



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Σχολή Μηχανικών

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

**Ανάπτυξη ασαφών συστημάτων με χρήση της εργαλειοθήκης FuzzyR
– εφαρμογή σε προβλήματα αναγνώρισης και πρόβλεψης
Διπλωματική Εργασία
του**

ΝΤΟΥΠΗ ΘΩΜΑ

Επιβλέπων Καθηγητής

Πάρις Μαστοροκόστας, *Καθηγητής*

Αθήνα-Αιγάλεω, Ιούλιος 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Σχολή Μηχανικών

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Ανάπτυξη ασαφών συστημάτων με χρήση της εργαλειοθήκης FuzzyR – εφαρμογή σε προβλήματα αναγνώρισης και πρόβλεψης

Όνοματεπώνυμο *Θωμάς Ντούπη*

Αριθμός Μητρώου 161169

Επιβλέπων Καθηγητής Πάρις Μαστοροκόστας
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14^η Ιουλίου 2022.

<i>Ιωάννης Αμοργίνος</i> <i>Λέκτορας Εφαρμογών</i>	<i>Πάρις Μαστοροκόστας</i> <i>Καθηγητής</i>	<i>Παναγιώτα Τσελέντη</i> <i>ΕΔΙΠ</i>
---	--	--

Αθήνα-Αιγάλεω, Ιούλιος 2022



UNIVERSITY OF WEST ATTICA

School of Engineering

Department of Informatics and Computer Engineering

**Development of fuzzy systems for system identification
and prediction via the FuzzyR Toolbox.**

Full Name Thomas Ntoupi

Identification Number 161169

Supervisor Paris Mastorocostas
Professor

Athens-Egaleo, July 2022

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Ντούπη Θωμάς** του Γρηγόρη, με αριθμό μητρώου **161169** φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του **Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών**, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου»

Ο Δηλών,

Θωμάς Ντούπη



Περίληψη

Στόχος της διπλωματικής αυτής, ήταν η ανάπτυξη δυο νευρο-ασαφών μοντέλων, που βασίζονται στο ANFIS και DENFIS, και η εφαρμογή τους σε προβλήματα πρόβλεψης χαοτικών χρονοσειρών και ταξινόμησης μαστογραφιών. Αρχικά αναπτύχθηκαν και εφαρμόστηκαν τα δυο αυτά μοντέλα στην χρονοσειρά Mackey-Glass, όπου χρησιμοποιήθηκε για αξιολόγηση των μοντέλων και εν συνεχεία στο σύνολο δεδομένων Μαστογραφίες.

Τα αποτελέσματα είναι αρκετά ενθαρρυντικά, καταφέροντας να κάνουν αξιόλογες προβλέψεις. Ενώ οι ταξινομήσεις καθώς αφορούν θέματα υγείας μπορούν να ληφθούν από έναν ειδικό σαν ένα συμβουλευτικό εργαλείο. Καλύτερες επιδόσεις μπορούν να επιτευχθούν, εφόσον υπάρχουν περιθώρια βελτίωσης.

Λέξεις Κλειδιά

Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, ANFIS, DENFIS, νευρο-ασαφές μοντέλα, ασαφής συστήματα, μαστογραφίες, χαοτικές χρονοσειρές, πρόβλεψη χρονοσειρών, ασαφείς κανόνες, ασαφής λογική, πρόβλημα ταξινόμησης μαστογραφίας.

Abstract

The purpose of this thesis was the development of two neuro-fuzzy models, based on ANFIS and DENFIS, and their application to chaotic time series prediction and classifying mammograms. Initially, these two models were developed and applied to the Mackey-Glass time series where it was used to evaluate the models and then to the Mammography dataset.

The results are quite encouraging, managing to make remarkable predictions. While the classifications, because they are health related, can be taken by an expert as a consulting tool. Better performance can be achieved since there is room for improvement.

Key words

Artificial Intelligence, Machine Learning, ANFIS, DENFIS, neuro-fuzzy models, fuzzy systems, mammograms, chaotic time series, time series prediction, fuzzy rules, fuzzy logic, mammogram classification problem.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του Προπτυχιακού Προγράμματος Σπουδών του Τμήματος Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής και σηματοδοτεί την ολοκλήρωση των σπουδών μου. Ως την ελάχιστη δυνατή μνεία, με την παρούσα παράγραφο θα ήθελα να ευχαριστήσω θερμά τους ανθρώπους με τους οποίους συνεργάστηκα και συνέβαλαν στην εκπόνηση της εργασίας αυτής.

Ευχαριστώ θερμά τον επιβλέπων καθηγητή μου, κύριο Πάρι Μαστοροκώστα, για την εμπιστοσύνη που μου έδειξε, αναθέτοντάς μου το συγκεκριμένο θέμα, την επιστημονική του καθοδήγηση, τις υποδείξεις του και τη συνεχή υποστήριξη από την αρχή μέχρι το τέλος.

Επιπλέον ευχαριστώ πολύ όλους τους συμφοιτητές μου για όλες τις όμορφες στιγμές και συζητήσεις που είχαμε καθ' όλη τη διάρκεια του πρώτου τίτλου σπουδών μου.

Τέλος, θέλω να ευχαριστήσω πολύ την οικογένεια μου, οι οποίοι πάντα υπήρξαν ένα ανεκτίμητο στήριγμα για εμένα. Είμαι ευγνώμον για την συμπαράσταση και την κατανόησή τους, καθ' όλη τη διάρκεια των σπουδών μου.

Πίνακας Περιεχομένων

Περίληψη	i
Abstract.....	ii
Ευχαριστίες.....	iii
Κατάλογος Πινάκων	vi
Κατάλογος Σχημάτων	vii
Κατάλογος Συνομογραφιών	ix
1. ANFIS	1
• Ασαφείς Αν-Τότε Κανόνες (If-Then Rules)	1
• Συστήματα Ασαφούς Συμπερασμού- Fuzzy Inference Systems (FIS).....	1
• Δομή ANFIS και Αλγόριθμος Εκμάθησης.....	3
(ANFIS - Adaptive-Network-based Fuzzy Inference System)	3
• Υβριδικός Αλγόριθμος μοντέλου ANFIS.....	6
• ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΕΛΑΧΙΣΤΩΝ ΤΕΤΡΑΓΩΝΩΝ (LSE).....	6
• Ο ΑΛΓΟΡΙΘΜΟΣ GRADIENT DESCENT	7
2. DENFIS.....	8
• Evolving Clustering Method ECM– Μέθοδος Εξελισσόμενης Ομαδοποίησης.....	8
• Online ECM	8
• Μαθησιακή διαδικασία στο DENFIS.....	15
3. Μετρικές αποδοτικότητας σε Προβλήματα ταξινόμησης.....	16
4. Εργαλειοθήκες	17
• FuzzyR	17
• FRBS	20
5. Πρόβλεψη της Χρονοσειράς Mackey-Glass.....	22
• Εισαγωγή.....	Error! Bookmark not defined.
6. Πρόβλεψη Mackey-Glass με ANFIS.....	22
7. Πρόβλεψη Mackey-Glass με DENFIS	25
8. Συμπεράσματα για MACKEY-GLASS Dataset	27
9. Μαστογραφία- Mammography	28
10. Εφαρμόζοντας το ANFIS σε Μαστογραφίες	28
• Πίνακας Αποτελεσμάτων του ANFIS σε 50 και 30 εποχές.....	32

11.	Δεύτερη Υλοποίηση ANFIS.....	33
	• Πίνακας Αποτελεσμάτων του ANFIS σε 50 και 30 εποχές.....	36
12.	Εφαρμόζοντας το DENFIS σε Μαστογραφίες	37
	• Πίνακας Αποτελεσμάτων του DENFIS σε 100, 50 και 30 εποχές.....	38
13.	Συμπεράσματα για Μαστογραφίες – Mammography Dataset.....	39
14.	Ανάλυση Πρωτευουσών Συνιστωσών (Principal Components Analysis, PCA)	39
15.	Εφαρμόζοντας το ANFIS σε PCA MAMMOGRAPHY-Μαστογραφίες	39
	• Πίνακας Αποτελεσμάτων του ANFIS σε 50 και 30 εποχές.....	43
16.	Εφαρμόζοντας το DENFIS σε PCA MAMMOGRAPHY-Μαστογραφίες.....	44
	• Πίνακας Αποτελεσμάτων του DENFIS σε 50 και 30 εποχές.....	46
17.	Συμπεράσματα για PCA Μαστογραφίες – Mammography Dataset.....	47
18.	ΕΠΙΛΟΓΟΣ	47
19.	Βιβλιογραφία	48
20.	Κώδικες	51
	• Mackey-Glass ANFIS.....	51
	• Mackey-Glass Denfis	55
	• Mammography ANFIS	59
	• Mammography DENFIS	70
	• Mammography ANFIS MF3	79
	• PCA Mammography ANFIS.....	89
	• PCA Mammography DENFIS.....	99

Κατάλογος Πινάκων

Πίνακας 4.1 Εργαλειοθήκη FUZZYR.....	20
Πίνακας 4.2 Εργαλειοθήκη FRBS.....	21
Πίνακας 11.1 Αποτελέσματα ANFIS με 2 ασαφείς συναρτήσεις στο Mammography dataset.....	32
Πίνακας 12.1 Αποτελέσματα ANFIS με 3 ασαφείς συναρτήσεις στο Mammography dataset.....	36
Πίνακας 13.1 Αποτελέσματα DENFIS στο Mammography dataset.....	38
Πίνακας 16.1 Αποτελέσματα ANFIS στο PCA Mammography dataset	43
Πίνακας 17.1 Αποτελέσματα DENFIS στο PCA Mammography dataset	46

Κατάλογος Σχημάτων

Εικόνα 1.1 Fuzzy Inference Systems (FIS).....	2
Εικόνα 1.2 Τύπου 3 Takagi και Sugeno’s Αν-τότε κανόνας	3
Εικόνα 1.3 Δομή ANFIS	4
Εικόνα 2.1 Σχηματική Αναπαράσταση Αλγορίθμου ECM	11
Εικόνα 2.2 Διαδικασία Δημιουργίας Συμπλέγματος.....	12
Εικόνα 2.3 ECM Online και off-line.....	14
Εικόνα 6.1 Αρχικές Συναρτήσεις Συμμετοχής Mackey-Glass ANFIS.....	23
Εικόνα 6.2 Τελικές Συναρτήσεις Συμμετοχής Mackey-Glass ANFIS	23
Εικόνα 6.3 Καμπύλη RMSE συνάρτηση εποχών Mackey-Glass ANFIS	24
Εικόνα 6.4 Έξοδος ANFIS(κόκκινο) και Πραγματική Έξοδος (μαύρο) Mackey-Glass ANFIS (2 Κανόνες, 0.01 ρυθμός μάθησης).	24
Εικόνα 6.5 Residuals - Διαφορά Μεταξύ Πραγματικής Εξόδου και Εξόδου ANFIS, Mackey-Glass ...	25
Εικόνα 7.1 Καμπύλη RMSE συνάρτηση εποχών Mackey-Glass DENFIS	26
Εικόνα 7.2 Έξοδος DENFIS(μπλέ) και Πραγματική Έξοδος (κόκκινο) Mackey-Glass DENFIS (0.01 ρυθμός μάθησης).	26
Εικόνα 7.3 Residuals - Διαφορά Μεταξύ Πραγματικής Εξόδου και Εξόδου DENFIS, Mackey-Glass .	27
Εικόνα 10.1 Αρχικές Συναρτήσεις Συμμετοχής Mammography ANFIS	28
Εικόνα 10.2 Τελικές Συναρτήσεις Συμμετοχής Mammography ANFIS	29
Εικόνα 10.3 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα ελέγχου Mammography	29
Εικόνα 10.4 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα εκπαίδευσης Mammography	30
Εικόνα 10.5 Confusion Matrix ANFIS των δεδομένων εκπαίδευσης Mammography.....	30
Εικόνα 10.6 Confusion Matrix ANFIS των δεδομένων ελέγχου Mammography.	31
Εικόνα 11.1 Αρχικές Συναρτήσεις Συμμετοχής MF3 ANFIS.....	33
Εικόνα 11.2 Τελικές Συναρτήσεις Συμμετοχής MF3 ANFIS	33
Εικόνα 11.3 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα ελέγχου Mammography MF3	34
Εικόνα 11.4 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα εκπαίδευσης Mammography MF3	34
Εικόνα 11.5 Confusion Matrix των δεδομένων εκπαίδευσης ANFIS Mammography MF3.	35
Εικόνα 11.6 Confusion Matrix των δεδομένων ελέγχου ANFIS Mammography MF3.	35
Εικόνα 12.1 Καμπύλη RMSE συνάρτηση εποχών DENFIS Mammography.	37

Εικόνα 12.2 Confusion Matrix DENFIS των δεδομένων εκπαίδευσης Mammography.	37
Εικόνα 12.3 Confusion Matrix DENFIS των δεδομένων ελέγχου Mammography	38
Εικόνα 15.1 Αρχικές Συναρτήσεις Συμμετοχής ANFIS PCA Mammography.....	40
Εικόνα 15.2 Τελικές Συναρτήσεις Συμμετοχής ANFIS PCA Mammography.....	40
Εικόνα 15.3 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα ελέγχου PCA Mammography	41
Εικόνα 15.4 Απεικόνιση RMSE για το μοντέλο ANFIS δεδομένα εκπαίδευσης PCA Mammography	41
Εικόνα 15.5 Confusion Matrix ANFIS των δεδομένων εκπαίδευσης PCA Mammography.	42
Εικόνα 15.6 Confusion Matrix ANFIS των δεδομένων ελέγχου PCA Mammography.	42
Εικόνα 16.1 Καμπύλη RMSE συνάρτηση εποχών DENFIS PCA Mammography.....	44
Εικόνα 16.2 Confusion Matrix DENFIS των δεδομένων εκπαίδευσης PCA Mammography.	45
Εικόνα 16.3 Confusion Matrix DENFIS των δεδομένων ελέγχου PCA Mammography.	45

Κατάλογος Συντομογραφιών

TN - AI Τεχνητή Νοημοσύνη - Artificial Intelligence
ANFIS Adaptive Neuro Fuzzy Inference System
DENFIS Dynamic Evolving Neural-Fuzzy Inference System
ECM Evolving Clustering Method
ECMc Evolving Clustering Method with constrained minimization
FAM Fuzzy Associative Memories
FIS Fuzzy Inference System
FN False Negatives
FP False Positives
LB Linguistic Label
LSE Least Square Estimate
MF Membership Function
MG Mackey Glass
ML Machine Learning – Μηχανική Μάθηση
PCA Principal Components Analysis
RMSE Root Mean Square Error
TN True Negatives
TP True Positives
TSK Takagi-Sugen-Kang

1. ANFIS

- Ασαφείς Αν-Τότε Κανόνες (If-Then Rules)

Οι Ασαφείς Αν-Τότε Κανόνες(Fuzzy If-Then Rules) είναι υποθετικές προτάσεις της μορφής:

Αν x είναι A Τότε y είναι B,

όπου A και B είναι λεκτικές τιμές (ασαφή σύνολα) των x και y, τα οποία ορίζονται στους χώρους X και Y αντίστοιχα και χαρακτηρίζονται από την κατάλληλη συνάρτηση συμμετοχής(Membership Functions). Λόγω της συνοπτικής τους μορφής, οι Ασαφείς Αν-Τότε Κανόνες χρησιμοποιούνται συχνά για την καταγραφή των ανακριβών τρόπων συλλογισμού, που παίζουν ουσιαστικό ρόλο στην ανθρώπινη ικανότητα να λαμβάνει αποφάσεις σε ένα περιβάλλον αβεβαιότητας και ανακρίβειας. Ένα παράδειγμα που περιγράφει ένα απλό γεγονός είναι:

Αν Πίεση είναι Υψηλό, Τότε Όγκος είναι Χαμηλό

Όπου Πίεση και Όγκος είναι λεκτικές μεταβλητές(linguistic variables), Χαμηλό, Μεσαίο και Υψηλό είναι λεκτικές τιμές (linguistic values or labels), όπου χαρακτηρίζονται από τις κατάλληλες συναρτήσεις συμμετοχής.

Μία άλλη μορφή των Ασαφών Αν-Τότε Κανόνων, προτάθηκε από τους Takagi και Sugeno, όπου τα ασαφή σύνολα εμπλέκονται μόνο στο τμήμα της υπόθεσης(premise part). Χρησιμοποιώντας τον Takagi και Sugeno Ασαφή Αν-Τότε κανόνα, μπορούμε να περιγράψουμε τη δύναμη της Αντίστασης ενός κινούμενου αντικειμένου όπως παρακάτω:

Αν Ταχύτητα είναι Υψηλό, Τότε Δύναμη = k * (Ταχύτητα)²

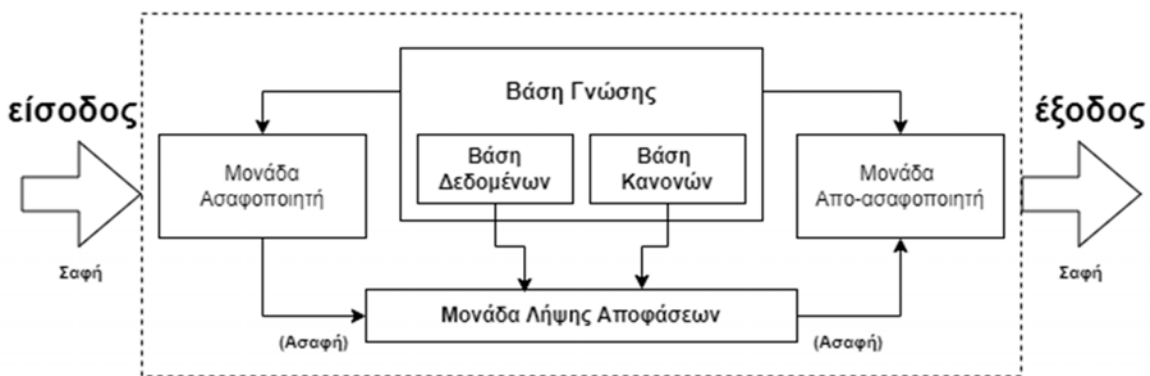
Όπου πάλι Υψηλό είναι λεκτική τιμή χαρακτηριζόμενη από μια συνάρτηση συμμετοχής. Ωστόσο, το τμήμα συμπεράσματος (consequence part) περιγράφεται από μια Μη-Ασαφή εξίσωση της μεταβλητής εισόδου, Ταχύτητα.

- Συστήματα Ασαφούς Συμπερασμού- Fuzzy Inference Systems (FIS)

Τα συστήματα ασαφούς συμπερασμού είναι γνωστά και ως συστήματα που βασίζονται σε ασαφείς κανόνες(fuzzy-rule-based systems), ασαφή μοντέλα(Fuzzy Models), ασαφείς συνειρμικές μνήμες (fuzzy associative memories-FAM), ή ασαφείς ελεγκτές όταν χρησιμοποιούνται σαν ελεγκτές. Πρακτικά ένα σύστημα ασαφούς συμπερασμού αποτελείται από πέντε λειτουργικά μπλόκ (Εικόνα 1.1):

- Μια βάση κανόνων που περιέχει έναν αριθμό ασαφών κανόνων Αν-Τότε.

- Μια βάση δεδομένων που ορίζει τις συναρτήσεις συμμετοχής των ασαφών συνόλων που χρησιμοποιούνται στους ασαφείς κανόνες.
- Μια μονάδα λήψης αποφάσεων που εκτελεί τις πράξεις εξαγωγής συμπερασμάτων στους κανόνες.
- Μια μονάδα ασαφοποιητή που μετατρέπει τις σαφείς εισόδους σε βαθμούς αντιστοιχίας με γλωσσικές τιμές.
- Μια μονάδα απο-ασαφοποιητή που μετατρέπει τα ασαφή αποτελέσματα του συμπεράσματος σε μια σαφή έξοδο.

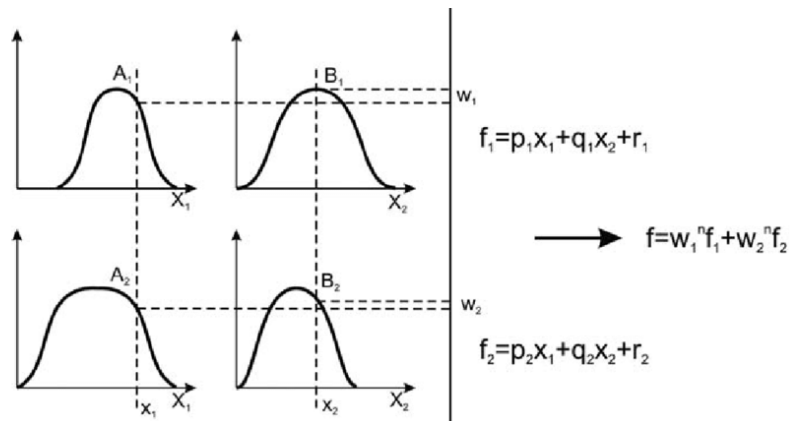


Εικόνα 1.1 Fuzzy Inference Systems (FIS)

Τα βήματα της ασαφούς συλλογιστικής που εκτελούνται από τα συστήματα ασαφούς συμπερασμού είναι:

1. Συγκρίνονται οι μεταβλητές εισόδου με τις συναρτήσεις συμμετοχής στο τμήμα υπόθεσης για να αποκτήσουν τις τιμές συμμετοχής κάθε λεκτικής τιμής. Αυτό το βήμα ονομάζεται ασαφοποίηση.
2. Συνδυάζονται (μέσω ενός συγκεκριμένου τελεστή T-norm, συνήθως πολλαπλασιασμού ή ελάχ.) οι τιμές συμμετοχής στο τμήμα υπόθεσης για να αποκτήσει βάρος ο κάθε κανόνας.
3. Δημιουργείται το κατάλληλο συμπέρασμα είτε ασαφές είτε σαφές κάθε κανόνα ανάλογα με το βάρος του.
4. Συγκεντρώνονται τα κατάλληλα συμπεράσματα για να παραχθεί μία σαφής έξοδος. Αυτό το βήμα ονομάζεται αποασαφοποίηση.

Στην ερευνα χρησιμοποιήθηκε ένας τύπος 3 Takagi και Sugeno's Αν-τότε κανόνας. Οι έξοδοι κάθε κανόνα είναι ένας γραμμικός συνδυασμός των μεταβλητών εισόδου συν έναν σταθερό όρο, και η τελική έξοδος είναι ο σταθμισμένος μέσος όρος της εξόδου κάθε κανόνα. (Εικόνα 1.2).

