4.66 <=t<=6.52	6.21 <=t<=8.70
3	3.62 <=t<=4.65	4.82 <=t<=6.20
4	t<3.62	t<4.82

Table 3.3 Scoring performance on test for 3-m/4-m walk

Another 50-foot walk test is introduced by Reuben and Siu [32]. The scoring system for the 50-foot gait speed test is as follow:

Score	50-foot walk(Seconds)
0	99
1	t>25
2	20 <t<=25
3	15 <t<=20
4	t<=15

Table 3.4 Scoring performance on test for 50-foot walk

A curve fitting method is proposed to find the relationship between the 50-foot walk and the 8 foot walk. To our knowledge, there is no formula in the literature for converting between the 50-foot walk and the 8-foot walk or 3-m/4-m walk. Here, we standardize on the original 8-foot walk. Figure 3.2 shows three polynomials which can be fitted based on the thresholds for the two scoring methods. In these polynomials, the quadratic has the best residuals, and it is chosen to convert 50-foot walking time to 8-foot walking time:

$$\text{50-foot walking time(seconds)}=0.014(\text{8-ft walking time })^2-0.31\left(\text{8-ft walking time}\right)+4.7$$

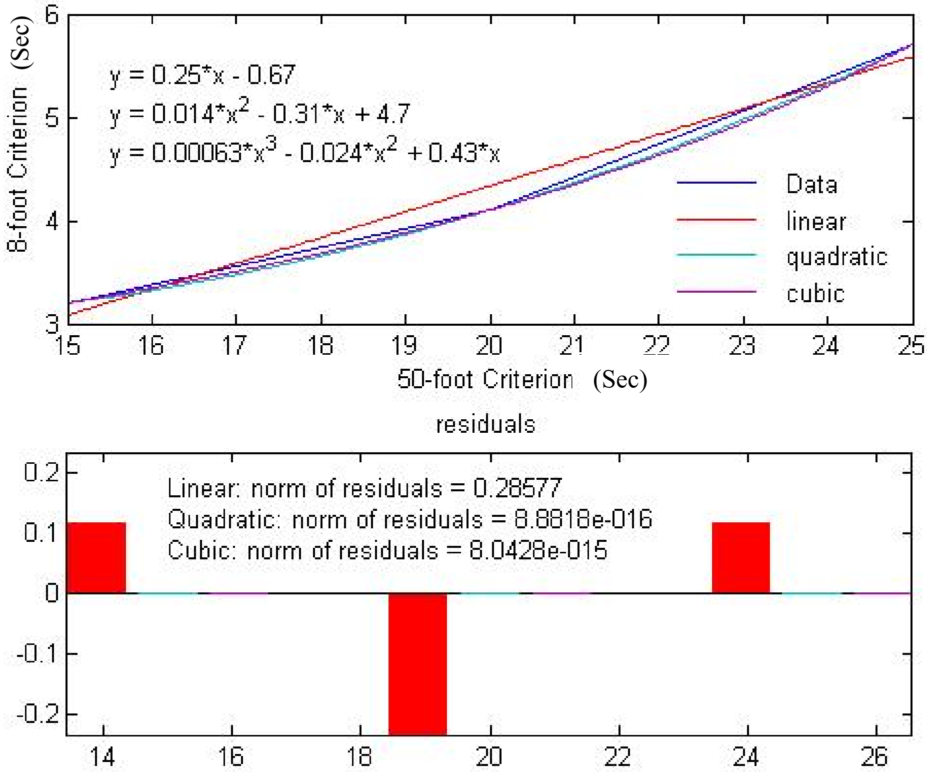


Figure 3.2 Curve fitting between 50-foot walk and 8-foot walk

3.3. Chair Stand Test

The final portion of the Short Physical Performance Battery is the chair stand test. In this test, participants are first instructed to fold their arms across their chest and to try to stand up one time from an armless chair placed against a wall. If the participants are successful rising from the chair once, they are then asked to stand up and sit down 5 times as quickly as possible.

The scoring system for the chair stand test is as follow [3]:

Score	Sit to Stand Time
1	t>=16.7 seconds
2	13.7seconds<=t<16.7seconds
3	11.2seconds<=t<13.7seconds
4	t<11.2seconds

Table 3.5 Scorning performance on chair stand test

4. Assessing Physical Performance of Elders Using Fuzzy Logic

4.1. Methodology

4.1.1. Design Criteria

For the SPPB test, there are three parts in the original scoring system, standing balance score, walking test score and chair test score. The overall flow chart of the implementation is shown in Figure 4.1. Fuzzy set theory offers us a wide variety of aggregation operators (including a fuzzy logic system) to combine the outputs of the three tests. However, because there is no standard for interpreting the sum of the test scores in the SPPB, we have not investigated a fuzzy aggregation to date.

A potential problem with aggregating these separate scores is that the tests are not completely independent. For example, the first physical performance test in the battery measures balance; however, subsequent test performance is also influenced by balance. In addition, the balance test is not as sensitive to variations in scoring because it is time-limited (a maximum of 10 seconds in each position). By summing the scores in to an

3aggregate score, we risk losing overall sensitivity. As a result, we look at individual sub-task scores using fuzzy logic and do not perform the aggregation in this study.

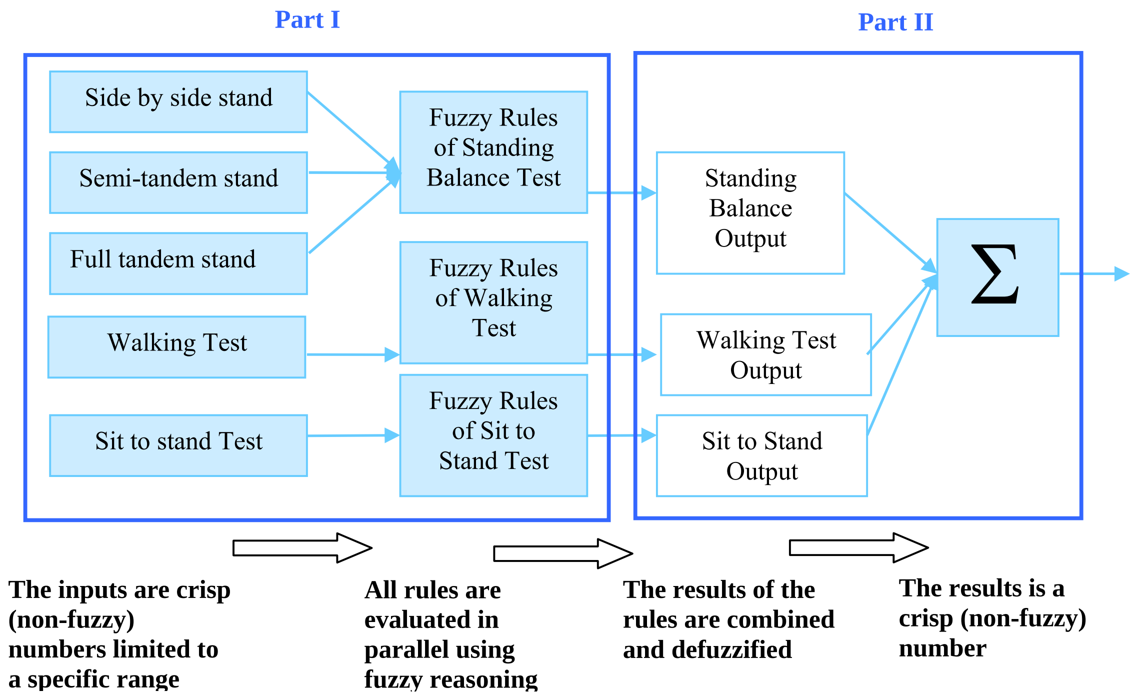


Figure 4.1 Overall Flow Chart of Fuzzy Scoring Implementation

4.1.2. Choice of Membership Functions

Based on a review of the data and discussions with the nursing members of our team, for the fuzzified scoring system of SPPB test, we needed membership functions with characteristics of smoothness, asymmetry, and zero on both extremes with a quick rise in the middle. Pi curves, as shown in Figure 4.2, satisfy our requirements.

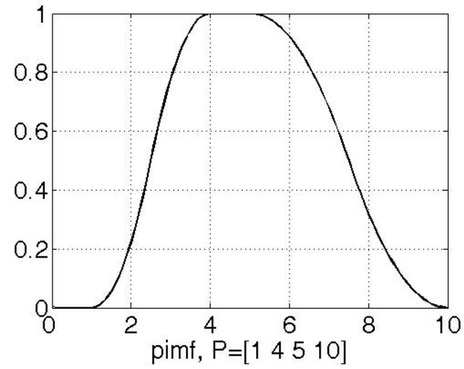


Figure 4.2 Pi Membership Function

A Pi membership function is determined by 4 parameters Pi [a,b,c,d]. The parameters a and d locate the "feet" of the curve, while b and c locate its "shoulders." The linguistic variables for our rule-based system are the same as for the original SPPB test. The linguistic values were created to model the crisp threshold ranges in [3]. Specific choices for membership functions to model the linguistic values were chosen experimentally (at this point) with an eye to soften the harsh boundaries while preserving the general flavor of the performance ranges from the standard scoring approach.

From Figure 4.1 we see that there are three inputs and one output in the standing balance test. Hence, we need linguistic values and membership functions for side by side stand, semi-tandem stand, and full tandem stand. Figure 4.3 displays the linguistic values for the three linguistic variables along with specific membership functions for the standing balance test as determined for this experiment. For all membership functions, the abscissa represents time in seconds to complete the task.

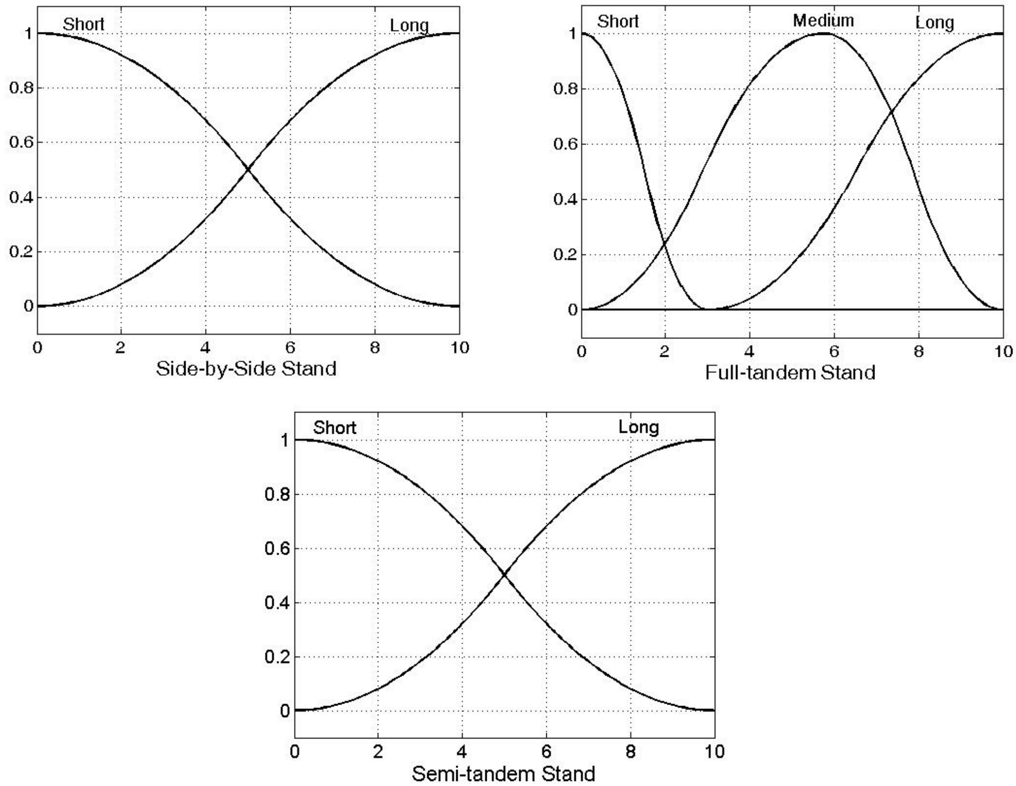


Figure 4.3 Membership Functions of Standing Balance Test

These functions were determined experimentally to preserve the meaning of the scoring categories while softening the boundaries.

In Figure 4.1 there is one input and one output in the walking test. The linguistic values with their membership functions are graphically displayed in Figure 4.4.

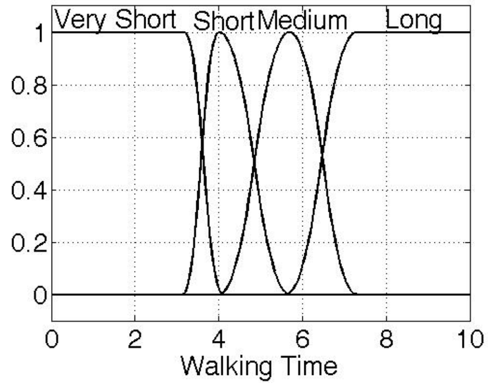


Figure 4.4 Membership Functions for Walking Test

In the sit to stand test also, there is one input and one output. Figure 4.5 shows the linguistic values and their membership functions for this test.

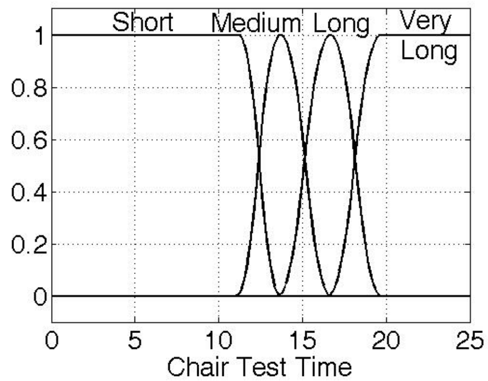


Figure 4.5 Membership Functions of Chair Test

4.1.3. Fuzzy Rules Decision

The fuzzy rules for the soft SPPB test should be decided by the nursing specialist based their knowledge, experience and expectations. Through experimentation, the rules for our initial prototype fuzzy logic system are the following.

Fuzzy Decision Rules for Balance Test

1. If (Side-by-Side_Stand_Time is SHORT) then (Standing_Test_Performance is VERY_POOR)

2. If (Side-by-Side_Stand_Time is LONG) and (Semi-Tandem_Stand_Time is SHORT) then (Standing_Test_Performance is POOR)

3. If (Side-by-Side_Stand_Time is LONG) and (Semi-Tandem_Stand_Time is LONG) and (Full-Tandem_Stand_Time is SHORT) then (Standing_Test_Performance is OK)

4. If (Side-by-Side_Stand_Time is LONG) and (Semi-Tandem_Stand_Time is LONG) and (Full-Tandem_Stand_Time is MEDIUM) then (Standing_Test_Performance is GOOD)

5. If (Side-by-Side_Stand_Time is LONG) and (Semi-Tandem_Stand_Time is LONG) and (Full-Tandem_Stand_Time is LONG) then (Standing_Test_Performance is EXCELLENT)

Fuzzy Decision Rules for Walking Test

1. If (WalkingTime is VERY_SHORT) then (Walking_Test_Performance is EXCELLENT)
2. If (WalkingTime is SHORT) then (Walking_Test_Performance is GOOD)
3. If (WalkingTime is MEDIUM) then (Walking_Test_Performance is OK)
4. If (WalkingTime is LONG) then (Walking_Test_Performance is POOR)

Fuzzy Decision Rules for Sit to Stand Test

1. If (Chair_Test_Time is SHORT) then (Chaire_Test_Performance is GOOD)
2. If (Chair_Test_Time is MEDIUM) then (Chaire_Test_Performance is OK)
3. If (Chair_Test_Time is LONG) then (Chaire_Test_Performance is POOR)
4. If (Chair_Test_Time is VERY_LONG) then (Chaire_Test_Performance is VERY_POOR)

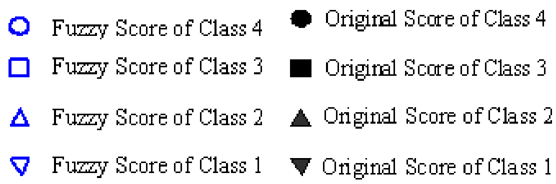
4.1.4. Choice of fuzzy inference system

The most popular models of fuzzy inference systems are the Mamdani models [10] and the Takagi-Sugeno-Kang (TSK) models [11]. Both models use linguistic variables to describe the antecedent part of fuzzy rules. For the fuzzified scoring system of SPPB test, we use a Sugeno constant inference system.

4.2. Results and Analysis

This fuzzy rule-based scoring system of SPPB test was developed with the Fuzzy Logic Toolbox of MATLAB [12]. We use data from Dr. Kathryn Burks, School of Nursing, University of Missouri-Columbia to test the fuzzy scoring system of SPPB test [33]. The data corresponds to the SPPB times recorded from 43 older adults with and without osteoarthritis of the knee collected as a part of a larger study. Figures 4.6, 4.7, and 4.8 display the output values for both the crisp and fuzzy scoring of the SPPB. Each figure has three parts. The first lists crisp and fuzzy scores for each subject for the individual tests while the second part shows the output of both scoring methods as a function of the input parameters.

The following symbols are used in Figures 4.6~4.8.



Fuzzy and Original Balance Score Comparison

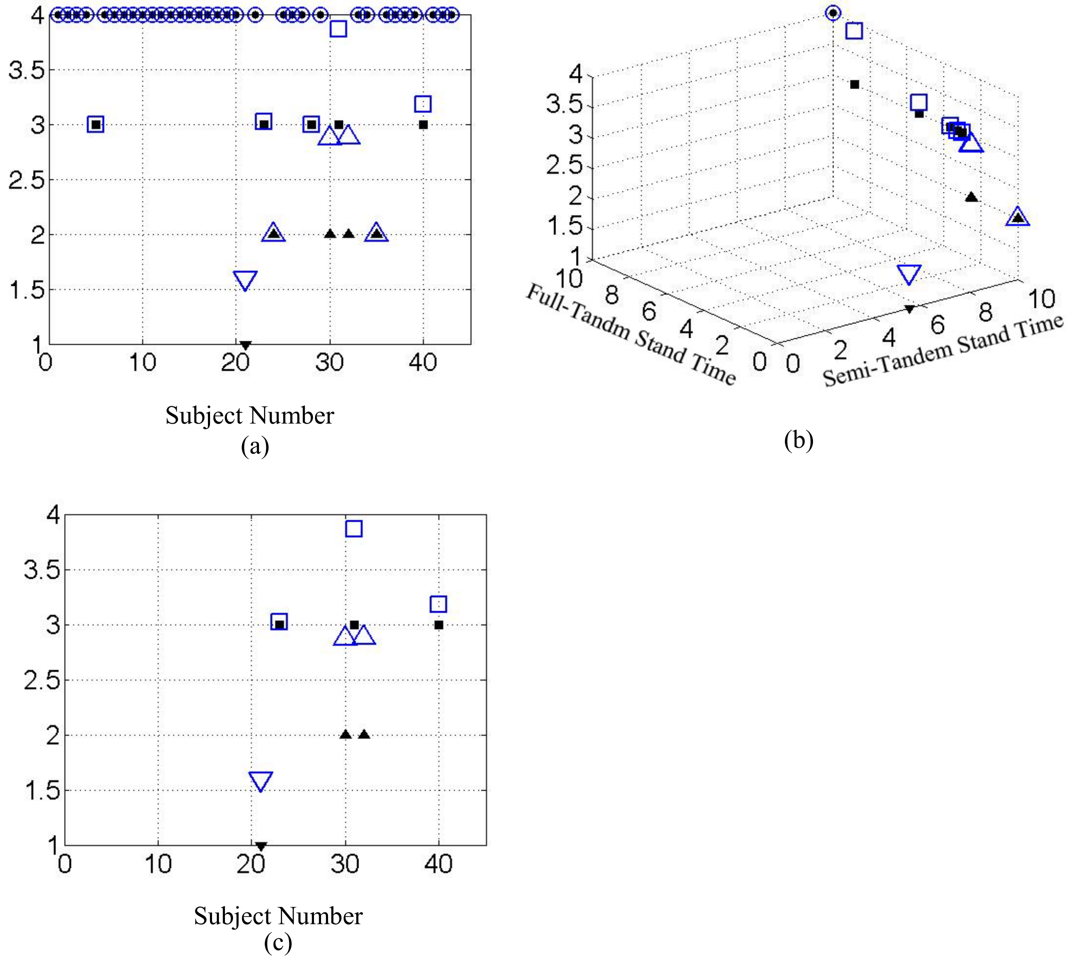


Figure 4.6 Comparisons of Fuzzy Balance Score and Original Balance Score; (a) viewed as function of subject, (b) viewed as function of input parameters, (c) only subjects with different fuzzy and original balance scores are shown.

Fuzzy and Original Walking Test Score Comparison

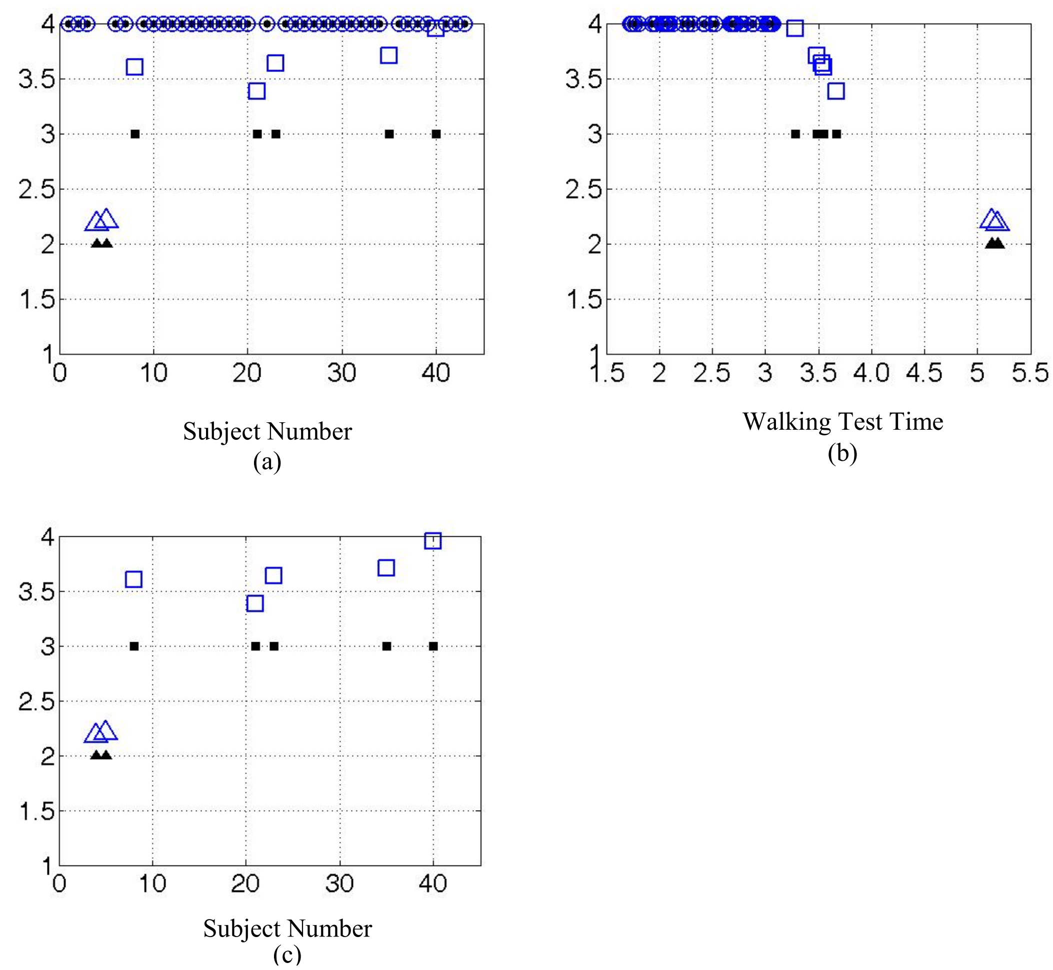


Figure 4.7 Comparisons of Fuzzy Walking Test Score and Original Walking Test Score; (a) viewed as function of subject, (b) viewed as function of input parameters, (c) only subjects with different fuzzy and original balance score are shown.

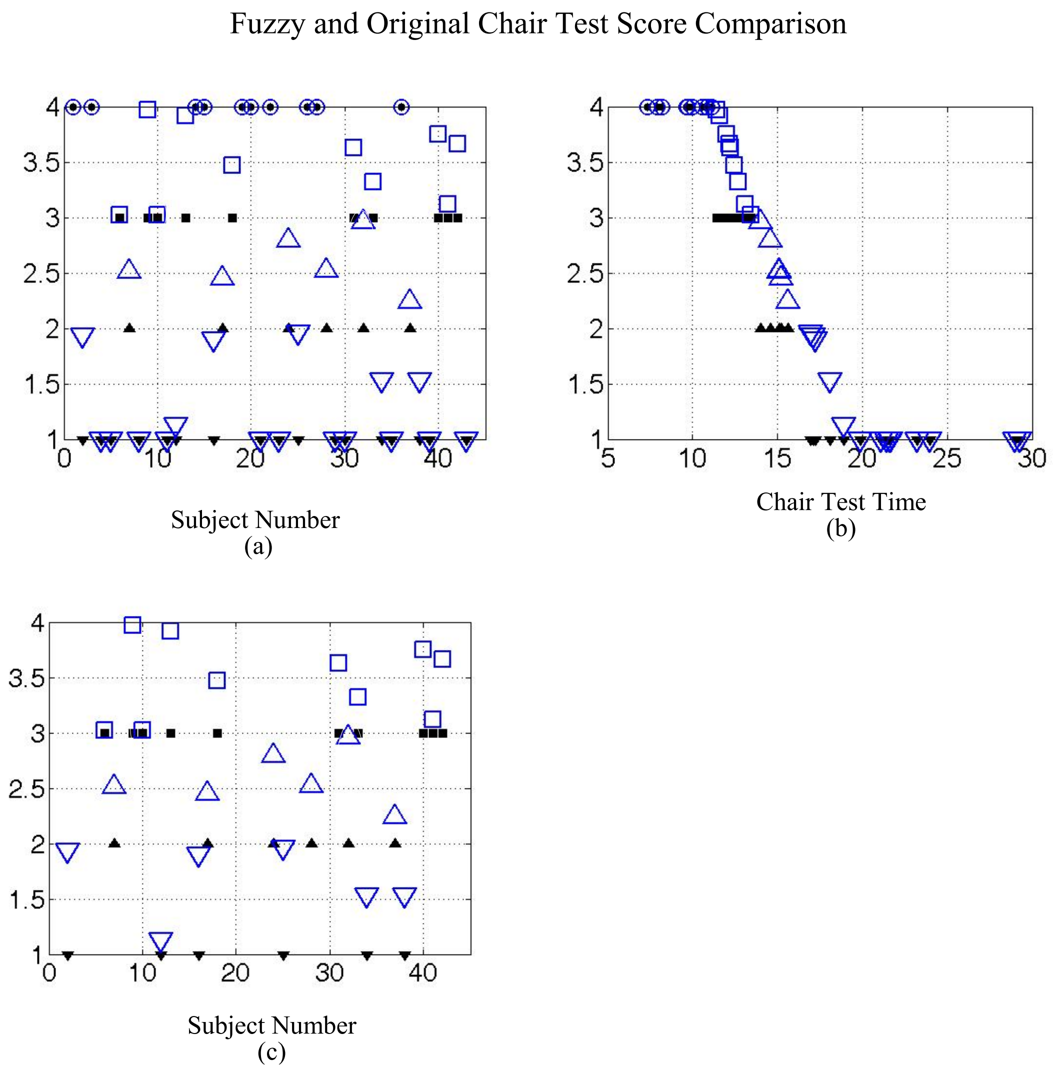


Figure 4.8 Comparisons of Fuzzy Chair Test and Original Chair Test Score; (a) viewed as function of subject, (b) viewed as function of input parameters, (c) only subjects with different fuzzy and original balance scores are shown.