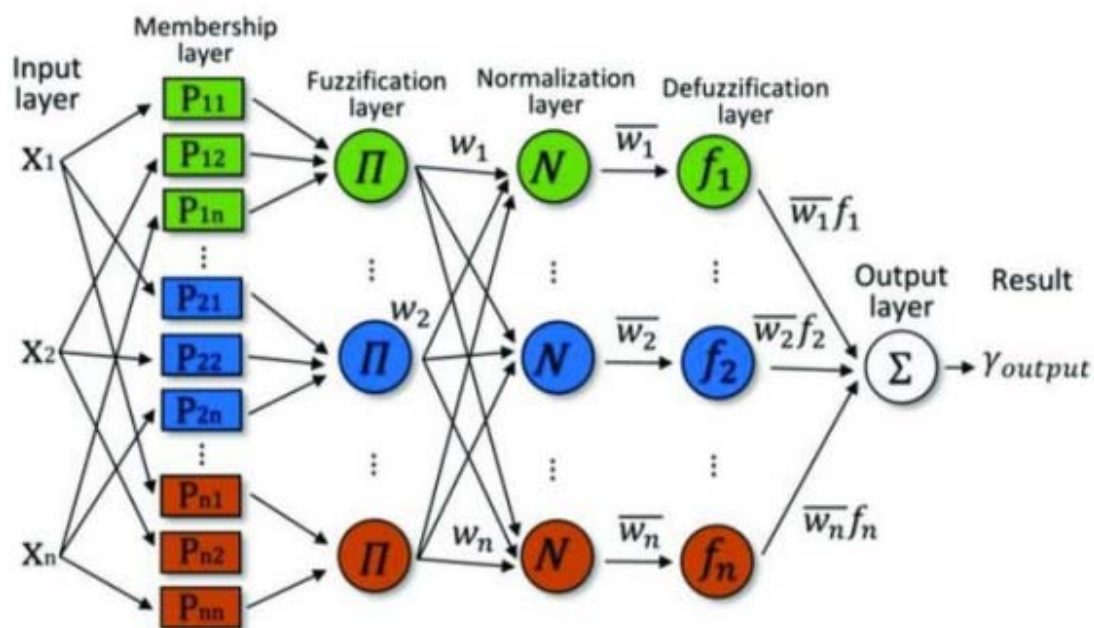


Εικόνα 1.2 Τύπου 3 Takagi και Sugeno's Αν-τότε κανόνας

- Δομή ANFIS και Αλγόριθμος Εκμάθησης (ANFIS - Adaptive-Neuro-based Fuzzy Inference System)

Το σύστημα ασαφούς συμπερασμού που βασίζεται σε προσαρμοστικό δίκτυο (ANFIS) είναι μια διαδικασία που βασίζεται σε δεδομένα που αντιπροσωπεύει μια προσέγγιση νευρωνικών δικτύων για την επίλυση προβλημάτων προσέγγισης συναρτήσεων. Οι διαδικασίες, που βασίζονται σε δεδομένα για τη σύνθεση δικτύων ANFIS βασίζονται τυπικά στη ομαδοποίηση ενός εκπαιδευτικού συνόλου αριθμητικών δειγμάτων της άγνωστης συνάρτησης που πρόκειται να προσεγγιστεί. Τα δίκτυα ANFIS έχουν εφαρμοστεί με επιτυχία σε εργασίες ταξινόμησης, έλεγχο διαδικασιών βάσει κανόνων, αναγνώριση προτύπων και παρόμοια προβλήματα. Υπάρχουν πολλοί τρόποι για τον συνδυασμό νευρωνικών δικτύων και ασαφών συστημάτων. Εμείς για την μοντελοποίηση χρησιμοποιήσαμε το **Υβριδικό Νεύρο-Ασαφές Σύστημα (Hybrid Neuro-Fuzzy System)**, που χρησιμοποιεί για την ανανέωση των παραμέτρων την **Gradient Descent** και **Least Square Estimate** μέθοδο. Δεδομένου ότι ο ορισμός της πρόγνωσης είναι μια πρόβλεψη της μελλοντικής πορείας και του αποτελέσματος, το περασμένο ιστορικό του αποτελέσματος θα πρέπει να διαμορφωθεί έτσι ώστε να βελτιστοποιήσει το μοντέλο που πρόκειται να αναπτυχθεί.

Η αρχιτεκτονική χρησιμοποιεί πέντε στρώματα πρόσθιας διάδοσης νευρωνικού δικτύου γνωστό και ως Προσαρμοζόμενο Νεύρο-Ασαφές Σύστημα Εξαγωγής Συμπερασμάτων (Adaptive Neuro-fuzzy Inference System). Σύμφωνα με την νεύρο-ασαφή εκτίμηση, ένα νευρωνικό δίκτυο προτείνεται να εφαρμοστεί στο ασαφές σύστημα, έτσι ώστε η δομή και η αναγνώριση των παραμέτρων της βάσης των ασαφών κανόνων να επιτυγχάνεται με τον καθορισμό, την προσαρμογή και την βελτιστοποίηση της τοπολογίας και οι παράμετροι του αντίστοιχου νεύρο-ασαφούς δικτύου με βάση μόνο τα διαθέσιμα δεδομένα. Το δίκτυο μπορεί να αναφερθεί και ως ένα προσαρμοζόμενο ασαφές σύστημα για την λήψη αποφάσεων με την ικανότητα της εκμάθησης ασαφών κανόνων από τα δεδομένα και ως διασυνδεδετική αρχιτεκτονική που παρέχεται με την γλωσσική έννοια.



Εικόνα 1.3 Δομή ANFIS

Στο παραπάνω σχήμα, οι κύκλοι απεικονίζουν σταθερό κόμβο ενώ τα τετράγωνα έναν προσαρμοζόμενο κόμβο.

Σε αυτή τη διασυνδεδετική δομή, υπάρχουν κόμβοι εισόδου και εξόδου και στα κρυφά στρώματα υπάρχουν κόμβοι που λειτουργούν ως συναρτήσεις συμμετοχής καθώς και κανόνες. Αυτό εξαλείφει το μειονέκτημα ενός κανονικού πολυστρωματικού δικτύου πρόσθιας φοράς, το οποίο είναι δύσκολο για έναν παρατηρητή να το κατανοήσει ή να το τροποποιήσει. Χάριν απλότητας, υποθέτουμε ότι το εξεταζόμενο ασαφές σύστημα εξαγωγής συμπερασμάτων έχει δυο εισόδους και μια έξοδο. Για ένα πρώτου βαθμού Sugeno ασαφές μοντέλο, ένα τυπικό σύνολο κανόνων με δυο ασαφείς κανόνες *If-Then* μπορεί να εκφραστεί ως :

Κανόνας 1: If x is A1 and y is B1, then f1 = p1 x + q1 y + r1

Κανόνας 2: If x is A2 and y is B2, then f2 = p2 x + q2 y + r2

Όπου x και y είναι δυο είσοδοι και A_i και B_i είναι γλωσσικές ετικέτες οι οποίες συνεργάζονται με την συνάρτηση του κόμβου.

Όπως βλέπουμε το σύστημα έχει συνολικά πέντε στρώματα η λειτουργία καθενός περιγράφεται παρακάτω :

Στρώμα 1 Κόμβος Εισόδου

Οι κόμβοι του επίπεδου αυτού εμπεριέχουν της συναρτήσεις συμμετοχής. Σε αυτό το επίπεδο βρίσκονται οι αναφερόμενες, σύμφωνα με τον κύριο Jang, υποθετικοί παράμετροι. Κάθε κόμβος i σε αυτό το στρώμα είναι ένας τετραγωνικός κόμβος, ο οποίος προσαρμόζεται σε μια συνάρτηση συμμετοχής:

$$O_i^1 = \mu_{A_i}(x)$$

Όπου x είναι η είσοδος στον κόμβο i και A_i είναι η γλωσσική ετικέτα (μικρό, μεγάλο) η οποία συσχετίζεται με την παραπάνω συνάρτηση κόμβου.

Με άλλα λόγια O_i^1 είναι η συνάρτηση συμμετοχής του A_i και ορίζει το βαθμό με τον οποίο ένα x ικανοποιεί την ποσοτικοποίηση A_i .

Στρώμα 2 - Κόμβος Κανόνων

Κάθε κόμβος σε αυτό το στρώμα είναι ένας κυκλικός κόμβος που χαρακτηρίζεται Π , ο οποίος πολλαπλασιάζει όλα τα σήματα από τα εισερχόμενα τμήματα και έχει ως έξοδο το γινόμενο.

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2$$

Κάθε έξοδος κόμβου αντιπροσωπεύει την ισχύ πυροδότησης ενός κανόνα.

Στρώμα 3 - Μέσος Όρος Κόμβων

Κάθε κόμβος σε αυτό το στρώμα είναι ένας κυκλικός κόμβος που χαρακτηρίζεται ως N . Ο κόμβος i υπολογίζει την αναλογία της δύναμης πυροδότησης του i κανόνα ως προς το άθροισμα της δύναμης πυροδότησης όλων των κανόνων. Κάθε κόμβος σε αυτά τα επίπεδα υπολογίζει το βάρος το οποίο κανονικοποιείται :

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2$$

Στρώμα 4 - Προκύπτοντες κόμβοι

Το επίπεδο αυτό περιλαμβάνει γραμμικές συναρτήσεις οι οποίες εξαρτώνται από τα σήματα εισόδου. Αυτό σημαίνει ότι υπολογίζεται η συνεισφορά του i κανόνα σε σχέση με την συνολική έξοδο ή την έξοδο του μοντέλου *AND/OR*.

Κάθε κόμβος i σε αυτό το στρώμα είναι ένας τετραγωνικός κόμβος με μια συνάρτηση κόμβου:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i)$$

Όπου \bar{w}_i είναι η έξοδος του επιπέδου 3 και $\{p_i, q_i, r_i\}$ είναι οι παράμετροι του κόμβου αυτού. Οι παράμετροι σε αυτό το στρώμα αναφέρονται και ως προκύπτοντες παράμετροι.

Στρώμα 5 - Κόμβος Εξόδου

Ο κυκλικός κόμβος σε αυτό το στρώμα είναι μόνος του και χαρακτηρίζεται ως Σ , ο οποίος υπολογίζει την συνολική έξοδο αθροίζοντας όλα τα εισερχόμενα σήματα :

$$O_i^5 = \text{overalloutput} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

- Υβριδικός Αλγόριθμος μοντέλου ANFIS

Για την εκπαίδευση του δικτύου χρησιμοποιούμε έναν υβριδικό αλγόριθμο, ο οποίος χωρίζεται σε δύο μέρη: σε μετάβαση προς τα εμπρός (forward pass) και μετάβαση προς τα πίσω (back-word pass). Λειτουργεί ως εξής: στο μπροστινό πέρασμα τα σήματα φτάνουν στο Επίπεδο 5 και υπολογίζονται με τη μέθοδο του ελαχίστου τετραγώνου στις προκύπτουσες παραμέτρους, καθώς διατηρούνται σταθερές. Στην μετάβαση προς τα πίσω ένας αλγόριθμος back-propagation διαδίδει την παράγωγο του σφάλματος από το επίπεδο εξόδου έως το επίπεδο 2, στη συνέχεια ενημερώνει τις υποθετικές παραμέτρους με την χρήση Gradient Descend, ενώ οι προκύπτουσες παράμετροι διατηρούν τις διορθώσεις. Η χρήση της προσέγγισης Batch είναι πιο απλή στην εφαρμογή από τη on-line εκμάθηση.

- ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΕΛΑΧΙΣΤΩΝ ΤΕΤΡΑΓΩΝΩΝ (LSE)

Αφού δοθούν τα δείγματα εκπαίδευσης, θα πρέπει να επιλέξουμε τις κατάλληλες τιμές των βαρών του δικτύου για να προσεγγίσουμε την συνάρτηση υπόθεσης. Ένα λογικό σενάριο θα ήταν να διαλέξουμε τα συναπτικά βάρη έτσι ώστε η συνάρτηση να βρίσκεται όσον τον δυνατό πιο κοντά στην ανεξάρτητη μεταβλητή y για τα δεδομένα εισόδου που έχουμε διαθέσιμα. Έτσι θα θέσουμε μία συνάρτηση ή οποία θα είναι σε θέση να «μετρά» για κάθε τιμή των βαρών το πόσο κοντά η συνάρτηση $h(x(i))$ βρίσκεται με τα αντίστοιχα $y(i)$. Με τον τρόπο αυτό ορίζεται η συνάρτηση κόστους ως εξής :

$$j(w) = \frac{1}{2m} \sum_{i=1}^m (h(x^{(i)}) - y^i)^2$$

Η παραπάνω συνάρτηση κόστους αποτελεί μία συνάρτηση ελαχίστων τετραγώνων, η οποία είναι σε θέση να μετρά την απόσταση μεταξύ των δειγμάτων της εξαρτημένης μεταβλητής και της συνάρτησης υπόθεσης, για κάθε ζεύγος εκπαίδευσης, ανεξάρτητα από το πρόσημο της διαφοράς σε κάθε διακριτό σημείο στο χώρο. Για το λόγο αυτό η διαφορά υψώνεται στο τετράγωνο, προσφέροντας μας μια πιο γενική εικόνα της προσέγγισης. Για να οδηγηθώ στο επιθυμητό ελάχιστο της συνάρτησης κόστους ακολουθώ τη κλίση της εφαπτομένης, της μερικής παραγώγου του σφάλματος ως προς τα συναπτικά βάρη.

- Ο ΑΛΓΟΡΙΘΜΟΣ GRADIENT DESCENT

Στη θεωρία που αναπτύσσουμε επιθυμούμε να διαλέξουμε τα συναπτικά βάρη w έτσι ώστε να ελαχιστοποιήσουμε το σφάλμα που παράγεται από τη συνάρτηση κόστους $J(w)$. Για να πραγματοποιηθεί αυτό, θα χρησιμοποιήσουμε έναν αλγόριθμο ανίχνευσης της κλίσης, μέσω της μερικής παραγώγου του σφάλματος ως προς τα συναπτικά βάρη, ώστε να εντοπίσουμε την τάση του σφάλματος. Ύστερα από αυτό θα ελαχιστοποιήσουμε τη συνάρτηση κόστους $J(w)$ για να συγκλίνει στις κατάλληλες τιμές. Ο αλγόριθμος αυτός ονομάζεται αλγόριθμος καθοδικής κλίσης (gradient descent), όπου κατά την εκκίνηση του τα συναπτικά βάρη κατέχουν τυχαίες τιμές και επανειλημμένα ανανεώνονται ακολουθώντας τον παρακάτω κανόνα ανανέωσης (update rule) :

$$w_j := w_j - a \frac{\partial J(w)}{\partial w_j}$$

Ο συμβολισμός $:=$ αναφέρεται στο ότι η νέα τιμή του εκάστοτε βάρους αριστερά του συμβολισμού, θα ισούται με την διαφορά της προηγούμενης τιμής του με το γινόμενο του βαθμού μάθησης (learning rate) a και της μερικής παραγώγου της συνάρτησης κόστους ως προς την παλιά τιμή του συναπτικού βάρους. Ο παραπάνω κανόνας εφαρμόζεται σε όλες τις τιμές του $\{j = 1, \dots, m\}$ ενώ πραγματοποιεί βήματα απότομης καθόδου (steepest descent) μέσω της παραγώγου του σφάλματος. Για να γίνει πιο κατανοητή η διαδικασία ας θεωρήσουμε ότι έχουμε μόνο ένα δείγμα εκπαίδευσης και συνεπώς μόνο ένα ζεύγος $(x(i), y(i))$, έτσι ώστε να μην επηρεαζόμαστε από το άθροισμα για όλα τα j δείγματα. Με την υπόθεση αυτή ισχύσει ότι :

$$\begin{aligned} \frac{\partial J(w)}{\partial w_j} &= \frac{\partial \frac{1}{2m} (h(x) - y)^2}{\partial w_j} = 2 \frac{1}{2} (h(x) - y) \frac{\partial (h(x) - y)}{\partial w_j} = \\ & (h(x) - y) \frac{\partial \left(\sum_{i=0}^m w_i x_i - y \right)}{\partial w_j} = (h(x) - y) x_j \end{aligned}$$

Στη περίπτωση ενός και μόνο δείγματος εκπαίδευσης, ο παραπάνω τύπος εξάγει τον εξής κανόνα ανανέωσης των συναπτικών βαρών:

$$w_j := w_j - \alpha (h(x^{(i)}) - y^{(i)}) x_j$$

Ο κανόνας ονομάζεται κανόνας LMS (least mean squares) και είναι επίσης γνωστός και σαν κανόνας μάθησης Widrow - Hoff. Ο κανόνας αυτός έχει αρκετές ιδιότητες, όπως ότι το μέτρο του ορίσματος ανανέωσης των συναπτικών βαρών είναι ανάλογο του σφάλματος ($y(i) - h(x(i))$). Εάν η πρόβλεψη παράγει ένα μεγάλο σφάλμα η αλλαγή στο εκάστοτε συναπτικό βάρος θα είναι μεγάλη, αντίστοιχα εάν το σφάλμα είναι μικρό, τότε η αλλαγή στο συναπτικό βάρους θα είναι μικρή.

2. DENFIS

- **Evolving Clustering Method ECM– Μέθοδος Εξελισσόμενης Ομαδοποίησης**

Η εξελισσόμενη, συνδεδεμένη(online), με βάση τη μέγιστη απόσταση(maximum distance-based) ομαδοποίηση ονομάζεται ECM, προτείνεται για την υλοποίηση ενός διαμερίσματος διασποράς του χώρου εισόδου με σκοπό τη δημιουργία ασαφών κανόνων συμπερασμάτων. Αυτή η μέθοδος έχει δύο τρόπους: η πρώτη συνήθως εφαρμόζεται σε συνδεδεμένα(online) συστήματα μάθησης και η δεύτερη είναι πιο κατάλληλη για συστήματα εκμάθησης εκτός σύνδεσης(offline). Το συνδεδεμένο ECM χρησιμοποιείται στο συνδεδεμένο μοντέλο DENFIS. Το ECM εκτός σύνδεσης με περιορισμένη ελαχιστοποίηση(constrained minimization) (ECMc) είναι μια επέκταση του συνδεδεμένου τρόπου. Λαμβάνει το αποτέλεσμα από την συνδεδεμένη λειτουργία(online mode) ως αρχικές τιμές. Στη συνέχεια εφαρμόζεται μια βελτιστοποίηση, η οποία κάνει μια προκαθορισμένη αντικειμενική συνάρτηση βασισμένη σε ένα μέτρο απόστασης για να φτάσει στην ελάχιστη τιμή που υπόκειται σε δεδομένους περιορισμούς.

- **Online ECM**

Χωρίς καμία βελτιστοποίηση, το online ECM είναι ένας γρήγορος αλγόριθμος με ένα πέρασμα(one-pass), για μια δυναμική εκτίμηση του αριθμού των συμπλεγμάτων σε ένα σύνολο δεδομένων και για την εύρεση των τρεχόντων κέντρων τους στο χώρο δεδομένων εισόδου. Είναι μια μέθοδος ομαδοποίησης που βασίζεται στην απόσταση συνδεδεσιμότητας. Με αυτήν τη μέθοδο, τα κέντρα συστάδων αντιπροσωπεύονται από εξελιγμένους κόμβους. Σε οποιοδήποτε σύμπλεγμα, η μέγιστη απόσταση, MaxDist, μεταξύ ενός σημείου παραδείγματος και του κέντρου συμπλέγματος, είναι μικρότερη από μια τιμή κατωφλίου, Dthr, που έχει οριστεί ως παράμετρος ομαδοποίησης και θα επηρεάσει τον αριθμό των συμπλεγμάτων που πρέπει να εκτιμηθούν.

Η απόσταση μεταξύ των διανυσμάτων \mathbf{x} και \mathbf{y} , δηλώνει μια γενική Ευκλείδεια απόσταση που ορίζεται ως :

$$\|\mathbf{x} - \mathbf{y}\| = \left(\sum_{i=1}^q |x_i - y_i|^2 \right)^{\frac{1}{2}} / q^{\frac{1}{2}}$$

Όπου x, y ανήκει R^d .

Στη διαδικασία ομαδοποίησης, τα παραδείγματα δεδομένων προέρχονται από μια ροή δεδομένων και η διαδικασία ξεκινά με ένα κενό σύνολο συμπλεγμάτων. Όταν δημιουργείται ένα νέο σύμπλεγμα, το κέντρο του συμπλέγματος, (Cluster Center-**CC**), ορίζεται και η ακτίνα του συμπλέγματος, (Cluster radius- **Ru**), που ορίζεται αρχικά στο μηδέν. Ορισμένα δημιουργημένα συμπλέγματα θα ενημερωθούν αλλάζοντας τις θέσεις των κέντρων τους και αυξάνοντας τις ακτίνες των συστάδων τους. Ποιο σύμπλεγμα θα ενημερωθεί και πόσο θα αλλάξει, εξαρτάται από τη θέση του τρέχοντος παραδείγματος στον χώρο εισαγωγής. Ένα σύμπλεγμα δεν θα ενημερώνεται πλέον όταν η ακτίνα του συμπλέγματος του, Ru , φτάσει την τιμή που ισούται με μια τιμή κατωφλίου (threshold value), D_{thr} .

Αυτό βασίζεται στην ιδέα της δυναμικής προσθήκης και τροποποίησης των συστάδων καθώς παρουσιάζονται νέα δεδομένα, όπου η τροποποίηση στα συμπλέγματα επηρεάζει τόσο τη θέση των συστάδων όσο και το μέγεθος του συμπλέγματος, όσον αφορά μια παράμετρο ακτίνας που σχετίζεται με κάθε συστάδα που καθορίζει τα όρια αυτού του συμπλέγματος. Η ECM έχει μόνο μία παράμετρο, η οποία οδηγεί την προσθήκη συστάδων, γνωστή ως κατώφλι απόστασης D_{thr} . Όταν προστίθενται νέα συμπλέγματα, τα κέντρα τους ορίζονται να ισούνται με το παράδειγμα που πυροδότησε τη δημιουργία τους και η ακτίνα R ενός νέου συμπλέγματος ορίζεται αρχικά στο μηδέν. Το R αυξάνεται καθώς περισσότερα διανύσματα κατανέμονται στο σύμπλεγμα. Λόγω του μηχανισμού με τον οποίο ενημερώνεται το R , δεν μπορεί να υπερβαίνει το D_{thr} . Ο αλγόριθμος ECM φαίνεται παρακάτω:

Δημιουργήστε το πρώτο κέντρο συμπλέγματος από το πρώτο παράδειγμα I_0

Για κάθε επόμενο διάνυσμα n **Κάνω**

Βρες την ελάχιστη απόσταση D_{min} μεταξύ I_n και κάθε κέντρου συμπλέγματος C_n

Αν το D_{min} είναι **Μικρότερο από** οποιαδήποτε ακτίνα συστάδας **Τότε**

Προσθέστε το n στο πλησιέστερο σύμπλεγμα

Αλλιώς

Βρες το σύμπλεγμα a με ελάχιστη τιμή $S_{i,j}$, όπου $S_{i,j} = D_{i,j} + R_{i,j}$, $D_{i,j}$ είναι η απόσταση μεταξύ του κέντρου του συμπλέγματος και του διανύσματος j και $R_{i,j}$ είναι η ακτίνα της συστάδας i

Αν $S_{i,a} > 2 D_{thr}$ **τότε**

Δημιούργησε ένα νέο σύμπλεγμα

Αλλιώς

Ενημέρωσε a

Τέλος Αν

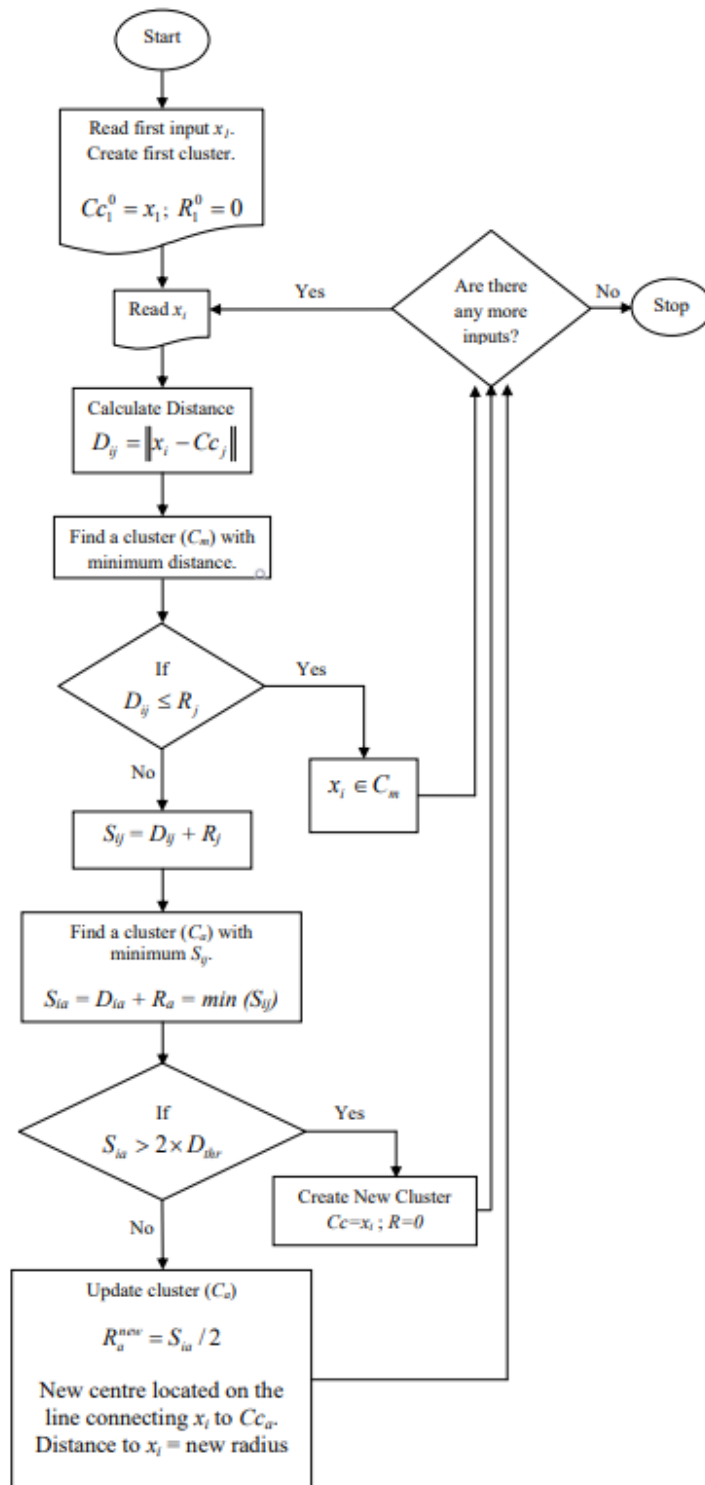
Τέλος Αν

Τέλος Για

Όταν το σύμπλεγμα a ενημερώνεται, το κέντρο του μετατοπίζεται πιο κοντά στο l_n και η ακτίνα του $R_{a(t+1)}$ ορίζεται σύμφωνα με την παρακάτω εξίσωση:

$$R_{a(t+1)} = S_{i,a}^2$$

Το νέο κέντρο του a , $C_{a(t+1)}$ ρυθμίζεται έτσι ώστε η απόστασή του να βρίσκεται στη γραμμή μεταξύ $C_{a(t)}$ και l_n σε απόσταση $R_{a(t+1)}$.



Εικόνα 2.1 Σχηματική Αναπαράσταση Αλγορίθμου ECM

Βήμα 0: Το πρώτο σύμπλεγμα C_1^0 δημιουργείται εκχωρώντας το πρώτο σημείο δεδομένων στην είσοδο ρεύμα ως το πρώτο κέντρο συστάδων C_1^0 και ορίζοντας την ακτίνα του συμπλέγματος $R_1 = 0$.

Βήμα 1: Εάν όλα τα σημεία δεδομένων της χρονοσειράς έχουν παρουσιαστεί στον αλγόριθμο, ο αλγόριθμος τερματίζεται. Διαφορετικά, η απόσταση μεταξύ του τρέχοντος σημείου δεδομένων x_i και όλων των n από τα υπάρχοντα κέντρα συμπλέγματος C_j υπολογίζονται ως

$$D_j = ||x_i - C_j|| \text{ όπου } j = 1, 2, \dots, n \text{ είναι ο δείκτης συμπλέγματος.}$$

Βήμα 2: Η απόσταση D_{ij} υπολογίζεται και συγκρίνεται με την ακτίνα των υπάρχοντων συστάδων R_j .

Αν τουλάχιστον για ένα R_j η συνθήκη $D_{ij} \leq R_j$ ικανοποιείται, τότε το x_i ανήκει σε ένα σύμπλεγμα

C_m με την ελάχιστη απόσταση $D_{im} = ||x_i - C_m|| = \min(||x_i - C_j||)$ για $D_{ij} \leq R_j$ ($j = 1, 2, \dots, n$). Σε αυτήν την περίπτωση, δεν δημιουργείται νέο σύμπλεγμα, ούτε ενημερώνεται το υπάρχον σύμπλεγμα και ο αλγόριθμος επιστρέφει στο Βήμα 1. Διαφορετικά, εάν $D_{ij} > R_j$, ο αλγόριθμος προχωρά στο επόμενο βήμα.

Βήμα 3: Για όλα τα υπάρχοντα συμπλέγματα, υπολογίζεται το $S_{ij} = D_{ij} + R_j$ ($j = 1, 2, \dots, n$). ο

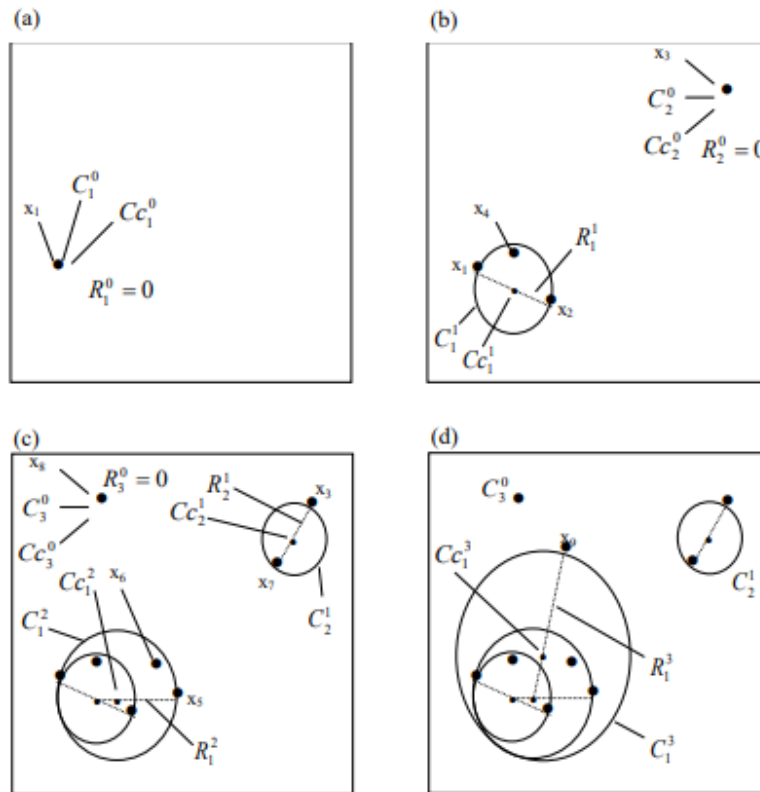
συστάδα C_a με την ελάχιστη τιμή $S_{ia} = D_{ia} + R_a = \min(S_{ij})$ όπου $j = 1, 2, \dots, n$

προσδιορίζεται.

Βήμα 3(i): Αν $S_{ia} > 2 \times D_{thr}$, το σημείο δεδομένων x_i δεν ανήκει σε κανένα υπάρχον σύμπλεγμα.

Έτσι, δημιουργείται ένα νέο σύμπλεγμα όπως περιγράφεται στο Βήμα 0 και ο αλγόριθμος επιστρέφει στο Βήμα 1. Διαφορετικά, ο αλγόριθμος προχωρά στο επόμενο βήμα.

Βήμα 3(ii): Το σύμπλεγμα C_a με $S_{ia} \leq 2 \times D_{thr}$ ενημερώνεται αυξάνοντας την ακτίνα R_a σε $R_a^{new} = S_{ia}/2$. Το νέο cluster center C_a^{new} βρίσκεται στο σημείο της γραμμής συνδέοντας το x_i και το C_a και η απόσταση από το σημείο x_i γίνεται ίση με R_a^{new} . Στη συνέχεια, ο αλγόριθμος επιστρέφει στο Βήμα 1.



Εικόνα 2.2 Διαδικασία Δημιουργίας Συμπλέγματος.

α) Το παράδειγμα x_1 προκαλεί το ECM να δημιουργήσει ένα καινούργιο σύμπλεγμα c_1^0 . β) x_2 : ενημερώνει το σύμπλεγμα $c_1^0 \rightarrow c_1^1$, x_3 : δημιουργεί καινούργιο σύμπλεγμα c_2^0 , x_4 : μην κάνεις τίποτα. γ) x_5 : ενημέρωσε το σύμπλεγμα $c_1^1 \rightarrow c_1^2$, x_6 : μην κάνεις τίποτα, x_7 : ενημέρωσε σύμπλεγμα $c_2^0 \rightarrow c_2^1$, x_8 : δημιουργεί καινούργιο σύμπλεγμα c_3^0 . δ) x_9 : ενημερώνει το σύμπλεγμα $c_1^2 \rightarrow c_1^3$.

Η ECM εκτός σύνδεσης(off-line), που ονομάζεται ECMc, εφαρμόζει μια διαδικασία βελτιστοποίησης στα προκύπτοντα κέντρα συμπλέγματος μετά την εφαρμογή του ECM. Το ECMc χωρίζει ένα σύνολο δεδομένων συμπεριλαμβανομένου του p διανύσματος x_i , $i = 1, 2, \dots, p$, σε n συμπλέγματα $C_{i,j} = 1, 2, \dots, n$, και βρίσκει ένα κέντρο συμπλέγματος σε κάθε σύμπλεγμα, για να ελαχιστοποιήσει την αντικειμενική συνάρτηση, η οποία βασίζεται σε ένα μέτρο απόστασης που υπόκειται σε δεδομένους περιορισμούς. Λαμβάνοντας τη γενική Ευκλείδεια απόσταση ως μέτρο μεταξύ π.χ. X_i διάνυσμα, σε συστάδα j και του αντίστοιχου κέντρου συστάδας C_{c_j} , η αντικειμενική συνάρτηση ορίζεται από την ακόλουθη εξίσωση:

$$J = \sum_{j=1}^n J_j = \sum_{j=1}^n \left(\sum_{x_i \in C_j} \|x_i - C_{c_j}\| \right) \quad (1)$$

Όπου $J_j = \sum_{x_i \in C_j} \|x_i - C_j\|$ είναι η αντικειμενική μέσα στο σύμπλεγμα j

$$i = 1, 2, \dots, p; \quad j = 1, 2, \dots, n.$$

και οι περιορισμοί ορίζονται από την επόμενη εξίσωση

$$\|x_i - C_j\| \leq D_{thr}, j = 1, 2, \dots, n.$$

(2)

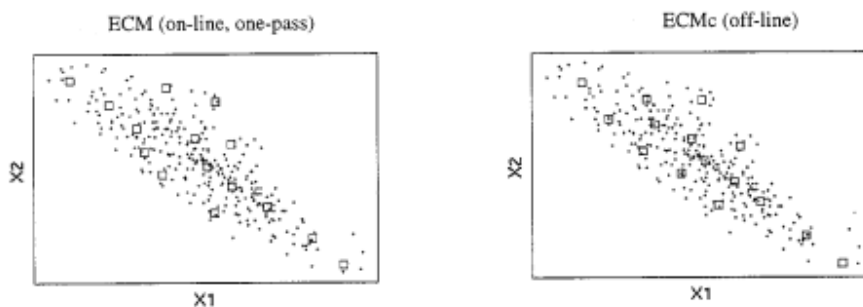
Τα διαμερισμένα συμπλέγματα τυπικά ορίζονται από έναν $p \times n$ πίνακα δυαδικών μελών U , όπου το στοιχείο u_{ij} είναι 1, εάν το i -στο σημείο δεδομένων x_i ανήκει στο σύμπλεγμα j , και 0 διαφορετικά. Αφού σταθεροποιηθούν τα κέντρα συμπλέγματος C_{cj} , η ελαχιστοποίηση u_{ij} για (1) και (2) προκύπτει ως εξής:

$$\forall \|x_i - C_{cj}\| \leq \|x_i - C_{ck}\|, \quad \text{για κάθε } j \neq k, \\ u_{ij} = 1, \text{ αλλιώς } u_{ij} = 0.$$

(3)

Για batch-mode λειτουργία, η μέθοδος καθορίζει τα κέντρα συμπλέγματος C_{cj} και τον πίνακα μελών U επαναληπτικά, χρησιμοποιώντας τα ακόλουθα βήματα.

- **Βήμα 1:** Αρχικοποίηση του κέντρου συμπλέγματος $C_{cj}, j = 1, 2, \dots, n$, που προέρχονται από το αποτέλεσμα της διαδικασίας ομαδοποίησης ECM.
- **Βήμα 2:** Προσδιορίστε τον πίνακα μελών U με το (3).
- **Βήμα 3:** Χρησιμοποιήστε τη μέθοδο ελαχιστοποίησης περιορισμένων με τα (1) και (2) για να αποκτήσει νέα κέντρα συμπλέγματος.
- **Βήμα 4:** Υπολογίστε την αντικειμενική συνάρτηση J σύμφωνα με (3). Σταματάει εάν το αποτέλεσμα είναι κάτω από μια συγκεκριμένη τιμή ανοχής ή η βελτίωσή του σε σχέση με την προηγούμενη επανάληψη είναι κάτω από ένα ορισμένο όριο ή ο αριθμός επαναλήψεων των λειτουργιών ελαχιστοποίησης είναι πάνω από μια συγκεκριμένη τιμή. Διαφορετικά, ο αλγόριθμος επιστρέφει στο Βήμα 2.



Εικόνα 2.3 ECM Online και off-line

- Μαθησιακή διαδικασία στο DENFIS

Το DENFIS χρησιμοποιεί σύστημα ασαφούς συμπερασμού τύπου Takagi-Sugeno (Takagi and Sugeno, 1985) όπου ο κανόνας i είναι:

If x_1 is R_{i1} and x_2 is R_{i2} and ... and x_q is R_{iq} , then y is $f_i(x_1, x_2, \dots, x_q)$

(1)

και το y είναι το κατά συνέπεια, το " x_j είναι R_{ij} " ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, q$) είναι $m \times q$ ασαφείς προτάσεις καθώς m προηγούμενα σχηματίζουν m ασαφείς κανόνες, q είναι ο αριθμός των εισόδων χρησιμοποιείται, x_j ($j = 1, 2, \dots, q$) είναι προηγούμενες μεταβλητές που ορίζονται στα U του λόγου, X_j ($j = 1, 2, \dots, q$), R_{ij} ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, q$) είναι ασαφή σύνολα που ορίζονται από τις ασαφείς συναρτήσεις συμμετοχής που δίνονται από: $\mu_{R_{ij}} : X_j \rightarrow [0,1]$. Η συνάρτηση συμμετοχής που χρησιμοποιείται είναι η Γκαουσιανή που δίνεται από:

$$\mu(x) = MF(x, \sigma, c) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right)$$

(2)

όπου c είναι το κέντρο του συμπλέγματος. Το σ είναι το πλάτος και συνήθως ορίζεται ως πολλαπλάσιο του D_{thr} , όπου το D_{thr} είναι η τιμή κατωφλίου (παράμετρος ομαδοποίησης) που ελέγχει το μέγεθος και επομένως τον αριθμό των συστάδων που θα δημιουργηθούν. Ο αριθμός των συναρτήσεων συμμετοχής εντοπίζεται αυτόματα κατά τη διάρκεια της εκπαίδευσης. Το DENFIS υιοθετεί τα ασαφή συστήματα συμπερασμάτων Takagi-Sugeno πρώτης τάξης όπου $f_i(x_1, x_2, \dots, x_q)$, $i = 1, 2, \dots, m$, είναι γραμμικές συναρτήσεις. Η εξίσωση για το επόμενο μέρος δίνεται από:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_q x_q$$

(3)

Όπου $\beta_0, \beta_1, \beta_2, \dots$, και β_q είναι παράμετροι που πρέπει να βελτιστοποιηθούν κατά τη διάρκεια της μάθησης.

Στο DENFIS, τα πρώτα n_0 ζεύγη δεδομένων, όπου $n_0 = q + 6$ (Kasabon and Song, 2002), χρησιμοποιούνται για τη δημιουργία των πρώτων m (m είναι ο αριθμός των αρχικών κανόνων) κέντρων συμπλέγματος με την εξελισσόμενη μέθοδο ομαδοποίησης. Για κάθε κέντρο συστάδας C_i ($i = 1, 2, \dots, m$) βρίσκονται τα πλησιέστερα p_i σημεία δεδομένων στο κέντρο. Χρησιμοποιώντας την Εξ. (2) και τη θέση του κέντρου, θα βρεθούν τα προηγούμενα του κανόνα που αντιστοιχούν στο συγκεκριμένο σύμπλεγμα. Τα προαναφερθέντα σημεία δεδομένων p_i και οι αποστάσεις τους από το κέντρο συστάδων C_i χρησιμοποιούνται στη συνέχεια για την εκτίμηση των επακόλουθων παραμέτρων m, n_0 και p_i .

Όταν ένα νέο ζεύγος δεδομένων παρουσιάζεται στο DENFIS, θα δημιουργηθεί ένας νέος ασαφής κανόνας, εάν η ECM δημιουργήσει ένα νέο κέντρο συμπλέγματος ($S_{ij} > 2 \times D_{thr}$) για αυτό το σημείο δεδομένων. Χρησιμοποιώντας τη θέση αυτού του νέου κέντρου συμπλέγματος και (2), θα διαμορφωθεί ο προηγούμενος του νέου κανόνα. Προκειμένου να εκτιμηθεί μια συνάρτηση αρχικής συνέπειας για τον πρόσφατα καθιερωμένο ασαφή κανόνα, ένας υπάρχων ασαφής κανόνας, όπου ο

κόμβος κανόνα είναι πιο κοντά στον νεοσυσταθέντα κόμβο κανόνων, θα βρεθεί. Από την άλλη πλευρά, εάν η απόσταση μεταξύ του νέου ζεύγους δεδομένων και ενός κόμβου κανόνα (S_{ij}) είναι μικρότερη από $2 \times D_{thr}$, τότε αυτός ο υπάρχων ασαφής κανόνας θα ενημερωθεί, χρησιμοποιώντας τη μέθοδο σταθμισμένου αναδρομικού εκτιμητή ελαχίστου τετραγώνου (LSE) με forgetting factor που παίρνει τιμές συνήθως μεταξύ 0,8 και 1 (Kasabov and Song, 2002). Επιπλέον, η ECM μπορεί να αλλάξει τη θέση του κόμβου κανόνα, όπως αναφέρεται στο βήμα 3(ii), οπότε ο αντίστοιχος κανόνας θα ενημερωθεί αλλάζοντας τα προηγούμενά του. Η επιλογή μικρών τιμών D_{thr} μπορεί να προκαλέσει πολλά περιττά συμπλέγματα που με τη σειρά τους μπορεί να οδηγήσουν σε υπερβολική προσαρμογή (overfitting). Η υπερβολική προσαρμογή μπορεί να ανιχνευθεί, όταν η απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης αυξάνεται, ενώ η απόδοση στην προσομοίωση του συνόλου δεδομένων δοκιμής μειώνεται.

3. Μετρικές αποδοτικότητας σε Προβλήματα ταξινόμησης

Συχνά στα προβλήματα ταξινόμησης χρησιμοποιούνται δύο μετρικές για την ποιότητα των μοντέλων, τις F1 Score και Accuracy. Και στις δυο μετρικές, όσο μεγαλύτερη είναι η τιμή τόσο πιο σωστά το μοντέλο μας καταφέρνει να ταξινομήσει τα δεδομένα μας σε κλάσεις. Παρόλα αυτά και οι δύο υπολογίζονται με διαφορετική φόρμουλα έχοντας το καθένα τα θετικά και τα αρνητικά του.

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

Μαθηματικός ορισμός του F1-Score

Αποσαφηνίζοντας τον τύπο: TP (True Positives) είναι τα στοιχεία που ανήκουν σε μια κλάση και τοποθετήθηκαν σωστά σε αυτήν. FP (False Positives) είναι τα στοιχεία που δεν ανήκουν σε μια κλάση αλλά παρόλα αυτά τοποθετήθηκαν σε αυτήν. TN (True Negatives) είναι τα στοιχεία που δεν ανήκουν σε μια κλάση και σωστά δεν τοποθετήθηκαν σε αυτήν. FN (False Negatives) είναι τα στοιχεία που εσφαλμένα θεωρήθηκε ότι δεν ανήκουν σε μια κλάση.

Precision είναι ο αριθμός των TP, διαιρώντας τον με τον αριθμό των FP συν τον αριθμό των TP, βρίσκουμε, δηλαδή, το ποσοστό επιτυχίας από τον αριθμό των στοιχείων που τοποθετήθηκαν σε μια κλάση.

Recall είναι ο αριθμός των TP, διαιρώντας τον με τον αριθμό των TP συν τον αριθμό των FN, βρίσκουμε, δηλαδή, το ποσοστό επιτυχίας από τον αριθμό των στοιχείων που τοποθετήθηκαν σωστά σε μια κλάση από το σύνολο των στοιχείων που ανήκουν σε αυτήν την κλάση.

Accuracy είναι το ποσοστό επιτυχίας από τα σωστά ταξινομημένα στοιχεία σε σχέση με όλα τα στοιχεία.

F1-Score είναι ο γεωμετρικός μέσος όρος του Precision και Recall. Όπως ο αριθμητικός μέσος όρος, έτσι και γεωμετρικός μέσος όρος η βαθμολογία F1 είναι μεταξύ του Precision και Recall. Αυτό σημαίνει ότι λαμβάνει υπόψη και τον τρόπο διανομής των δεδομένων.

4. Εργαλειοθήκες

- FuzzyR

Function	Description	Arguments	Value	Usage
fis.builder()	To build a one-output TSK FIS by automatically generating the input membership functions and the fuzzy rules	<p>x.range a vector/matrix as the range of input(s)</p> <p>input.num the number of inputs</p> <p>input.mf.num a list of the number of membership functions for all inputs</p> <p>input.mf.type designed for different membership function types, however, currently, 'T1' for gbellmf, else 'it2gbellmf'</p> <p>rule.num the number of rules</p> <p>rule.which selected rules to be used in the full rule list, for example, c(1,2,3) specify the first three rules</p> <p>defuzzMethod "default"</p> <p>params.ante parameter settings for initialising antecedent membership functions</p> <p>params.conse</p>	A Fis object.	<pre> fis.builder(x.range, input.num, input.mf.num, input.mf.type, rule.num = prod(input.mf.num), rule.which = NULL, defuzzMethod = "default", params.ante, params.conse) </pre>

		parameter settings for initialising consequent membership functions		
anfis.builder()	To build an ANFIS model from an existing FIS model	fis A fuzzy inference system model initialised by newfis	An ANFIS model.	anfis.builder(fis)
Plotmf()	Plots a 2D graph of all membership functions from the specified variable which must be part of a fis object.	<p>fis Requires an existing fis as an argument.</p> <p>varType Can be either 'input' or 'output', representing the type of variable.</p> <p>varIndex A numerical integer, representing the index of the input or output variable whose membership functions shall be plotted (base 1).</p> <p>xx primary inputs for extra lines</p> <p>timelimit for perturbation</p> <p>xlab X axis label using font, size and color</p> <p>ylab Y axis label, same font attributes as xlab</p> <p>main The main title (on top)</p>	A two dimensional graph displaying all the membership functions of a given variable	plotmf(fis, varType, varIndex, xx = NULL, timelimit = 0, xlab = NULL, ylab = NULL, main = NULL)
anfis.optimise()	To optimise the performance of a given ANFIS model by learning the parameters in L1 and L4.	<p>anfis The given ANFIS model</p> <p>data.trn The input and output data pairs as training data</p> <p>data.chk The input and output data pairs as checking (validation) data</p>	The optimised ANFIS model.	anfis.optimise(anfis, data.trn, data.chk = NULL, epoch.total = 100, stepsize = 0.1, rate.inc = 1.1, rate.dec = 0.9, method = c("gradient", "lse"), err.log = F, online = 0, lambda = 1,

		<p>epoch.total The total training epochs.</p> <p>stepsize The initial stepsize</p> <p>rate.inc increasing rate of the stepsize</p> <p>rate.dec decreasing rate of the stepsize</p> <p>method The learning algorithms for Layer 1 and Layer 4 respectively. default method=c("gradient", "lse")</p> <p>err.log T or F, the flag indicate whether to save the error log.</p> <p>online 0 – batch; 1 – online; 2 – semi-online lambda The forgetting rate for the LSE algorithm</p> <p>opt.by To optimise the ANFIS model by: err.opt – optimisation error; err.trn – training error; err.chk – checking (validation) error.</p> <p>err.trn.fix T or F. When KM defuzzification is used for IT2 ANFIS, err.trn is not equal to err.opt. Hence, this flag is used for users to choose whether to fix this issue. The default value is set to T for the compatibility with previous built IT2 models. For T1 ANFIS, this flag can be set to F for speed improvement</p>		<p>opt.by = "err.opt", err.trn.fix = T)</p>
--	--	--	--	--

anfis.eval()	To evaluate a ANFIS model with input data.	anfis The given ANFIS model input.stack The input data	The output of the anfis for given input data.	anfis.eval(anfis, input.stack)
---------------------	--	---	---	--------------------------------

Πίνακας 4.1 Εργαλειοθήκη FUZZYR

- FRBS

Function	Description	Arguments	Value	Usage
frbs.learn()	This is one of the central functions of the package. This function is used to generate/learn the model from numerical data using fuzzy rule-based systems.	data.train a data frame or matrix (m × n) of data for the training process, where m is the number of instances and n is the number of variables; the last column is the output variable. It should be noted that the training data must be expressed in numbers (numerical data). And, especially for classification tasks, the last column representing class names/symbols isn't allowed to have values 0 (zero). In the other words, the categorical values 0 should be replaced with other values. range.data a matrix (2 × n) containing the range of the data, where n is the number of variables, and first	The frbs-object	frbs.learn(data.train, range.data = NULL, method.type = c("WM"), control = list())