Several observations can be made. First, in many cases, the crisp and fuzzy scoring are consistent, with the fuzzy scores close to the integer crisp classification. This is particularly true for the balance test, where most subjects received 4.0 scores, a potential cause for confusion if the scores from the three subtests are simply added. This situation also indicates that the fuzzy scoring preserves the essential meaning of the SPPB for persons who are “nominal” members of the crisp classes. The fuzzy scores gradually deviate from the crisp number as the subject’s time moves towards the threshold boundaries. The real differences can be seen near the boundaries of the time thresholds. For the few cases in the balance test where the crisp and fuzzy scoring differ (figure 4.6(c)), the fuzzy score is higher than the standard, indicating somewhat better balance than what is reflected in the crisp score. In the sit-to-stand test, subject 10’s time was 13.44 sec. while that for subject 32 was 13.99 sec. These are highlighted in Figure 4.8(a). Within the time range allowed for this test, there is little difference between these two times and yet the crisp scores are 3 and 2, respectively, indicating a substantial difference in performance. Our fuzzy logic scoring system produces values of 3.02 and 2.96, conveying to the caregiver the nearest of the relative physical performance. From the standpoint of frequent automated analysis of physical performance, one of our goals, having gradual scores provides better information to the caregiver to monitor changes in an elder’s physical capabilities. This should allow for timely intervention, perhaps suggesting some physical therapy to slow the decline before it becomes drastic. Crisp class labels then can still have a nominal meaning but this allows for shades of gray to indicate how far from the “prototype” member of that physical category a particular elder performs. Also, higher performance vs. lower performance within a category is made obvious. Figures 4.6(c), 4.7(c), and 4.8(c) show the number of deviations from the crisp categories.

Remarkably, the fuzzy scores on the SPPB provide more sensitive information regarding the status of physical performance in these individuals than is available when using only the crisp scores. This is especially true for the sit-to-stand test. In the first two tests, there is only one score of 1, whereas the sit-to-stand test shows 17 subjects whose crisp score is 1 (Figure 4.8 (c)). Further, the fuzzy scores in Figure 4.8 (b) have more variability. This indicates that fuzzification more realistically shows partial loss of physical performance and potentially at an earlier stage. We conjecture that this soft scoring might be a mechanism to start a treatment intervention that will slow the loss of functionality.

The membership functions and rules of the fuzzified SPPB scoring system are supplied by human experts. For hand-drawn membership functions, it may be difficult to provide an accurate description of membership functions which are general enough to cover the possible variations. The ANFIS method can address this problem. One of the main advantages of the ANFIS is that it is possible to use human knowledge to build an initial system, and then use measured input/output relations to optimize the system. In Chapter 5, the ANFIS method is used to optimize the fuzzified SPPB scoring system.

5. Learning the Fuzzified SPPB Scoring System

5.1. Methodology

5.1.1. Overview

The ANFIS combines fuzzy system and learning methods. It is possible to use relations of measured data to learn a fuzzy system. In Chapter 4, the manually fuzzified SPPB test scoring system is created. But membership functions and rules which are designed manually can not describe the nature variables perfectly. The rules and membership functions which are learned from the ANFIS method are often more precise because the learning algorithm may find better solutions. The ANFIS method with measured data from real tests can be used to learn membership functions and rules for the SPPB test. Before the learning process, the performance of the ANFIS method needs to be evaluated.

5.1.2. Evaluation of the ANFIS

5.1.2.1. Evaluation Methods

Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on ``new'' data. This is the basic idea for cross validation. In n-fold cross-validation, the original sample is partitioned into n subsamples. In the n subsamples, a single subsample is retained as the validation data for testing the model, and the remaining (n-1) subsamples are used as training data. The cross-validation process is then repeated n times (the folds), with each of the n subsamples used exactly once as the validation data. The n results from the folds then can be averaged to produce a single estimation. To evaluate the performance of the ANFIS system, 10-cross validation is used, and training data and test data are in three different methods (see detail in *Input method 1, 2 and 3* of sec. 5.1.1.2). The data used in this validation is from the Health Connection.

Data set 1: test results of the SPPB test (87 subjects). Inputs in these data sets are from Health Connection, and outputs in these data sets are the fuzzy scores which come from the hand-drafted fuzzy rules (Chapter 4) .

Data set 2: a repeat of the SPPB tests by the same subjects as in Data set 1, collected several months later. Inputs in these data sets are from Health Connection, and outputs are the fuzzy scores which come from the hand-drafted fuzzy rules (Chapter 4).

5.1.2.2. Evaluation Results

Input Method 1

Data set 1 and data set 2 are used together to calculate resubstitution error (1 fold) and do 10-fold cross validation. Balance Test, Walking Test and Chair Test are processed separately.

a) 10% of data is used to test, and 90% of data is used to train.
Terminate the loop when TRAINING ERROR DIFFERENCE BETWEEN EVERY 10 EPOCHS (Trn termination)<0.0001.

b) 10% data is used to test, and the other 90% data is divided into checking data (20%) and training data (70%).
Terminate the loop when CHECKING ERROR DIFFERENCE BETWEEN EVERY 10 EPOCHS (Chk termination)<0.0001

The results are summarized here; more detailed results are provided in Appendix A.

The validation results of method 1 show a low error rate (Table 5.1). The performance of the ANFIS training system is very good. Three subtests scoring system learning with the ANFIS are validated separately. Two different loop conditions (Trn termination and Chk termination) are validated separately also. Figure 5.1 shows the training data error and test data error. From this figure, we can see that the loop condition of Trn termination has a lower error rate than Chk termination for each subtest. Figure 5.2 displays the number of epochs' comparison between two loop conditions. Chk termination has fewer epochs for each subtest. In other words, the loop condition of Chk termination can learn faster than the other one.

	Balance Test		Walking Test		Chair Test	
	1) Trn Termination	2) Chk Termination	1) Trn Termination	2) Chk Termination	1) Trn Termination	2) Chk Termination
Avg. Trn Data Err	0.0287	0.0463	0.0183	0.0174	0.0436	0.0594
Avg. Test Data	0.0846	0.1060	0.0682	0.1124	0.0474	0.0786
Epochs	720	199	1382	1167	1579	1158

Table 5.1 Results Comparison of Method 1

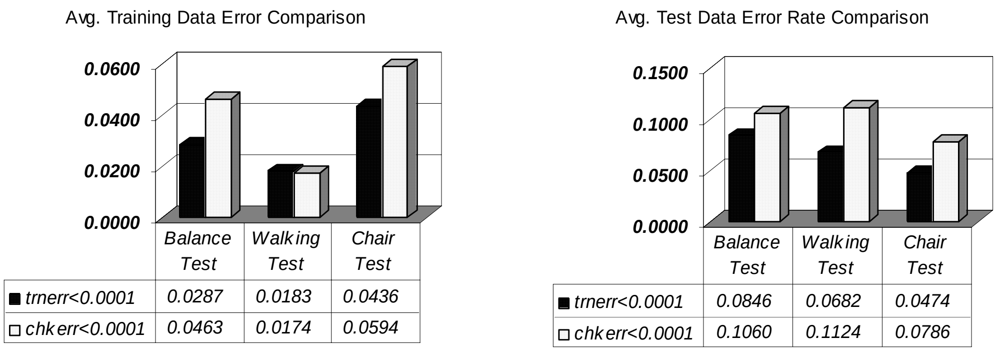


Figure 5.1 Error Rate Comparison of Method 1

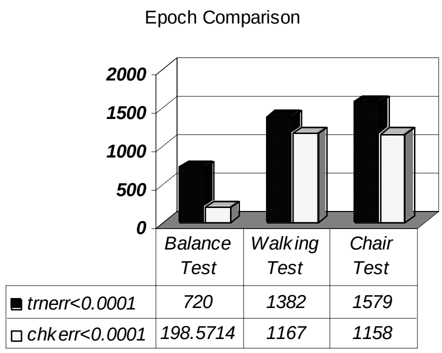


Figure 5.2 Epochs Comparison of Method 1

Input Method 2

Data set 1 is used as training data and Data set 2 is used as checking data and test data to calculate the resubstitution error (1 fold) and do a 10-fold cross validation. Balance Test, Walking Test and Chair Test are processed separately.

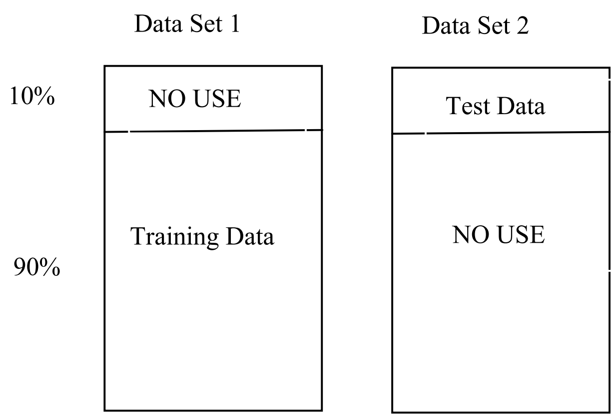


Figure 5.3 Data Selection Structure for (a)

a) 10% data from data set 2 is used to test, and the other 90% data in the corresponding position of data set 1 is used to train. Figure 5.4 displays the structure of data selection.

Terminate the loop when TRAINING ERROR DIFFERENCE BETWEEN EVERY 10 EPOCHS (Trn termination) <0.0001.

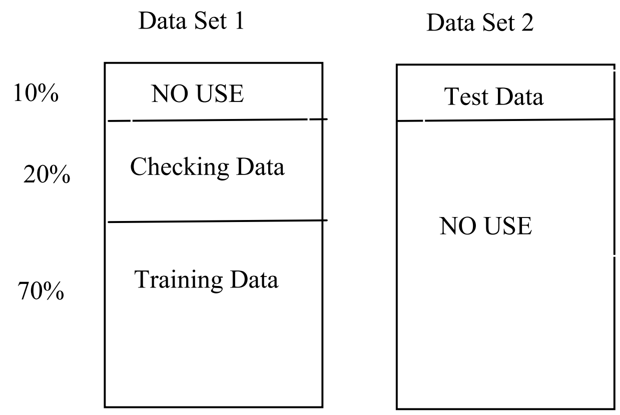


Figure 5.4 Data Selection Structure (b)

b) 10% data from data set 2 is used to test, and other 90% data in the corresponding position of data set 1 is used to train and check (20% for checking and 70% for training), as shown in Figure 5.5.

Terminate the loop when CHECKING ERROR DIFFERENCE BETWEEN EVERY 10 EPOCHS (Chk termination) <0.0001

Again, the results are summarized here; more detailed results can be found in Appendix B.

Table 5.2 shows the validation results of method 2. As shown in the results, the performance of the ANFIS is very good also. Figure 5.6 shows the training data error rate, and we can see that the loop condition of Trn termination has a lower training error rate than the Chk termination for each subtest. Figure 5.7 shows the test data error rate, and both loop condition have similar performance. Comparing the 1 fold chair test results of Figure 5.6 and 5.7, the training error rate of Trn termination (the black one) is much lower than the test error rate; this may indicate a over feeding problem. The loop criterion of Chk termination can address this problem. Figure 5.8 shows the epochs' comparison between two loop conditions, and Chk termination has much fewer epochs for each subtest.

	Balance Test		Walking Test		Chair Test	
	1) Trn Termination	2) Chk Termination	1) Trn Termination	2) Chk Termination	1) Trn Termination	2) Chk Termination
Trn Error (1 fold)	0.0578	0.0618	0.0014	0.0013	0.0506	0.1649
Test Error (1 Fold)	0.0422	0.0483	0.0143	0.0115	0.2425	0.1855
Epochs (1 Fold)	620	40	2790	860	3840	30
Avg. Trn Error (10 Fold)	0.0581	0.06013	0.0029	0.0107	0.0539	0.0588
Avg.Test Error (10 Fold)	0.0396	0.0550	0.0174	0.0664	0.1707	0.2162
Epochs (10 Fold)	580	460	2762	1031	3048	1780

Table 5.2 Results Comparison of Method 2

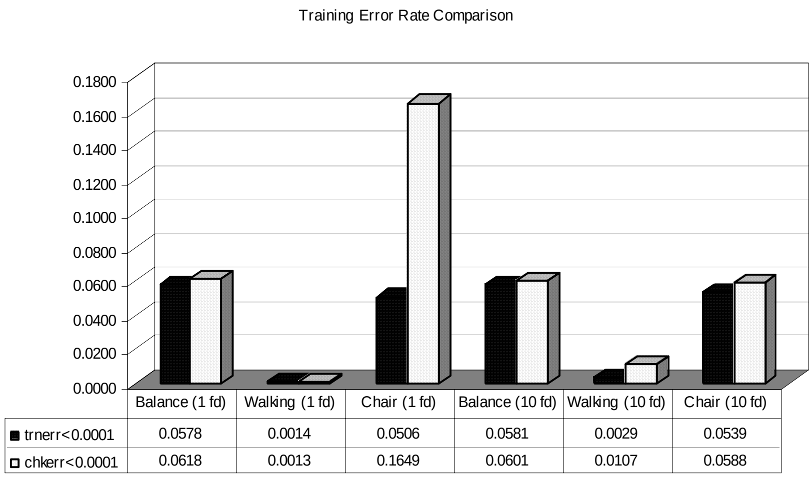
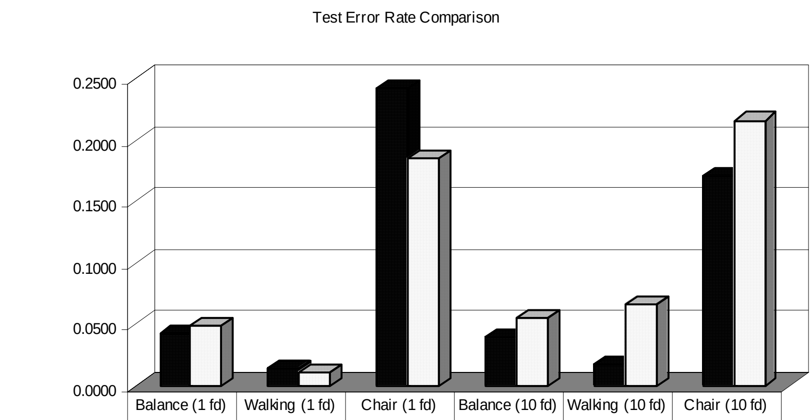
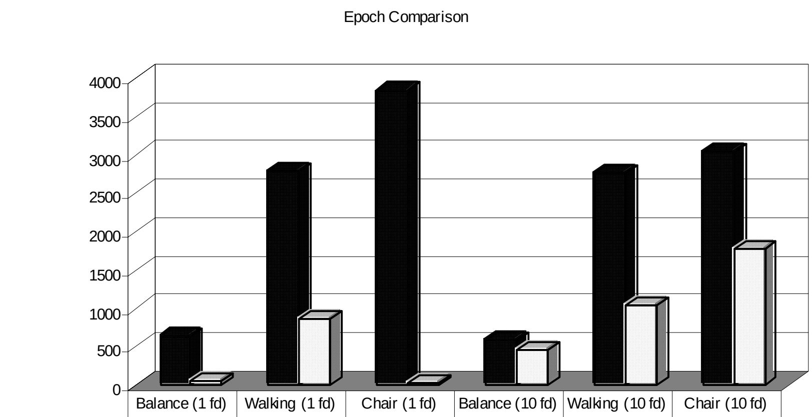


Figure 5.5 Training Error Rate Comparison of Method 2



	Balance (1 fd)	Walking (1 fd)	Chair (1 fd)	Balance (10 fd)	Walking (10 fd)	Chair (10 fd)
trnerr<0.0001	0.0422	0.0143	0.2425	0.0396	0.0174	0.1707
chkerr<0.0001	0.0483	0.0115	0.1855	0.0550	0.0664	0.2162

Figure 5.6 Test Error Rate Comparison of Method 2



	Balance (1 fd)	Walking (1 fd)	Chair (1 fd)	Balance (10 fd)	Walking (10 fd)	Chair (10 fd)
trnerr>0.0001	620	2790	3840	580	2762	3048
chkerr>0.0001	40	860	30	460	1031	1780

Figure 5.7 Epochs Comparison of Method 2.

Input Method 3

In method 3, data set 1 and data set 2 are swapped. Data set 2 is used as training data and Data set 1 is used as checking data and test data to calculate resubstitution error (1 fold) and do 10-fold cross validation. The same observations and conclusions can be made from Table 5.3 and Figure 5.9~5.11 as in method 2. More detailed results can be found in Appendix C.

	Balance Test		Walking Test		Chair Test	
	1) Training Termination	2) Checking Termination	1) Training Termination	2) Checking Termination	1) Training Termination	2) Checking Termination
Training Error (1 fold)	0.0334	0.0421	0.0084	0.0017	0.1592	0.0141
Test Error (1 Fold)	0.2142	0.2497	0.0135	0.0122	0.1216	0.0150
Epochs (1 Fold)	1940	270	1340	1710	1520	1800
Avg. Training Error (10 Fold)	0.0324	0.0418	0.0174	0.0257	0.1575	0.0728
Avg.Test Error (10 Fold)	0.1138	0.1823	0.0708	0.1170	0.1152	0.0727
Epochs (10 Fold)	2032	270	1284	1007	1655	1051

Table 5.3 Results Comparison of Method 3.

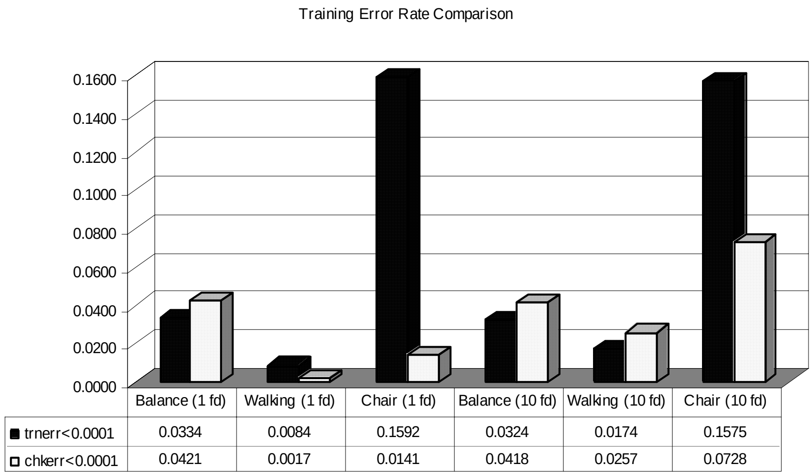


Figure 5.8 Training Error Rate Comparison of Method 3

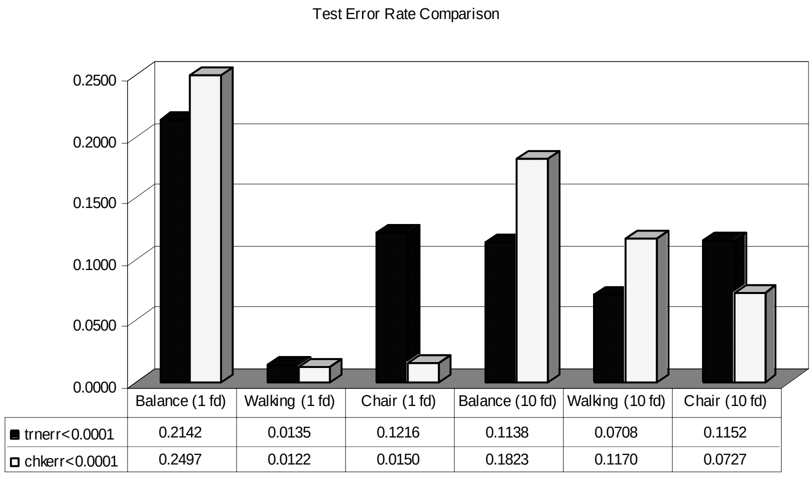


Figure 5.9 Training Error Rate Comparison of Method 3

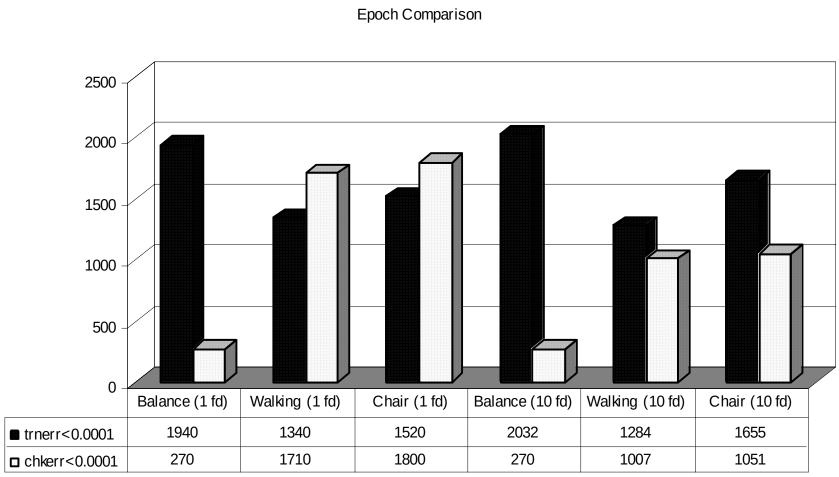


Figure 5.10 Epochs Comparison of Method 3

5.1.2.3. Evaluation Conclusion

The performance of the ANFIS system is very good. The loop condition of Chk termination can run faster. The performance of Chk termination is better when the data set is large enough. For small data sets, a loop condition of Trn termination is a better choice, because its performance in small data sets is better although it consumes time.

Because of the small data sets we have, the ANFIS with loop condition of Trn termination is used in the following experiments.

5.2. The Results of Fuzzified SPPB Test Learned from ANFIS

From the former validation results, we can see that the ANFIS method has an excellent performance. Because the ANFIS method can use measured input/output relations to learn a system, it can be used to learn membership functions and rules of the fuzzified SPPB test scoring system. For this learning procedure, the data set from the Health Connection is used as input training data; the output of training data is the result of the Health Connection data fired in the manually fuzzified SPPB test.

The Heath Connection data use 50-foot gait speed test. The relationship between the 50-foot walk and the 8-foot walk (discussed in sec. 3.2) can be expressed as:

$$\text{50-foot walking time(seconds)}=0.014(\text{8-ft walking time })^2-0.31\left(\text{8-ft walking time}\right)+4.7$$

which is used to convert the 50-foot walk time to a 8-foot walk time.

For the training program, the loop terminates at the difference of training error rate less than 0.1% between every ten epochs.

5.2.1. Balance Test Results

Figure 5.12 displays the system structure of the balance test. There are three inputs and 1 output in the balance system. Input1 is the side-by-side stand with 2 membership functions, input2 is the semi-tandem stand with two membership functions, and input3 is the full-tandem stand with three membership functions. After training with the ANFIS, there are 12 rules in the system. The Membership functions are shown in Figure 5.13. The training error of this system is 0.34%. Data from Dr. Kathryn Burks, School of Nursing, University of Missouri-Columbia is used to test the system, and test error is 0.36%.