		<p>and second rows are the minimum and maximum values, respectively. It will be assigned as min/max of training data if it is omitted.</p> <p>method.type this parameter determines the learning algorithm to be used.</p> <p>control a list containing all arguments, depending on the learning algorithm to use.</p>		
predict.frbs()	This is the main function to obtain a final result as predicted values.	<p>object an frbs-object.</p> <p>newdata a data frame or matrix (m×n) of data for the prediction process, where m is the number of instances and n is the number of input variables. It should be noted that the testing data must be expressed in numbers (numerical data).</p>	The predicted values.	predict(object, newdata)

Πίνακας 4.2 Εργαλειοθήκη FRBS

5. Πρόβλεψη της Χρονοσειράς Mackey-Glass

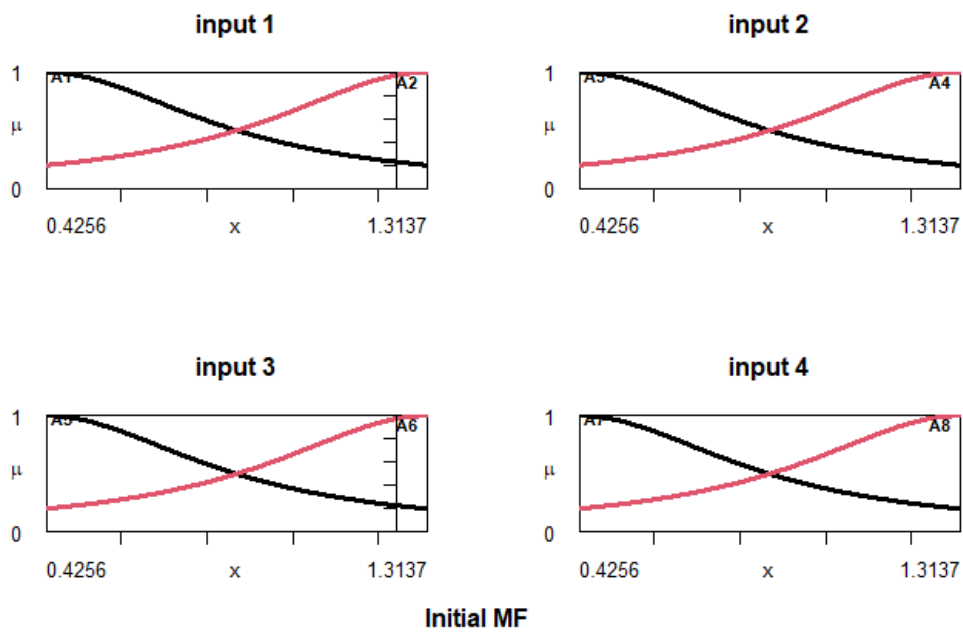
Η μοντελοποίηση δυναμικών μη-γραμμικών χρονοσειρών αποτελεί κεντρικό πρόβλημα σε πλήθος επιστημονικών περιοχών. Η χαοτική χρονοσειρά Mackey-Glass αποτελεί benchmark των διαφόρων μεθόδων πρόβλεψης και συναντάται πολύ συχνά στη βιβλιογραφία που ασχολείται με τα νευρο-ασαφή συστήματα. Η χρονοσειρά προκύπτει από την ακόλουθη διαφορική εξίσωση διαφορών

$$\dot{x}(t) = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t).$$

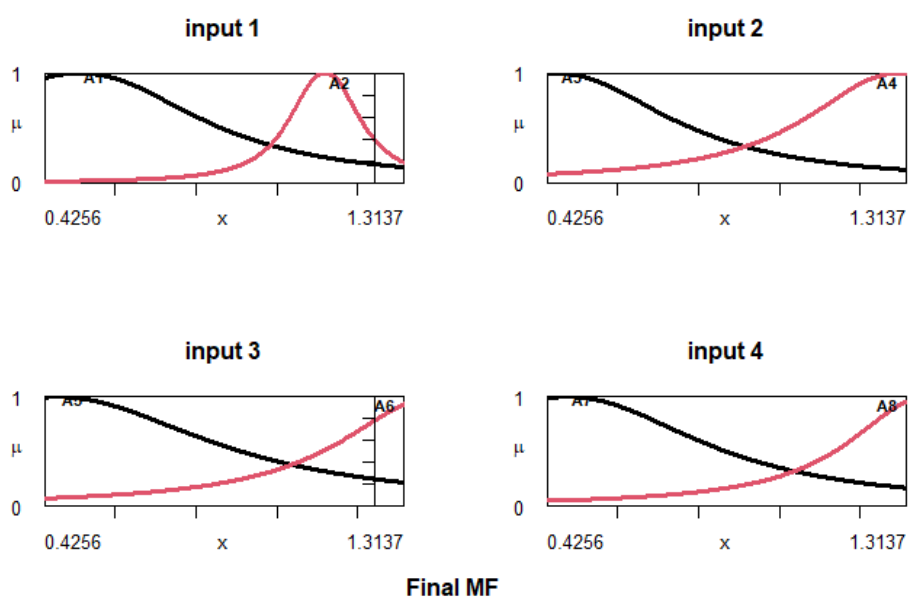
Το όνομα της πάρθηκε από τα ονόματα των δημιουργών της, Michael C. Mackey και Leon Glass. Ο βιομαθηματικός καθηγητής του πανεπιστημίου McGill Mackey ανέπτυξε πολλαπλά μαθηματικά μοντέλα περιγραφής, ασθενειών, γενετικών ρυθμίσεων σε όργανα και άλλες διεργασίες σε μοριακό, κυτταρικό και μυϊκό επίπεδο. Με την βοήθεια ενός συναδέλφου καθηγητή του ίδιου πανεπιστημίου Leon Glass ο οποίος συνέβαλε εξίσου στον χώρο της μαθηματικής ιατρικής, δημιούργησαν την διαφορική εξίσωση Mackey Glass. Η εξίσωση Mackey-Glass είναι μία μη γραμμική διαφορική εξίσωση χρονικής καθυστέρησης, όπου ανάλογα με τις τιμές των παραμέτρων, αυτή η εξίσωση εμφανίζει ένα εύρος περιοδικής και χαοτικής δυναμικής.

6. Πρόβλεψη Mackey-Glass με ANFIS

Το μοντέλο ANFIS χρησιμοποίησε 500 ζευγάρια από την χρονοσειρά, από τα οποία το 80% χρησιμοποιήθηκαν για την εκπαίδευση και το υπόλοιπο 20% για τον έλεγχο. Το σύστημα είναι τεσσάρων εισόδων, μιας εξόδου και οι ασαφής συναρτήσεις συμμετοχής ορίζονται σε 2 για κάθε είσοδο, οπότε έχουμε 16 ασαφή κανόνες.

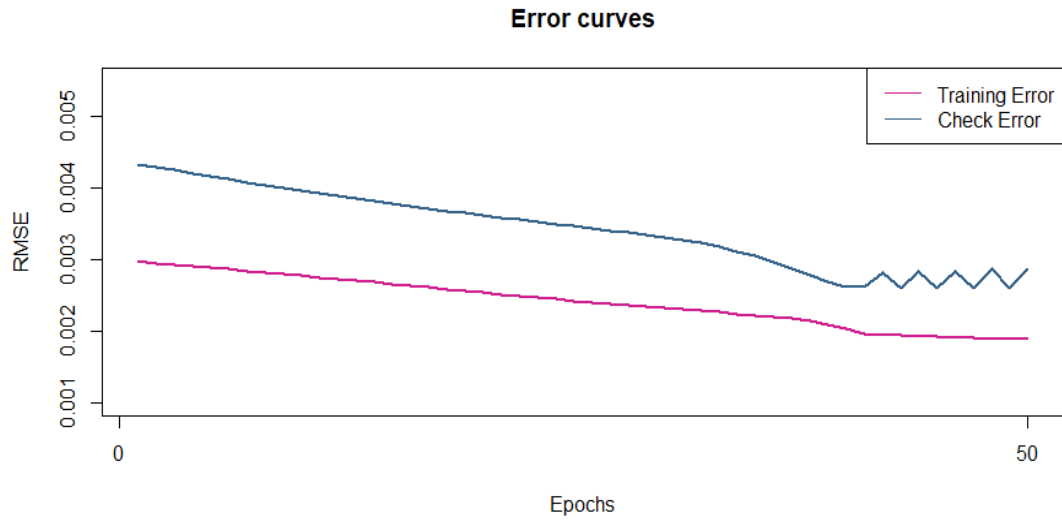


Εικόνα 6.1 Αρχικές Συναρτήσεις Συμμετοχής Mackey-Glass ANFIS



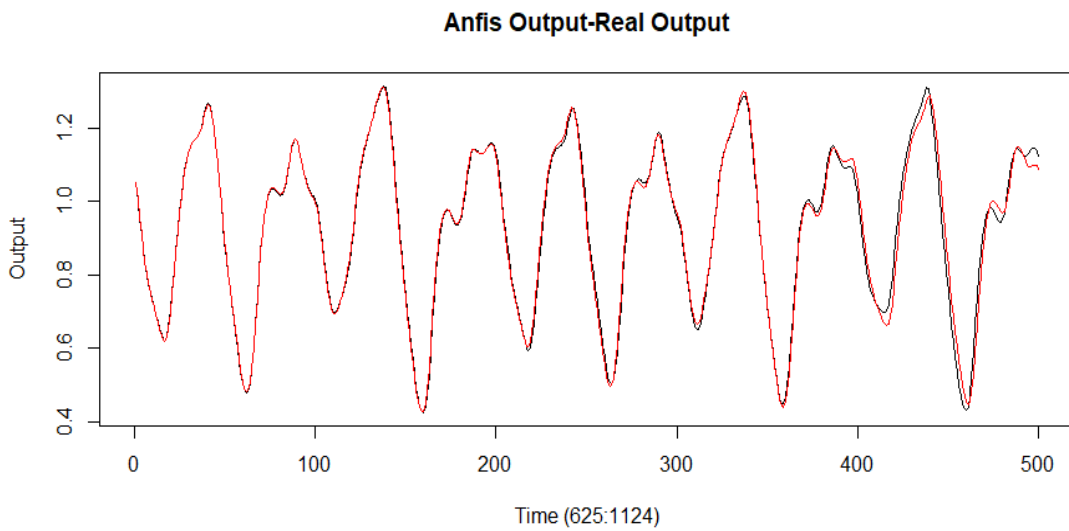
Εικόνα 6.2 Τελικές Συναρτήσεις Συμμετοχής Mackey-Glass ANFIS

Στο παρακάτω σχήμα βλέπουμε την καμπύλη του μέσου τετραγωνικού σφάλματος(RMSE) για ένα μοντέλο Anfis με 2 κανόνες σε διάστημα 50 εποχών, βήμα 0.01 και με τον υβριδικό τρόπο εκμάθησης.

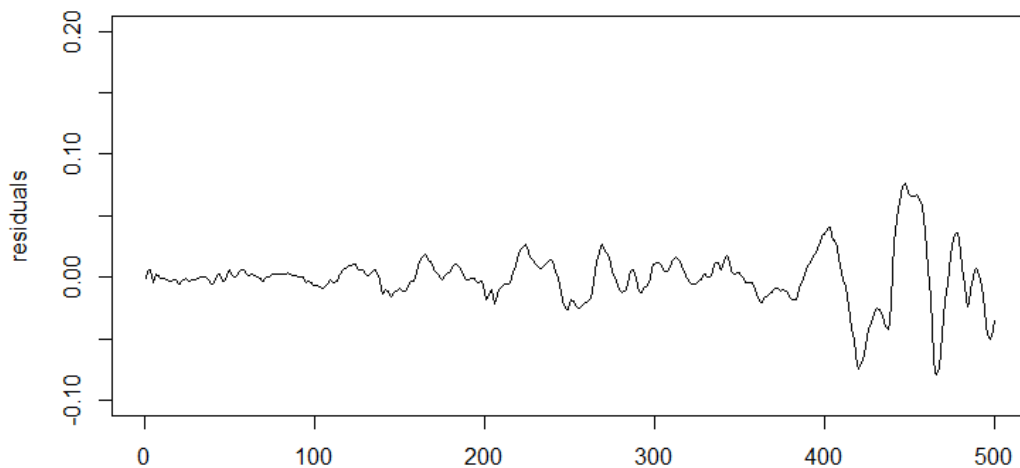


Εικόνα 6.3 Καμπύλη RMSE συνάρτηση εποχών Mackey-Glass ANFIS

Εν συνεχεία βλέπουμε την απόδοση της εξόδου του μοντέλου σε σχέση με την πραγματική έξοδο.



Εικόνα 6.4 Έξοδος ANFIS(κόκκινο) και Πραγματική Έξοδος (μαύρο) Mackey-Glass ANFIS (2 Κανόνες, 0.01 ρυθμός μάθησης).

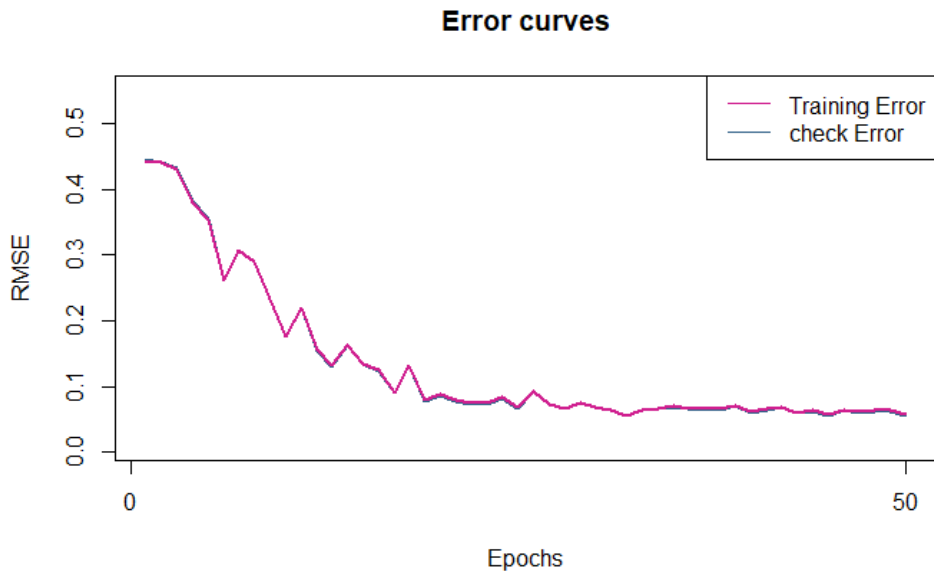


Εικόνα 6.5 Residuals - Διαφορά Μεταξύ Πραγματικής Εξόδου και Εξόδου ANFIS, Mackey-Glass

Τελικός έχουμε για το μοντέλο ANFIS με δυο συναρτήσεις συμμετοχής σε διάστημα 50 εποχών και ρυθμό εκμάθησης 0.01. Παρατηρούμε να εμφανίζει μέσω τετραγωνικό σφάλμα για τα δεδομένα εκπαίδευσης 0.00188146 και αντιστοίχως 0.0025942 στα δεδομένα ελέγχου. Το μοντέλο για αυτήν την υλοποίηση χρειάζεται 45.42 δευτερόλεπτα για να ολοκληρωθεί και να υπολογίσει.

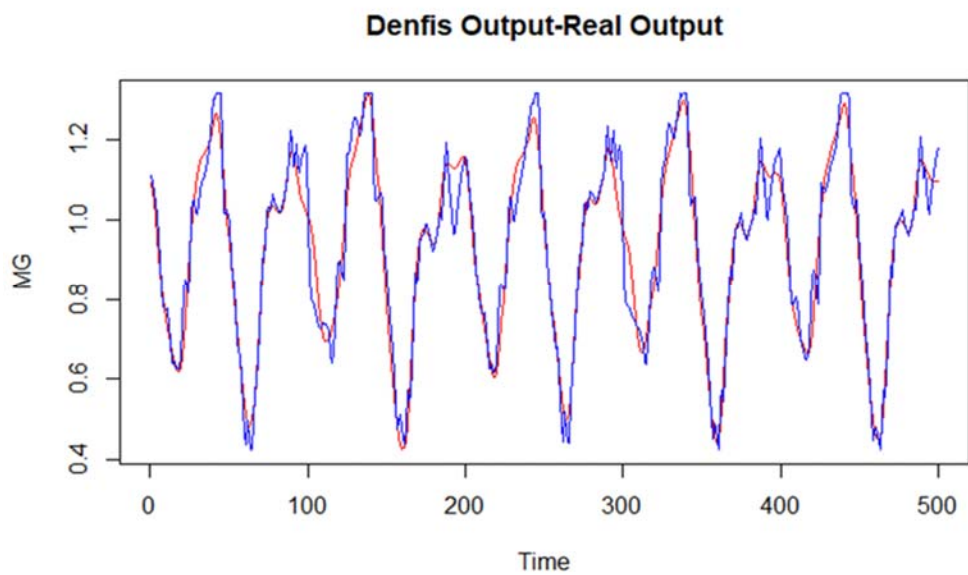
7. Πρόβλεψη Mackey-Glass με DENFIS

Το μοντέλο DENFIS χρησιμοποίησε 500 ζευγάρια από την χρονοσειρά για εκπαίδευση και για τον έλεγχο. Το σύστημα είναι τεσσάρων εισόδων, μιας εξόδου και δημιουργεί 22 κέντρα συστάδων.

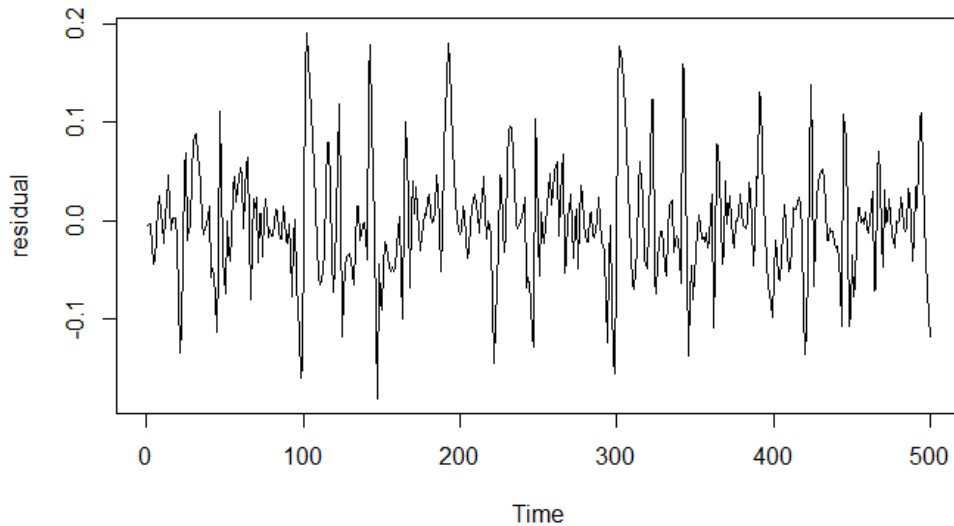


Εικόνα 7.1 Καμπύλη RMSE συνάρτηση εποχών Mackey-Glass DENFIS

Στο παραπάνω σχήμα βλέπουμε την καμπύλη του μέσου τετραγωνικού σφάλματος(RMSE) για ένα μοντέλο DENFIS σε διάστημα 50 εποχών, βήμα 0.01. Και στην συνέχεια πως αποτυπώνεται το σφάλμα συγκρίνοντας τα πραγματικά δεδομένα με την έξοδο του DENFIS.



Εικόνα 7.2 Έξοδος DENFIS(μπλέ) και Πραγματική Έξοδος (κόκκινο) Mackey-Glass DENFIS (0.01 ρυθμός μάθησης).



Εικόνα 7.3 Residuals - Διαφορά Μεταξύ Πραγματικής Εξόδου και Εξόδου DENFIS, Mackey-Glass

Το μοντέλο DENFIS με είκοσι δυο Κέντρα Συστάδων σε διάστημα 50 εποχών και ρυθμό εκμάθησης 0.01 παρατηρούμε να εμφανίζει μέσω τετραγωνικό σφάλμα για τα δεδομένα εκπαίδευσης 0.0562924 και αντίστοιχως 0.05496114 στα δεδομένα ελέγχου. Το μοντέλο για αυτήν την υλοποίηση χρειάζεται 10.93 δευτερόλεπτα για να ολοκληρωθεί και να υπολογίσει.

8. Συμπεράσματα για MACKEY-GLASS Dataset

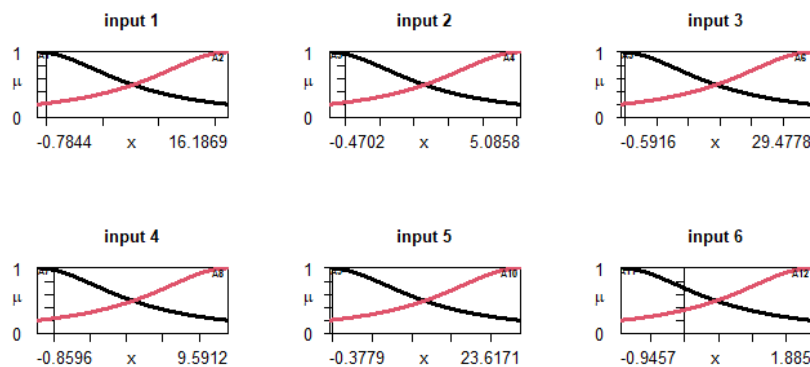
Έχοντας υλοποιήσει και εφαρμόσει δυο διαφορετικά μοντέλα για πρόβλεψη της χαοτικής χρονοσειράς Mackey-Glass, τα συμπεράσματα είναι θετικά. Ξεκινώντας με το μοντέλο ANFIS μπορούμε να παρατηρήσουμε ότι όχι μόνο κατάφερε να πιάσει την χρονοσειρά αλλά είναι και σε θέση να την προβλέψει και με μεγάλη ακρίβεια. Συγκεκριμένα τα δεδομένα εκμάθησης είχανε μέσω τετραγωνικό σφάλμα 0.00188146 και 0.0025942 τα δεδομένα ελέγχου. Τα νούμερα αυτά επιβεβαιώνουνε την επιτυχία του ANFIS. Το δεύτερο μοντέλο που χρησιμοποιήθηκε, το DENFIS, είχε εξίσου ενθαρρυντικά αποτελέσματα καταφέροντάς να ακολουθήσει πιστά την χρονοσειρά έχοντας όμως χαμηλότερη απόδοση από το προηγούμενο μοντέλο. Μιλώντας ποιο συγκεκριμένα, παρατηρήθηκε μέσω τετραγωνικό σφάλμα 0.0562924 στα δεδομένα εκμάθησης και 0.0549611 στα δεδομένα ελέγχου. Τα αποτελέσματα αυτά είναι σαφώς χειρότερα, παρόλα αυτά το DENFIS χρειάστηκε 10.93 δευτερόλεπτα σε αντίθεση με το ANFIS που χρειάζεται 45.42, δηλαδή σχεδόν 4 φορές πιο γρήγορο, ωστόσο το RMSE του DENFIS είναι σχεδόν 20 φορές χειρότερο αντιπαραβάλλοντας το άλλο μοντέλο. Συμπερασματικά καταλαβαίνουμε ότι την χαοτική χρονοσειρά Mackey-Glass είμαστε σε θέση να την προβλέψουμε με μεγάλη ακρίβεια και σίγουρα να την ακολουθήσουμε πιστά και με τα δύο μοντέλα.

9. Μαστογραφία- Mammography

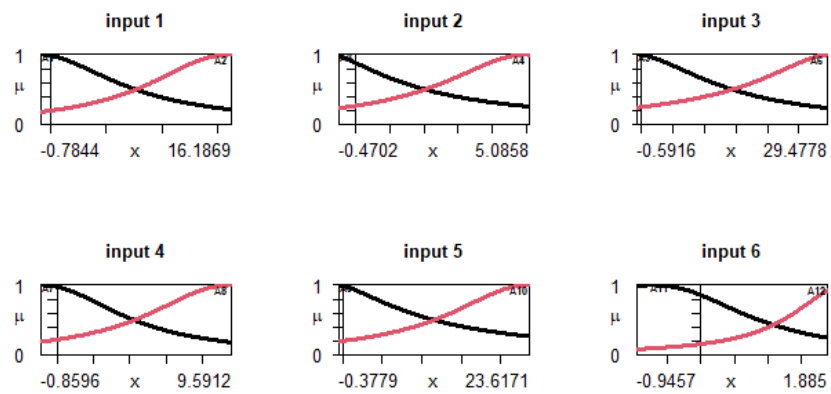
Μαστογραφία ονομάζεται ακτινολογική διαγνωστική εξέταση που χρησιμοποιείται με στόχο την ανίχνευση καρκίνου του μαστού. Χρησιμοποιείται σχετικά μικρή δόση ακτίνων X προκειμένου να διερευνηθούν πιθανές παθολογικές αλλοιώσεις του μαστού. Τα φωτόνια που χρησιμοποιούνται είναι σχετικά μικρών ενεργειών για να απεικονίσουν πιο έντονα τις αλλαγές στους ιστούς και το λίπος. Επίσης ο στόχος που χρησιμοποιείται στη μαστογραφία είναι από μολυβδένιο και όχι από βολφράμιο, όπως στην ακτινογραφία. Η αξία της μαστογραφίας στην αντιμετώπιση του καρκίνου του μαστού έχει αποδειχτεί ανεκτίμητη. Μια μαστογραφία μπορεί να εντοπίσει αναπτυσσόμενους καρκινικούς όγκους που θα γίνονταν αντιληπτοί και ψηλαφητοί από την ασθενή ή το γιατρό περίπου δύο χρόνια αργότερα.

10. Εφαρμόζοντας το ANFIS σε Μαστογραφίες

Το μοντέλο ANFIS χρησιμοποίησε 4473 μαστογραφίες για την εκπαίδευση και 6710 για τον έλεγχο. Το σύστημα είναι 6 εισόδων, μιας εξόδου και οι ασαφείς συναρτήσεις ορίζονται σε 2 για κάθε είσοδο με συνέπεια να έχουμε 64 ασαφή κανόνες. Ο χρόνος που χρειάστηκε να ολοκληρωθεί ήταν 23.1 λεπτά για διάστημα 50 εποχών και βήμα εκμάθησης 0.01. Οι αρχικές και τελικές συναρτήσεις συμμετοχής απεικονίζονται παρακάτω.

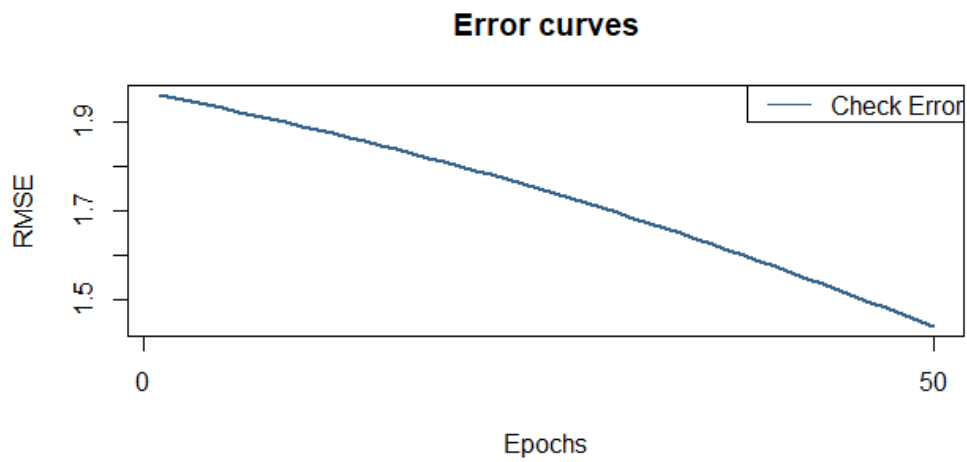


Εικόνα 10.1 Αρχικές Συναρτήσεις Συμμετοχής Mammography ANFIS

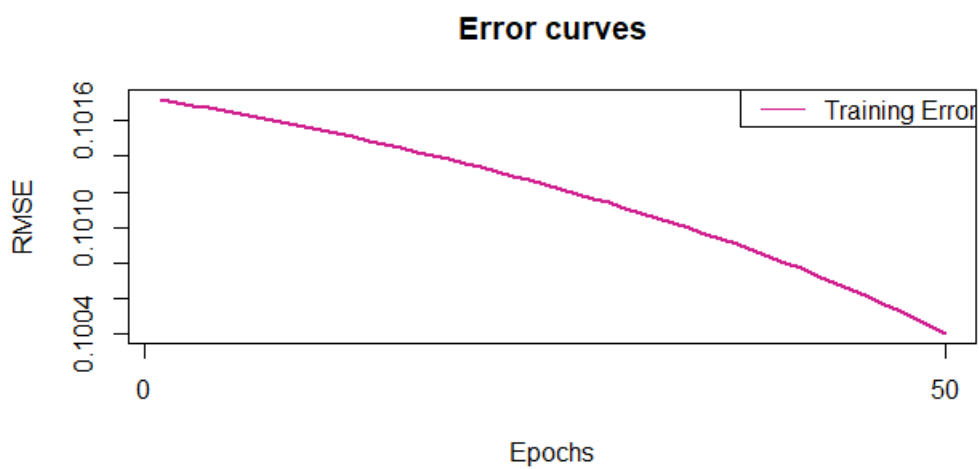


Εικόνα 10.2 Τελικές Συναρτήσεις Συμμετοχής Mammothography ANFIS

Απεικόνιση των σφαλμάτων σε διάστημα 50 εποχών.

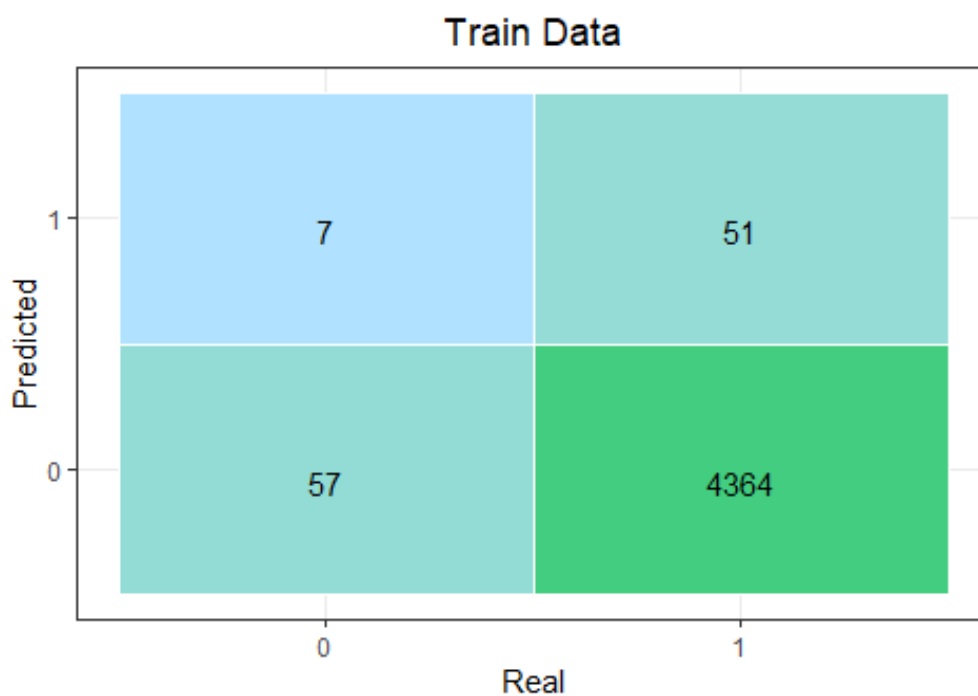


Εικόνα 10.3 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα ελέγχου Mammothography

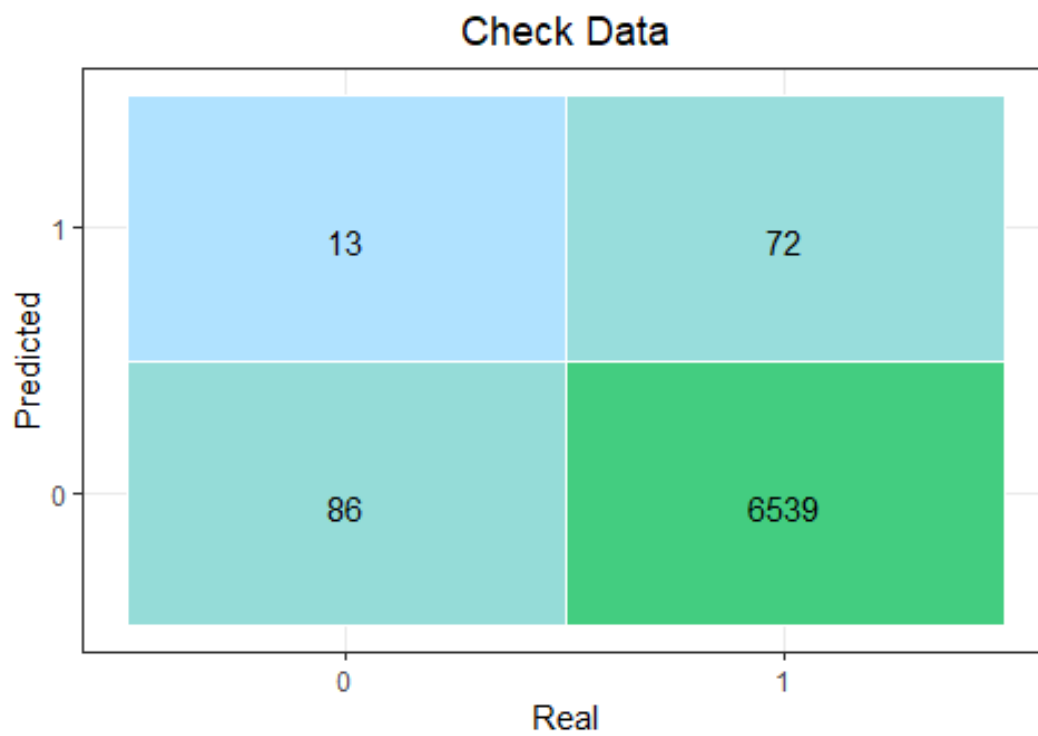


Εικόνα 10.4 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα εκπαίδευσης Mammography

Απεικόνιση των Confusion Matrix των δεδομένων εκπαίδευσης και ελέγχου.



Εικόνα 10.5 Confusion Matrix ANFIS των δεδομένων εκπαίδευσης Mammography.



Εικόνα 10.6 Confusion Matrix ANFIS των δεδομένων ελέγχου Mammography.

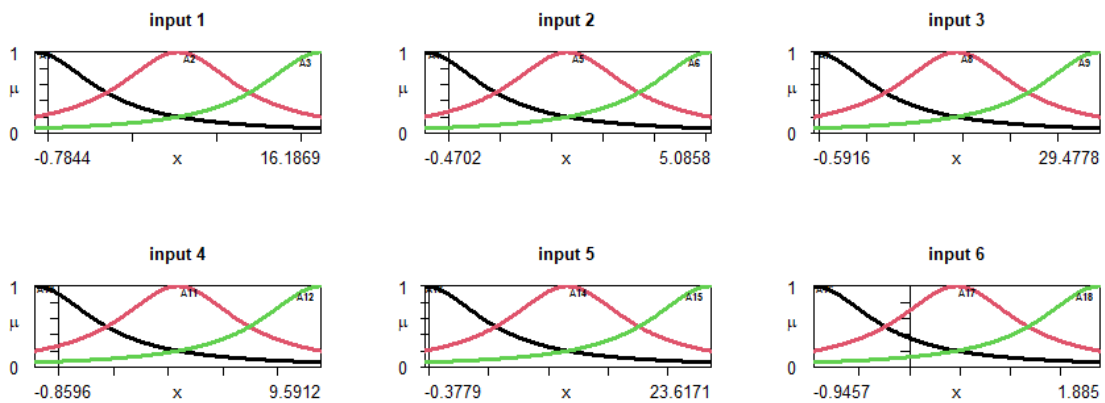
- Πίνακας Αποτελεσμάτων του ANFIS σε 50 και 30 εποχές

Εποχές		50		30
FN_check		86		85
FP_check		13		17
TP_check		72		73
TN_check		6539		6535
FN_train		57		55
FP_train		7		7
TP_train		51		50
TN_train		4364		4364
F1 score_check		0.58333		0.5887
Accuracy_check		0.98509		0.9847
Recall_check		0.44303		0.462
Precision_check		0.8536		0.8111
F1 score_train		0.6144		0.6172
Accuracy_train		0.9857		0.9861
Recall_train		0.4722		0.4761
Precision_train		0.8793		0.8771
RMSE_check		1.57406		1.76488
RMSE_train		0.100447		0.10171

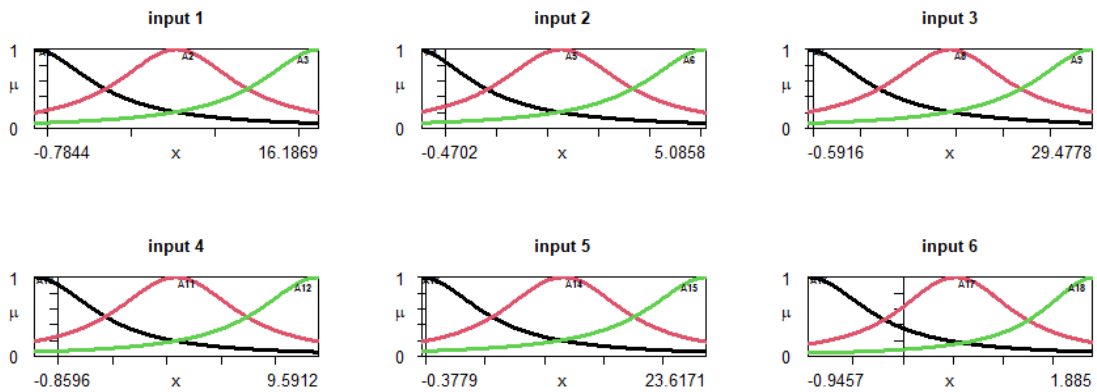
Πίνακας 11.1 Αποτελέσματα ANFIS με 2 ασαφείς συναρτήσεις στο Mammography dataset

11. Δεύτερη Υλοποίηση ANFIS

Το μοντέλο ANFIS χρησιμοποίησε 4473 μαστογραφίες για την εκπαίδευση και 6710 για τον έλεγχο. Το σύστημα είναι 6 εισόδων, μιας εξόδου και οι ασαφείς συναρτήσεις ορίζονται σε 3 για κάθε είσοδο με συνέπεια να έχουμε 729 ασαφείς κανόνες, αυξάνοντας την πολυπλοκότητα. Ο χρόνος που χρειάστηκε το μοντέλο για να ολοκληρωθεί ήταν 3.5 ώρες για διάστημα 50 εποχών και 1.4 ώρα για διάστημα 30 εποχών με βήμα εκμάθησης 0.01. Οι αρχικές και τελικές συναρτήσεις συμμετοχής απεικονίζονται παρακάτω.

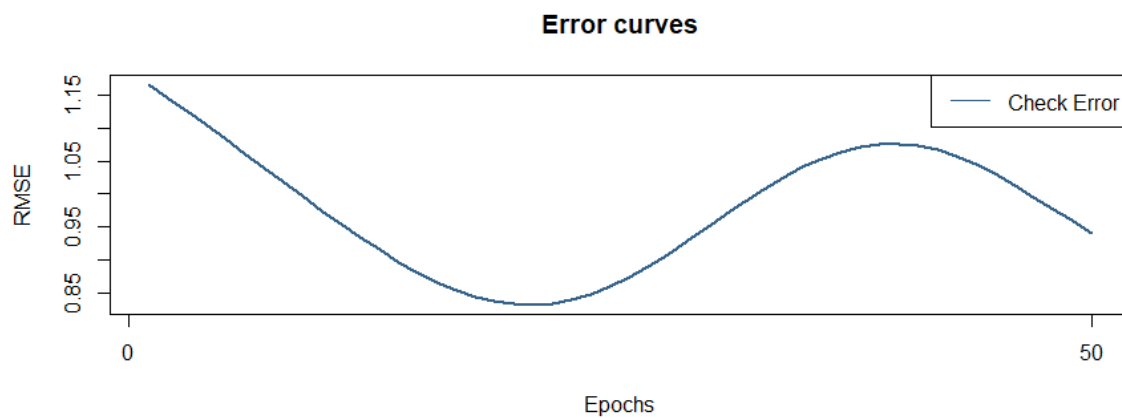


Εικόνα 11.1 Αρχικές Συναρτήσεις Συμμετοχής MF3 ANFIS



Εικόνα 11.2 Τελικές Συναρτήσεις Συμμετοχής MF3 ANFIS

Απεικόνιση των σφαλμάτων σε διάστημα 50 εποχών.

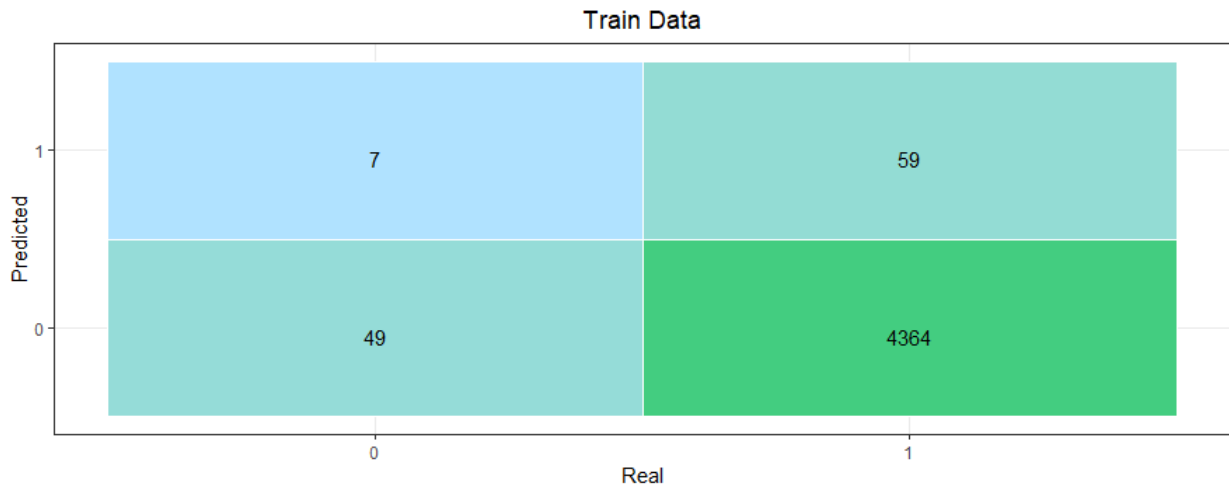


Εικόνα 11.3 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα ελέγχου Mammography MF3

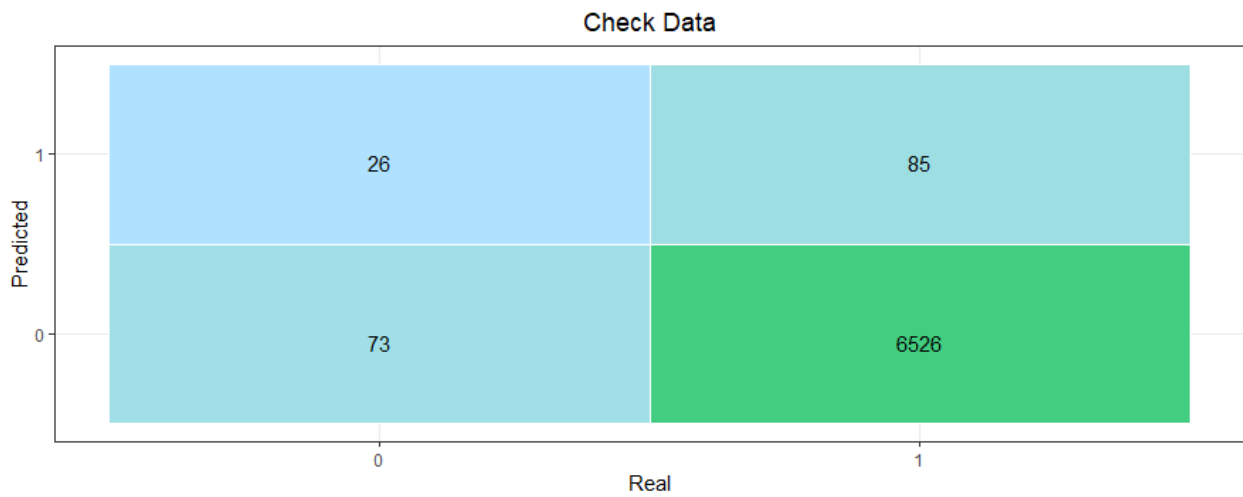


Εικόνα 11.4 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα εκπαίδευσης Mammography MF3

Απεικόνιση των Confusion Matrix των δεδομένων εκπαίδευσης και ελέγχου



Εικόνα 11.5 Confusion Matrix των δεδομένων εκπαίδευσης ANFIS Mammography MF3.



Εικόνα 11.6 Confusion Matrix των δεδομένων ελέγχου ANFIS Mammography MF3.

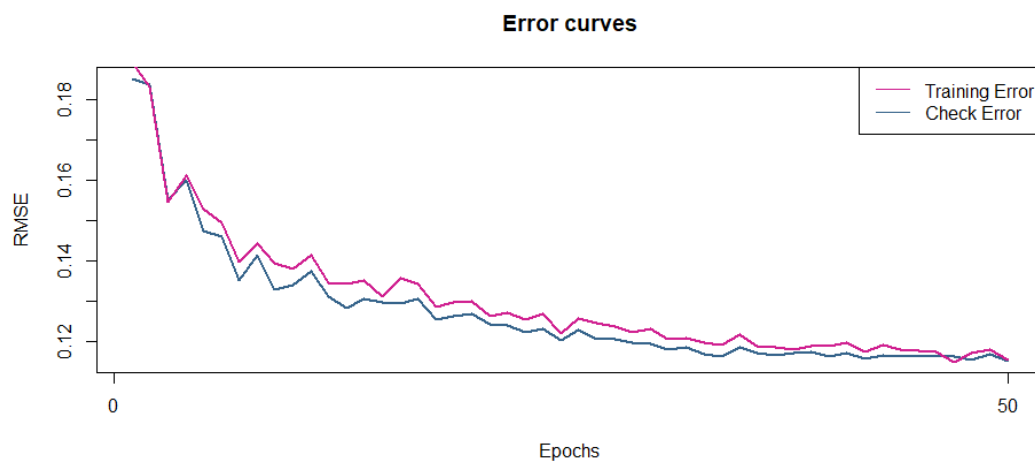
- Πίνακας Αποτελεσμάτων του ANFIS σε 50 και 30 εποχές

Εποχές		50		30
FN_check		73		75
FP_check		26		24
TP_check		85		83
TN_check		6526		6528
FN_train		49		49
FP_train		7		7
TP_train		59		57
TN_train		4364		4364
F1 score_check		0.6319		0.6264
Accuracy_check		0.9852		0.9852
Recall_check		0.5379		0.5253
Precision_check		0.7657		0.7757
F1 score_train		0.6781		0.6705
Accuracy_train		0.9874		0.9874
Recall_train		0.5462		0.5377
Precision_train		0.8939		0.8906
RMSE_check		0.831468		0.95353
RMSE_train		0.091827		0.09417

Πίνακας 12.1 Αποτελέσματα ANFIS με 3 ασαφείς συναρτήσεις στο Mammography dataset

12. Εφαρμόζοντας το DENFIS σε Μαστογραφίες

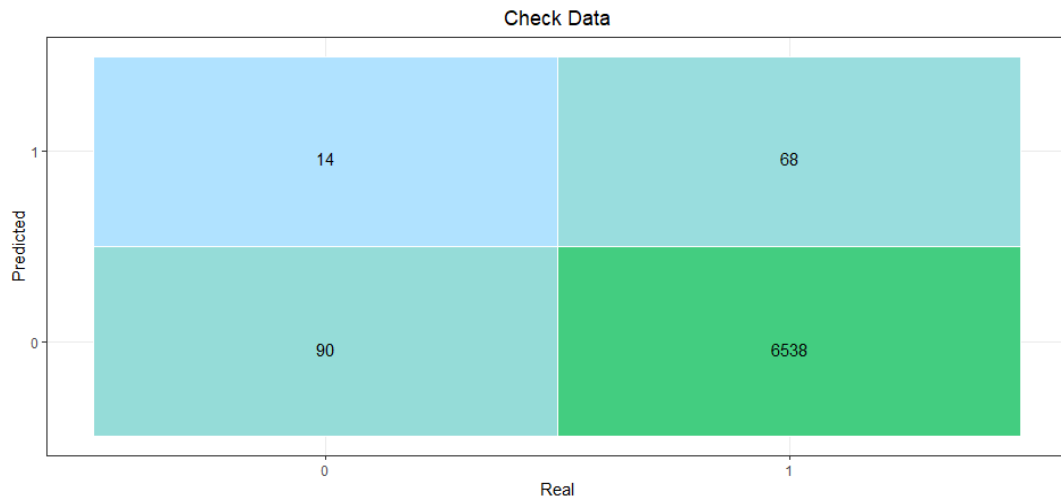
Το μοντέλο DENFIS χρησιμοποίησε 4473 μαστογραφίες για την εκπαίδευση και 6710 για τον έλεγχο. Το σύστημα είναι 6 εισόδων, μιας εξόδου και τα κέντρα συστάδων που δημιουργούνται είναι 28. Ο χρόνος που χρειάστηκε να ολοκληρώσει ήταν 2.38 λεπτά σε διάστημα 50 εποχών και βήμα εκμάθησης 0.01.