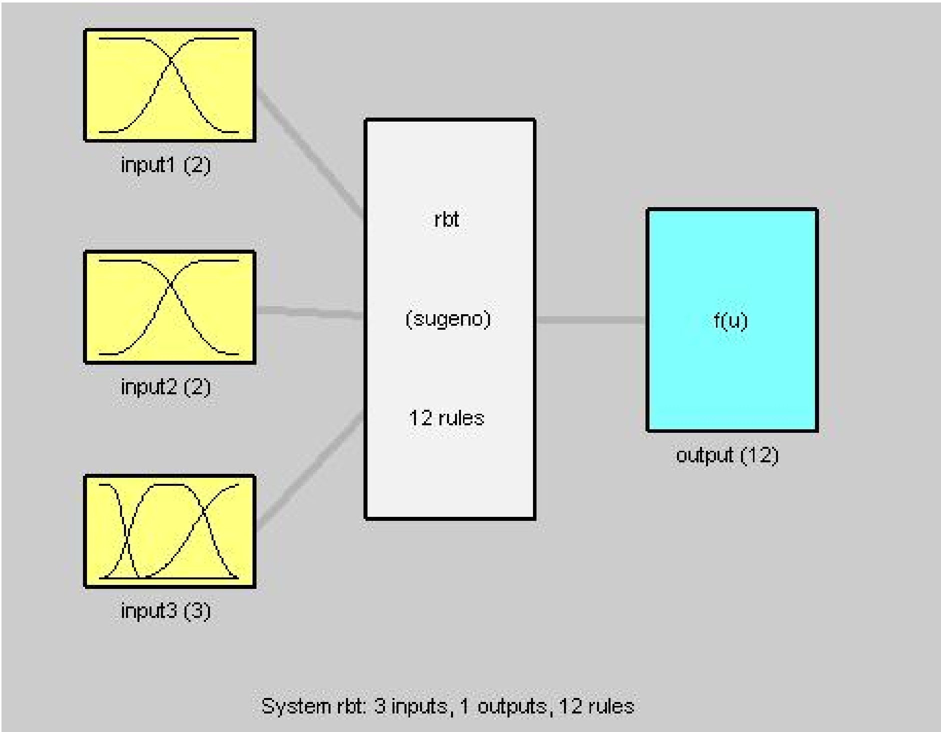


Figure 5.11 System Structure of Balance Test

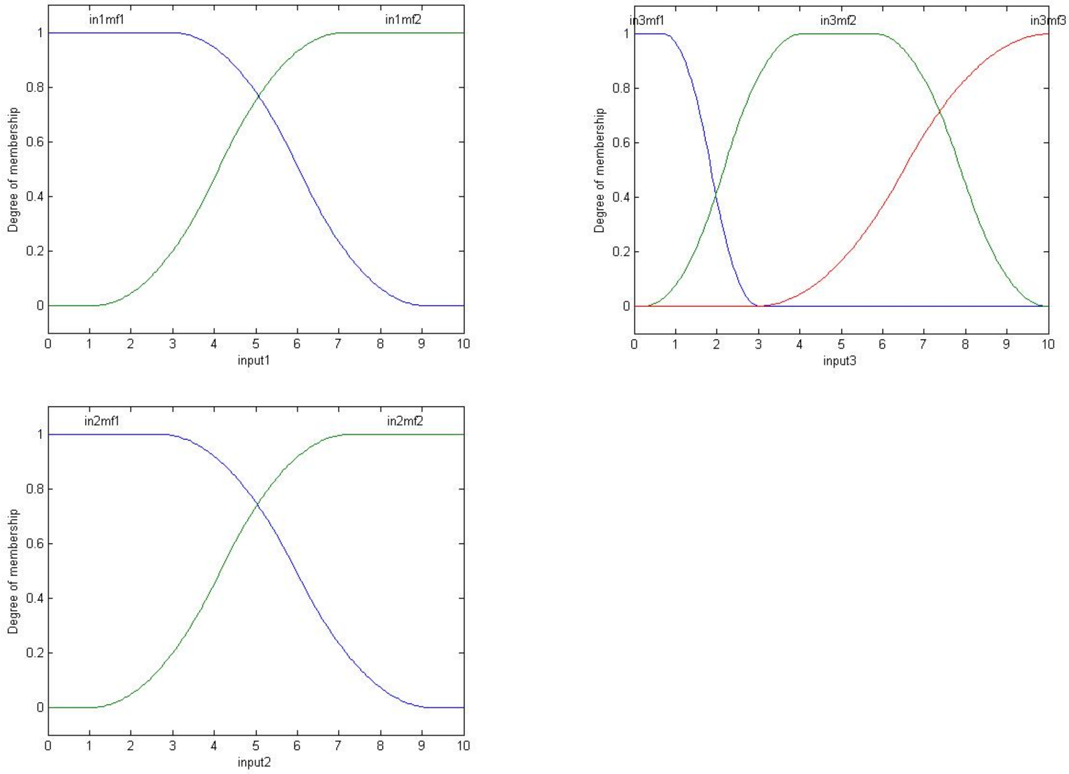


Figure 5.12 Membership Functions of Balance Test

5.2.2. Gait Speed Test Results

Figure 5.14 displays the system structure of the gait speed test. There are one input and one output in this system. Input1 is the time of the gait speed test. After training with the ANFIS, there are 4 rules in the system. As shown in Figure 5.15, input1 has four membership functions. The training error of this system is 0.28%. Data from Dr. Kathryn Burks, School of Nursing, University of Missouri-Columbia is used to test the system, and test error is close to 0.00%.

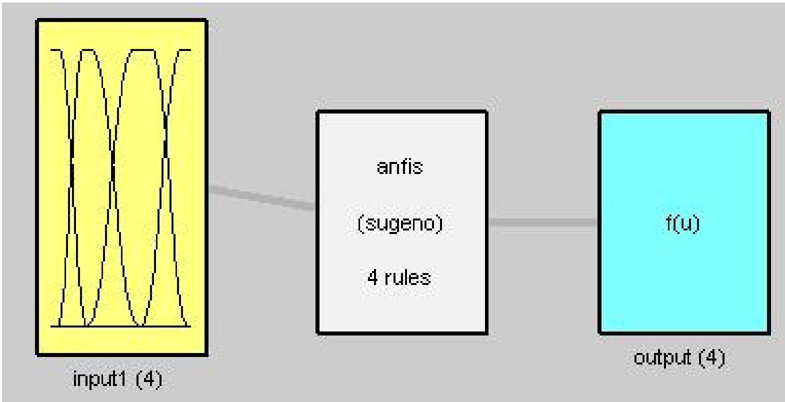


Figure 5.13 System Structure of Gait Speed test

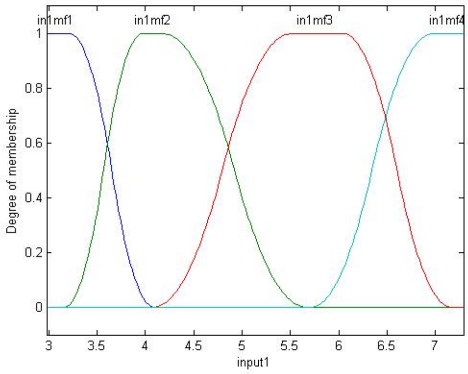


Figure 5.14 Membership Functions of Gait Speed Test

5.2.3. Chair Stand Test Results

Figure 5.16 displays the system structure of the chair stand test. There are one input and one output in this system. Input1 is the time of the chair stand test. After training with the ANFIS, there are 5 rules in the system. As shown in Figure 5.17, input1 has five membership functions. The training error of this system is 3.38%. Data from Dr. Kathryn Burks, School of Nursing, University of Missouri-Columbia is used to test the system, and test error rate is 4.00 %. Because the training data variations are very limited for this test, the error rate of this system is higher than the others.

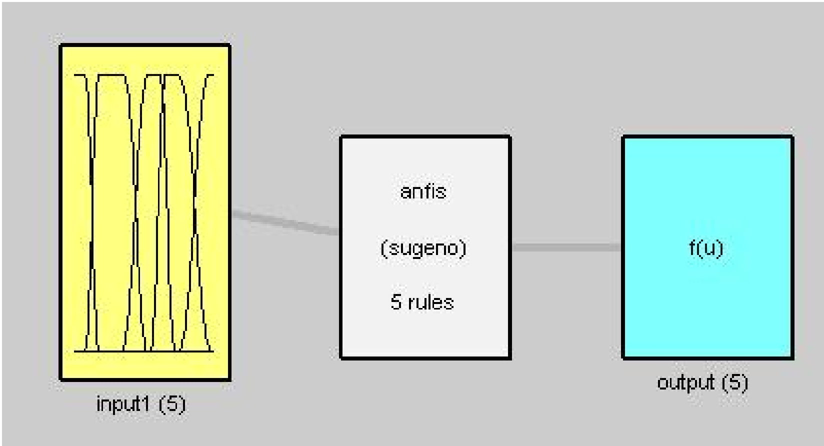


Figure 5.15 System Structure of Chair Stand Test

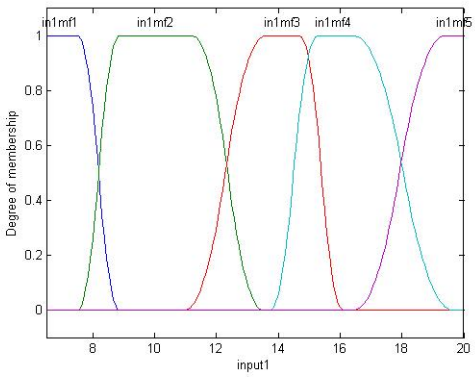


Figure 5.16 Membership Functions of Chair Stan Test

6. Tuning the Fuzzified SPPB Test

6.1. Introduction

The basic motivation of our tuning system is to provide a flexible system that can handle changes in the assessment system, to reflect the expert opinion of nurses and physical therapists. The tuning system should dynamically change the physical performance score in a meaningful pattern. In such a tuning system, the score boundaries need to be fixed, such as, 0 as the lower boundary and 4 as the higher, and the shift of a score should be uniform and smooth. This problem is generalized as an equation solving problem. For this purpose, the particle swarm optimization algorithm (PSO) is used to solve this problem. PSO algorithms are especially useful for parameter optimization in continuous, multi-dimensional search spaces. It is mainly inspired by social behavior patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees.

6.2. Methodology

6.2.1. PSO

Particle swarm optimization (PSO) is a technology to address this problem. The particle swarm concept originated as a simulation of a simplified social system. The original intent was to graphically simulate the choreography of a flock of birds or a school of fish. However, it was found that the particle swarm model can be used as an optimizer. Particle swarm optimization (PSO) is an evolutionary computation technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [34]. It has been compared to genetic algorithms for efficiently finding optimal or near-optimal solutions in large search spaces. The technique involves simulating social behavior among individuals (particles) “flying” through a multidimensional search space, each particle representing a single intersection of all search dimensions. The particles evaluate their positions relative to a goal (fitness) at every iteration, and particles in a local neighborhood share memories of their “best” positions, and then use those memories to adjust their own velocities, and thus subsequent positions.

Similar to genetic algorithms (GA), PSO is a population based optimization tool. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, are “flown” through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. Most of evolutionary techniques have the following procedure: first, random generation of an initial population; second, reckoning of a fitness value for each subject, which will directly depend on the distance to the optimum; third, reproduction of the population based on fitness values; fourth if requirements are met, then stop. Otherwise go back to the second step.

The flowchart of the PSO algorithm is given as follows:

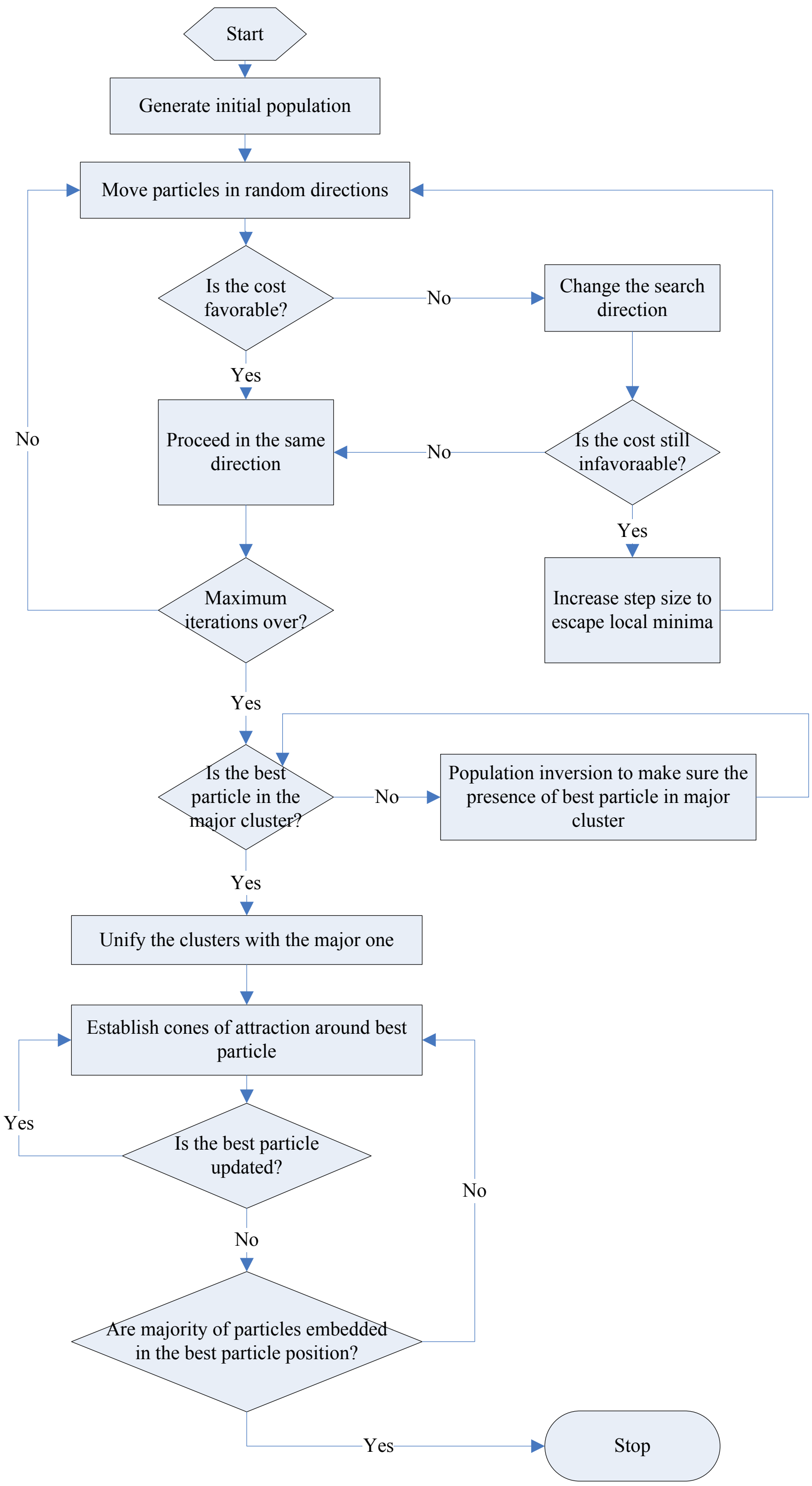


Figure 6.1 Flowchart of the PSO algorithm

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two “best” values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pBest. Another “best” value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best and is called gBest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lBest.

After finding the two best values, the particle updates its velocity and positions with following formulas.

$$v = v + c_1 \times rand() \times (pBest - present) + c_2 \times rand() \times (gBest - present)$$
$$present = present + v$$

Where v is the particle’s velocity, present the current position of the particle (solution). pBest and gBest are defined as stated before. Rand() is a random number between 0 and 1. c_1 and c_2 are learning factors.

Following is the pseudo code of the procedure

```
PSO(){
    For each particle {
        Initialize particle
    }
    Do {
        For each particle {
            Calculate fitness value
```

If the fitness value is better than the best fitness value (pBest) in history {

Set current value as the new pBest

}

Choose the particle with the best fitness value of all the particles as the gBest

For each particle {

Calculate particle velocity and update particle position

according the above formula

}

} While maximum iterations or minimum error criteria is not attained

} // end of PSO()

Particles' velocities on each dimension are clamped to a maximum velocity V_{max}. If the sum of accelerations would cause the velocity on that dimension to exceed V_{max}, which is a parameter specified by the user, then the velocity on that dimension is limited to V_{max}.

6.2.2. Experiment Design

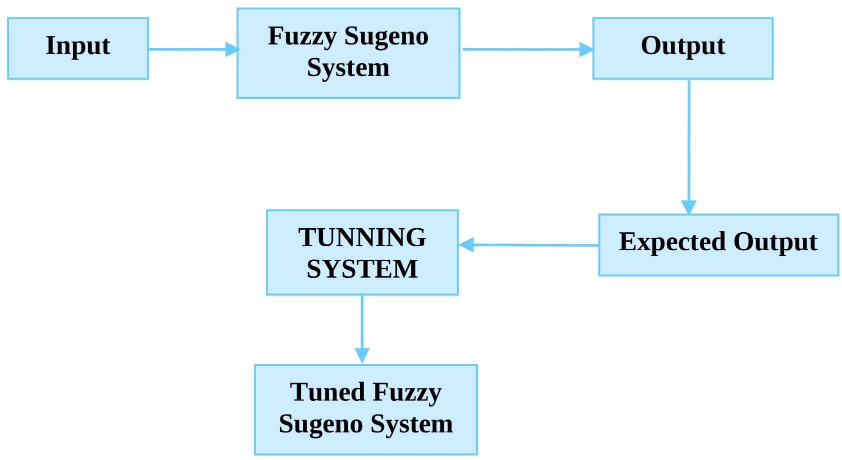


Figure 6.2 Tuning Procedure

Figure 6.2 shows the tuning procedure. In the TUNING SYSTEM, the PSO algorithm is used to look for several key parameters. These key parameters will be sent back to the fuzzy system to generate a tuned fuzzy system. The output of a fuzzy Sugeno system is the weighted average of all rule outputs, computed as:

$$\text{Final Output=A=}\frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{6.1}$$

$$\sum_{i=1}^{N} w_i z_i = A\sum_{i=1}^{N} w_i \tag{6.2}$$

In the tuning procedure, the two end points of the output scale need to be fixed, so z_1 and z_N are unchanged. In the equation 6.2 w_i, z_1, z_N are known and moved to the right of equation. Then equation 6.2 can be written as equation 6.3. The PSO algorithm looks for a sequence of z_i which fits for the equation 6.3. This sequence of z_i constitute the key parameters which need to be sent back to the fuzzy system.

$$\sum_{i=2}^{N-1} w_i z_i = A\sum_{i=1}^{N} w_i - w_1 z_1 - w_N z_N \tag{6.3}$$

PSO has shown to be an efficient optimizer, especially in large, convoluted search space. Since solving equations, linear or nonlinear, is by itself an optimization problem, PSO is a candidate to solve such systems.

For the equation $\sum_{i=2}^{N-1} w_i z_i = \text{Constant}$ (equation 6.3), there are infinite answers for z_i. It is decided by another fitness function which z_i is better. Equation 6.4 is the fitness function that makes the new z_i sequence fit the tuned point. According to the requirements, a system must change uniformly and smoothly, so we can use the fitness function in equation 6.5 to guarantee it. Therefore, there are two fitness functions in the system as follows:

$$\sum_{i=2}^{N-1} w_i z_i^{'} - A\sum_{i=1}^{N} w_i - w_1 z_1 - w_N z_N = 0 \tag{6.4}$$

$$\min\left(\sqrt{\frac{\sum_{i=2}^{N-1} w_i (z_i^{'} - z_i)^2}{\sum_{i=2}^{N-1} w_i}}\right) \tag{6.5}$$

where $z_i^{'}$ represents the new z_i sequence after tuned.

6.3. Results and Discussion

Two fitness functions are decided above. Equation 6.4 and equation 6.5 are combined together as one fitness function in program, the model can be written as:

$$\min\left\{\alpha\left(\left|\sum_{i=2}^{N-1} w_i z_i^{'} - A\sum_{i=1}^{N} w_i - w_1 z_1 - w_N z_N\right|\right) + \beta\sqrt{\frac{\sum_{i=2}^{N-1} w_i (z_i^{'} - z_i)^2}{\sum_{i=2}^{N-1} w_i}}\right\} \quad \text{where } \alpha + \beta = 1 \qquad (6.6)$$

The parameters α and β are weights for the fitness functions. They help to control the convergence speed of each fitness function. In this application, fitness function 1 (equation 6.4) converges to zero faster than fitness function 2 (equation 6.5), so α must be a little larger than β. Several tests about the parameters α and β have been done, and the results can be found in Appendix D. When α =0.67 and β =0.33, the effect of fitness function 1 (equation 6.4) overtakes fitness function 2 (equation 6.5). When $0.7 \le \alpha \le 0.9$ and $0.1 \le \beta \le 0.3$, the performance of the system is acceptable. When α is too large compared to β, fitness function 2 (equation 6.5) overtakes fitness function 1 (equation 6.4). Users can select α and β according to their requirements.

A tuning system with the above fitness function is tested on the fuzzified SPPB system. Table 6.1 shows one test result on the fuzzified balance test system. The input side-by-side test is 10 seconds, the semi-tandem test is 9 seconds and the full tandem test is 2 seconds. The output of the fuzzy score is 2.46. Suppose the output 2.46 needs to be tuned to 2.74.

For output 2.46, according to equation 6.4 we can get following answer:

$$\sum_{i=2}^{4} w_i z_i = 1.1669 \qquad (6.7)$$

For output 2.7374, according to equation 6.4 we can get following answer:

$$\sum_{i=2}^{4} w_i z_i' = 1.3 \qquad (6.8)$$

The tuning system needs to look for a sequence of $z'(z_2', z_3', z_4')$ which satisfies equation 6.8. Table 6.1 shows the results. The sequence of $z'(z_2', z_3', z_4')$ is distributed uniformly, and standard deviations are very small. Similar tests were done on variety of points, and all results are very good. More detailed results are provided in Appendix D.

Input: [10 9 2]

2.46→2.74 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.1669 to 1.3)

Test #	Result	z_2	z_3	z_4
1	1.3	1.28	2.28	3.28
2	1.3	1.27	2.28	3.28
3	1.3	1.28	2.28	3.28
4	1.3	1.09	2.29	3.29
5	1.3	1.25	2.29	3.27
6	1.3	1.26	2.27	3.29
7	1.3	1.30	2.22	3.28
8	1.3	1.30	2.24	3.32
9	1.3	1.27	2.31	3.26
10	1.3	1.28	2.28	3.28
STD	0	0.06	0.02	0.01

Table 6.1 Tuning Test on the Fuzzified Balance Test System (2.46→2.74)

Figure 6.3 displays a fuzzy balance system with input [10 9 2] and output [2.46]. Figure 6.4 shows a fuzzy balance system tuned from Figure 6.3. The output is tuned from [2.46] to [3.3]. The output membership functions are updated.

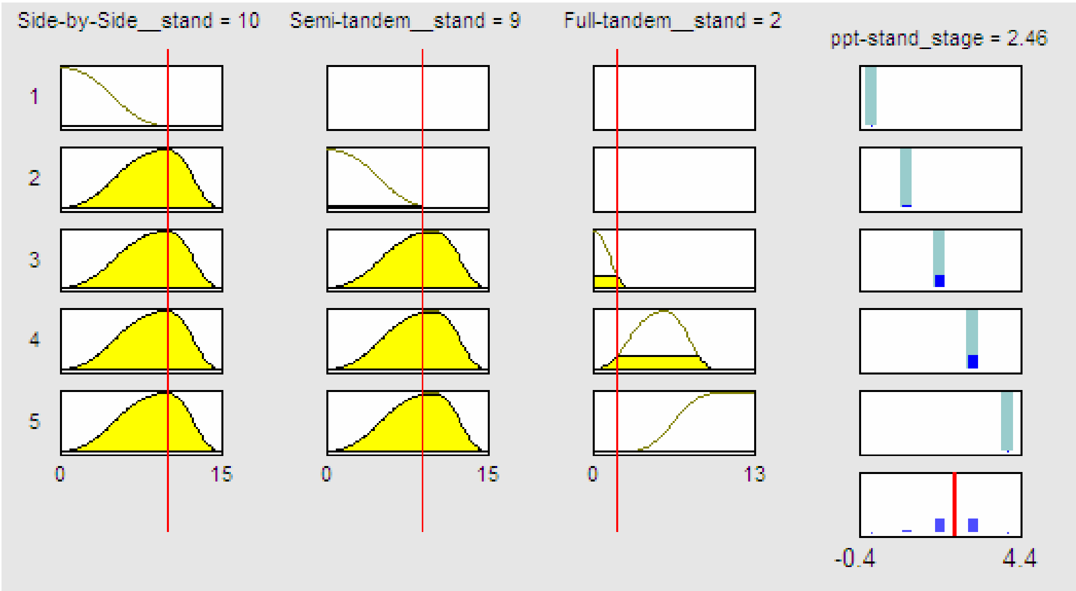


Figure 6.3 Fuzzy SPPB Balance Test System

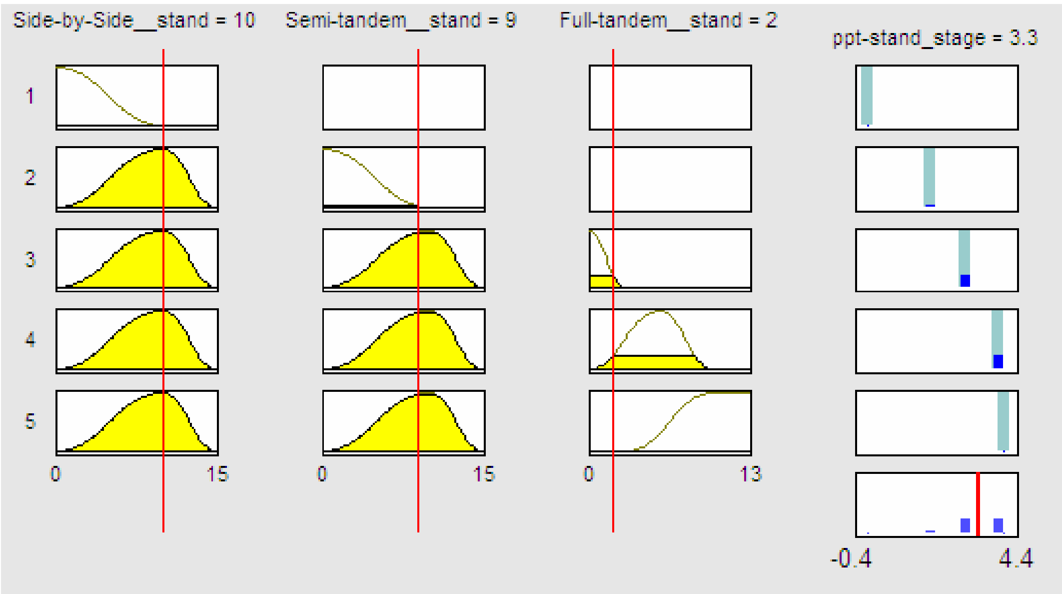


Figure 6.4 Fuzzy SPPB Balance Test System after Tuned

Figure 6.5 and Figure 6.6 display the surfaces of a tuned and untuned fuzzy balance test system. The tuned fuzzy system keeps most of the character of the original system.

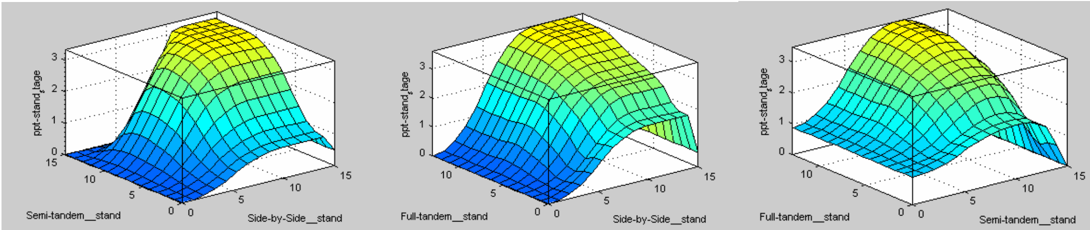


Figure 6.5 Fuzzy SPPB Balance Test Surface

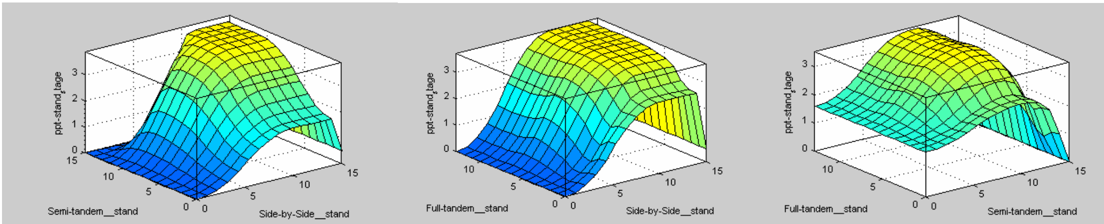


Figure 6.6 Fuzzy SPPB Balance Test System Surface after Tuned

6.4. User interface and functionalities of the program

The tuning program is implemented using Matlab 7.0. The following figure shows the interface of the program, which was done using the Matlab fuzzy toolbox.