Εικόνα 12.1 Καμπύλη RMSE συνάρτηση εποχών DENFIS Mammography.



Εικόνα 12.2 Confusion Matrix DENFIS των δεδομένων εκπαίδευσης Mammography.



Εικόνα 12.3 Confusion Matrix DENFIS των δεδομένων ελέγχου Mammography.

- Πίνακας Αποτελεσμάτων του DENFIS σε 100, 50 και 30 εποχές

Εποχές	50	30	100
FN_check	89	87	92
FP_check	18	22	15
TP_check	69	71	66
TN_check	6534	6530	6537
FN_train	58	56	60
FP_train	15	19	14
TP_train	4	47	46
TN_train	4356	4352	4357
F1 score_check	0.5632	0.5657	0.5523
Accuracy_check	0.9840	0.9837	0.984
Recall_check	0.4367	0.4493	0.4177
Precision_check	0.7931	0.7634	0.8148
F1 score_train	0.7401	0.5562	0.5542
Accuracy_train	0.9925	0.9832	0.9834
Recall_train	0.7230	0.4563	0.4339
Precision_train	0.7580	0.7121	0.7666
RMSE_check	0.11518	0.1194	0.1151
RMSE_train	0.11496	0.12152	0.1148

Πίνακας 13.1 Αποτελέσματα DENFIS στο Mammography dataset

13. Συμπεράσματα για Μαστογραφίες – Mammography Dataset

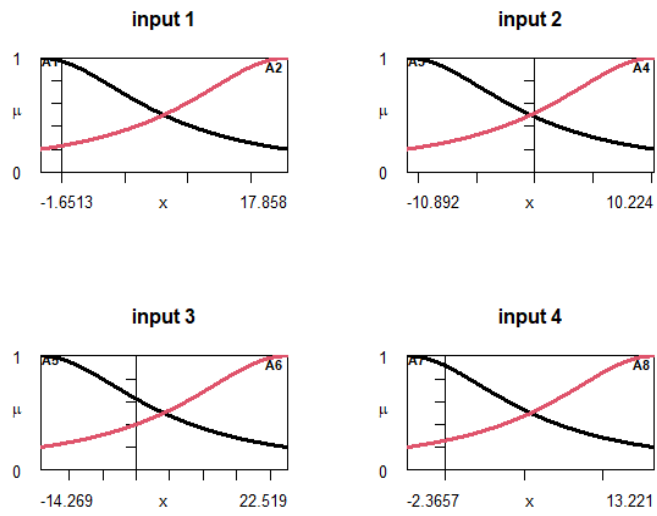
Υλοποιώντας δυο διαφορετικά μοντέλα σε ένα πρόβλημα ταξινόμησης, τα μοντέλα ANFIS DENFIS, τα αποτελέσματα εκ πρώτης άποψης θα μπορούσαμε να πούμε ότι δεν είναι ενθαρρυντικά, αφού επιτύχαμε F1-score 0.58 και 0.56 αντίστοιχα (ANFIS-DENFIS) για τα δεδομένα ελέγχου. Αντιθέτως το Accuracy έχει μια πολύ υψηλή τιμή 0.985 και 0.984, χωρίς όμως να είναι αντιπροσωπευτικό το αποτέλεσμα, καθώς η κατανομή των αποτελεσμάτων από το dataset (0,1) δεν έχουν ισόποσο πλήθος. Όσο για το ANFIS δοκιμάστηκε εξίσου μια περαιτέρω παραμετροποίηση στις εισόδους, αντί για δυο, τρεις συναρτήσεις συμμετοχής με μικρή βελτίωση στα αποτελέσματα και με χρόνο ολοκλήρωσης 17.5 μεγαλύτερο. Είναι σημαντικό να αναφερθεί ο δείκτης Precision, οποίος είναι 0.8536 και 0.7931 αντίστοιχα. Αυτό συνδυάζοντας το με τον δείκτη recall 0.44 και 0.43, παρατηρούμε ότι από τα 100 περιπτώσεις βρίσκουν σχεδόν τις 44 με αποτελεσματικότητα 85 και 79 τοις εκατό. Καταλαβαίνουμε ότι ένας ειδικός αν χρησιμοποιήσει αυτά τα μοντέλα στον πραγματικό κόσμο θα μπορέσει να λάβει μια πολύ γρήγορη συμβουλευτική άποψη, καθώς αν βγει θετικό το αποτέλεσμα αυτό σημαίνει ότι είναι 80 τοις εκατό σίγουρο ότι ισχύει. Παρόλα αυτά το κόστος για λάθος σε θέματα υγείας είναι υψηλό και δεν πρέπει σε καμία περίπτωση να χρησιμοποιούνται αυτά τα μοντέλα στην θέση του ειδικού, αλλά μόνο για συμβουλευτικό σκοπό.

14. Ανάλυση Πρωτευουσών Συνιστωσών (Principal Components Analysis, PCA)

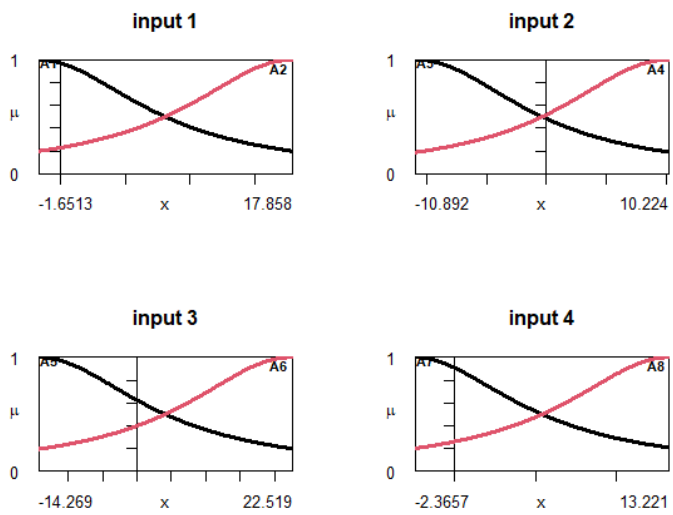
Η PCA έχει ως σκοπό τον εντοπισμό της δομής εξάρτησης που συνδέει τις παρατηρήσεις μίας πολυμεταβλητής διαδικασίας, ώστε να πετύχουμε μια πλήρη περιγραφή του συστήματος που εξετάζουμε. Η περιγραφή περιέχει n συνιστώσες, ενώ κανονικά για την περιγραφή απαιτούνται m συνιστώσες. Η διάσταση n ονομάζεται επιφανειακή (superficial dimensionality) ενώ η m ονομάζεται εσωτερική διάσταση (intrinsic dimensionality). Η PCA επιδιώκει την εκπλήρωση του αξιώματος του φειδωλού μοντέλου, σύμφωνα με το οποίο όσο απλούστερο είναι το μαθηματικό μοντέλο ενός συστήματος τόσο αποδοτικότερη είναι η περιγραφή της δομής εξάρτησης. Και τελικώς έχει ως στόχο την ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος ανακατασκευής.

15. Εφαρμόζοντας το ANFIS σε PCA MAMMOGRAPHY-Μαστογραφίες

Το μοντέλο ANFIS χρησιμοποίησε 4473 μαστογραφίες για την εκπαίδευση και 6710 για τον έλεγχο. Το σύστημα είναι 4 εισόδων, μιας εξόδου και οι ασαφείς συναρτήσεις ορίζονται σε 2 για κάθε είσοδο με συνέπεια να έχουμε 16 ασαφείς κανόνες. Ο χρόνος που χρειάστηκε να ολοκληρώσει ήταν 7.71 λεπτά για διάστημα 50 εποχών και βήμα εκμάθησης 0.01. Οι αρχικές και τελικές συναρτήσεις συμμετοχής απεικονίζονται παρακάτω.

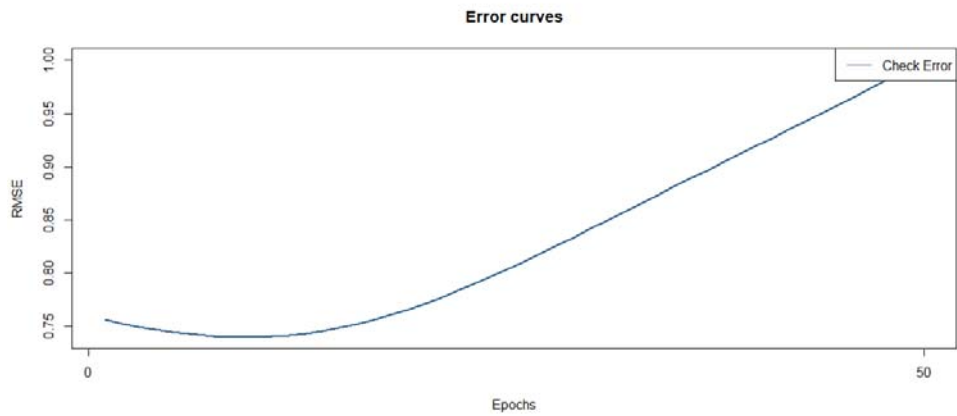


Εικόνα 15.1 Αρχικές Συναρτήσεις Συμμετοχής ANFIS PCA Mammography.

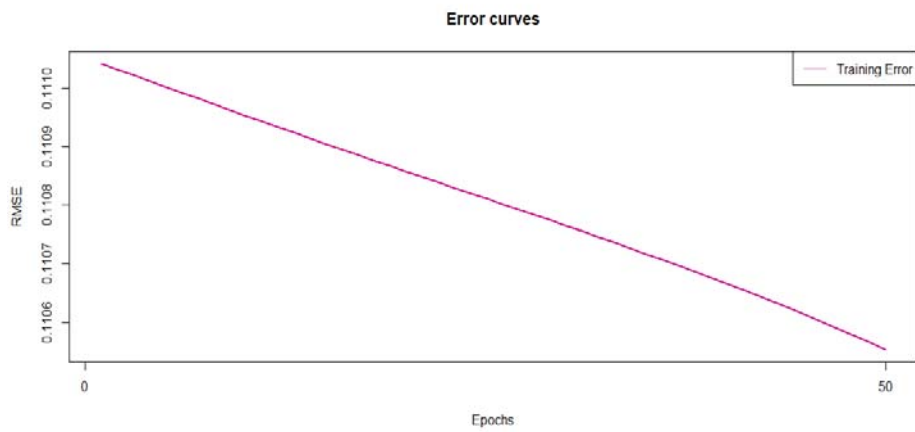


Εικόνα 15.2 Τελικές Συναρτήσεις Συμμετοχής ANFIS PCA Mammography.

Απεικόνιση των σφαλμάτων σε διάστημα 50 εποχών.

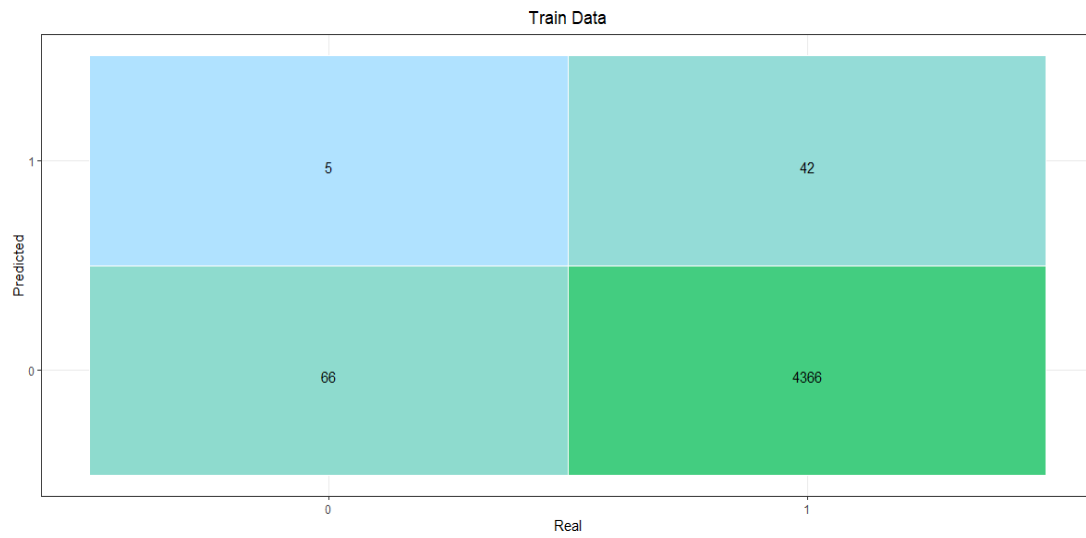


Εικόνα 15.3 Απεικόνιση RMSE για το μοντέλο ANFIS στα δεδομένα ελέγχου PCA Mammography

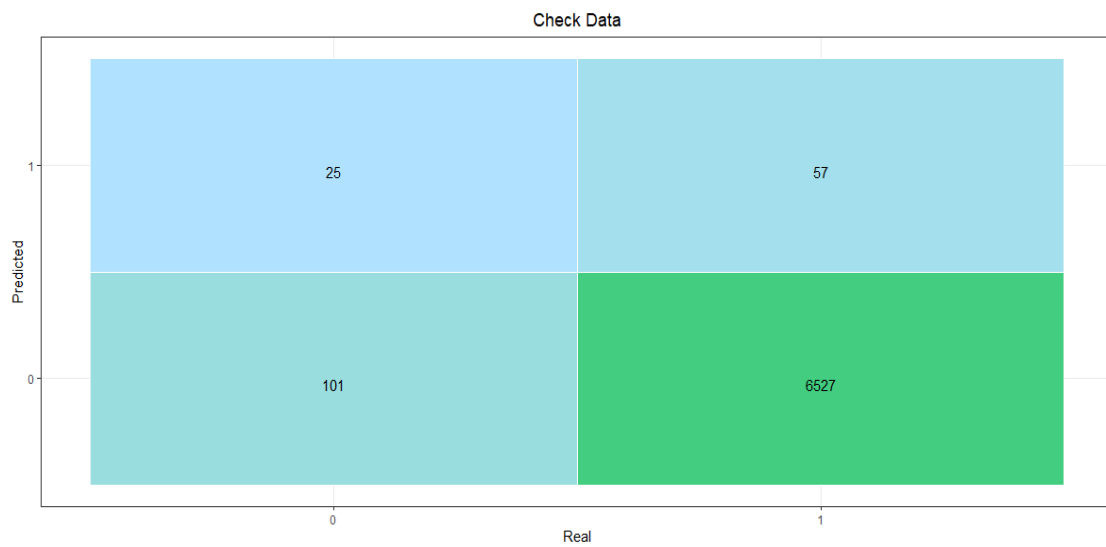


Εικόνα 15.4 Απεικόνιση RMSE για το μοντέλο ANFIS δεδομένα εκπαίδευσης PCA Mammography

Απεικόνιση των Confusion Matrix των δεδομένων εκπαίδευσης και ελέγχου.



Εικόνα 15.5 Confusion Matrix ANFIS των δεδομένων εκπαίδευσης PCA Mammography.



Εικόνα 15.6 Confusion Matrix ANFIS των δεδομένων ελέγχου PCA Mammography.

- Πίνακας Αποτελεσμάτων του ANFIS σε 50 και 30 εποχές

Εποχές		50		30
FN_check		101		101
FP_check		25		25
TP_check		57		57
TN_check		6527		6527
FN_train		66		66
FP_train		5		5
TP_train		42		42
TN_train		4366		4366
F1 score_check		0.475		0.475
Accuracy_check		0.9812		0.9812
Recall_check		0.3607		0.3607
Precision_check		0.6951		0.6951
F1 score_train		0.5419		0.5419
Accuracy_train		0.9841		0.9841
Recall_train		0.3888		0.3888
Precision_train		0.8936		0.8936
RMSE_check		0.739822		0.739822
RMSE_train		0.110552		0.110765

Πίνακας 16.1 Αποτελέσματα ANFIS στο PCA Mammography dataset

16. Εφαρμόζοντας το DENFIS σε PCA MAMMOGRAPHY-Μαστογραφίες

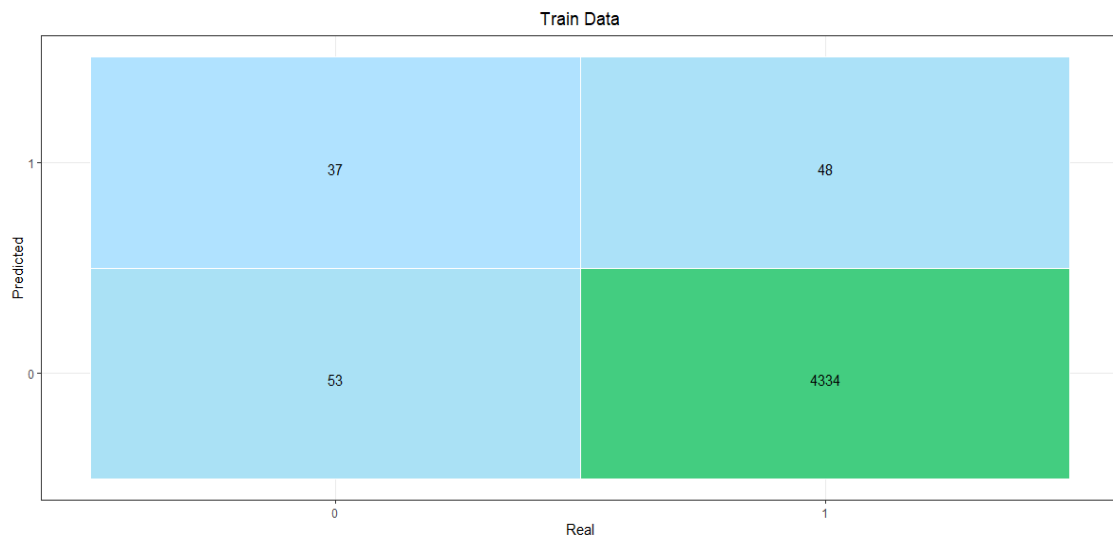
Το μοντέλο DENFIS χρησιμοποίησε 4473 μαστογραφίες για την εκπαίδευση και 6710 για τον έλεγχο. Το σύστημα είναι τεσσάρων εισόδων, μιας εξόδου και τα κέντρα συστάδων που δημιουργούνται είναι 13. Ο χρόνος που χρειάστηκε να ολοκληρωθεί ήταν 50.58 δευτερόλεπτα σε διάστημα 50 εποχών και βήμα εκμάθησης 0.01.

Απεικόνιση των σφαλμάτων σε διάστημα 50 εποχών.

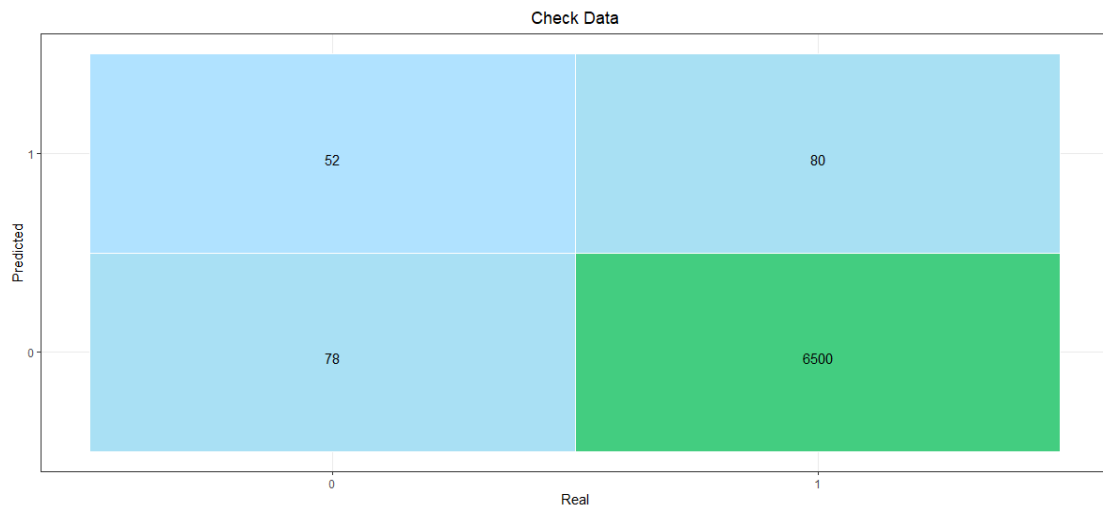


Εικόνα 16.1 Καμπύλη RMSE συνάρτηση εποχών DENFIS PCA Mammography.

Απεικόνιση των Confusion Matrix των δεδομένων εκπαίδευσης και ελέγχου.



Εικόνα 16.2 Confusion Matrix DENFIS των δεδομένων εκπαίδευσης PCA Mammography.



Εικόνα 16.3 Confusion Matrix DENFIS των δεδομένων ελέγχου PCA Mammography.

- Πίνακας Αποτελεσμάτων του DENFIS σε 50 και 30 εποχές

Εποχές		50		30
FN_check		78		84
FP_check		52		44
TP_check		80		74
TN_check		6500		6508
FN_train		53		56
FP_train		37		27
TP_train		48		46
TN_train		4344		4344
F1 score_check		0.5517		0.5362
Accuracy_check		0.9806		0.9809
Recall_check		0.5063		0.4683
Precision_check		0.606		0.6271
F1 score_train		0.5161		0.5257
Accuracy_train		0.9798		0.9814
Recall_train		0.4752		0.4509
Precision_train		0.5647		0.6301
RMSE_check		0.133578		0.133584
RMSE_train		0.134095		0.133259

Πίνακας 17.1 Αποτελέσματα DENFIS στο PCA Mammography dataset

17. Συμπεράσματα για PCA Μαστογραφίες – Mammography Dataset

Εφαρμόζοντας την στατιστική μέθοδο PCA στο dataset Μαστογραφίες-Mammography δεν λάβαμε τα επιθυμητά αποτελέσματα. Ενώ αυτή η μέθοδος διαπραγματεύεται να εμφανίσει καλύτερα scores με λιγότερες μεταβλητές άρα και μειωμένη πολυπλοκότητα, τελικώς είδαμε και στα δύο μοντέλα ANFIS-DENFIS χαμηλή αποδοτικότητα σε όλες τις μετρικές.