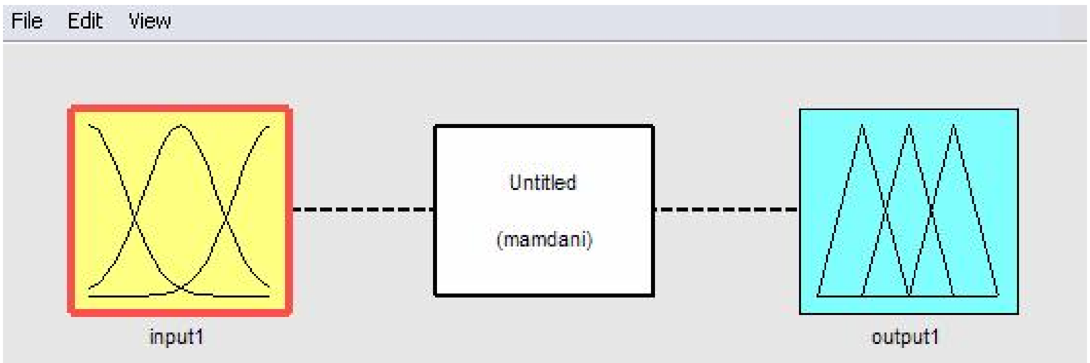


Figure 6.7 User interface of the program

Load the fuzzy stem you want to tune from here (figure 6.7). This interface looks the same as the original fuzzy toolbox in Matlab, but some functions are changed.

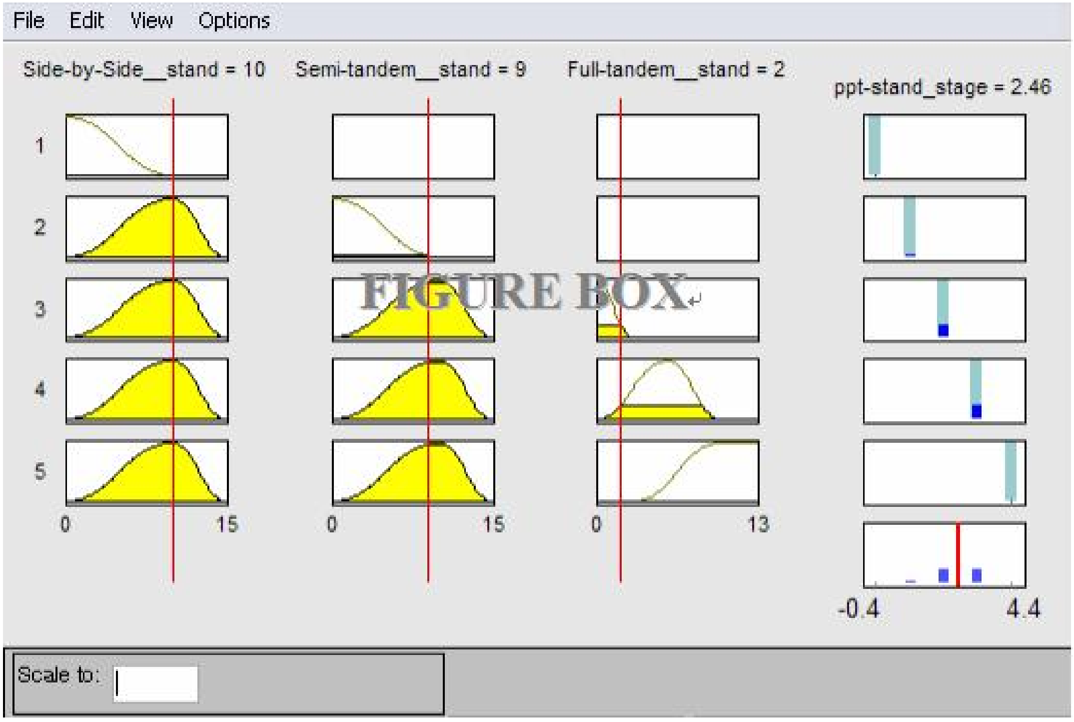


Figure 6.8 User interface of the program

From view → Rules, you can find this interface. According to the current result, the user can input the value to be tuned in the SCALE TO box. The results of the system after tuning will display in the figure box (as shown in Figure 6.8). When you enter data into the INPUT box, the system will produce the result from the tuned fuzzy system. All other interface usage is the same with the Matlab fuzzy toolbox 7.0.1.

7. Conclusion

The intent of this thesis is to introduce the fuzzified SPPB test scoring system and its tuning system. For the SPPB walk test, it is standardized on the 8-foot walk. The conversion between 50-foot walk and 8-foot walk is done based on the curve fitting method.

The crisp nature of the SPPB scoring system leads to common anomalies near threshold boundaries and does not possess a fine enough granularity. A fuzzy logic rule-based system can preserve the original design of the SPPB but addresses these two shortcomings. In Chapter 4, the fuzzified SPPB scoring system with the manually crafted rules and membership functions is created. The test results show that this system preserves the original design of the SPPB, and the fuzzy scores provide more sensitive information regarding the status of physical performance.

For hand-crafted membership functions, it may be difficult to provide an accurate description of membership functions which are general enough to cover the possible variations. The ANFIS method can use measured input/output relations to train membership functions and rules. The rules and membership functions which are learned from the ANFIS method are often more precise because the learning algorithm may find better solutions. In Chapter 5, the ANFIS method is introduced. The performance of the ANFIS is validated, and it shows good performance. Then, the ANFIS method is used to learn the rules and membership functions of the fuzzified SPPB system. The learned fizzified SPPB system shows good performance and has a low error rate.

In Chapter 6, the PSO method is used to create the tuning system for the fuzzified SPPB test. The fuzzified SPPB tuning system provides a flexible system that can handle changes in the assessment system.

In this thesis, there are several things that can be improved. Although the ANFIS shows good performance, it is still a problem to find the optimum number of fuzzy rules in the fuzzy model. To improve the number of rules, a pruning technique can be applied to the ANFIS in the future. For the fuzzified SPPB tuning system, a tuning system which can handle changing of several points without fixed end points can be created. In addition, nursing specialists of our research team are working on an expert panel study in which the subjects are ranked according to the knowledge of specialists. After this study is finished, a more precise fuzzified SPPB system can be learned from ANFIS.

The major goal of our extended research team is to introduce advanced sensors, novel signal/image processing, and high level reasoning to enhance the independence and safety of older people while maintaining privacy and minimizing interference. Our goal is consistent with Zadeh's definition of Recognition Technology to provide a "quantum jump in the capabilities of today's recognition systems" [35]. Keeping track of the day-to-day physical performance capabilities plays a major role in this effort. One of the benchmark methods for measuring such performance is the Short Physical Performance Battery. Since we will be able to perform similar measurements frequently, though perhaps with more variation than is seen in the caregiver assisted exam, a more robust method of scoring, with finer sensitivity, is needed.

BIBLIOGRAPHY

[1] M. Rantz, K. Marek, M. Aud, R. Johnson, D. Otto, R. Porter, "TigerPlace: A New Future for Older Adults," *Journal of Nursing Care Quality*, vol. 20, no. 1, 2005, pp. 1-4.

[2] M. Skubic, "AI Technologies in TigerPlace," the AAAI Fall 2005 Symposium Workshop on Caring Machines: AI in Eldercare, Arlington, VA., November 2005.

[3] J. Guralnik, E. Simonsick, L. Ferrucci, et al., "A short physical performance battery assessing lower extremity function: association with self-reported disability and prediction of mortality and nursing home admission," *J Gerontol Med Sci.,* vol.;49, 1994, M85–M94.

[4] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.

[5] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed Representation of Fuzzy Rules and Its Application to Pattern Classification," *Fuzzy Sets and Systems*, Vol. 52, pp. 21-32, 1992.

[6] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY: Plenum, 1981.

[7] J. C. Bezdek, S. K. Chuah, and D. Leep, "Generalized K-Nearest Neighbor Rules," *Fuzzy Sets and Systems*, Vol. 18, No. 3, pp. 237-256, 1986.

[8] J. M. Keller, M. R. Gray, and J. A. Givens, "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 15, No. 4, pp. 580-585, 1985.

[9] R. L. Chang and T. Pavlidis, "Fuzzy Decision Tree Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 7, No. 1, pp. 28-35, 1977.

[10] E. H. Mamdani, "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant," *IEEE Proceedings*, Vol. 121, No. 12, pp. 1585-1588, 1974.

[11] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Application to Modeling and Control," *IEEE Transactions on System, Man, Cybernetics*, Vol. 15, No. 1, pp. 116-132, 1985.

[12] The Math Work (1995). Fuzzy Logic Toolbox for use with MATLAB – User's Guide. The Math Works, Massachusetts, US.

[13] M. Sugeno and G. T. Kang, "Structure Identification of Fuzzy Model," *Fuzzy Sets and Systems*, Vol. 28, No. 1, pp. 15-33, 1988.

[14] C. L. Seplaki, N. Goldman, M. Weinstein, and Y. Lin, "How Are Biomarkers Related to Physical and Mental Well-Being?" *Journals of Gerontology Series A*, vol 59(3), pp. 201-17, Mar. 2004.

[15] D. Wieland, V. Lamb, H. Wanf, S. Sutton, GP. Eleazer, J. Egbert, "Participants in the Program of All-Inclusive care for the Elderly (PACE) demonstration: developing disease-impairment-disability profiles," *Gerontologist.* 40(2), pp. 219-27, Apr. 2000.

[16] F. Portrait, M. Lindeboom, D. Deeg, "Health and mortality of the elderly: the grade of membership method classification and determination," *Health Economics.* 8(5), pp. 441-57, 1999.

[17] KG. Manton, ES. Cornelius, MA. Woodbur, "Nursing home residents: a multivariate analysis of their medical, behavioral, psychosocial, and service use characteristics," *Journals of Gerontology Series A- Biological Sciences & Medical Sciences.* 50(5), pp. M242-51, 1995.

[18] RP. Erickson, PM. Di Lorenz, MA. Woodbury, "Classification of teste responses in brain stem: membership in fuzzy sets," *Journal of Neurophysiology*. 71(6), pp. 2139-50, Jun. 1994.

[19] K. Marton, M. Woodbury, "Grade of Membership generalization and aging research," *Exper. Aging Research.* 17(4), pp. 217-26,1991.

[20] Jang, J. S. R., Sun, C. T., and Mizutani, E.: *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997.

[21] J.-S. R. Jang. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transaction on Systems, Man and Cybernetice*, pages 665-685, 1993.

[22] S. Khanmohammadi L. Hassanzadeh and J. Jiang Gh. Alizadeh. Imlementation of a Functional Link Net_ANFIS Controller for a Robot Manipulator. T*hird International Workshop on Robot Motion and Control*, pages 399-404, 2002.

[23] Jettrey T. Drake and Nadipuram R. Prasad. ANFIS for parameter estimation in coherent communications phase synchronization. *IEEE Signal Processing Society Workshop*, page 433-442, 2001.

[24] Chia-Chi Chen Wen-Liang and Shing-Chia Chen. ANFIS based PRML system for read-out RF signal. *IFSA World congress and 20^th^ NAFIPS International conference*, page 912-917, 2001.

[25] K. Polat, S.Gunes**, “**Automatic determination of diseases related to lymph system from lymphography data using principles component analysis (PCA), fuzzy weighting pre-processing and ANFIS,**”** *Expert Systems with Applications: An International Journa*l, vol. 33 , 2007, Pages: 636-641

[26] Shahina Begum, Jerker Westin, Peter Funk, Mark Dougherty, “Induction of an Adaptive Neuro-Fuzzy Inference System for Investigating Fluctuation in Parkinson’s Disease”, *In SAIS 2006*, pages 67-72

[27] Manish Kakar, Håkan Nyström, Lasse Rye Aarup, Trine Jakobi Nøttrup and Dag Rune Olsen, “Respiratory motion prediction by using the adaptive neuro fuzzy inference system (ANFIS)”, *Phys Med Biol.* 2005 Oct 7;50(19):4721-8

[28] J. Guralnik, T. Seeman, M. Tinetti, M. Nevitt, L. Berkman, “Validation and use of performance measures of functioning in a nondisabled older population,” *Aging Clin Exp Res,* vol. 6, 1994, pp. 410–419.

[29] J.Guralnik, L. Ferrucci, E. Simonsick, M. Salive, R. Wallace, “Lower-extremity function in persons over the age of 70 years as a predictor of subsequent disability,” *N Engl J Med.* 1995, 332:556–561.

[30] G. V. Ostir, S. Volpato, L. P. Fried, P. Chaves, J. M. Guralnik, “Reliability and sensitivity to change assessed for a summary measure of lower body function Results from the Women’s Health and Aging Study,” *Journal of Clinical Epidemiology* 55, 2002, 916–921

[31] Guralnik, J.M., Ferrucci, L., Pieper, C.F., Leveille, S.G., Markides, K.S., Ostir, G.V., Studenski, S., Berkman, L.F. & Wallace, R.B. (2000), “Lower extremity function and subsequent disability: Consistency across studies , predictive models and value of gait speed alone compared with the short physical performance battery”, *Journal of Gerontology,* 55(4), 221-231

[32] Reuben, D.B., & Siu, A.L. (1990), “ An objective measure of physical function of elderly outpatients: the physical performance test”, *Journal of the American Geriatrics Society*. 38: 1105-12.

[33] K. Burks, K. Keegan, “Objective Measurement of Stiffness in Knee Osteoarthritis(OA)”, *Journal of Orthopaedic Nursing*, in Press, 2006.

[34] J. Kennedy, R. Eberhart, Particle swarm optimization, P*roceeding of the IEEE International Conference on Neural Networks*, 1995, pp. 1942-1947

[35] L. Zadeh, "Soft Computing, Fuzzy Logic and Recognition Technology," *Proceedings, IEEE International Conference on Fuzzy Systems, Anchorage,* AK, May, 1998, pp. 1678-1679.

APPENDIX

A. Results of Case 1, ANFIS Learning

a. Balance Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the data: balancedatafor1cv2edition.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
Training Data Error Rate 0.050986
Test Data Error Rate 0.050983
Epochs 280

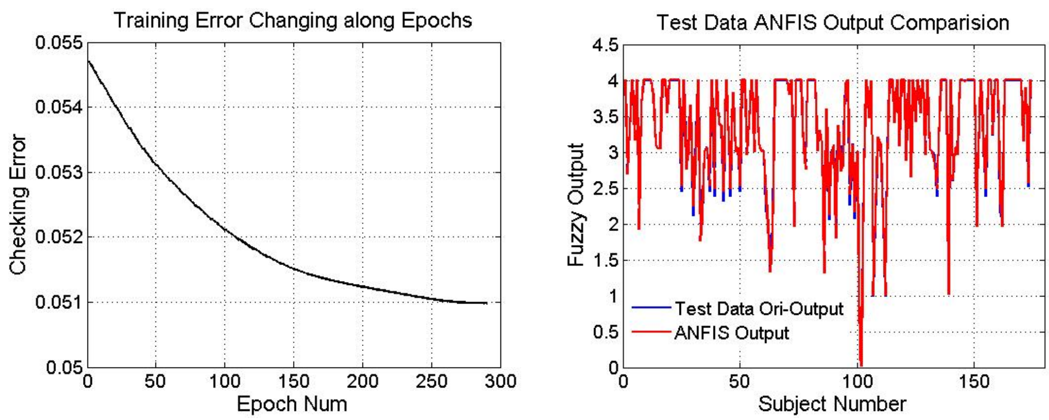


Figure A. 1 1-Fold CV results of Balance Test for Trn Err Criteria

b) 10 fold

Please input file name of the data: balancedatafor1cv2edition.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Training Data Error Rate	Test Data Error Rate	Epochs
Fold 1	1	9	others	0.002230	0.323718	1270
Fold 2	10	18	others	0.047266	0.076835	290
Fold 3	19	27	others	0.049991	0.062618	260
Fold 4	28	36	others	0.048283	0.074407	260
Fold 5	37	45	others	0.002262	0.123863	1260
Fold 6	46	54	others	Can not set up MFs, because all side-by-side stand time are 10		
Fold 7	55	63	others	0.002086	0.006397	1260
Fold 8	64	72	others	0.002226	0.030766	1320
Fold 9	73	81	others	0.051302	0.048732	280
Fold 10	82	87	others	0.052697	0.014441	280
Average				0.028705	0.084642	720

Table A. 1 10-Fold CV results of Balance Test for Trn Err Criteria

In 10-fold cross validation, we tracked training error changing along epochs and compared test data original output and ANFIS output. Figure A.2 is the figures for 1st. fold of 10 folds. In the following experiment results, we only list the results of one or two folds which are representative.

Fold 1 Results:

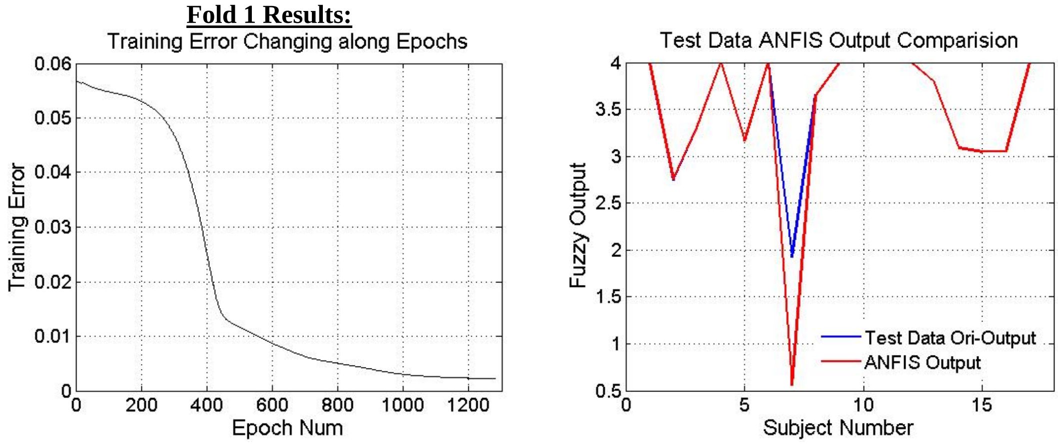


Figure A. 2 Fold 1 of 10-Fold CV results of Balance Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the data: balancedatafor1cv2edition.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.050307
Checking Data Error Rate 0.050302
Test Data Error Rate 0.050302
Epochs 290

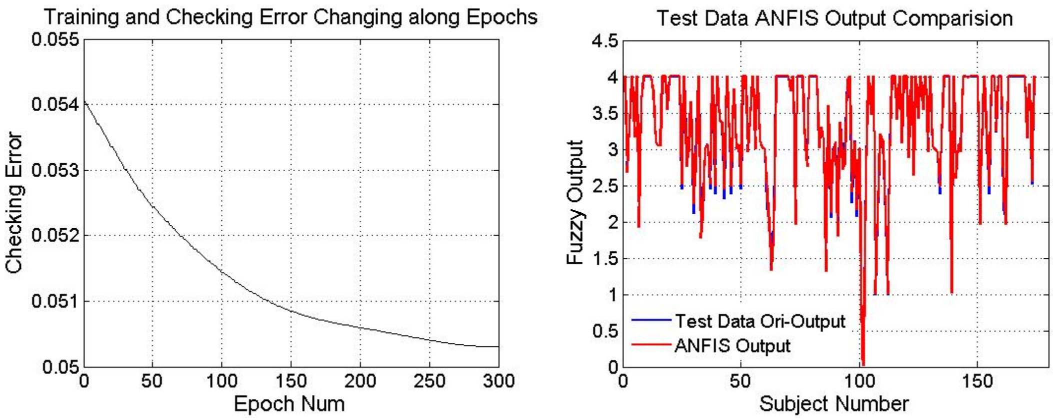


Figure A. 3 1-Fold CV results of Balance Test for Chk Err Criteria

b) 10 fold

Please input file name of the data:balancedatafor1cv2edition.txt

Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Trn	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.049048	0.067933	0.454570	260
Fold 2	10	18	19	36	others	0.040530	0.069858	0.080124	200
Fold 3	19	27	28	45	others	0.050789	0.358636	0.063065	80
Fold 4	28	36	37	54	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 5	37	45	46	63	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 6	46	54	55	72	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 7	55	63	64	81	others	0.054554	0.050878	0.048484	110
Fold 8	64	72	73	90	others	0.013659	0.010292	0.035153	450
Fold 9	73	81	82	99	others	0.058611	0.321626	0.049021	150
Fold 10	82	87	88	99	others	0.057244	0.393875	0.011654	140
Average						0.046347	0.181871	0.106010	198.571

Table A. 2 10-Fold CV results of Balance Test for Chk Err Criteria

Fold 1 Results:

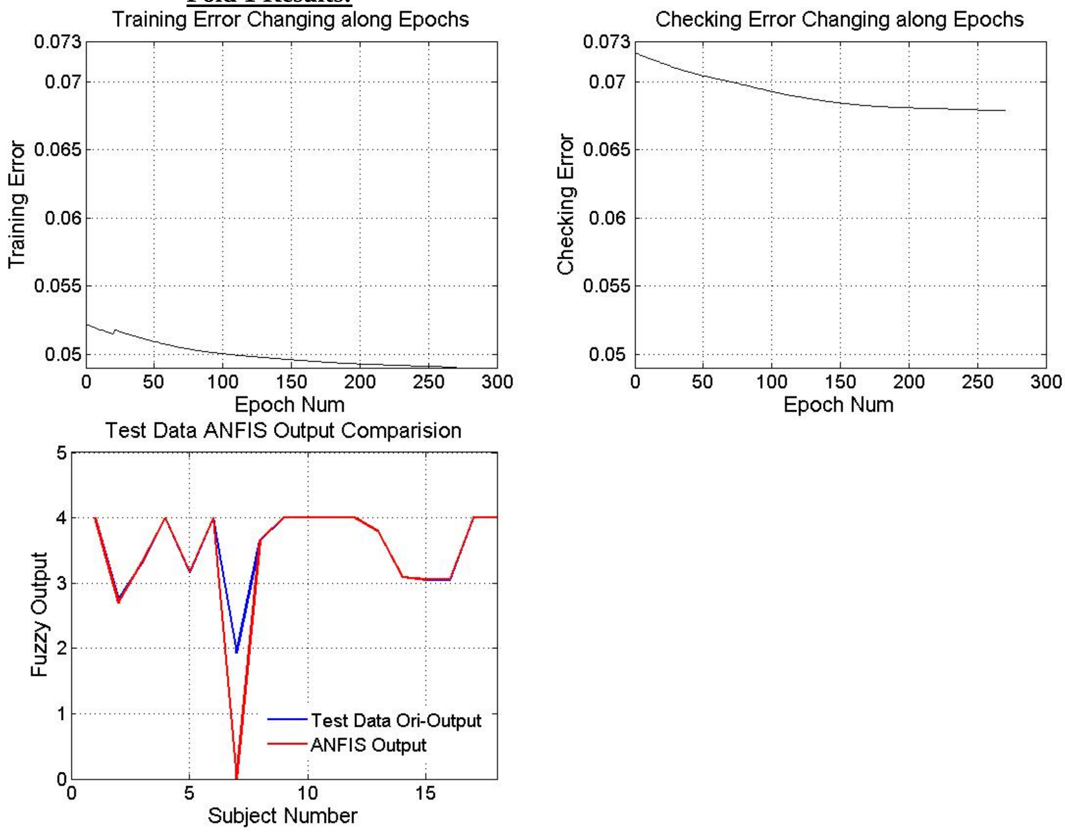


Figure A. 4 Fold 1 of 10-Fold CV results of Balance Test for Chk Err Criteria

b. Walking Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the data:50walkingdatafor1cv.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
Training Data Error Rate 0.009361
Test Data Error Rate 0.009363
Epochs 1440

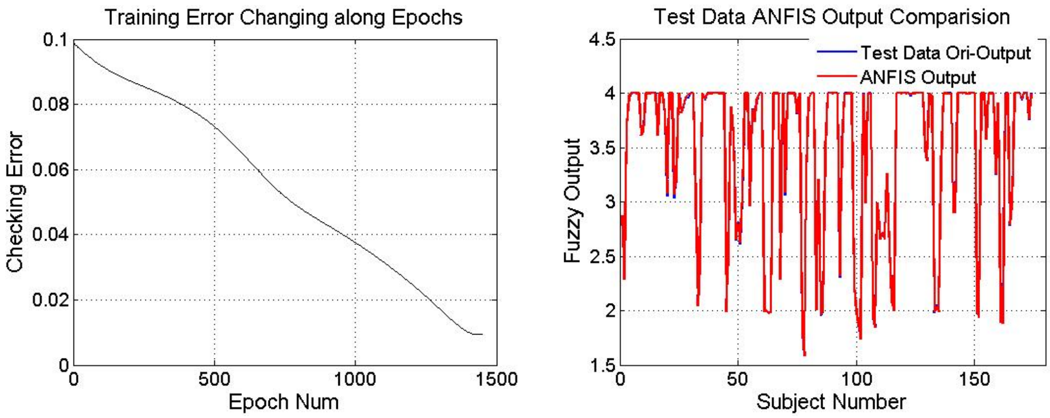


Figure A. 5 1-Fold CV results of Walking Test for Trn Err Criteria

b) 10 fold

Please input file name of the data:50walkingdatafor1cv.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
Training Data Error Rate 0.009361
Test Data Error Rate 0.009363
Epochs 1440
Please input file name of the data:50walkingdatafor1cv.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Training Data Error Rate	Test Data Error Rate	Epochs
Fold 1	1	9	others	0.009220	0.011516	1400
Fold 2	10	18	others	0.008526	0.016242	1640
Fold 3	19	27	others	0.009030	0.011753	1300
Fold 4	28	36	others	0.009555	0.008137	1620
Fold 5	37	45	others	0.038730	0.036609	1430
Fold 6	46	54	others	0.039415	0.473349	1190
Fold 7	55	63	others	0.009467	0.008421	1310
Fold 8	64	72	others	0.009395	0.009470	1330
Fold 9	73	81	others	0.040411	0.097426	1590
Fold 10	82	87	others	0.009330	0.009104	1010
Average				0.0183079	0.0682027	1382

Table A. 3 10-Fold CV results of Walking Test for Trn Err Criteria

Fold 1 Results:

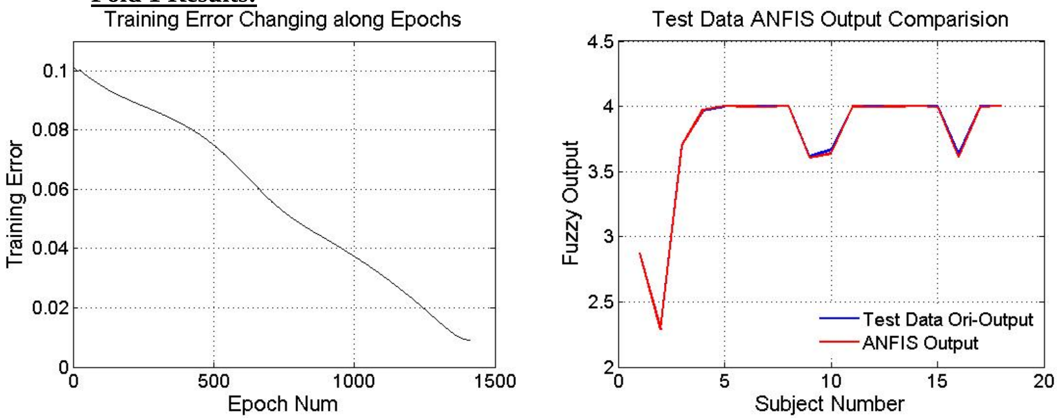


Figure A. 6 Fold 1 of 10-Fold CV results of Walking Test for Trn Err Criteria

Fold 9 Results:

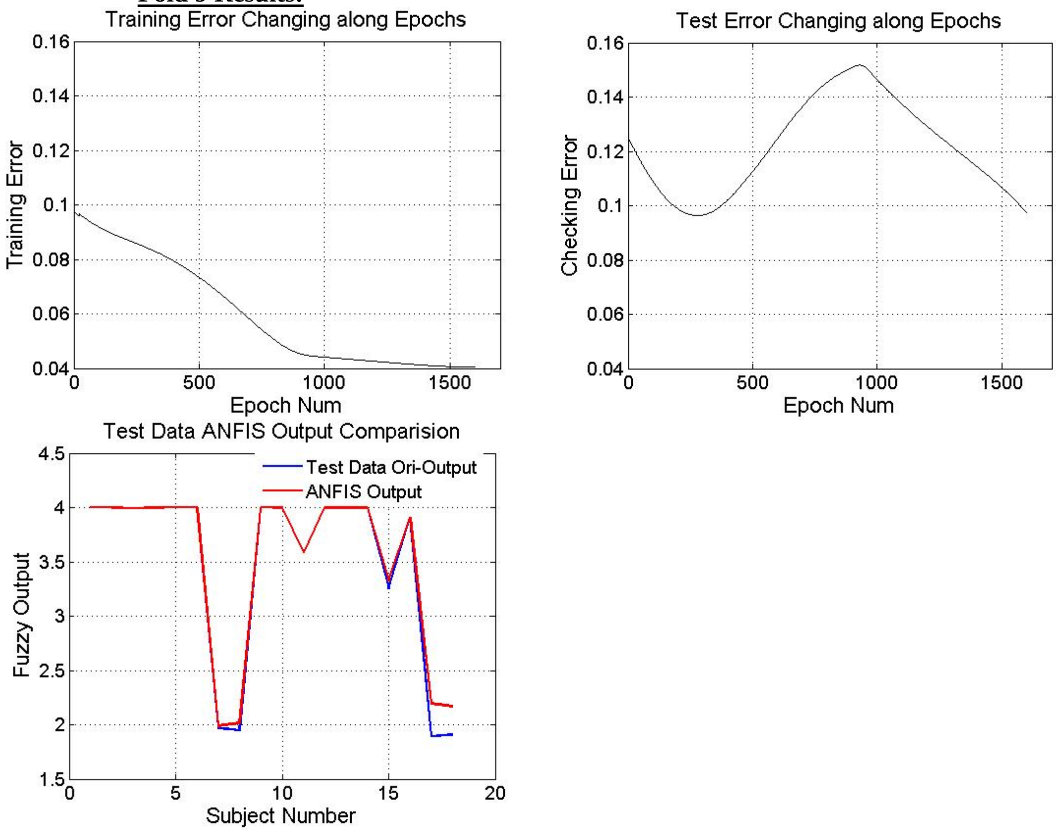


Figure A. 7 Fold 9 of 10-Fold CV results of Walking Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the data:50walkingdatafor1cv.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87

checkStart 1 checkEnd 87
Training Data Error Rate 0.009361
Checking Data Error Rate 0.009363
Test Data Error Rate 0.009354
Epochs 1440

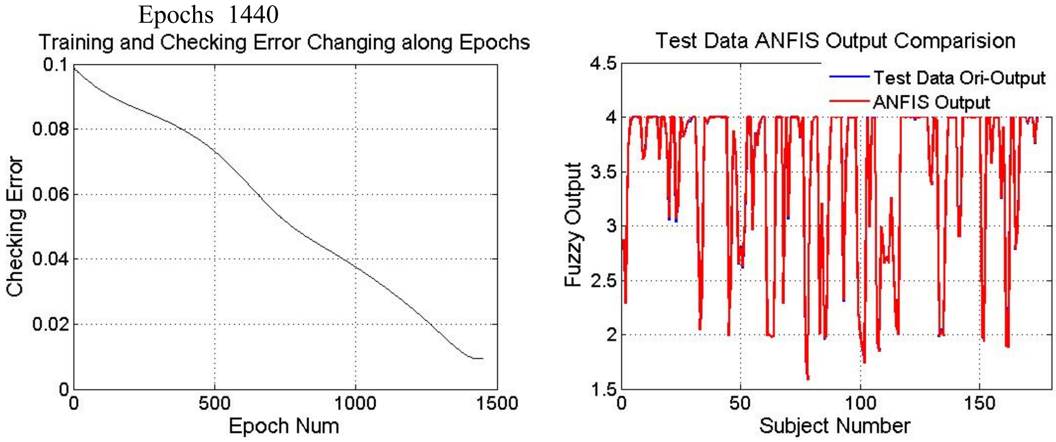


Figure A. 8 1-Fold CV results of Walking Test for Chk Err Criteria

b) 10 fold
Please input file name of the data:50walkingdatafor1cv.txt
Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Trn	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.007501	0.014428	0.010946	1330
Fold 2	10	18	19	36	others	0.007625	0.011680	0.017937	1480
Fold 3	19	27	28	45	others	0.038629	0.034116	0.043542	1540
Fold 4	28	36	37	54	others	0.005512	0.018550	0.005144	1120
Fold 5	37	45	46	63	others	0.042634	0.032365	0.307092	660
Fold 6	46	54	55	72	others	0.035477	0.053321	0.704730	1130
Fold 7	55	63	64	81	others	0.009653	0.010660	0.008966	1360
Fold 8	64	72	73	90	others	0.009781	0.009255	0.010284	990
Fold 9	73	81	82	99	others	0.008952	0.011556	0.008707	1060
Fold 10	82	87	88	99	others	0.008693	0.013086	0.006234	1000
Average						0.017446	0.020901	0.112358	1167

Table A. 4 10-Fold CV results of Walking Test for Chk Err Criteria

Fold 1 Results:

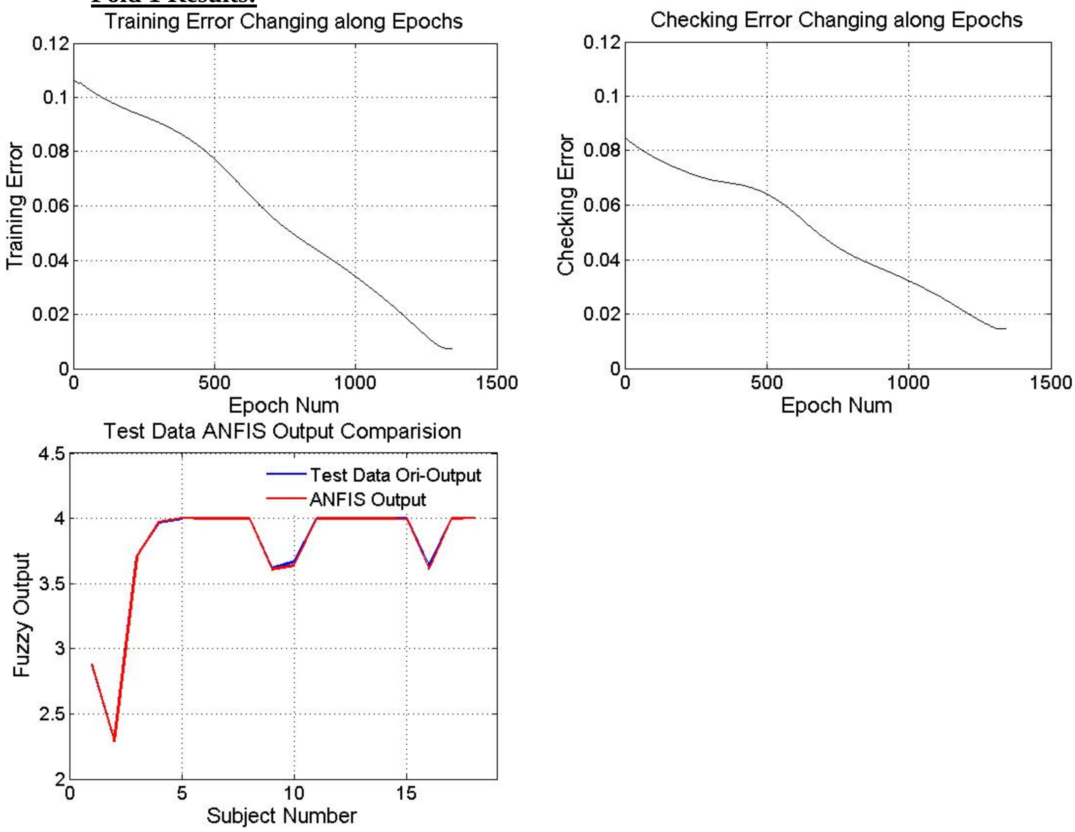


Figure A. 9 Fold 1 of 10-Fold CV results of Walking Test for Chk Err Criteria

c. Chair Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the data: 5chairdatafor1cv.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
Training Data Error Rate 0.048534
Test Data Error Rate 0.048525
Epochs 1830

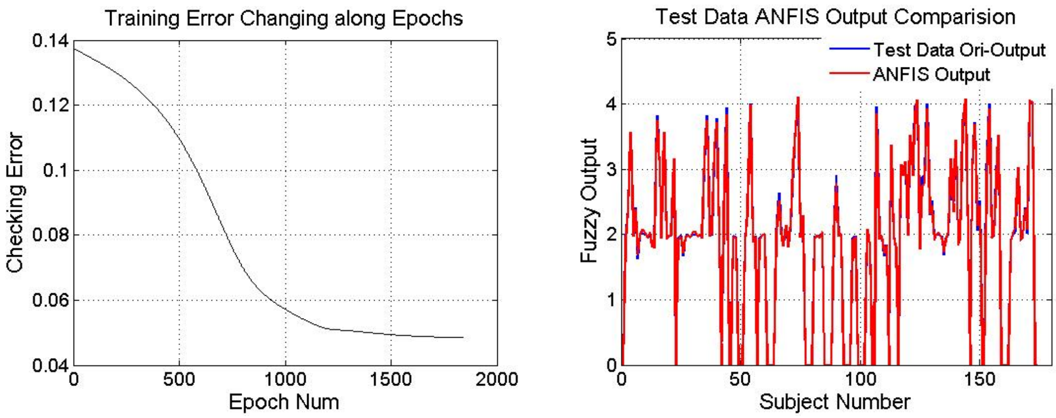


Figure A. 10 1-Fold CV results of Chairg Test for Trn Err Criteria

b) 10 fold

Please input file name of the data:5chairdatafor1cv.txt

Num of fold cross validation:10

	Test Start	Test End	Train	Training Data Error Rate	Test Data Error Rate	Epochs
Fold 1	1	9	others	0.049646	0.037667	1840
Fold 2	10	18	others	0.050387	0.058442	1230
Fold 3	19	27	others	0.048473	0.049466	1890
Fold 4	28	36	others	0.049376	0.040545	1840
Fold 5	37	45	others	0.049137	0.046878	1760
Fold 6	46	54	others	0.050095	0.032746	1840
Fold 7	55	63	others	0.045734	0.068856	1840
Fold 8	64	72	others	0.046409	0.076215	1850
Fold 9	73	81	others	0.047591	0.058897	1750
Fold 10	82	87	others	0.049163	0.042190	1790
Average				0.0436365	0.0474235	1579

Table A. 5 10-Fold CV results of Chair Test for Trn Err Criteria

<u>Fold 1 Results:</u>

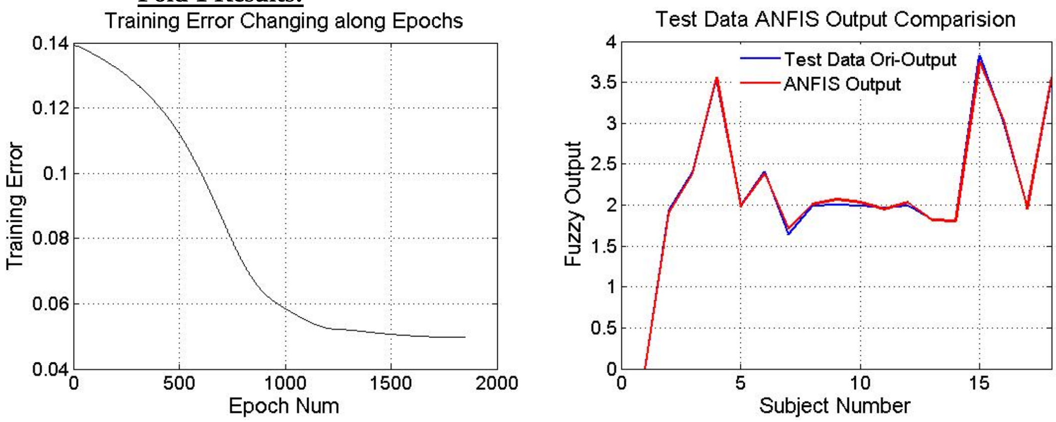


Figure A. 11 Fold 1 of 10-Fold CV results of Walking Test for Trn Err Criteria

Fold 5 Results:

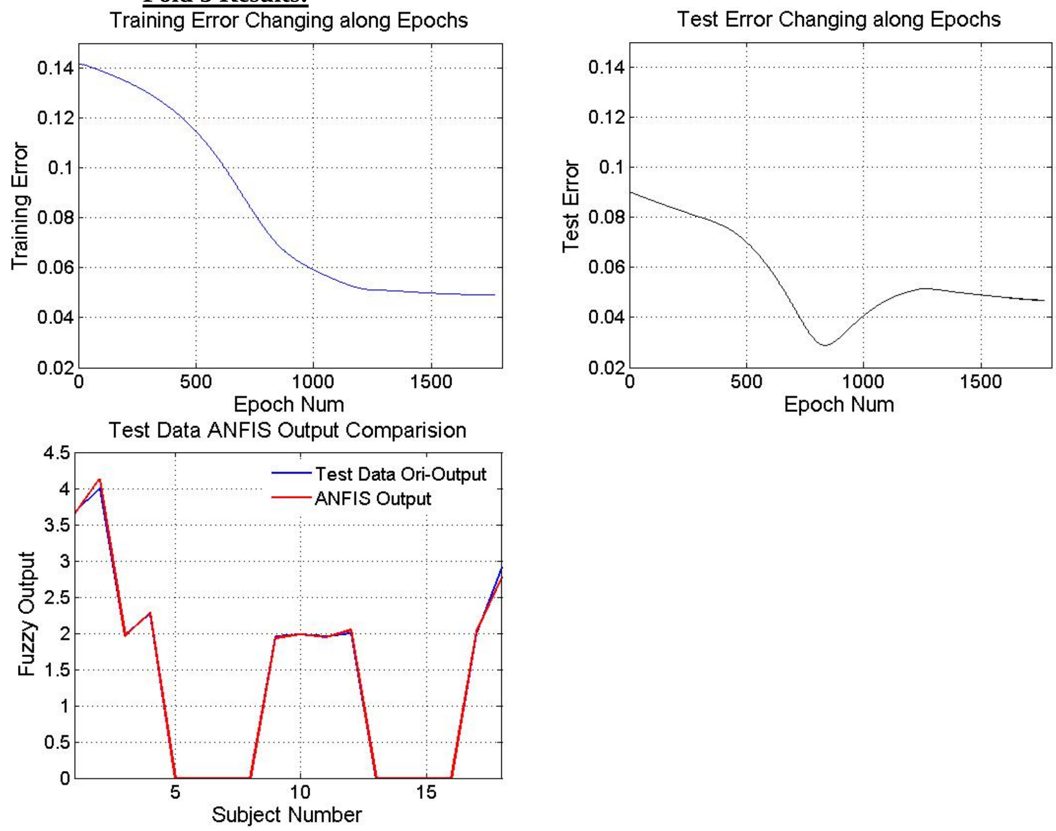


Figure A. 12 Fold 5 of 10-Fold CV results of Balance Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the data:5chairdatafor1cv.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.048534
Checking Data Error Rate 0.048525
Test Data Error Rate 0.048525
Epochs 1830

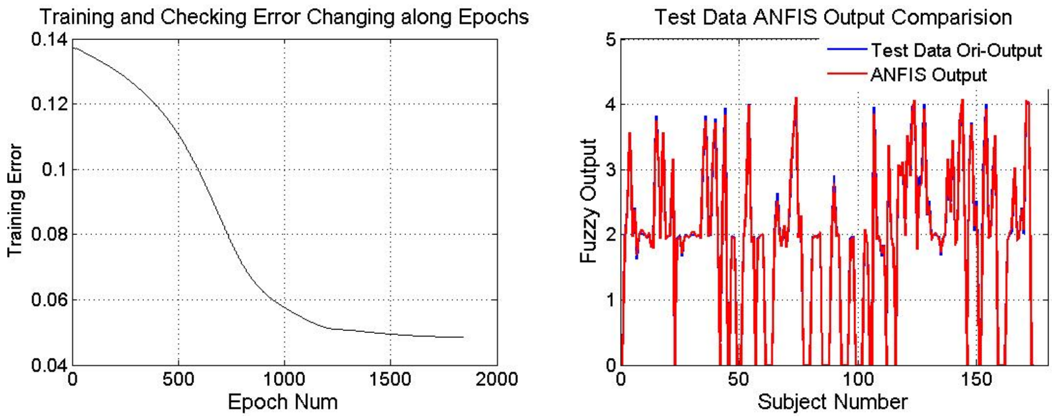


Figure A. 13 1-Fold CV results of Balance Test for Chk Err Criteria

b) 10 fold

Please input file name of the data:5chairdatafor1cv.txt

Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Trn	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.051240	0.060668	0.042110	1170
Fold 2	10	18	19	36	others	0.014277	0.013011	0.017825	1880
Fold 3	19	27	28	45	others	0.050454	0.041951	0.047752	2140
Fold 4	28	36	37	54	others	0.052206	0.039060	0.041054	2190
Fold 5	37	45	46	63	others	0.048175	0.052455	0.046129	1770
Fold 6	46	54	55	72	others	0.125645	0.205914	0.087333	60
Fold 7	55	63	64	81	others	0.122153	0.205568	0.096705	90
Fold 8	64	72	73	90	others	0.077031	0.207938	0.355124	670
Fold 9	73	81	82	99	others	0.052552	0.040386	0.058913	1250
Fold 10	82	87	88	99	others	0.051355	0.041029	0.035311	1530
Average						0.059384	0.084731	0.078614	1158

Table A. 6 10-Fold CV results of Balance Test for Chk Err Criteria

Fold 1 Results:

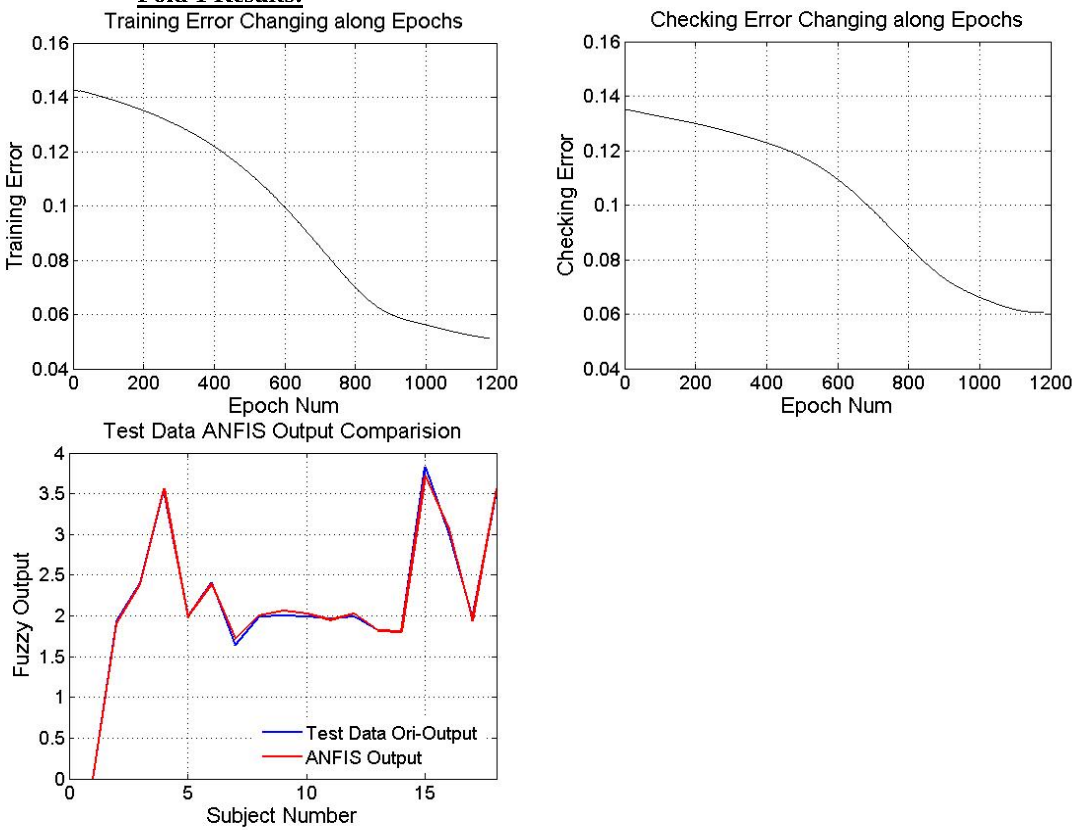


Figure A. 14 Fold 1 of 10-Fold CV results of Balance Test for Chk Err Criteria

B. Results of Case 2, ANFIS Learning

a. Balance Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the training data:balancedata1.txt
Please input file name of the test data:balancedata2.txt
Num of fold cross validation:1
trainStart at trainData 1 trainEnd at trainData 87
testStart at testData 1 testEnd at testData 87
Training Data Error Rate 0.057824
Test Data Error Rate 0.042174
Epochs 620

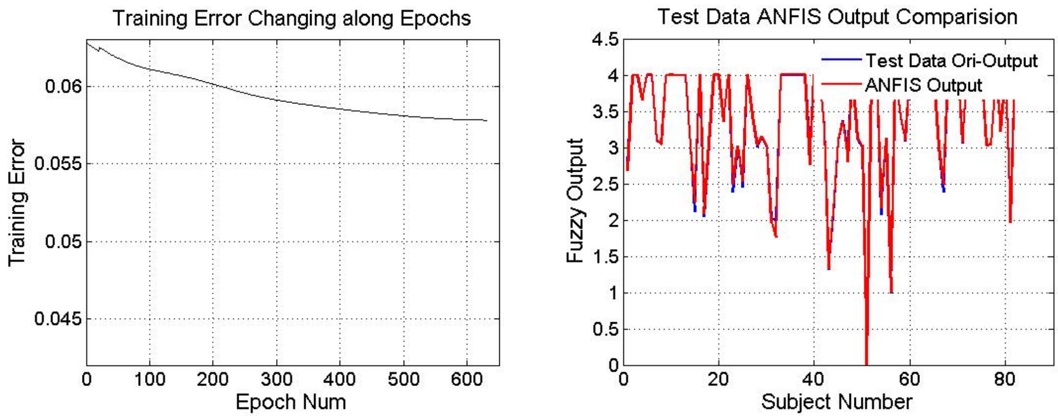


Figure B. 1 1-Fold CV results of Balance Test for Trn Err Criteria

b) 10 fold

Please input file name of the training data:balancedata1.txt
Please input file name of the test data:balancedata2.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Training Data	Test Data	Epochs
Fold 1	1	9	others	0.061086	0.091722	590
Fold 2	10	18	others	0.049757	0.033447	650
Fold 3	19	27	others	0.056703	0.050846	650
Fold 4	28	36	others	0.059235	0.076564	260
Fold 5	37	45	others	0.060928	0.043475	610
Fold 6	46	54	others	Can not set up MFs,because all side-by-side		
Fold 7	55	63	others	0.057751	0.006862	590
Fold 8	64	72	others	0.060791	0.035196	590
Fold 9	73	81	others	0.056734	0.014989	650
Fold 10	82	87	others	0.059673	0.003044	630
Average				0.058073	0.039572	580

Table B. 1 10-Fold CV results of Balance Test for Trn Err Criteria

Fold 1 Results:

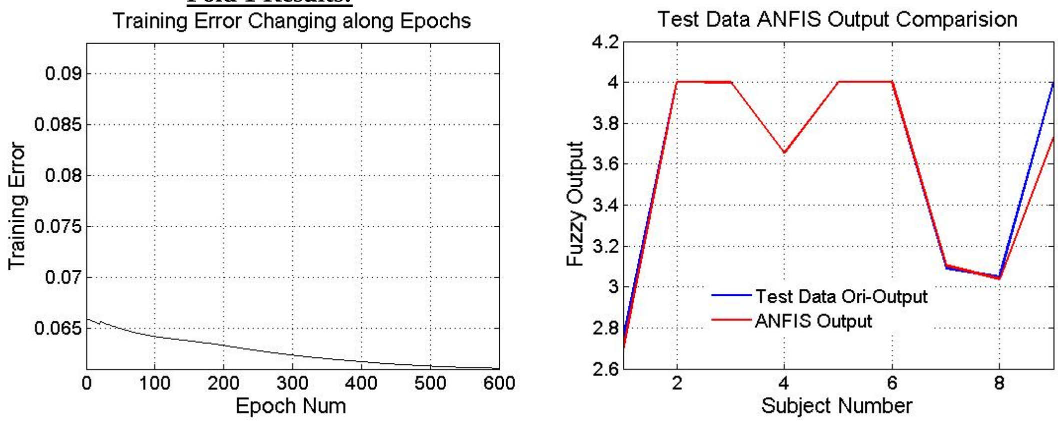


Figure B. 2 Fold 1 of 10-Fold CV results of Balance Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the training data:balancedata1.txt
Please input file name of the test & checking data:balancedata2.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.061826
Checking Data Error Rate 0.048348
Test Data Error Rate 0.048348
Epochs 40