18. ΕΠΙΛΟΓΟΣ

Τα μοντέλα που αναπτύχθηκαν, το ANFIS και το DENFIS, εμφανίζουν αξιόλογα αποτελέσματα στα προβλήματα που τους τέθηκαν. Αρχικά δημιουργήθηκαν και εφαρμόστηκαν με επιτυχία στην πρόβλεψη χαοτικών χρονοσειρών, όπου το ANFIS καταφέρνει να την προβλέψει με μεγάλη ακρίβεια και το DENFIS με λιγότερη, αλλά έχοντας εξίσου αξιόλογες επιδόσεις, καθιστώντας τα δύο αυτά συστήματα, σε προβλήματα πραγματικού κόσμου, εργαλεία συνεργατικά και όχι αποκλειστικά. Εν συνεχεία, χρησιμοποιήθηκαν σε πρόβλημα ταξινόμησης (classification) και συγκεκριμένα σε Μαστογραφίες. Τα αποτελέσματα ήταν αρκετά ενδιαφέροντα, καθώς και τα δυο νευρο-ασαφή συστήματα βρίσκανε περίπου το 44% των θετικών περιπτώσεων και όταν το σύστημα είχε σαν έξοδο θετικό υπήρχε 80% πιθανότητα να αληθεύει. Ορίζοντας έτσι τα εργαλεία εξ αρχής, από μια γρήγορη άποψη ο ειδικός, θα πρέπει να κάνει της απαραίτητες ενέργειες κανονικά έχοντας ή μη αυτά τα μοντέλα. Για αυτόν το λόγο δοκιμάστηκε να αυξηθούν οι εποχές εκμάθησης, επηρεάζοντας εξ άλλου τον χρόνο εκτέλεσης σε μικρό βαθμό, χωρίς να παίρνουμε πιο ενθαρρυντικά αποτελέσματα. Επίσης αυξήσαμε την πολυπλοκότητα στο ANFIS χρησιμοποιώντας αντί για δύο συναρτήσεις συμμετοχής τρεις, εκτινάζοντας τον χρόνο εκτέλεσης, από 23 λεπτά σε 3.5 ώρες, και καταφέροντας ουσιαστικά να ανεβάσει από το 44% στο 53% την εύρεση θετικών περιπτώσεων, κάνοντας όμως περισσότερα λάθη, καθώς από αυτά που βγάζει θετικά το 76% αντί του 80% είναι σωστά. Στις Μαστογραφίες χρησιμοποιήθηκε και η ανάλυση πρωτευουσών συνιστωσών (PCA), για να εντοπιστεί η δομή εξάρτησης που συνδέει μια πολυμεταβλητή διαδικασία για να δημιουργηθεί ένα απλούστερο σύνολο δεδομένων με αποτέλεσμα να υπάρχει μεγαλύτερη αποδοτικότητα. Η διαδικασία αυτή δεν παρήγε τα αναμενόμενα, ολοκληρώνοντας στα 7 λεπτά, που είναι αρκετά γρηγορότερα, επιτυγχάνει χειρότερα αποτελέσματα, αλλά πολύ καλά για τον χρόνο που υλοποιείται. Περαιτέρω βελτιστοποιήσεις μπορούν να γίνουν για να αυξηθεί η αποδοτικότητα. Τα μοντέλα θα μπορούσαν να δοκιμαστούν και σε online μάθηση για να αξιολογηθεί η αποδοτικότητά τους. Εξίσου ενδιαφέρουσες θα ήταν οι υλοποιήσεις στον τομέα τραπεζική και διοίκηση. Αναπτύσσοντας ένα ANFIS το οποίο θα αποφασίζει, εάν θα δοθεί ένα δάνειο σε έναν πελάτη εξετάζοντας διάφορες παραμέτρους. Επίσης θα μπορούσε να δημιουργεί μέσω e-banking το προφίλ κάθε πελάτη αναλόγως των κινήσεων του εκάστοτε κατηγοριοποιώντας τον, που αυτό σαν συνέχεια δημιουργεί αρκετές εφαρμογές υλοποίησης, προσφέροντας καλύτερα προϊόντα και υπηρεσίες και για την τράπεζα προστιθέμενη αξία.

Βιβλιογραφία

- [1] Zadeh, L. A. Fuzzy sets. *Information and Control* 8 (1965), 338–353.
- [2] Zadeh, L. A. Quantitative fuzzy semantics. *Information Sciences* 3 (1971), 159–176.
- [3] Zadeh, L. A. Similarity relations and fuzzy orderings. *Information Sciences* 3 (1971), 177–200.
- [4] Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning-i. *Information Sciences* 8 (1975), 199–249.
- [5] Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning-ii. *Information Sciences* 8 (1975), 301–357.
- [6] Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning-iii. *Information Sciences* 9 (1975), 43–80.
- [7] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [8] L.-X. Wang and J. M. Mendel. Fuzzy basis function, universal approximation, and orthogonal least squares learning. *IEEE Trans. on Neural Networks*, 3(5):807–814, September 1992.
- [9] Thomas Wood, F-Score. [Ηλεκτρονικό]. Available: <https://deepai.org/machine-learning-glossary-and-terms/f-score>
- [10] Y. Sasaki, The truth of the F-measure (2007), December 15, 2019
- [11] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988
- [12] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 15:116–132, 1985.
- [13] T. Takagi and M. Sugeno. Derivation of fuzzy control rules from human operator’s control actions. *Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis*, pages 55–60, July 1983.
- [14] H. Takagi and I. Hayashi. NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, 5(3):191–212, 1991.
- [15] C.-T Sun and J.-S. Roger Jang. Adaptive network based fuzzy classification. In *Proc. of the Japan-U.S.A. Symposium on Flexible Automation*, July 1992.
- [16] C. Chen, J. M. Garibaldi, and T. Razak, “FuzzyR: Fuzzy Logic Toolkit for R,” <https://cran.r-project.org/package=FuzzyR>. January 18, 2021
- [17] L. S. Riza, C. Bergmeir, F. Herrera, and J. M. Benitez, “Fuzzy Rule-Based Systems for Classification and Regression Tasks,” <https://cran.r-project.org/package=frbs>. December 15, 2019
- [18] C. Wagner, S. Miller, and J. M. Garibaldi, “A fuzzy toolbox for the R programming language,” *Proceedings IEEE International Conference on Fuzzy Systems*, pp. 1185–1192, 2011.
- [19] J. Alcalá-Fdez and J. M. Alonso, “A Survey of Fuzzy Systems Software: Taxonomy, Current Research Trends, and Prospects,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 40–56, feb 2016.
- [20] R. Babuska, “Fuzzy Modelling and Identification,” *Control Engineering Laboratory, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands*, version, vol. 3, 2000.
- [21] J. M. Alonso and L. Magdalena, “Generating Understandable and Accurate Fuzzy Rule-Based Systems in a Java Environment,” in *Fuzzy Logic and Applications*, A. M. Fanelli, W. Pedrycz, and A. Petrosino, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 212–219.
- [22] C. Wagner, M. Pierfitt, and J. McCulloch, “Juzzy online: An online toolkit for the design, implementation, execution and sharing of Type-1 and Type-2 fuzzy logic systems,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2014, pp. 2321–2328.
- [23] C. Knott, L. Hovell, N. Karimian, and J. M. Garibaldi, “FuzzyToolkitUoN: Type 1 Fuzzy Logic Toolkit,” <https://CRAN.R-project.org/package=FuzzyToolkitUoN>, 2013.
- [24] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

- [25] S. Amari and N. Kasabov, Eds., *Brain-Like Computing and Intelligent Information Systems*. New York: Springer Verlag, 1997.
- [26] S. Amari, "Mathematical foundations of neuro-computing," *Proc. IEEE*, vol. 78, Sept. 1990.
- [27] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [28] J. Bezdek, Ed., *Analysis of Fuzzy Information*. Boca Raton, FL: CRC Press, 1987, vol. 3.
- [29] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden Day, 1970.
- [30] G. Carpenter and S. Grossberg, *Pattern Recognition by Self-Organizing Neural Networks*. Cambridge, MA: MIT Press, 1991
- [31] S. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intel. Fuzzy Syst.*, vol. 2, no. 3, Sept. 1994.
- [32] R. S. Croder III, "Predicting the Mackey-glass timeseries with cascade—Correlation learning," in *Proc. 1990 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. Pittsburgh, PA, 1990, pp. 117–123.
- [33] D. Deng and N. Kasabov, "Evolving self-organizing maps for online learning, data analysis and modeling," in *Proc. IJCNN'2000 Neural Networks Neural Computing: New Challenges Perspectives New Millennium*, vol. VI, S.-I. Amari, C. L. Giles, M. Gori, and V. Piuri, Eds. New York, 2000, pp. 3–8.
- [34] C. Fahlman and C. Lebiere, "The cascade—Correlation learning architecture," in *Advances in Neural Information Processing Systems*, D. Turetzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, vol. 2, pp. 524–532.
- [35] J. D. Farmer and J. J. Sidorowitch, "Predicting chaotic time series," *Phys. Rev. Lett.*, vol. 59, p. 845, 1987.
- [36] R. Goodman, C. M. Higgins, J. W. Miller, and P. Smyth, "Rule-based neural networks for classification and probability estimation," *Neural Comput.*, vol. 14, pp. 781–804, 1992.
- [37] T. Hashiyama, T. Furuhashi, and Y. Uchikawa, "A decision making model using a fuzzy neural network," in *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks*, Iizuka, Japan, 1992, pp. 1057–1060.
- [38] T. C. Hsia, *System Identification: Least-Squares Methods*. Boston, MA: D.C. Heath, 1977.
- [39] M. Ishikawa, "Structural learning with forgetting," *Neural Networks*, vol. 9, pp. 501–521, 1996.
- [40] R. Jang, "ANFIS: Adaptive network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 665–685, May 1993.
- [41] Kasabov, N. K. Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems* 10 (2002), 144–154.
- [42] Kasabov, N. K. Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems. *Fuzzy Sets and Systems* 82 (1996), 135–149.
- [43] Sugeno, M., and Kang, G. Structure identification of fuzzy model. *Fuzzy Sets and Systems* 28 (1988), 15–33.
- [44] N. Kasabov, "Adaptable connectionist production systems," *Neurocomputing*, vol. 13, no. 2–4, pp. 95–117, 1996.
- [45] N. Kasabov, "Evolving fuzzy neural networks for online, adaptive, knowledge-based learning," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 902–918, Dec. 2001. [38], "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems," *Fuzzy Sets Syst.*, vol. 82, no. 2, pp. 2–20, 1996.
- [46] N. Kasabov,, "ECOS: A framework for evolving connectionist systems and the eco learning paradigm," in *Proc. ICONIP'98*. Kitakyushu, Japan, Oct. 1998, pp. 1222–1235.
- [47] N. Kasabov, "Evolving fuzzy neural networks—Algorithms, applications and biological motivation," in *Methodologies for the Conception, Design and Application of Soft Computing*, T. Yamakawa and G. Matsumoto, Eds, Singapore: World Scientific, 1998, pp. 271–274.
- [48] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Cambridge, MA: MIT Press, 1996.
- [49] N. Kasabov, J. S. Kim, M. Watts, and A. Gray, "FuNN/2—A fuzzy neural network architecture for adaptive learning and knowledge acquisition," *Inform. Sci.—Appl.*, vol. 101, no. 3–4, pp. 155–175, 1997.

- [50] N. Kasabov and B. Woodford, "Rule insertion and rule extraction from evolving fuzzy neural networks: Algorithms and applications for building adaptive, intelligent expert systems," Proc. FUZZ-IEEE, Aug. 1999.
- [51] N. Kasabov, "Adaptive Learning System and Method," PCT WO01/78003, April 20, 2000.
- [52] S. Kawahara and T. Saito, "On a novel adaptive self-organizing network," Cellular Neural Networks Appl., pp. 41–46, 1996.
- [53] J. Mandziuk and L. Shastri, "Incremental Class Learning Approach and Its Applications to Hand-Written Digit Recognition," International Computer Science Institute, CA, TR-98-015, 1998.

19. Κώδικες

- Mackey-Glass ANFIS

```
require(FuzzyR)
require(ggplot2)
require(reshape2)
require(shiny)

data <- read.table("C:/Users/thoma/Desktop/THESIS/DATA/trnData.txt",
                  quote="\\"", comment.char="")
data<-as.matrix(data)

#split train-check
din <- dim(data)[1]
tl <- din*80/100 #train data length
cl <- din*10/100
tel <- din*10/100
train <- data[1:tl,] #train data
check <- data[(tl+1):(tl+cl),] #check data
#fit <- data[-(1:(tl+cl)),]
#fit<-as.matrix(fit)

#membership functions
input.num<-4
input.mf.num<-rep(2,input.num)

#rules
rule.num<-input.num^input.mf.num
rule.which<-sort(sample(1:prod(input.mf.num),rule.num))
```

```

#range
x.range <- matrix(0, ncol=2, nrow=input.num)
for(i in 1:input.num) {
  x.range[i,] <- range(train[,1:input.num][,i])
}

#Building the anfis model
fis<-fis.builder(      x.range,
                      input.num,
                      input.mf.num,
                      input.mf.type = 'T1',
                      rule.num      = rule.num,
                      rule.which    = rule.which)

#showGUI(fis)

epochs<-50
step<-0.01

anfis1<-anfis.builder(fis)

par(mfrow=c(2,2))

anfis.plotmf(anfis1,'input',1,main="input 1")
anfis.plotmf(anfis1,'input',2,main="input 2")
anfis.plotmf(anfis1,'input',3,main="input 3")
anfis.plotmf(anfis1,'input',4,main="input 4")
title("Initial MF", line = -26.5, outer = TRUE)

g <- rnorm(100000)
h <- rep(NA, 100000)

```

```

# Start the clock!
ptm <- proc.time()

anfis1<-anfis.optimise(anfis1, train, check,
                      epoch.total = epochs,
                      stepsize    = step,
                      rate.inc    = 1.1,
                      rate.dec    = 0.9,
                      err.log     = T,
                      lambda      = 1,
                      opt.by     = "err.chk",
                      online      = 0, # 0-Batch
                      method     = c("gradient", "lse"))

# Stop the clock
proc.time() - ptm

par(mfrow=c(2,2))
anfis.plotmf(anfis1, 'input', 1, main="input 1")
anfis.plotmf(anfis1, 'input', 2, main="input 2")
anfis.plotmf(anfis1, 'input', 3, main="input 3")
anfis.plotmf(anfis1, 'input', 4, main="input 4")
title("Final MF", line = -26.5, outer = TRUE)

rmse_trn <- err.trn[,2]
rmse_chk <- err.chk[,2]

from_which_epoch = 30

#plot error curves
ep <- from_which_epoch:epochs
par(mfrow=c(1,1))
plot(ep, rmse_chk[from_which_epoch:epochs], type='l', col="steelblue4", xlab="Epochs",
     ylab="RMSE", main="Error curves", lwd=2, ylim=c(0.001,0.081),xaxt='n')

```

```

axis(1, at=seq((from_which_epoch),epochs,by=(epochs-(from_which_epoch))), labels=T)
lines(ep, rmse_trn[from_which_epoch:epochs], lwd=2, col="violetred")
legend(x = "topright",
       col=c("violetred", "steelblue4"), lty = 1, lwd = 1,
       legend=c("Training Error", "Check Error"))

#evaluate
mgdata      <-      read.table("C:/Users/thoma/Desktop/THESIS/R_CODE/FUZZYR/mgdata.dat",
quote="\\"", comment.char="")
x<-mgdata[ ,2]
time<-mgdata[ ,1]
#plot(time, x, type='l')

anfis_out <- anfis.eval(anfis1, data[ ,1:4])
real <-x[625:1124]
residuals <- (real - anfis_out)
plot( type = "l", xlab="", residuals)
index = (1124-499):1124

# num_anfis_out = nrow(anfis_out)
# index = (1124 - num_anfis_out) : 1124

par(mfrow=c(1,1))
plot(time(index), anfis_out, type='l', main='Anfis Output-Real Output', xlab='Time
(625:1124)', ylab='Output')
lines(time(index), x[index], col='red')

min(rmse_chk)
min(rmse_trn)

```

- Mackey-Glass Denfis

```
g <- rnorm(100000)
h <- rep(NA, 100000)

# Start the clock!
ptm <- proc.time()

# Stop the clock
proc.time() - ptm

library(frbs)

trainFP <- "C:/Users/thoma/Desktop/THESIS/DATA/trnData.txt"
checkFP <- "C:/Users/thoma/Desktop/THESIS/DATA/chkData.txt"

my_data_train <- read.table(trainFP, header = FALSE)
my_data_check <- read.table(checkFP, header = FALSE)

## Define interval of data min max
range.data_train <- matrix(apply(my_data_train, 2, range), nrow = 2)

data.train <- my_data_train
data.fit <- data.train[,1:(ncol(data.train)-1)]
real.train <- matrix(my_data_train[, (ncol(my_data_train))], ncol=1)

data.check <- my_data_check[,1:(ncol(my_data_check)-1)]
real.val <- matrix(my_data_check[, (ncol(my_data_check))], ncol=1)

epochs <- 50
```

```

RMSE_check_root <- matrix()
RMSE_train_root <- matrix()
for(i in 50:epochs)
{
## Set the method and its parameters
  print(i)
control.DENFIS <- list(Dthr= 0.1, #the threshold value for the evolving clustering
method (ECM), between 0 and 1
                        max.iter = i,
                        step.size = 0.01,
                        d = 2,
                        name = "MG")

method.type <- "DENFIS"
## Generate fuzzy model
g <- rnorm(100000)
h <- rep(NA, 100000)

# Start the clock!
ptm <- proc.time()
object.DENFIS <- frbs.learn(data.train,
                           range.data_train,
                           method.type,
                           control.DENFIS)

# Stop the clock
proc.time() - ptm

## Predicting step
res.check <- predict(object.DENFIS, data.check)
res.train <- predict(object.DENFIS, data.train[,1:4])

## error calculation

## check
y.pred_check <- res.check

```



```

y.real_check <- real.val
residuals_check <- (y.real_check - y.pred_check)
RMSE_check_root[i] <- sqrt(mean(residuals_check^2))

##train
y.pred_train <- res.train
y.real_train <- real.train
residuals_train <- (y.real_train - y.pred_train)
RMSE_train_root[i] <- sqrt(mean(residuals_train^2))
}

#RMSE <- sqrt(mean(residuals^2))

#plot error curves

from_which_epoch = 30
ep <- from_which_epoch:epochs

plot(ep, RMSE_check_root[from_which_epoch:epochs], type='l', col="steelblue4",
      xlab="Epochs",
      ylab="RMSE", main="Error curves", lwd=2, ylim=c(0.000,0.0810),xaxt='n')

axis(1, at=seq(from_which_epoch,epochs,by=(epochs-from_which_epoch)), labels=T)
lines(ep, RMSE_train_root[from_which_epoch:epochs], lwd=2, col="violetred")
legend(x = "topright",
       col=c("violetred", "steelblue4"), lty = 1, lwd = 1,
       legend=c("Training Error", "Check Error"))

## Comparing between simulation and real data

bench <- cbind(real.val, res.check)
x2 <- seq(from = 1, to = nrow(bench))

plot(x2, bench[, 1], col="red", main = "Denfis Output-Real Output", type = "l",
      xlab='Time', ylab = "MG")
lines(x2, bench[, 2], col="blue", type = "l")

```

```
plot( type = "l", ylab="residual", xlab="Time", residuals_check)
```

```
summary(object.DENFIS)
```

```
min(RMSE_check_root)
```

```
min(RMSE_train_root)
```

- Mammography ANFIS

```

require(FuzzyR)
require(ggplot2)
require(reshape2)
require(shiny)
library(ggplot2)

dataTrain <-
read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
raining_dataset/mammo_trn_binary01.dat",
           quote="\\"", comment.char="")

dataCheck <-
read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
esting_dataset/mammo_tst_binary01.dat",
           quote="\\"", comment.char="")

dataTrain <- as.matrix(dataTrain)
dataCheck <- as.matrix(dataCheck)

train <- dataTrain #train data
check <- dataCheck #check data

allData <- (rbind(train, check))

#####TRUE- all the 1 ouput data#####
k=1
t1 <- matrix(ncol = 7)
for( i in 1:nrow(allData))
{
  if(allData[i,7] == 1)
  {
    t1 <- rbind(t1, allData[i,1:7])
    k = k+1
  }
}
t1 <- t1[-1,]
#####

```

```

#####FALSE- all the 0 output data#####
k=1
f1 <- matrix(ncol = 7)
for( i in 1:nrow(allData))
{
  if(allData[i,7] == 0)
  {
    f1 <- rbind(f1, allData[i,1:7])
    k = k+1
  }
}
f1 <- f1[-1,]
#####

#membership functions
input.num<-6
input.mf.num<-rep(2,input.num)

#rules
rule.num<-input.mf.num^input.num
rule.which<-sort(sample(1:prod(input.mf.num),rule.num)) #sample

#range
x.range <- matrix(0, ncol=2, nrow=input.num)
for(i in 1:input.num) {
  x.range[i,] <- range(train[,1:input.num][,i])
}

#Building the anfis model
fis<-fis.builder(x.range, input.num, input.mf.num,
                 input.mf.type = 'T1',
                 defuzzMethod="KM",
                 rule.num      = rule.num,
                 rule.which     = rule.which)

```

```

epochs <- 50
step <- 0.01

anfis1<-anfis.builder(fis)

par(mfrow=c(2,3))

#plotmf(fis,'input',1,main="input 1")

anfis.plotmf(anfis1,'input',1,main="input 1")
anfis.plotmf(anfis1,'input',2,main="input 2")
anfis.plotmf(anfis1,'input',3,main="input 3")
anfis.plotmf(anfis1,'input',4,main="input 4")
anfis.plotmf(anfis1,'input',5,main="input 5")
anfis.plotmf(anfis1,'input',6,main="input 6")

g <- rnorm(100000)
h <- rep(NA, 100000)

# Start the clock!
ptm <- proc.time()

# Loop through the vector, adding one

anfis1<-anfis.optimise(anfis1, train, check,
                      epoch.total = epochs,
                      stepsize     = step,
                      rate.inc     = 1.1,
                      rate.dec     = 0.9,
                      err.log      = T,
                      lambda       = 1,
                      opt.by       = "err.chk",
                      online       = 0,
                      method       = c("gradient", "lse"),
                      err.trn.fix = F)

```

```

# Stop the clock
proc.time() - ptm

#Here we propose a hybrid learning rule which combines the
#gradient method and the least squares estimate (LSE) to identify parameters

par(mfrow=c(2,3))

anfis.plotmf(anfis1, 'input', 1, main="input 1")
anfis.plotmf(anfis1, 'input', 2, main="input 2")
anfis.plotmf(anfis1, 'input', 3, main="input 3")
anfis.plotmf(anfis1, 'input', 4, main="input 4")
anfis.plotmf(anfis1, 'input', 5 ,main="input 5")
anfis.plotmf(anfis1, 'input', 6 ,main="input 6")

rmse_trn <- err.trn[,2]
rmse_chk <- err.chk[,2]

min(rmse_chk)
min(rmse_trn)

#which.min(rmse_chk) ## to check which epoch has the lowest rmse

#plot error curves
ep <- 1:epochs

par(mfrow=c(1,1))
##check curve plot
plot(ep, rmse_chk, type='l', col="steelblue4", xlab="Epochs",
      ylab="RMSE", main="Error curves" ,lwd=2,xaxt='n')
legend(x = "topright",
       col=c( "steelblue4"), lty = 1, lwd = 1,
       legend=c( "Check Error"))
axis(1, at=seq(0,epochs,by=50), labels=T)

```

```

##train curve plot
plot(ep, rmse_trn, type='l', col="violetred", xlab="Epochs",
      ylab="RMSE", main="Error curves" ,lwd=2,xaxt='n')
legend(x = "topright",
       col=c("violetred"), lty = 1, lwd = 1,
       legend=c( "Training Error"))
axis(1, at=seq(0,epochs,by=50), labels=T)

#####--CHECK--#####-##-#---#-
#####

# #evaluate

#anfis_out_TRUE <- anfis.eval(anfis1, t1[ ,1:input.num])## ??NFIS OUTPUT ONLY FOR THE
TRUE=1 OUTPUTS

#anfis_out_FALSE <- anfis.eval(anfis1, f1[ ,1:input.num])## ANFIS OUTPUT ONLY FOR THE
FALSE=0 OUTPUTS

anfis_out_check <- anfis.eval(anfis1, dataCheck[,1:input.num])## ANFIS OUTPUT FOR THE
CHECK DATA

plot(dataCheck[,7])

plot(anfis_out_check)
for(i in 1 : nrow(anfis_out_check))
{
  if(anfis_out_check[i] >= 0.5)

    anfis_out_check[i] <- 1
  else
  {
    anfis_out_check[i] <- 0
  }
}

```

```

TP_matrix <- matrix(ncol = 3)
FN_matrix <- matrix(ncol = 3)
FP_matrix <- matrix(ncol = 3)
TN_matrix <- matrix(ncol = 3)
TP = 0 #True positive
FN = 0 #False negative
FP = 0 #False positive
TN = 0 #True negative
for(i in 1 : nrow(anfis_out_check))
{
  #True Possitive
  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 1)
  {
    TP = TP + 1
    TP_matrix <- rbind(TP_matrix, c(i, 1, 1))
  }
  #False Negative
  if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 1)
  {
    FN = FN + 1
    FN_matrix <- rbind(FN_matrix, c(i, 0, 1))
  }
  #False Positive
  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 0)
  {
    FP = FP + 1
    FP_matrix <- rbind(FP_matrix, c(i, 1, 0))
  }

  #True Negative
  if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 0)
  {
    TN = TN + 1
    TN_matrix <- rbind(TN_matrix, c(i, 0, 0))
  }
}

```



```

}
#we delete the first row because is NA from the init
TP_matrix <- TP_matrix[-1,]
FN_matrix <- FN_matrix[-1,]
FP_matrix <- FP_matrix[-1,]
TN_matrix <- TN_matrix[-1,]

#### Precision ####
### Formula -> tp / tp + fp
# Precision is the number that were correctly picked,
# divided by all that the model picked.
precision = TP / (TP + FP)

### Recall #####
### Formula -> tp / tp + fn
# Recall is the number that were correctly picked,
# divided by the total number

recall = TP / (TP + FN)

##### f1-score #####
### Formula -> tp / ( tp + 1/2(fp + Fn) )

f1_score = TP / ( TP + 1/2 *(FP + FN))
#Second way
#f1_score2 = 2 * ((precision * recall) / (precision + recall))

### Accuracy ###
## Formula -> (tp + tn) / tp + tn + fp + fn
## Accuracy is defined as the ratio of correctly classified
## examples among all examples
accuracy = (TP + TN ) / (TP + TN + FP + FN)