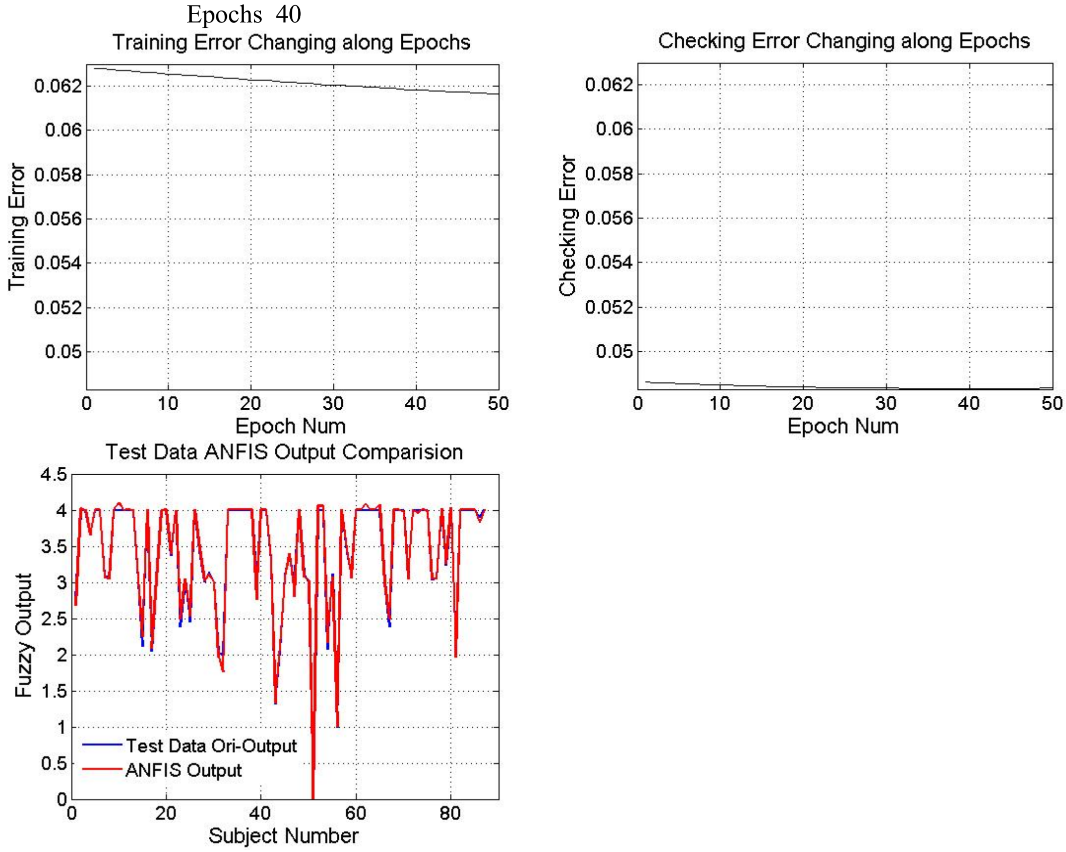


Figure B. 3 1-Fold CV results of Balance Test for Chk Err Criteria

b) 10 fold

Please input file name of the training data:balancedata1.txt
Please input file name of the test & checking data:balancedata2.txt
Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Trn	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.052620	0.069844	0.091512	660
Fold 2	10	18	19	36	others	0.049977	0.072866	0.060217	300
Fold 3	19	27	28	45	others	0.062674	0.305336	0.058770	80
Fold 4	28	36	37	54	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 5	37	45	46	63	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 6	46	54	55	72	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 7	55	63	64	81	others	0.061996	0.077570	0.015746	200
Fold 8	64	72	73	90	others	0.063954	0.014004	0.039760	1100
Fold 9	73	81	82	99	others	0.064806	0.079647	0.114888	390
Fold 10	82	87	88	99	others	0.064858	0.079205	0.003916	490
Average						0.060126	0.099781	0.054972	460

Table B. 2 10-Fold CV results of Balance Test for Chk Err Criteria

<u>Fold 1 Results:</u>

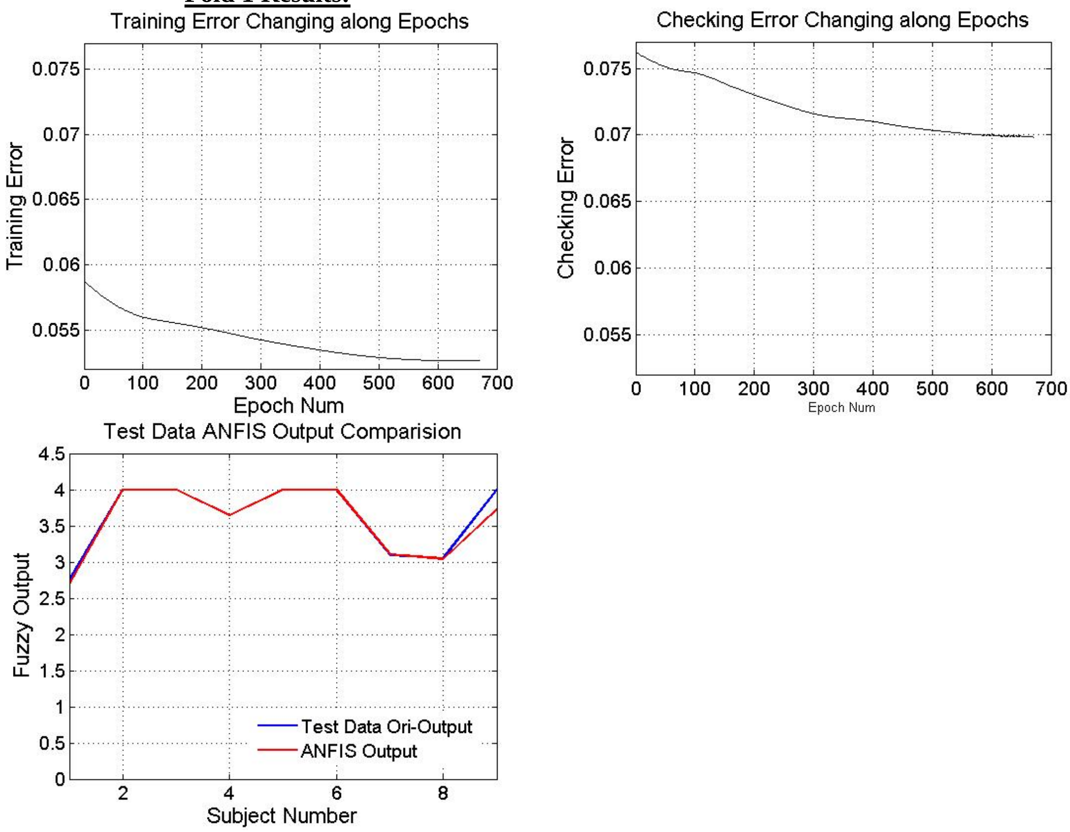


Figure B. 4 Fold 1 of 10-Fold CV results of Balance Test for Chk Err Criteria

b. Walking Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the training data:50walkingdata1.txt
Please input file name of the test data:50walkingdata2.txt
Num of fold cross validation:1
trainStart at trainData 1 trainEnd at trainData 87
testStart at testData 1 testEnd at testData 87
Training Data Error Rate 0.001431
Test Data Error Rate 0.014260
Epochs 2790

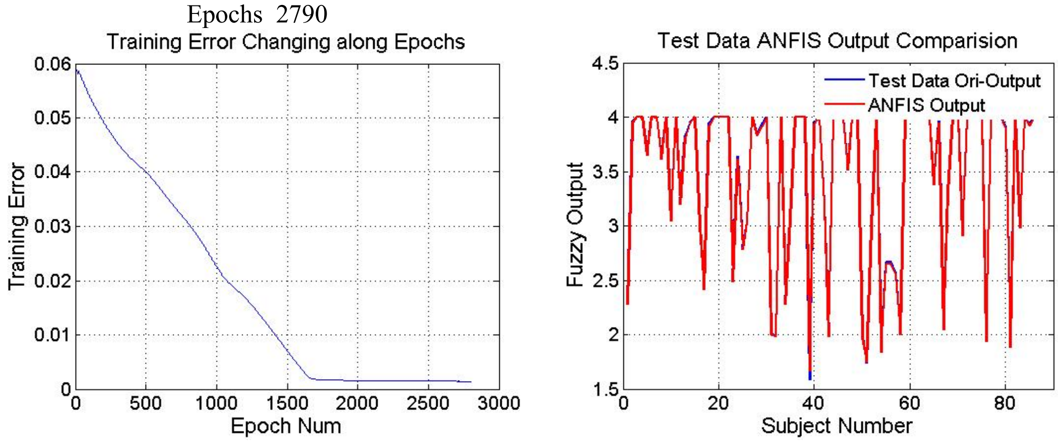


Figure B. 5 1-Fold CV results of Walking Test for Trn Err Criteria

b) 10 fold

Please input file name of the training data:50walkingdata1.txt
Please input file name of the test data:50walkingdata2.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Training Data Error Rate	Test Data Error Rate	Epochs
Fold 1	1	9	others	0.001439	0.012637	3050
Fold 2	10	18	others	0.001203	0.012598	4260
Fold 3	19	27	others	0.001345	0.011449	3510
Fold 4	28	36	others	0.001395	0.010795	2670
Fold 5	37	45	others	0.001042	0.026926	1240
Fold 6	46	54	others	0.001088	0.042192	2240
Fold 7	55	63	others	0.001452	0.011912	3010
Fold 8	64	72	others	0.016939	0.028003	1760
Fold 9	73	81	others	0.001227	0.008194	2810
Fold 10	82	87	others	0.001461	0.009220	3070
Average				0.002859	0.017393	2762

Table B. 3 10-Fold CV results of Walking Test for Trn Err Criteria

Fold 1 Results:

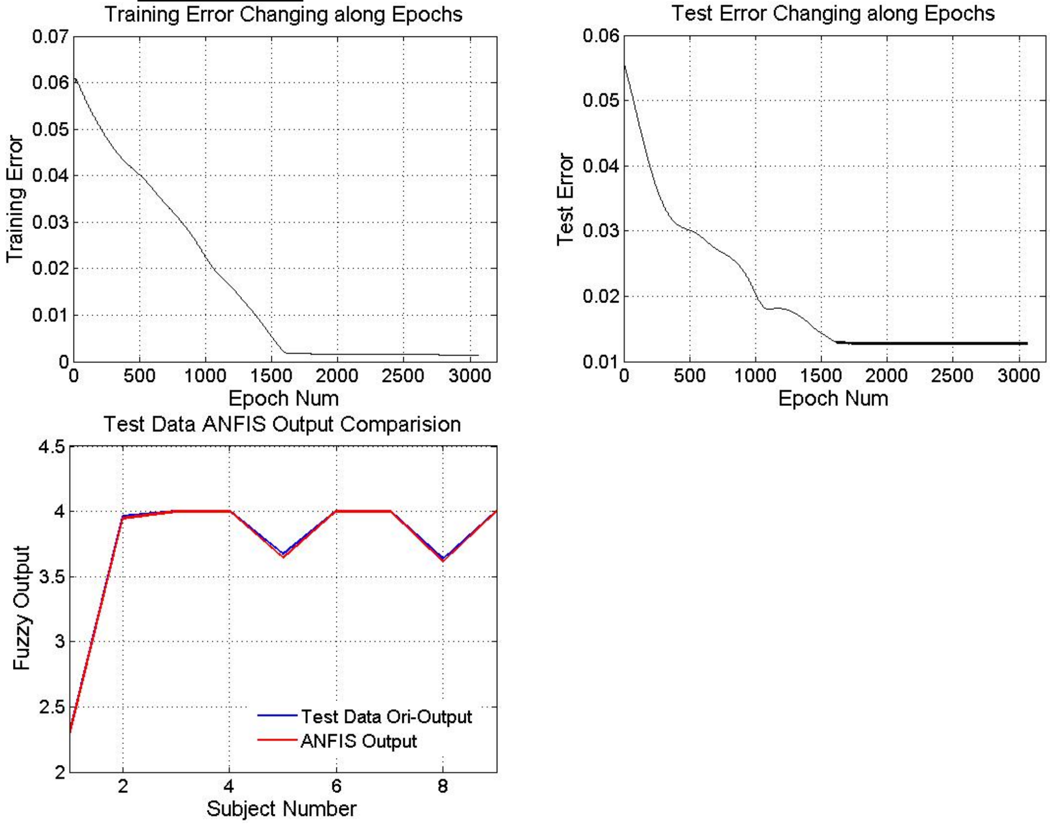


Figure B. 6 Fold 1 of 10-Fold CV results of Walking Test for Trn Err Criteria

Fold 5 Results:

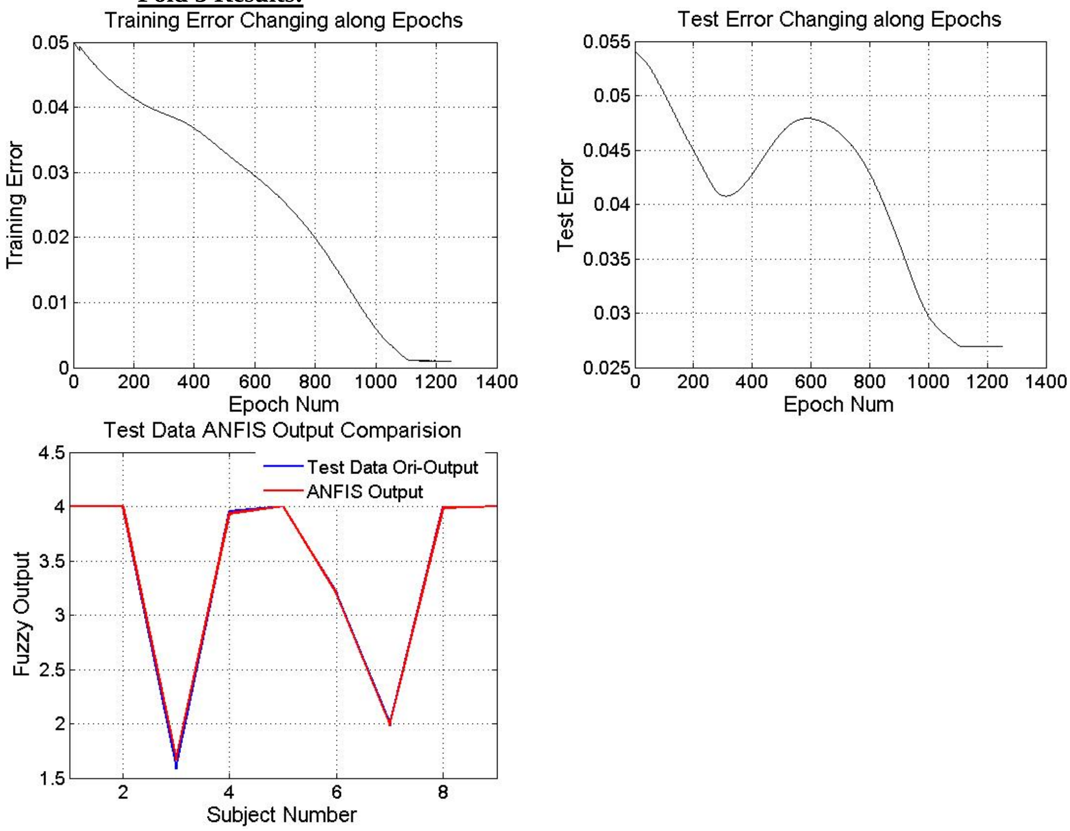


Figure B. 7 Fold 5 of 10-Fold CV results of Walking Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the training data:50walkingdata1.txt
Please input file name of the test & checking data:50walkingdata2.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.001250
Checking Data Error Rate 0.011456
Test Data Error Rate 0.011456
Epochs 860

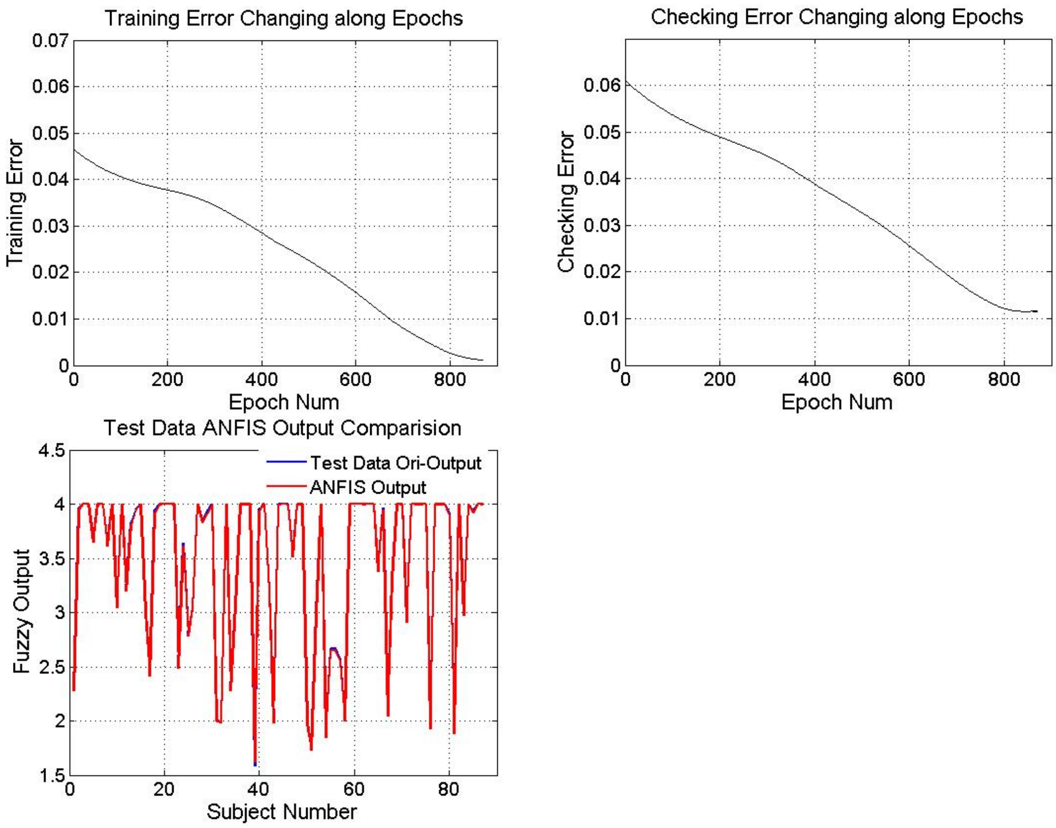


Figure B. 8 1-Fold CV results of Walking Test for Chk Err Criteria

b) 10 fold

Please input file name of the training data:50walkingdata1.txt

Please input file name of the test & checking data:50walkingdata2.txt

Num of fold cross validation:10

	Test Start	Test End	ChK Start	Chk End	Trn	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.000776	0.011829	0.012717	1560
Fold 2	10	18	19	36	others	0.000902	0.015593	0.016600	1000
Fold 3	19	27	28	45	others	0.000640	0.063680	0.011281	1620
Fold 4	28	36	37	54	others	0.053070	0.334781	0.061930	390
Fold 5	37	45	46	63	others	0.048286	0.179360	0.429657	320
Fold 6	46	54	55	72	others	0.000693	0.010308	0.096433	990
Fold 7	55	63	64	81	others	0.000755	0.009066	0.012683	960
Fold 8	64	72	73	90	others	0.000823	0.009512	0.006349	700
Fold 9	73	81	82	99	others	0.000630	0.011131	0.008883	1300
Fold 10	82	87	88	99	others	0.000779	0.011473	0.007766	1470
Average						0.010735	0.065673	0.06643	1031

Table B. 4 10-Fold CV results of Walking Test for Chk Err Criteria

Fold 1 Results:

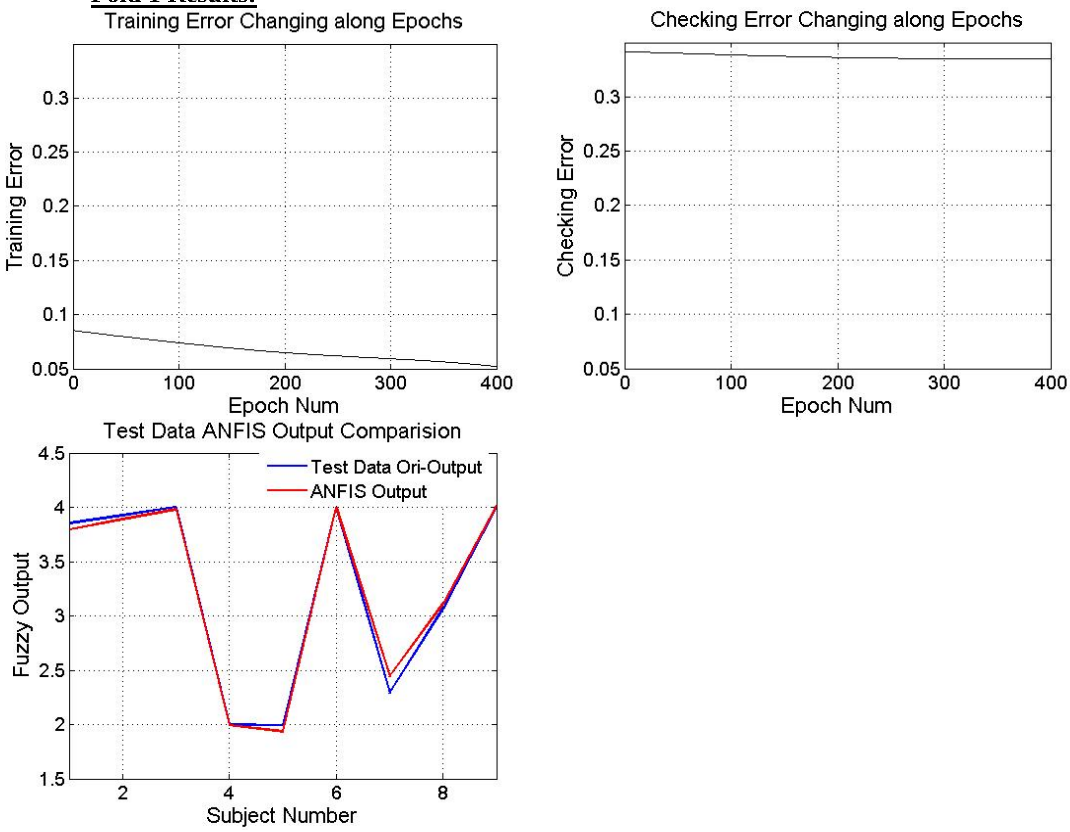


Figure B. 9 Fold 1 of 10-Fold CV results of Balance Test for Chk Err Criteria

c. Chair Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the training data:chairdata1.txt
Please input file name of the test data:chairdata2.txt
Num of fold cross validation:1
trainStart at trainData 1 trainEnd at trainData 87
testStart at testData 1 testEnd at testData 87
Training Data Error Rate 0.050582
Test Data Error Rate 0.242481
Epochs 3840

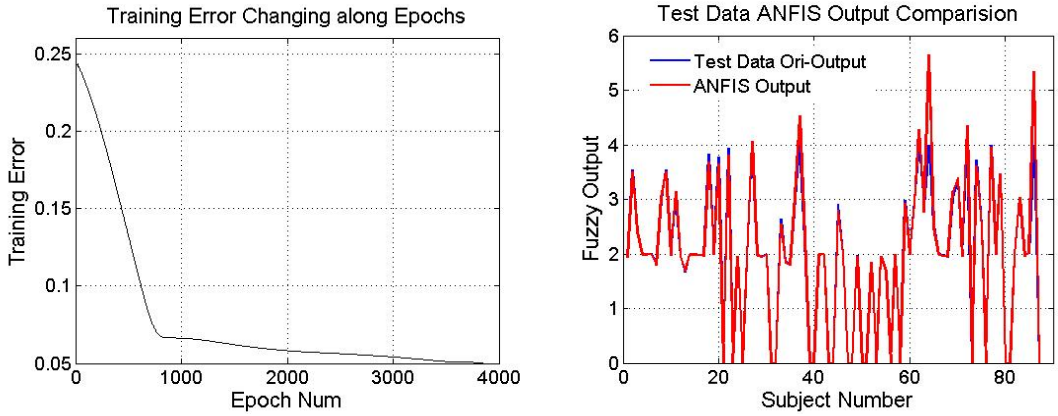


Figure B. 10 1-Fold CV results of Chair Test for Trn Err Criteria

b) 10 fold

Please input file name of the training data:chairdata1.txt
Please input file name of the test data:chairdata2.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Training Data Error Rate	Test Data Error Rate	Epochs
Fold 1	1	9	others	0.063726	0.029996	880
Fold 2	10	18	others	0.051774	0.054149	3950
Fold 3	19	27	others	0.051645	0.061552	3850
Fold 4	28	36	others	0.053274	0.032230	3950
Fold 5	37	45	others	0.052534	0.179156	3830
Fold 6	46	54	others	0.051314	0.003936	3810
Fold 7	55	63	others	0.045958	0.074524	4160
Fold 8	64	72	others	0.064316	0.569815	870
Fold 9	73	81	others	0.050745	0.045151	3960
Fold 10	82	87	others	0.054029	0.656617	1220
Average				0.0539315	0.1707126	3048

Table B. 5 10-Fold CV results of Chair Test for Trn Err Criteria

Fold 1 Results:

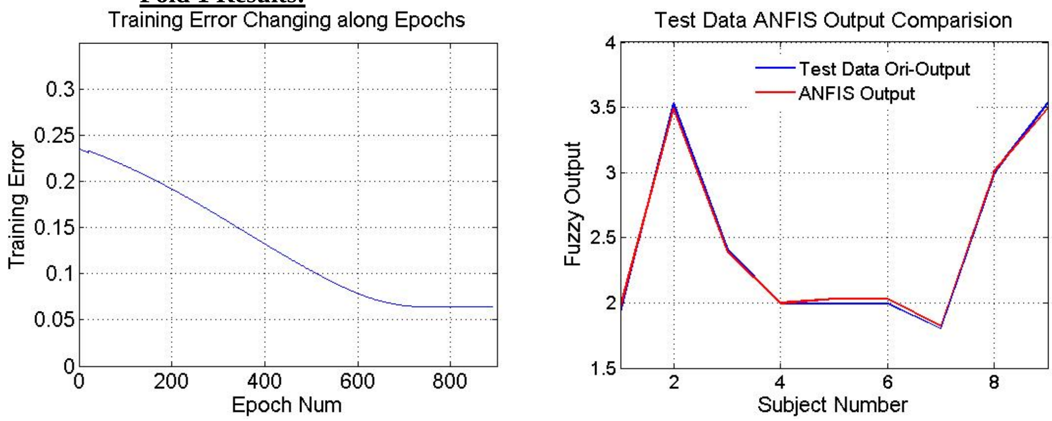


Figure B. 11 Fold 1 of 10-Fold CV results of Chair Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the training data:chairdata1.txt
Please input file name of the test & checking data:chairdata2.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.164851
Checking Data Error Rate 0.185503
Test Data Error Rate 0.185503
Epochs 30

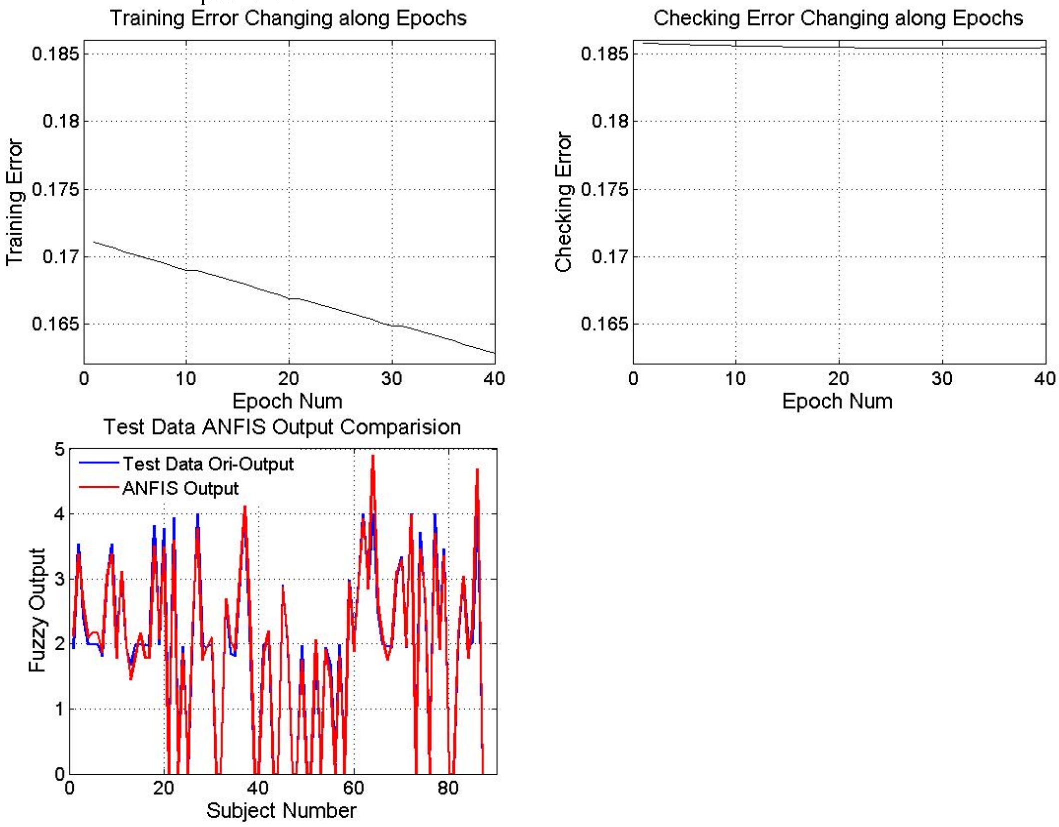


Figure B. 12 1-Fold CV results of Chair Test for Chk Err Criteria

b) 10 fold

Please input file name of the training data:chairdata1.txt
Please input file name of the test & checking data:chairdata2.txt
Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Trn	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.054257	0.060222	0.016914	1870
Fold 2	10	18	19	36	others	0.056533	0.048076	0.055502	1620
Fold 3	19	27	28	45	others	0.132116	0.119339	0.138412	220
Fold 4	28	36	37	54	others	0.076162	0.116438	0.055396	730
Fold 5	37	45	46	63	others	0.053463	0.062548	0.160445	1520
Fold 6	46	54	55	72	others	0.043062	0.307920	0.009526	2570
Fold 7	55	63	64	81	others	0.068493	0.365615	0.092249	700
Fold 8	64	72	73	90	others	0.019105	0.619002	1.070760	4420
Fold 9	73	81	82	99	others	0.037942	0.436496	0.056081	2700
Fold 10	82	87	88	99	others	0.046563	0.035617	0.507172	1450
Average						0.0587696	0.217127	0.216245	1780

Table B. 6 10-Fold CV results of Chair Test for Chk Err Criteria

Fold 1 Results:

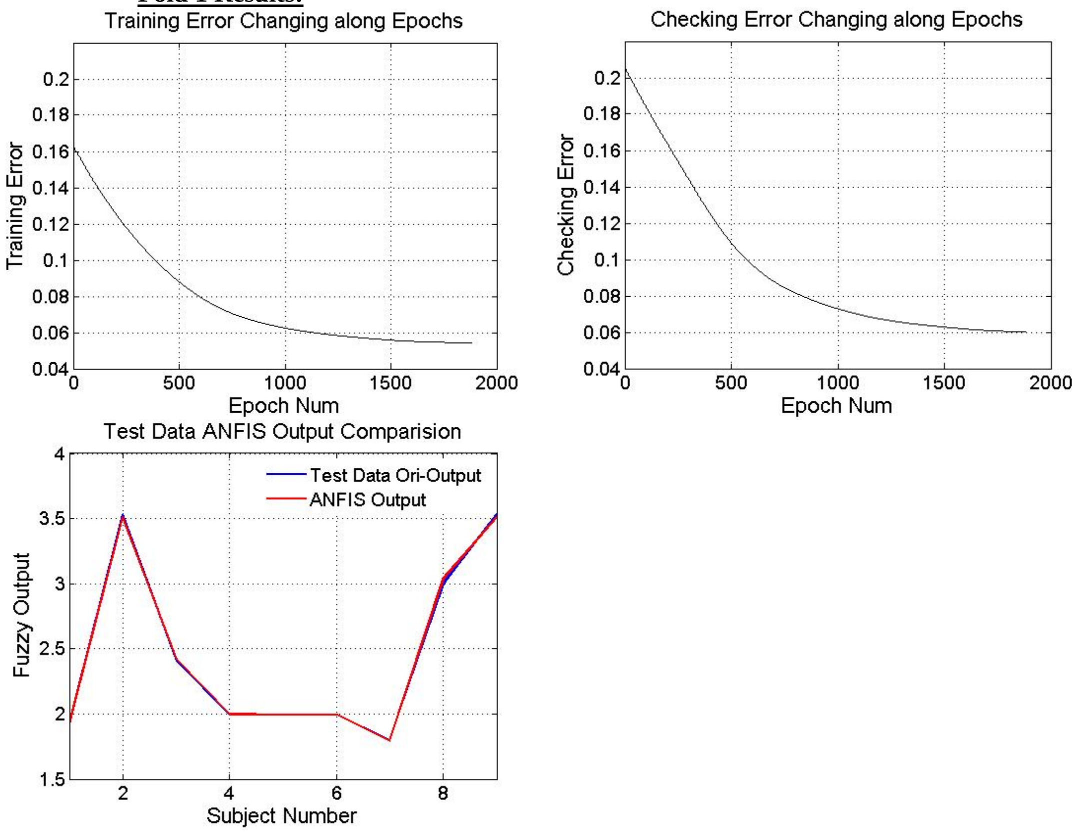


Figure B. 13 Fold 1 of 10-Fold CV results of Chair Test for Chk Err Criteria

C. Results of Case3, ANFIS Learning

a.   Balance Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the training data:balancedata2.txt
Please input file name of the test data:balancedata1.txt
Num of fold cross validation:1
trainStart at trainData 1 trainEnd at trainData 87
testStart at testData 1 testEnd at testData 87
Training Data Error Rate 0.033437
Test Data Error Rate 0.214215
Epochs 1940

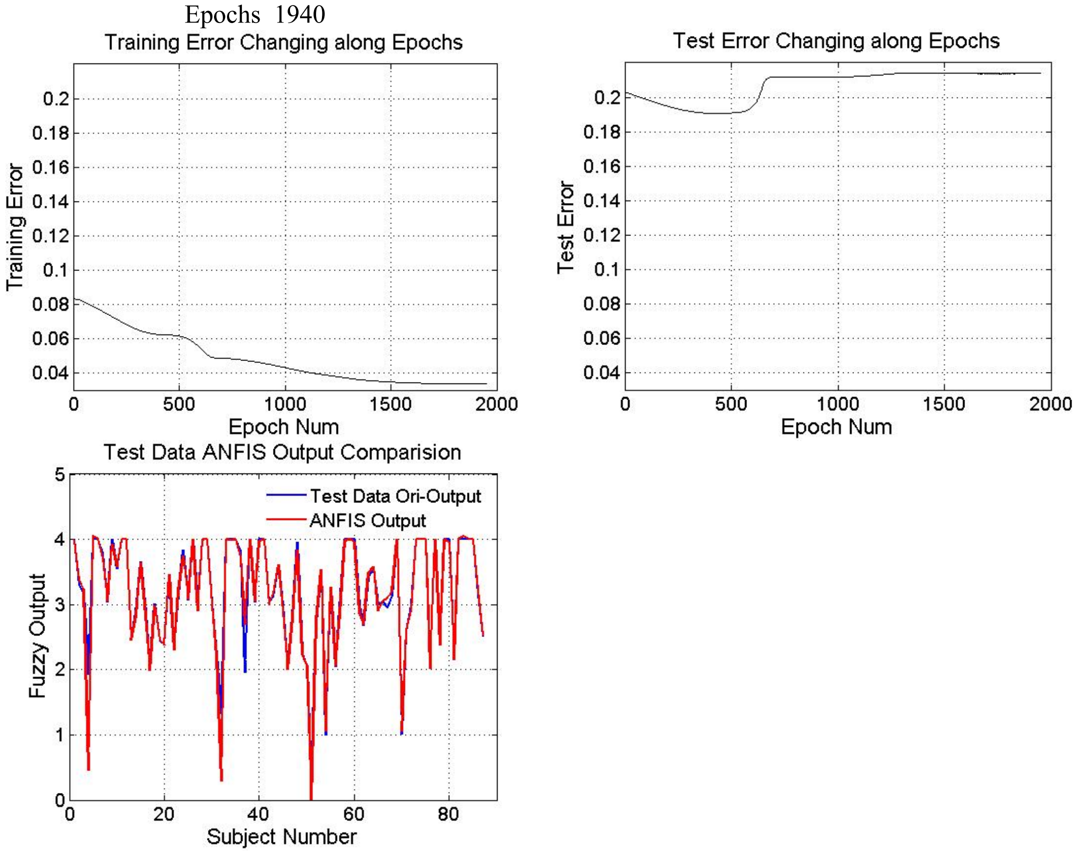


Figure C. 1 1-Fold CV results of Balance Test for Trn Err Criteria

b) 10 fold

Please input file name of the training data:balancedata1.txt
Please input file name of the test data:
Num of fold cross validation:
Please input file name of the training data:balancedata2.txt
Please input file name of the test data:balancedata1.txt
Num of fold cross validation:10

	Test	Test	Train	Trn Data	Test Data	Epochs
Fold 1	1	9	others	0.034754	0.492764	1900
Fold 2	10	18	others	0.020068	0.017125	1940
Fold 3	19	27	others	0.041154	0.071923	2110
Fold 4	28	36	others	0.029494	0.040696	2470
Fold 5	37	45	others	0.033416	0.249898	2050
Fold 6	46	54	others	Can not set up MFs,because all		
Fold 7	55	63	others	0.033776	0.057064	2000
Fold 8	64	72	others	0.057280	0.042455	400
Fold 9	73	81	others	0.031632	0.035959	1570
Fold 10	82	87	others	0.009655	0.016153	3850
Average				0.032359	0.113782	2032.2

Table C. 1 10-Fold CV results of Balance Test for Trn Err Criteria

<u>Fold 1 Results:</u>

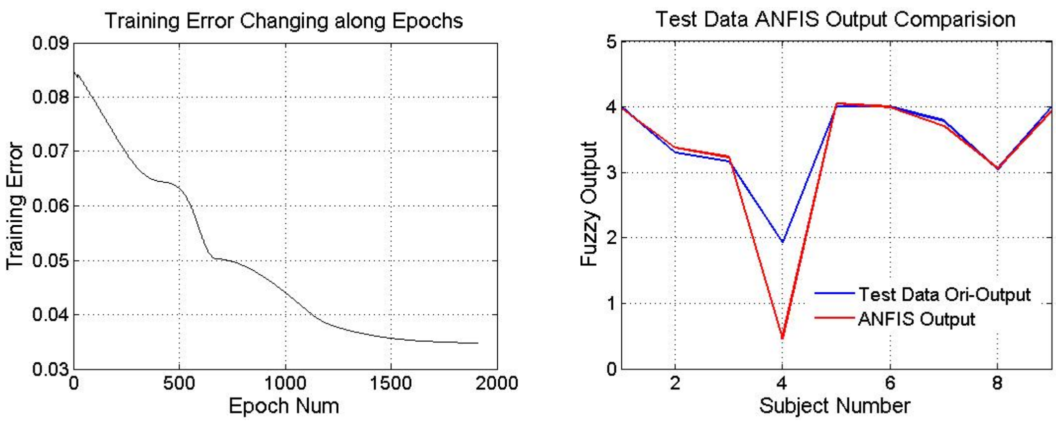


Figure C. 2 Fold 1 of 10-Fold CV results of Balance Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the training data:balancedata2.txt
Please input file name of the test & checking data:balancedata1.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.042062
Checking Data Error Rate 0.249730
Test Data Error Rate 0.249730
Epochs 270

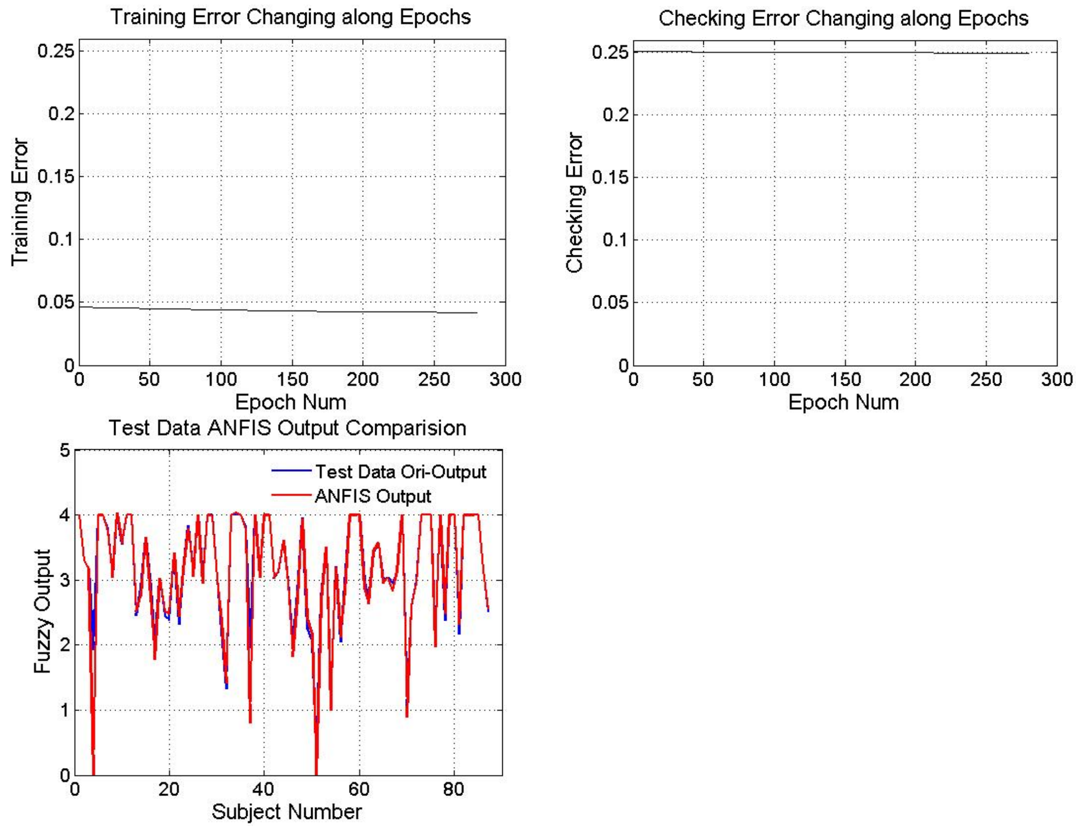


Figure C. 3 1-Fold CV results of Balance Test for Chk Err Criteria

b) 10 fold

Please input file name of the training data:balancedata2.txt

Please input file name of the test & checking data:balancedata1.txt

Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Train	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.044765	0.082334	0.643057	10
Fold 2	10	18	19	36	others	0.025318	0.065186	0.116395	1100
Fold 3	19	27	28	45	others	0.028517	0.418207	0.061303	280
Fold 4	28	36	37	54	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 5	37	45	46	63	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 6	46	54	55	72	others	Can not set up MFs,because all side-by-side stand time are 10.			
Fold 7	55	63	64	81	others	0.051177	0.246279	0.064416	10
Fold 8	64	72	73	90	others	0.047572	0.045632	0.320119	150
Fold 9	73	81	82	99	others	0.048698	0.454779	0.058918	190
Fold 10	82	87	88	99	others	0.046398	0.556822	0.011642	150
Average						0.041778	0.267034	0.182264	270

Table C. 2 10-Fold CV results of Balance Test for Chk Err Criteria

Fold 3 Results:

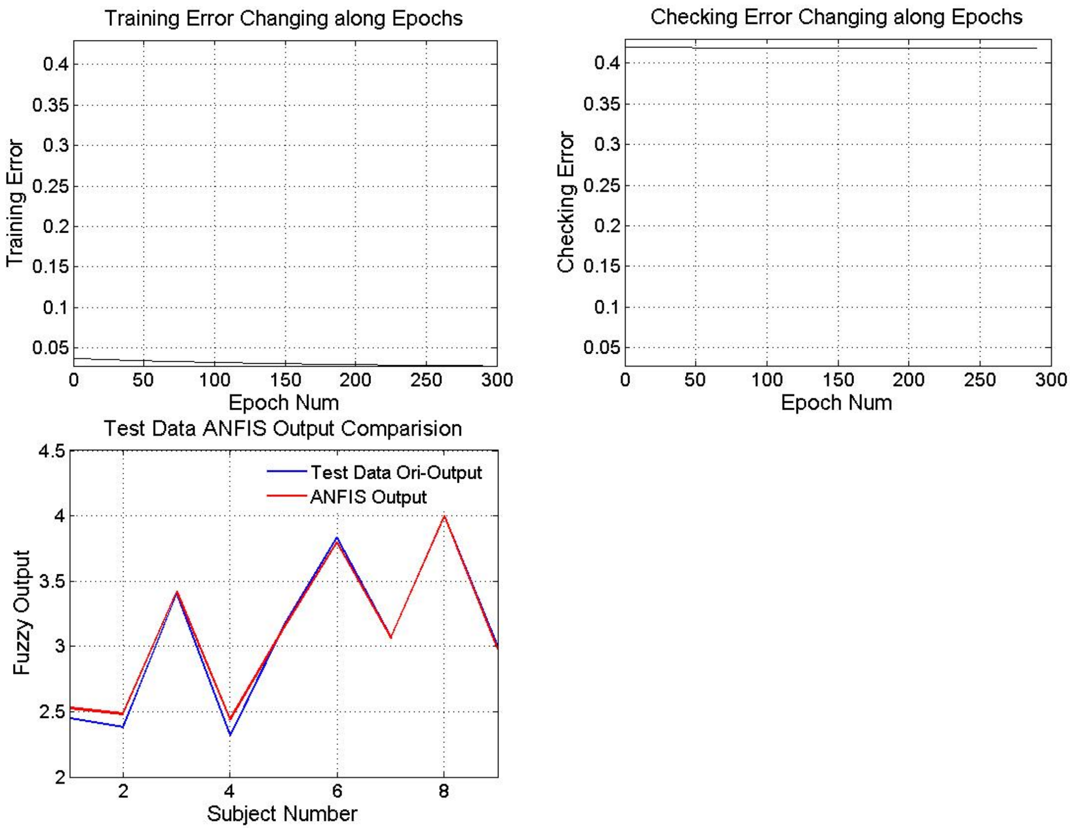


Figure C. 4 Fold 3 of 10-Fold CV results of Balance Test for Chk Err Criteria

b. Walking Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the training data:50walkingdata2.txt
Please input file name of the test data:50walkingdata1.txt
Num of fold cross validation:1
trainStart at trainData 1 trainEnd at trainData 87
testStart at testData 1 testEnd at testData 87
Training Data Error Rate 0.008443
Test Data Error Rate 0.013481
Epochs 1340

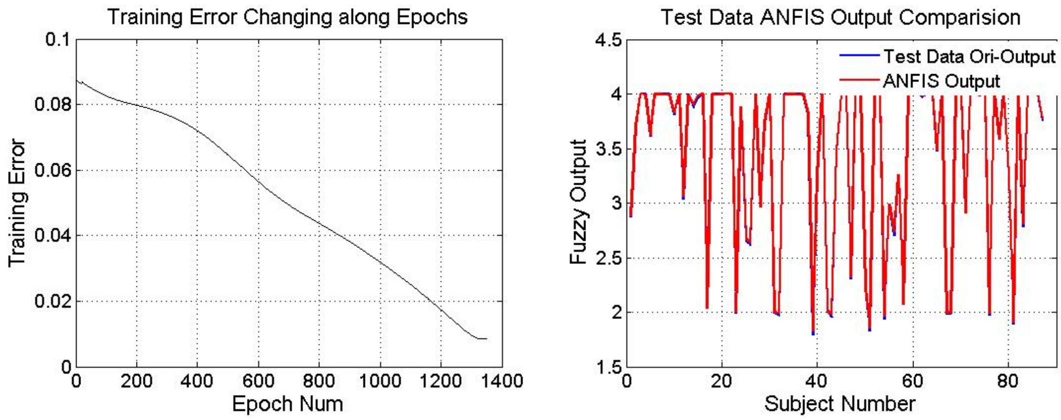


Figure C. 5 1-Fold CV results of Walking Test for Trn Err Criteria

b) 10 fold

Please input file name of the training data:50walkingdata2.txt
Please input file name of the test data:50walkingdata1.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Trn Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	others	0.008195	0.005518	1420
Fold 2	10	18	others	0.008078	0.019680	1330
Fold 3	19	27	others	0.008179	0.014360	1290
Fold 4	28	36	others	0.008423	0.008834	1400
Fold 5	37	45	others	0.037406	0.043278	1860
Fold 6	46	54	others	0.008488	0.015412	1600
Fold 7	55	63	others	0.008823	0.015372	1260
Fold 8	64	72	others	0.007588	0.006786	1310
Fold 9	73	81	others	0.071061	0.565560	250
Fold 10	82	87	others	0.008206	0.012853	1120
Average				0.0174447	0.0707653	1284

Table C. 3 10-Fold CV results of Walking Test for Trn Err Criteria

Fold 1 Results:

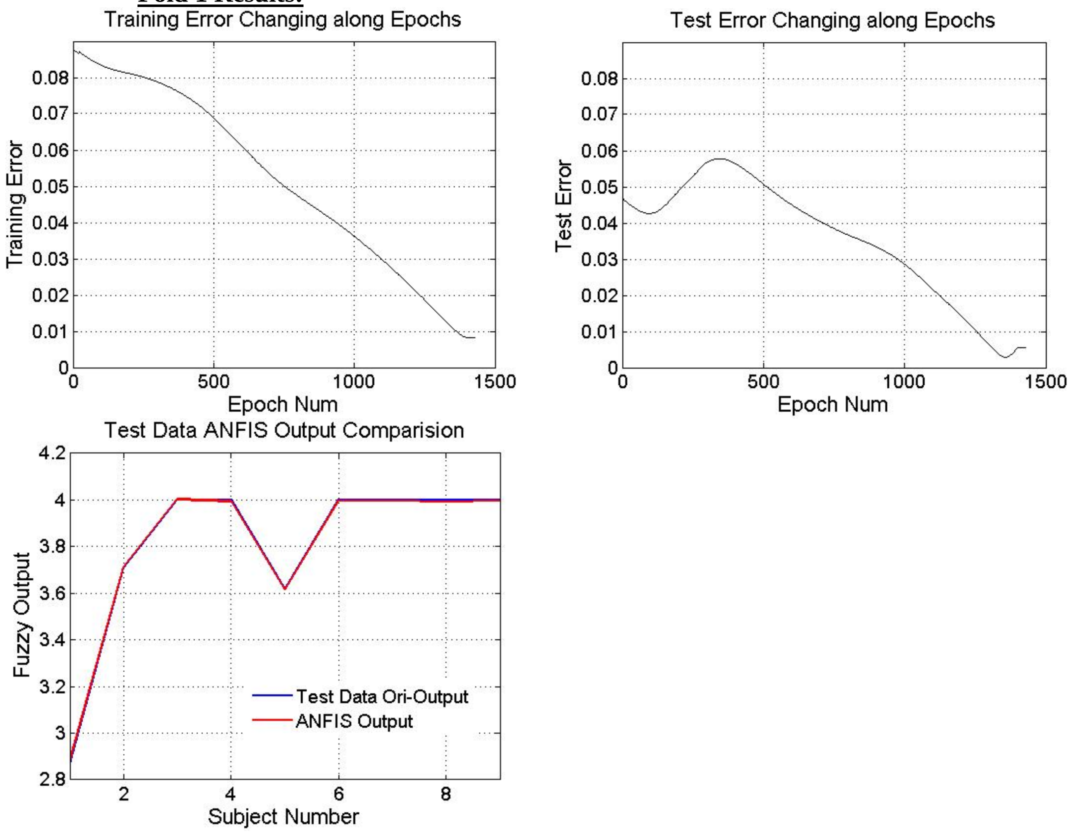


Figure C. 6 Fold 1 of 10-Fold CV results of Walking Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the training data:50walkingdata2.txt
Please input file name of the test & checking data:50walkingdata1.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.001712
Checking Data Error Rate 0.012232
Test Data Error Rate 0.012232
Epochs 1710

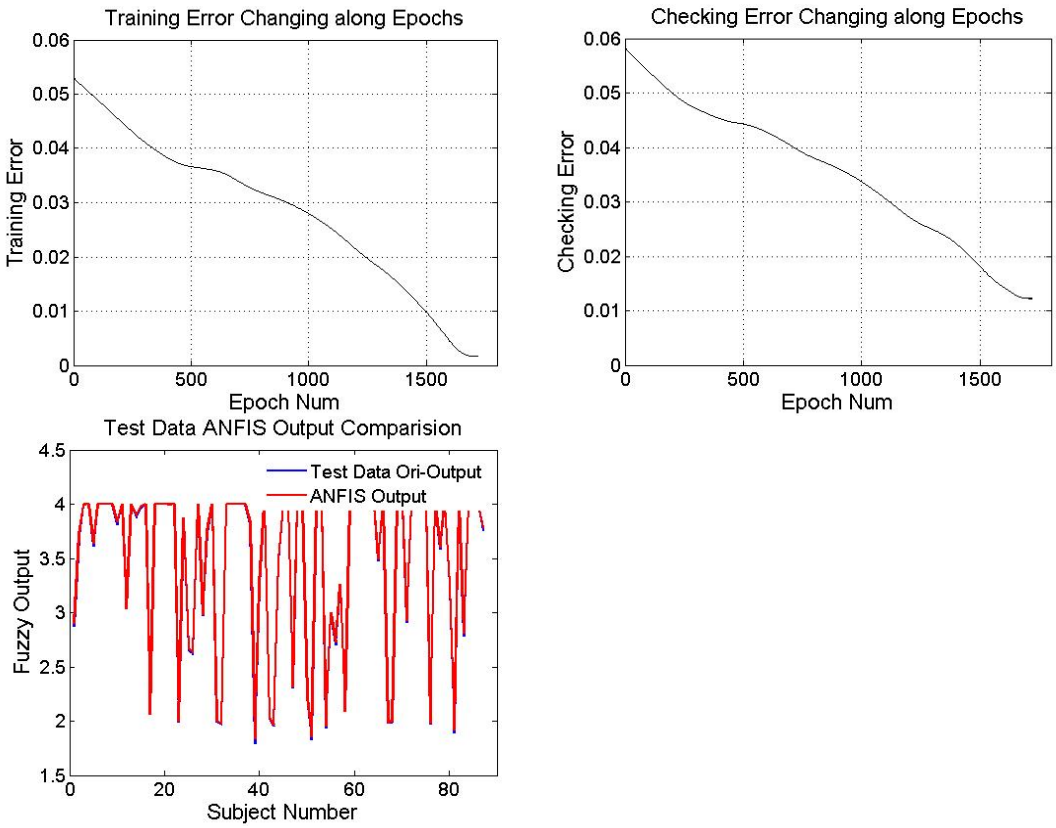


Figure C. 7 1-Fold CV results of Walking Test for Chk Err Criteria

b) 10 fold

Please input file name of the training data:50walkingdata2.txt
Please input file name of the test & checking data:50walkingdata1.txt
Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Train	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.001612	0.012506	0.013020	1640
Fold 2	10	18	19	36	others	0.033482	0.051390	0.051623	500
Fold 3	19	27	28	45	others	0.001395	0.041392	0.012484	1150
Fold 4	28	36	37	54	others	0.022256	0.126834	0.130546	690
Fold 5	37	45	46	63	others	0.029644	0.102703	0.384780	2280
Fold 6	46	54	55	72	others	0.029918	0.053028	0.110345	1250
Fold 7	55	63	64	81	others	0.031516	0.084192	0.057747	910
Fold 8	64	72	73	90	others	0.031489	0.200386	0.040830	870
Fold 9	73	81	82	99	others	0.038769	0.040341	0.335897	540
Fold 10	82	87	88	99	others	0.037356	0.036213	0.032758	240
Average						0.0257437	0.0748985	0.117003	1007

Table C. 4 10-Fold CV results of Walking Test for Chk Err Criteria

Fold 1 Results:

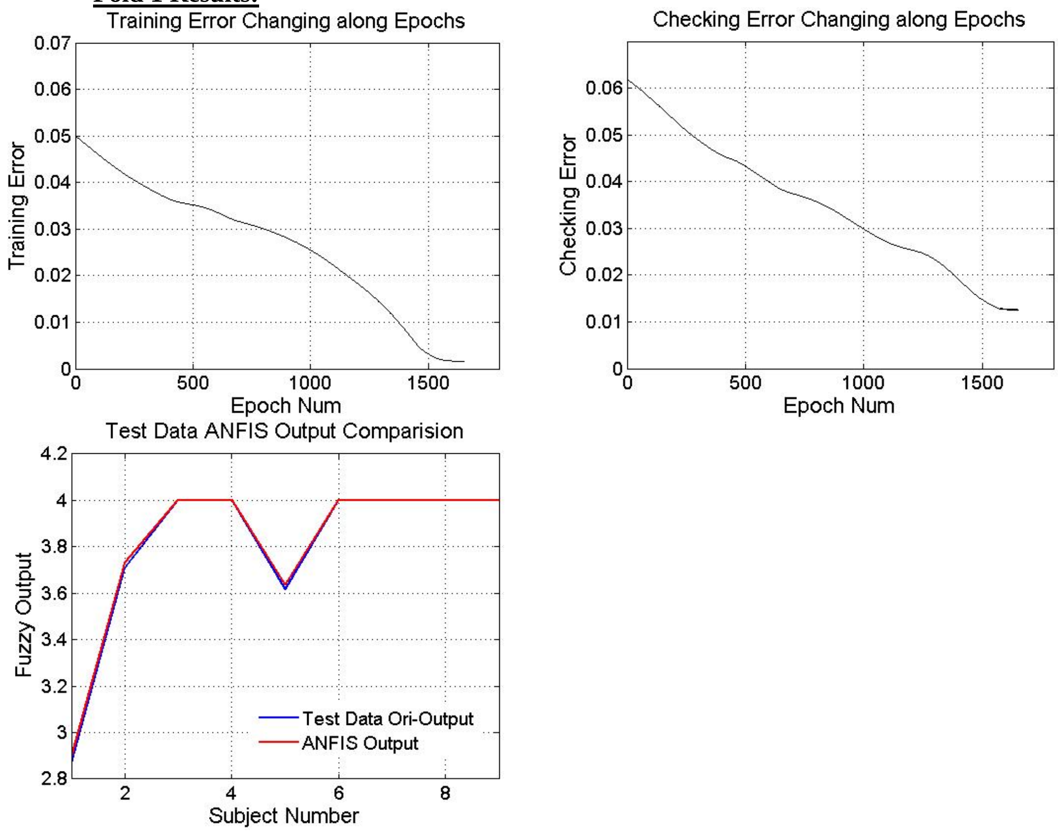


Figure C. 8 Fold 1 of 10-Fold CV results of Walking Test for Chk Err Criteria

c. Chair Test

1). Program running with training data and test data. Stop criteria: training error<0.0001

a) 1 fold

Please input file name of the training data:chairdata2.txt
Please input file name of the test data:chairdata1.txt
Num of fold cross validation:1
trainStart at trainData 1 trainEnd at trainData 87
testStart at testData 1 testEnd at testData 87
Training Data Error Rate 0.159205
Test Data Error Rate 0.121570
Epochs 1520

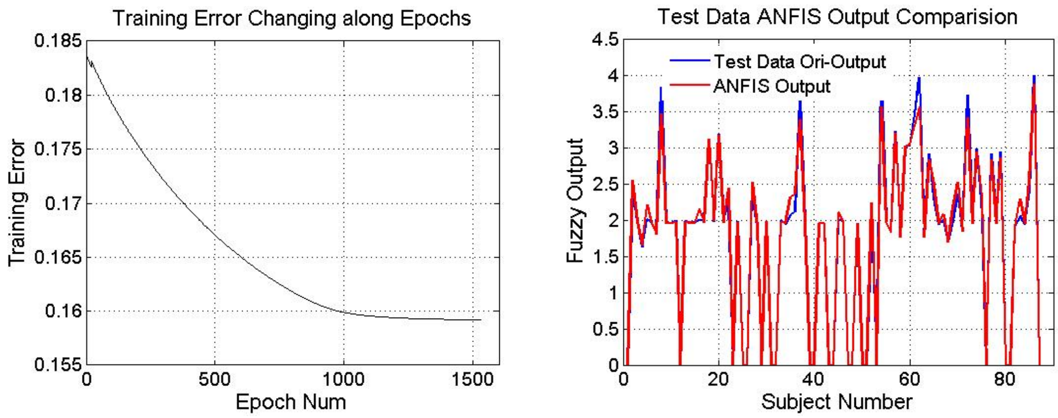


Figure C. 9 1-Fold CV results of Chair Test for Trn Err Criteria

b) 10 fold

Please input file name of the training data:chairdata1.txt
Please input file name of the test data:chairdata2.txt
Num of fold cross validation:10

	Test Start	Test End	Train	Trn Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	others	0.162007	0.155772	1430
Fold 2	10	18	others	0.161949	0.057245	1560
Fold 3	19	27	others	0.151769	0.088872	1450
Fold 4	28	36	others	0.167947	0.117677	1760
Fold 5	37	45	others	0.164928	0.101538	1480
Fold 6	46	54	others	0.168126	0.151745	1540
Fold 7	55	63	others	0.166914	0.149320	1670
Fold 8	64	72	others	0.129597	0.120804	2580
Fold 9	73	81	others	0.156745	0.094171	1510
Fold 10	82	87	others	0.144593	0.115311	1570
Average				0.1574575	0.115246	1655

Table C. 5 10-Fold CV results of Chair Test for Trn Err Criteria

Fold 1 Results:

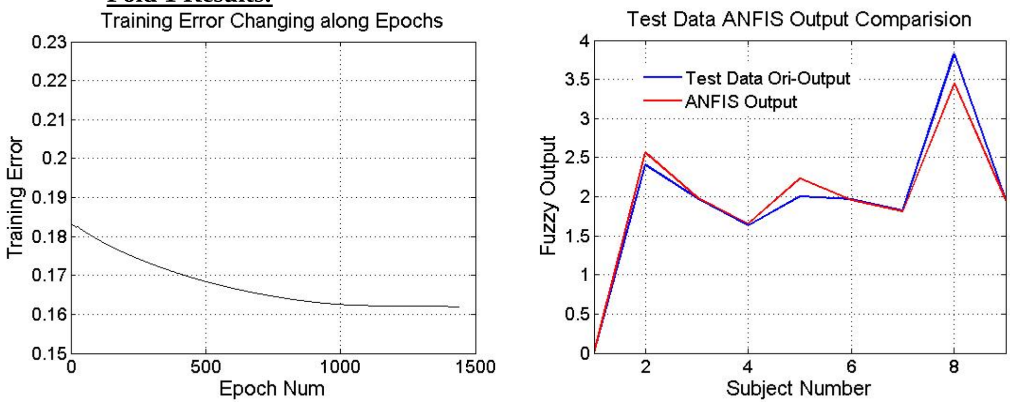


Figure C. 10 Fold 1 of 10-Fold CV results of Chair Test for Trn Err Criteria

2). Program running with training data checking data and test data. Stop criteria: checking error<0.0001

a) 1 fold

Please input file name of the training data:chairdata2.txt
Please input file name of the test & checking data:chairdata1.txt
Num of fold cross validation:1
trainStart 1 trainEnd 87
testStart 1 testEnd 87
checkStart 1 checkEnd 87
Training Data Error Rate 0.014120
Checking Data Error Rate 0.014960
Test Data Error Rate 0.014960
Epochs 1800

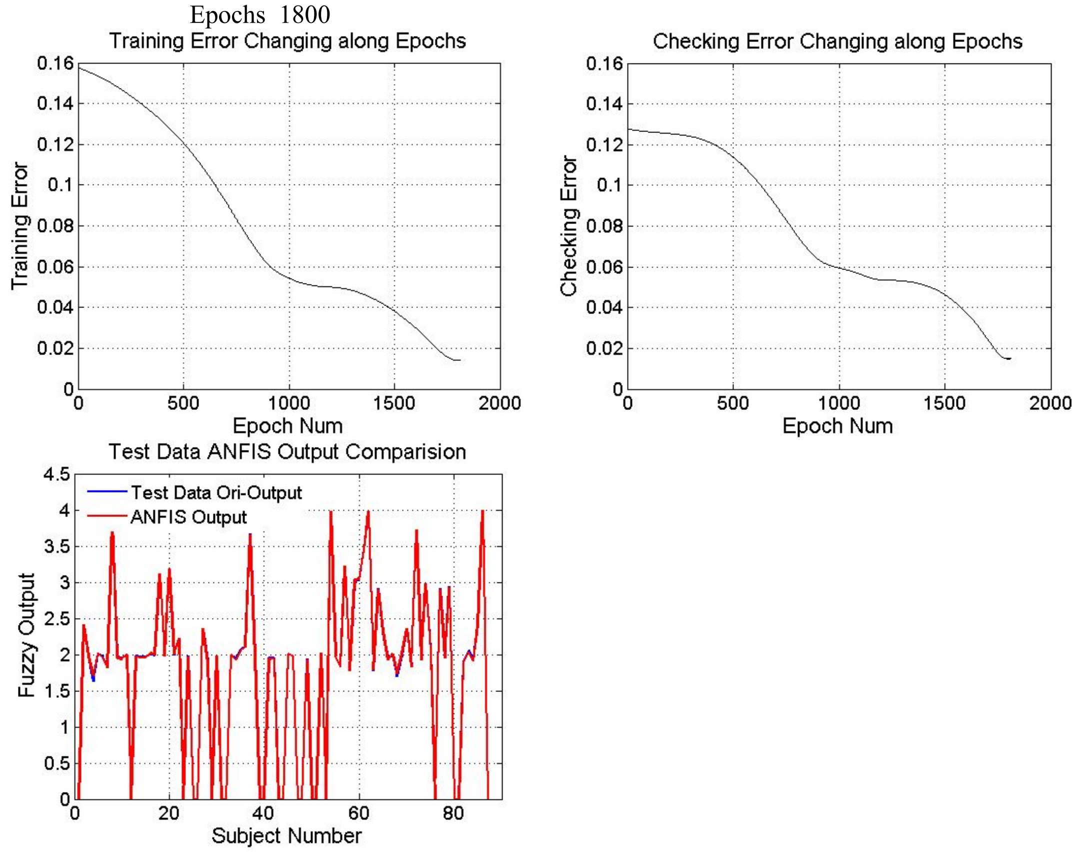


Figure C. 11 1-Fold CV results of Chair Test for Chk Err Criteria

b) 10 fold

Please input file name of the training data:chairdata2.txt
Please input file name of the test & checking data:chairdata1.txt
Num of fold cross validation:10

	Test Start	Test End	Chk Start	Chk End	Train	Trn Data Err Rate	Chk Data Err Rate	Test Data Err Rate	Epochs
Fold 1	1	9	10	27	others	0.144594	0.096679	0.180102	60
Fold 2	10	18	19	36	others	0.155044	0.114251	0.090452	60
Fold 3	19	27	28	45	others	0.056194	0.033589	0.060246	960
Fold 4	28	36	37	54	others	0.015130	0.011359	0.016439	1890
Fold 5	37	45	46	63	others	0.013945	0.009266	0.010316	1810
Fold 6	46	54	55	72	others	0.044163	0.061403	0.037202	1730
Fold 7	55	63	64	81	others	0.042161	0.059940	0.074385	1500
Fold 8	64	72	73	90	others	0.075976	0.073231	0.091908	890
Fold 9	73	81	82	99	others	0.045383	0.044862	0.052895	1340
Fold 10	82	87	88	99	others	0.135355	0.128135	0.113443	270
Average						0.0727945	0.063272	0.072739	1051

Table C. 6 10-Fold CV results of Chair Test for Chk Err Criteria

Fold 1 Results:

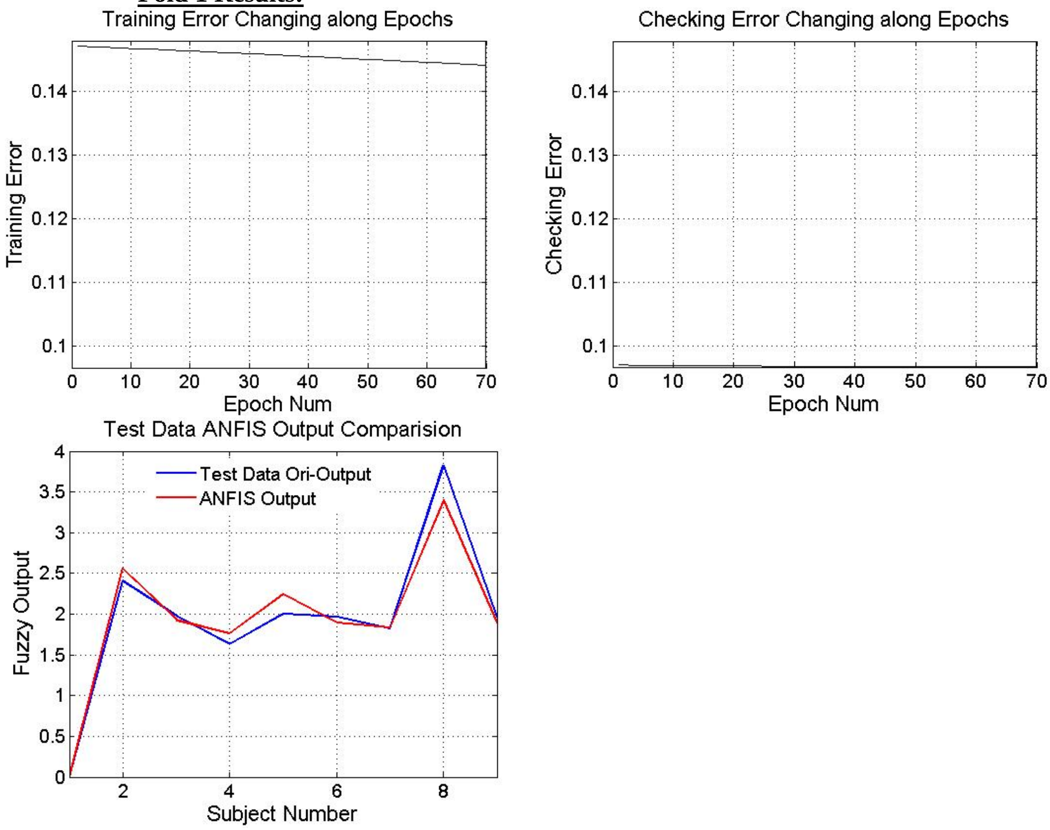


Figure C. 12 Fold 1 of 10-Fold CV results of Chair Test for Chk Err Criteria

D. Test on Tuning the Fuzzified SPPB system

a. Test on different point

For all following tests, parameter α is 0.75 and parameter β is 0.25.

Input: [10 9 2]

2.46→1.90 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.17 to 0.9)

Test #	Result			
1.00	0.90	0.45	1.45	2.43
2.00	0.90	0.43	1.44	2.44
3.00	0.90	0.43	1.44	2.44
4.00	0.90	0.44	1.43	2.44
5.00	0.90	0.43	1.44	2.44
6.00	0.90	0.45	1.44	2.43
7.00	0.90	0.46	1.43	2.44
8.00	0.90	0.43	1.44	2.44
9.00	0.90	0.44	1.44	2.44
10.00	0.90	0.43	1.44	2.44
STD	0.00	0.01	0.00	0.00

Figure D. 1 Tuning Test on the Fuzzified SPPB Balance Test (2.46→1.90)

Input: [10 8 1]

1.97→1.6 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.68 to 1.36)

Test #	Result	z_2	z_3	z_4
1.00	1.36	0.63	1.63	2.62
2.00	1.36	0.64	1.62	2.63
3.00	1.36	0.64	1.63	2.62
4.00	1.36	0.63	1.63	2.63
5.00	1.36	0.64	1.63	2.62
6.00	1.36	0.63	1.63	2.62
7.00	1.36	0.64	1.62	2.65
8.00	1.36	0.66	1.62	2.62
9.00	1.36	0.64	1.62	2.64
10.00	1.36	0.64	1.63	2.62
STD	0.00	0.01	0.00	0.01

Figure D. 2 Tuning Test on the Fuzzified SPPB Balance Test (1.97→1.60)

Input: [10 1 1]

1.02→1.3 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.02 to 1.3)

Test #	Result	z_2	z_3	z_4
1.00	1.30	1.29	2.29	3.24
2.00	1.30	1.29	2.28	3.24
3.00	1.30	1.29	2.27	3.28
4.00	1.30	1.29	2.33	3.28
5.00	1.30	1.29	2.30	3.29
6.00	1.30	1.29	2.29	3.55
7.00	1.30	1.29	2.29	3.31
8.00	1.30	1.29	2.26	3.30
9.00	1.30	1.29	2.29	3.33
10.00	1.30	1.29	2.26	3.28
STD	0.00	0.00	0.02	0.09

Figure D. 3 Tuning Test on the Fuzzified SPPB Balance Test (1.02→1.30)

Input: [10 8 2]

2.28→2.6 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.16 to 1.32)

Test #	Result	z_2	z_3	z_4
1.00	1.32	1.32	2.32	3.32
2.00	1.32	1.32	2.32	3.32
3.00	1.32	1.32	2.32	3.32
4.00	1.32	1.31	2.33	3.32
5.00	1.32	1.32	2.32	3.32
6.00	1.32	1.32	2.32	3.32
7.00	1.32	1.32	2.32	3.32
8.00	1.32	1.32	2.32	3.32
9.00	1.32	1.32	2.32	3.32
10.00	1.32	1.32	2.32	3.32
STD	0.00	0.00	0.00	0.00

Figure D. 4 Tuning Test on the Fuzzified SPPB Balance Test (2.28→2.6)

Input: [10 8 2]

2.28→1.7 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.16 to 0.86)

Test #	Result	z_2	z_3	z_4
1.00	0.86	0.42	1.42	2.41
2.00	0.86	0.41	1.41	2.42
3.00	0.86	0.42	1.41	2.42
4.00	0.86	0.42	1.41	2.42
5.00	0.86	0.40	1.42	2.41
6.00	0.86	0.41	1.42	2.41
7.00	0.86	0.41	1.42	2.41
8.00	0.86	0.40	1.42	2.41
9.00	0.86	0.41	1.41	2.42
10.00	0.86	0.43	1.40	2.42
STD	0.00	0.01	0.01	0.00

Figure D. 5 Tuning Test on the Fuzzified SPPB Balance Test (2.28→1.7)

Input: [10 1 2]

1.01→1.2 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.00 to 1.19)

Test #	Result	z_2	z_3	z_4
1.00	1.19	1.19	2.20	3.21
2.00	1.19	1.19	2.19	3.18
3.00	1.19	1.19	2.19	3.18
4.00	1.19	1.19	2.16	3.18
5.00	1.19	1.19	2.19	3.19
6.00	1.19	1.19	2.22	3.22
7.00	1.19	1.19	2.16	3.15
8.00	1.19	1.19	2.20	3.19
9.00	1.19	1.19	2.20	3.18
10.00	1.19	1.19	2.19	3.19
STD	0.00	0.00	0.02	0.02

Figure D. 6 Tuning Test on the Fuzzified SPPB Balance Test (1.01→1.2)

Input: [10 9 2]

2.46-→2.51 (in the program, $\sum_{i=2}^{4} w_i z_i$ change from 1.17 to 1.19)

Test #	Result	z_2	z_3	z_4
1.00	1.19	1.05	2.05	3.05
2.00	1.19	1.05	2.05	3.05
3.00	1.19	1.05	2.05	3.05
4.00	1.19	1.05	2.05	3.05
5.00	1.19	1.05	2.05	3.05
6.00	1.19	1.05	2.05	3.05
7.00	1.19	1.05	2.05	3.05
8.00	1.19	1.05	2.05	3.05
9.00	1.19	1.05	2.05	3.05
10.00	1.19	1.05	2.05	3.05
STD	0.00	0.00	0.00	0.00

Figure D. 7 Tuning Test on the Fuzzified SPPB Balance Test (2.46→2.51)

b. Test on Parameters α and β

For all following tests, parameter, input is [10 9 2]; output changes from 2.46 to 1.90; in the program it is changed from 1.17 to 0.9.

α =0.67, β =0.33

Test #	Result	z_2	z_3	z_4
1.00	1.17	1.00	2.00	3.00
2.00	1.17	1.00	2.00	3.00
3.00	1.17	1.00	2.00	3.00
4.00	1.17	1.00	2.00	3.00
5.00	1.17	1.00	2.00	3.00
6.00	1.17	1.00	2.00	3.00
7.00	1.17	1.00	2.00	3.00
8.00	1.17	1.00	2.00	3.00
9.00	1.17	1.00	2.00	3.00
10.00	1.17	1.00	2.00	3.00
STD	0.00	0.00	0.00	0.00

Figure D. 8 Tuning Test on the Fuzzified SPPB Balance Test (α =0.67, β =0.33)

α =0.75, β =0.25

Test #	Result	z_2	z_3	z_4
1.00	0.90	0.45	1.45	2.43
2.00	0.90	0.43	1.44	2.44
3.00	0.90	0.43	1.44	2.44
4.00	0.90	0.44	1.43	2.44
5.00	0.90	0.43	1.44	2.44
6.00	0.90	0.45	1.44	2.43
7.00	0.90	0.46	1.43	2.44
8.00	0.90	0.43	1.44	2.44
9.00	0.90	0.44	1.44	2.44
10.00	0.90	0.43	1.44	2.44
STD	0.00	0.01	0.00	0.00

Figure D. 9 Tuning Test on the Fuzzified SPPB Balance Test (α =0.75, β =0.25)

α =0.85, β =0.15

Test #	Result	z_2	z_3	z_4
1.00	0.90	0.43	1.45	2.43
2.00	0.90	0.48	1.43	2.44
3.00	0.90	0.45	1.44	2.44
4.00	0.90	0.43	1.43	2.45
5.00	0.90	0.45	1.43	2.44
6.00	0.90	0.48	1.44	2.44
7.00	0.90	0.46	1.44	2.43
8.00	0.90	0.44	1.44	2.43
9.00	0.90	0.43	1.41	2.46
10.00	0.90	0.41	1.44	2.44
STD	0.00	0.02	0.01	0.01

Figure D. 10 Tuning Test on the Fuzzified SPPB Balance Test (α =0.85, β =0.15)

α =0.9, β =0.1

Test #	Result	z_2	z_3	z_4
1.00	0.90	0.48	1.45	2.43
2.00	0.90	0.47	1.40	2.47
3.00	0.90	0.51	1.48	2.40
4.00	0.90	0.51	1.42	2.45
5.00	0.90	0.49	1.51	2.37
6.00	0.90	0.53	1.43	2.44
7.00	0.90	0.43	1.45	2.43
8.00	0.90	0.45	1.44	2.43
9.00	0.90	0.43	1.47	2.41
10.00	0.90	0.52	1.46	2.41
STD	0.00	0.04	0.03	0.03

Figure D. 11 Tuning Test on the Fuzzified SPPB Balance Test (α =0.9, β =0.1)

α =0.99, β =0.01

Test #	Result	z_2	z_3	z_4
1.00	0.90	2.48	1.37	2.32
2.00	0.90	1.28	2.31	1.56
3.00	0.90	1.05	1.66	0.18
4.00	0.90	2.08	1.43	2.31
5.00	0.90	0.50	1.61	2.28
6.00	0.90	0.64	1.48	2.37
7.00	0.90	3.15	1.51	2.15
8.00	0.90	0.56	2.62	1.34
9.00	0.90	3.17	1.23	2.40
10.00	0.90	0.55	0.82	3.00
STD	0.00	1.08	0.52	0.78

Figure D. 12 Tuning Test on the Fuzzified SPPB Balance Test (α =0.99, β =0.01)