```

```

# me 30 epohes exoume 69 Tp, 89 FN, 12FP, TN 6540
# me 44 epohes exoume 70 TP, 88 FN, 9 FP
# me 50 epohes exoume 69 TP, 89 FN, 13FP, TN 6539
# me 70 epohes exoume 70 TP, 85 FN, 14 FP, TN 6538
#####--CHECK--
#####

#####--TRAIN--
#####

# #evaluate
anfis_out_train <- anfis.eval(anfis1, dataTrain[,1:input.num])## ANFIS OUTPUT FOR THE
CHECK DATA

plot(dataTrain[,7])

plot(anfis_out_train)
for(i in 1 : nrow(anfis_out_train))
{
  if(anfis_out_train[i] >= 0.5)

    anfis_out_train[i] <- 1
  else
  {
    anfis_out_train[i] <- 0
  }
}

TP_train = 0 #True positive
FN_train = 0 #False negative
FP_train = 0 #False positive
TN_train = 0 #True negative
for(i in 1 : nrow(anfis_out_train))
{
  #True Possitive

```

```

if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 1)
  TP_train = TP_train + 1

#False Negative
if((anfis_out_check[i] == 0 ) & train[i, ncol(train)] == 1)
  FN_train = FN_train + 1

#False Positive
if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 0)
  FP_train = FP_train + 1

#True Negative
if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 0)
  TN_train = TN_train + 1
}

#### Precision ####
precision_train = TP_train / (TP_train + FP_train)

### Recall #####
recall_train = TP_train / (TP_train + FN_train)

##### f1-score #####
f1_score_train = TP_train / ( TP_train + 1/2 *(FP_train + FN_train))

### Accuracy ###
accuracy_train = (TP_train + TN_train ) / (TP_train + TN_train + FP_train + FN_train)

#####--TRAIN--
#####

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#V1 = Position, V2 = Prediction, V3 = Real
#TP_FN_FP_check_matrices <- rbind(TP_matrix,

```

```

# FN_matrix,
#FP_matrix)

#Sort by Position
#TP_FN_FP_check_matrices
TP_FN_FP_check_matrices[order(TP_FN_FP_check_matrices[,1],decreasing=FALSE),]

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN, FP, TN, TP)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Check Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN_train, FP_train, TN_train, TP_train)
df <- data.frame(Real, Predicted, Y)

```

```

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Train Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#
# # num_anfis_out = 1:nrow(anfis_out)
#
# par(mfrow=c(1,1))
# plot(num_anfis_out, anfis_out, type='l', main='Anfis Output-Real Output', xlab='Time
(625:1124)', ylab='Output')
# lines(time(index), x[index], col='red')

## Comparing between simulation and real data
#op <- par(mfrow = c(2, 1))
##x1 <- seq(from = 1, to = nrow(res.fit))
#result.fit <- cbind(data.train[, 5], res.fit)
#plot(x1, result.fit[, 1], col="red", main = "Fitting phase (Training Data(red) Vs
Result(blue))", type = "l", ylab = "MG")
#lines(x1, result.fit[, 2], col="blue")

```

- Mammography DENFIS

```
library(frbs)

dataTrain <-
read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
raining_dataset/mammo_trn_binary01.dat",
           quote="\\"", comment.char="")

dataCheck <-
read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
esting_dataset/mammo_tst_binary01.dat",
           quote="\\"", comment.char="")

my_data_train <- dataTrain
my_data_check <- dataCheck

## Define interval of data min max
range.data_train <- matrix(apply(my_data_train, 2, range), nrow = 2)

data.train <- my_data_train
data.fit <- data.train[,1:(ncol(data.train)-1)]
real.train <- matrix(my_data_train[, (ncol(my_data_train))], ncol=1)

data.check <- my_data_check[,1:(ncol(my_data_check)-1)]
real.val <- matrix(my_data_check[, (ncol(my_data_check))], ncol=1)

##-epochs-##
epochs <- 50
#####
#####-- Set Denfis-for a specific epoch--####
control.DENFIS <- list(Dthr= 0.1,
                      max.iter = epochs,
                      step.size = 0.01,
                      d = 2,
                      name = "Mammography")

method.type <- "DENFIS"
```

```

g <- rnorm(100000)
h <- rep(NA, 100000)
# Start the clock!
ptm <- proc.time()

## Generate fuzzy model
object.DENFIS <- frbs.learn(data.train,
                           range.data_train,
                           method.type,
                           control.DENFIS)

# Stop the clock
proc.time() - ptm

## Predicting step
res.check <- predict(object.DENFIS, data.check)
res.train <- predict(object.DENFIS, data.train[,1:6])

## error calculation

## check
y.pred_check <- res.check
y.real_check <- real.val
residuals_check <- (y.real_check - y.pred_check)
RMSE_check_root_epoch <- sqrt(mean(residuals_check^2))

##train
y.pred_train <- res.train
y.real_train <- real.train
residuals_train <- (y.real_train - y.pred_train)
RMSE_train_root_epoch <- sqrt(mean(residuals_train^2))

min(RMSE_check_root_epoch)
min(RMSE_train_root_epoch)

#####-- Set Denfis-for a specific epoch--#####

```

```

RMSE_check_root <- matrix()
RMSE_train_root <- matrix()
for(i in 1:epochs)
{
  ## Set the method and its parameters
  print(i)
  control.DENFIS <- list(Dthr= 0.1,
                        max.iter = i,
                        step.size = 0.01,
                        d = 2,
                        name = "Mammography")

  method.type <- "DENFIS"
  ## Generate fuzzy model
  object.DENFIS <- frbs.learn(data.train,
                              range.data_train,
                              method.type,
                              control.DENFIS)

  ## Predicting step
  res.check <- predict(object.DENFIS, data.check)
  res.train <- predict(object.DENFIS, data.train[,1:6])

  ## error calculation

  ## check
  y.pred_check <- res.check
  y.real_check <- real.val
  residuals_check <- (y.real_check - y.pred_check)
  RMSE_check_root[i] <- sqrt(mean(residuals_check^2))

  ##train
  y.pred_train <- res.train
  y.real_train <- real.train
  residuals_train <- (y.real_train - y.pred_train)

```



```

    RMSE_train_root[i] <- sqrt(mean(residuals_train^2))
}
#####---plot error curves--#####
ep <- 1:epochs

plot(ep, RMSE_check_root, type='l', col="steelblue4", xlab="Epochs",
      ylab="RMSE", main="Error curves", lwd=2,xaxt='n')

axis(1, at=seq(0,epochs,by=50), labels=T)
lines(ep, RMSE_train_root, lwd=2, col="violetred")
legend(x = "topright",
       col=c("violetred", "steelblue4"), lty = 1, lwd = 1,
       legend=c("Training Error", "Check Error"))
min(RMSE_check_root)
min(RMSE_train_root)
#####---plot error curves--#####

#####--Check--#####
#####

## Predicting step
anfis_out_check <- predict(object.DENFIS, data.check[,1:6])
check <- my_data_check

for(i in 1 : nrow(anfis_out_check))
{
  if(anfis_out_check[i] > 0.5)

    anfis_out_check[i] <- 1
  else
  {
    anfis_out_check[i] <- 0
  }
}

TP = 0 #True positive
FN = 0 #False negative

```

```

FP = 0 #False positive
TN = 0 #True negative
for(i in 1 : nrow(anfis_out_check))
{
  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 1)
    TP = TP + 1

  if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 1)
    FN = FN + 1

  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 0)
    FP = FP + 1

  if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 0)
    TN = TN + 1
}

#### Precision ####
### Formula -> tp / tp + fp
# Precision is the number that were correctly picked,
# divided by all that the model picked.
precision = TP / (TP + FP)

### Recall #####
### Formula -> tp / tp + fn
# Recall is the number that were correctly picked,
# divided by the total number

recall = TP / (TP + FN)

##### f1-score #####
### Formula -> tp / ( tp + 1/2(fp + Fn) )

f1_score = TP / ( TP + 1/2 *(FP + FN))
#Second way

```

```

#f1_score2 = 2 * ((precision * recall) / (precision + recall))

### Accuracy ###
## Formula -> (tp + tn) / tp + tn + fp + fn
## Accuracy is defined as the ratio of correctly classified
## examples among all examples
accuracy = (TP + TN ) / (TP + TN + FP + FN)

#####--Check--
#####

#####--TRAIN--
#####

# #evaluate
anfis_out_train <- predict(object.DENFIS, data.train[, 1:6])
train <- my_data_train

for(i in 1 : nrow(anfis_out_train))
{
  if(anfis_out_train[i] >= 0.5)

    anfis_out_train[i] <- 1
  else
  {
    anfis_out_train[i] <- 0
  }
}

TP_train = 0 #True positive
FN_train = 0 #False negative
FP_train = 0 #False positive
TN_train = 0 #True negative
for(i in 1 : nrow(anfis_out_train))
{
  #True Possitive
  if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 1)
    TP_train = TP_train + 1
}

```

```

#False Negative
if((anfis_out_check[i] == 0 ) & train[i, ncol(train)] == 1)
  FN_train = FN_train + 1

#False Positive
if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 0)
  FP_train = FP_train + 1

#True Negative
if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 0)
  TN_train = TN_train + 1
}

#### Precision ####
precision_train = TP_train / (TP_train + FP_train)

### Recall #####
recall_train = TP_train / (TP_train + FN_train)

##### f1-score #####
f1_score_train = TP_train / ( TP_train + 1/2 *(FP_train + FN_train))

### Accuracy ###
accuracy_train = (TP_train + TN_train ) / (TP_train + TN_train + FP_train + FN_train)

#####--TRAIN--
#####

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#V1 = Position, V2 = Prediction, V3 = Real
#TP_FN_FP_check_matrices <- rbind(TP_matrix,
# FN_matrix,
#FP_matrix)
#Sort by Position

```

```

#TP_FN_FP_check_matrices
TP_FN_FP_check_matrices[order(TP_FN_FP_check_matrices[,1],decreasing=FALSE),]

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN, FP, TN, TP)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Check Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN_train, FP_train, TN_train, TP_train)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +

```

```

scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
theme_bw() + theme(legend.position = "none") +
ggtitle("Train Data") +
theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#
# ## error calculation
# y.pred <- res.test
# y.real <- real.val
# bench <- cbind(y.pred, y.real)
# colnames(bench) <- c("pred. val.", "real. val.")
# print("Comparison DENFIS Vs Real Value on Mackey Glass Data Set")
# print(bench)
#
# residuals <- (y.real - y.pred)
# MSE <- mean(residuals^2)
# RMSE <- sqrt(mean(residuals^2))
# SMAPE <- mean(abs(residuals)/(abs(y.real) + abs(y.pred))/2)*100
#
# err <- c(MSE, RMSE, SMAPE)
# names(err) <- c("MSE", "RMSE", "SMAPE")
# print("DENFIS: Error Measurement: ")
# print(err)
#
# plot( type = "l", xlab="", residuals)
summary(object.DENFIS)

```

- Mammography ANFIS MF3

```
require(FuzzyR)
require(ggplot2)
require(reshape2)
require(shiny)
library(ggplot2)

dataTrain <-
read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
raining_dataset/mammo_trn_binary01.dat",
           quote="\\"", comment.char="")

dataCheck <-
read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
esting_dataset/mammo_tst_binary01.dat",
           quote="\\"", comment.char="")

dataTrain <- as.matrix(dataTrain)

dataCheck <- as.matrix(dataCheck)

train <- dataTrain #train data
check <- dataCheck #check data

allData <- (rbind(train, check))

#####TRUE- all the 1 ouput data#####
k=1
t1 <- matrix(ncol = 7)
for( i in 1:nrow(allData))
{
  if(allData[i,7] == 1)
  {
    t1 <- rbind(t1, allData[i,1:7])
    k = k+1
  }
}
```

```

}
t1 <- t1[-1,]
#####

#####FALSE- all the 0 output data####
k=1
f1 <- matrix(ncol = 7)
for( i in 1:nrow(allData))
{
  if(allData[i,7] == 0)
  {
    f1 <- rbind(f1, allData[i,1:7])
    k = k+1
  }
}
f1 <- f1[-1,]
#####
#membership functions
input.num<-6
input.mf.num<-rep(3,input.num)

#rules
rule.num<-input.num^input.mf.num
rule.which<-sort(sample(1:prod(input.mf.num),rule.num)) #sample

#range
x.range <- matrix(0, ncol=2, nrow=input.num)
for(i in 1:input.num) {
  x.range[i,] <- range(train[,1:input.num][,i])
}
#Building the anfis model
fis<-fis.builder(x.range, input.num, input.mf.num,
                 input.mf.type = 'T1',
                 defuzzMethod="KM",
                 rule.num      = rule.num,
                 rule.which     = rule.which)

```



```

epochs <- 30
step   <- 0.01

anfis1<-anfis.builder(fis)

par(mfrow=c(2,3))
anfis.plotmf(anfis1,'input',1,main="input 1")
anfis.plotmf(anfis1,'input',2,main="input 2")
anfis.plotmf(anfis1,'input',3,main="input 3")
anfis.plotmf(anfis1,'input',4,main="input 4")
anfis.plotmf(anfis1,'input',5,main="input 5")
anfis.plotmf(anfis1,'input',6,main="input 6")

g <- rnorm(100000)
h <- rep(NA, 100000)
# Start the clock!
ptm <- proc.time()

# Loop through the vector, adding one

anfis1<-anfis.optimise(anfis1, train, check,
                      epoch.total = epochs,
                      stepsize    = step,
                      rate.inc    = 1.1,
                      rate.dec    = 0.9,
                      err.log     = T,
                      lambda      = 1,
                      opt.by      = "err.chk",
                      online      = 0,
                      method      = c("gradient", "lse"),
                      err.trn.fix = F)

# Stop the clock
proc.time() - ptm

```

```

#Here we propose a hybrid learning rule which combines the
#gradient method and the least squares estimate (LSE) to identify parameters

par(mfrow=c(2,3))
anfis.plotmf(anfis1, 'input', 1, main="input 1")
anfis.plotmf(anfis1, 'input', 2, main="input 2")
anfis.plotmf(anfis1, 'input', 3, main="input 3")
anfis.plotmf(anfis1, 'input', 4, main="input 4")
anfis.plotmf(anfis1, 'input', 5, main="input 5")
anfis.plotmf(anfis1, 'input', 6, main="input 6")

rmse_trn <- err.trn[,2]
rmse_chk <- err.chk[,2]

min(rmse_chk)
min(rmse_trn)
#which.min(rmse_chk) ## to check which epoch has the lowest rmse

#plot error curves
ep <- 1:epochs

par(mfrow=c(1,1))
##check curve plot
plot(ep, rmse_chk, type='l', col="steelblue4", xlab="Epochs",
      ylab="RMSE", main="Error curves" ,lwd=2,xaxt='n')
legend(x = "topright",
       col=c( "steelblue4"), lty = 1, lwd = 1,
       legend=c( "Check Error"))
axis(1, at=seq(0,epochs,by=50), labels=T)
##train curve plot
plot(ep, rmse_trn, type='l', col="violetred", xlab="Epochs",
      ylab="RMSE", main="Error curves" ,lwd=2,xaxt='n')
legend(x = "topright",
       col=c("violetred"), lty = 1, lwd = 1,
       legend=c( "Training Error"))
axis(1, at=seq(0,epochs,by=50), labels=T)

```

```

#####--CHECK--
#####

# #evaluate

#anfis_out_TRUE <- anfis.eval(anfis1, t1[,1:input.num])## ??NFIS OUTPUT ONLY FOR THE
TRUE=1 OUTPUTS

#anfis_out_FALSE <- anfis.eval(anfis1, f1[,1:input.num])## ANFIS OUTPUT ONLY FOR THE
FALSE=0 OUTPUTS

anfis_out_check <- anfis.eval(anfis1, dataCheck[,1:input.num])## ANFIS OUTPUT FOR THE
CHECK DATA

plot(dataCheck[,7])

plot(anfis_out_check)
for(i in 1 : nrow(anfis_out_check))
{
  if(anfis_out_check[i] >= 0.5)

    anfis_out_check[i] <- 1
  else
  {
    anfis_out_check[i] <- 0
  }
}

TP_matrix <- matrix(ncol = 3)
FN_matrix <- matrix(ncol = 3)
FP_matrix <- matrix(ncol = 3)
TN_matrix <- matrix(ncol = 3)
TP = 0 #True positive
FN = 0 #False negative
FP = 0 #False positive
TN = 0 #True negative
for(i in 1 : nrow(anfis_out_check))
{
  #True Possitive

```

```

if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 1)
{
  TP = TP + 1
  TP_matrix <- rbind(TP_matrix, c(i, 1, 1))
}
#False Negative
if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 1)
{
  FN = FN + 1
  FN_matrix <- rbind(FN_matrix, c(i, 0, 1))
}
#False Positive
if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 0)
{
  FP = FP + 1
  FP_matrix <- rbind(FP_matrix, c(i, 1, 0))
}

#True Negative
if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 0)
{
  TN = TN + 1
  TN_matrix <- rbind(TN_matrix, c(i, 0, 0))
}
}

#we delete the first row because is NA from the init
TP_matrix <- TP_matrix[-1,]
FN_matrix <- FN_matrix[-1,]
FP_matrix <- FP_matrix[-1,]
TN_matrix <- TN_matrix[-1,]

#### Precision ####
### Formula -> tp / tp + fp
# Precision is the number that were correctly picked,
# divided by all that the model picked.
precision = TP / (TP + FP)

```

```

### Recall #####
### Formula -> tp / tp + fn
# Recall is the number that were correctly picked,
# divided by the total number

recall = TP / (TP + FN)

##### f1-score #####
### Formula -> tp / ( tp + 1/2(fp + Fn) )

f1_score = TP / ( TP + 1/2 *(FP + FN))
#Second way
#f1_score2 = 2 * ((precision * recall) / (precision + recall))

### Accuracy ###
## Formula -> (tp + tn) / tp + tn + fp + fn
## Accuracy is defined as the ratio of correctly classified
## examples among all examples
accuracy = (TP + TN ) / (TP + TN + FP + FN)

# me 30 epohes exoume 69 Tp, 89 FN, 12FP, TN 6540
# me 44 epohes exoume 70 TP, 88 FN, 9 FP
# me 50 epohes exoume 69 TP, 89 FN, 13FP, TN 6539
# me 70 epohes exoume 70 TP, 85 FN, 14 FP, TN 6538
#####--CHECK--
#####

#####--TRAIN--
#####

# #evaluate

anfis_out_train <- anfis.eval(anfis1, dataTrain[,1:input.num])## ANFIS OUTPUT FOR THE
CHECK DATA

plot(dataTrain[,7])
plot(anfis_out_train)
for(i in 1 : nrow(anfis_out_train))
{

```

```

if(anfis_out_train[i] >= 0.5)

    anfis_out_train[i] <- 1
else
{
    anfis_out_train[i] <- 0
}
}

TP_train = 0 #True positive
FN_train = 0 #False negative
FP_train = 0 #False positive
TN_train = 0 #True negative
for(i in 1 : nrow(anfis_out_train))
{
    #True Possitive
    if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 1)
        TP_train = TP_train + 1

    #False Negative
    if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 1)
        FN_train = FN_train + 1

    #False Positive
    if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 0)
        FP_train = FP_train + 1

    #True Negative
    if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 0)
        TN_train = TN_train + 1
}

#### Precision ####
precision_train = TP_train / (TP_train + FP_train)

```

```

### Recall #####
recall_train = TP_train / (TP_train + FN_train)

##### f1-score #####
f1_score_train = TP_train / ( TP_train + 1/2 *(FP_train + FN_train))

### Accuracy ###
accuracy_train = (TP_train + TN_train ) / (TP_train + TN_train + FP_train + FN_train)

#####--TRAIN--
#####

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#V1 = Position, V2 = Prediction, V3 = Real
#TP_FN_FP_check_matrices <- rbind(TP_matrix,
# FN_matrix,
#FP_matrix)
#Sort by Position
#TP_FN_FP_check_matrices
TP_FN_FP_check_matrices[order(TP_FN_FP_check_matrices[,1],decreasing=FALSE),] <-

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN, FP, TN, TP)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Check Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title
#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

```

```

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN_train, FP_train, TN_train, TP_train)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Train Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#
# # num_anfis_out = 1:nrow(anfis_out)
#
# par(mfrow=c(1,1))
# plot(num_anfis_out, anfis_out, type='l', main='Anfis Output-Real Output', xlab='Time
(625:1124)', ylab='Output')
# lines(time(index), x[index], col='red')

## Comparing between simulation and real data
#op <- par(mfrow = c(2, 1))
##x1 <- seq(from = 1, to = nrow(res.fit))
#result.fit <- cbind(data.train[, 5], res.fit)
#plot(x1, result.fit[, 1], col="red", main = "Fitting phase (Training Data(red) Vs
Result(blue))", type = "l", ylab = "MG")
#lines(x1, result.fit[, 2], col="blue")

```


- PCA Mammography ANFIS

```

require(FuzzyR)
require(ggplot2)
require(reshape2)
require(shiny)
library(ggplot2)

dataTrain <- read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
raining_dataset/mammo_trn_binary01.dat",
                        quote="\\"", comment.char="")

dataCheck <- read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/Initial_datasets_6_inputs/T
esting_dataset/mammo_tst_binary01.dat",
                        quote="\\"", comment.char="")

dataTrain <- as.matrix(dataTrain)
dataCheck <- as.matrix(dataCheck)

train <- dataTrain #train data
check <- dataCheck #check data

allData <- rbind(train, check)
#####TRUE- all the 1 ouput data#####
k=1
t1 <- matrix(ncol = 7)
for( i in 1:nrow(allData))
{
  if(allData[i,7] == 1)
  {
    t1 <- rbind(t1, allData[i,1:7])
    k = k+1
  }
}
t1 <- t1[-1,]
#####

```

```

#####FALSE- all the 0 output data#####
k=1
f1 <- matrix(ncol = 7)
for( i in 1:nrow(allData))
{
  if(allData[i,7] == 0)
  {
    f1 <- rbind(f1, allData[i,1:7])
    k = k+1
  }
}
f1 <- f1[-1,]
#####
#membership functions
input.num<-6
input.mf.num<-rep(3,input.num)

#rules
rule.num<-input.num^input.mf.num
rule.which<-sort(sample(1:prod(input.mf.num),rule.num)) #sample

#range
x.range <- matrix(0, ncol=2, nrow=input.num)
for(i in 1:input.num) {
  x.range[i,] <- range(train[,1:input.num][,i])
}

#Building the anfis model
fis<-fis.builder(x.range, input.num, input.mf.num,
                input.mf.type = 'T1',
                defuzzMethod="KM",
                rule.num      = rule.num,
                rule.which    = rule.which)

epochs <- 30
step   <- 0.01

```

```

anfis1<-anfis.builder(fis)

par(mfrow=c(2,3))
anfis.plotmf(anfis1,'input',1,main="input 1")
anfis.plotmf(anfis1,'input',2,main="input 2")
anfis.plotmf(anfis1,'input',3,main="input 3")
anfis.plotmf(anfis1,'input',4,main="input 4")
anfis.plotmf(anfis1,'input',5,main="input 5")
anfis.plotmf(anfis1,'input',6,main="input 6")

g <- rnorm(100000)
h <- rep(NA, 100000)

# Start the clock!
ptm <- proc.time()

# Loop through the vector, adding one

anfis1<-anfis.optimise(anfis1, train, check,
                      epoch.total = epochs,
                      stepsize     = step,
                      rate.inc     = 1.1,
                      rate.dec     = 0.9,
                      err.log      = T,
                      lambda       = 1,
                      opt.by       = "err.chk",
                      online       = 0,
                      method       = c("gradient", "lse"),
                      err.trn.fix = F)

# Stop the clock
proc.time() - ptm

#Here we propose a hybrid learning rule which combines the
#gradient method and the least squares estimate (LSE) to identify parameters

```

```

par(mfrow=c(2,3))
anfis.plotmf(anfis1, 'input', 1, main="input 1")
anfis.plotmf(anfis1, 'input', 2, main="input 2")
anfis.plotmf(anfis1, 'input', 3, main="input 3")
anfis.plotmf(anfis1, 'input', 4, main="input 4")
anfis.plotmf(anfis1, 'input', 5 ,main="input 5")
anfis.plotmf(anfis1, 'input', 6 ,main="input 6")

rmse_trn <- err.trn[,2]
rmse_chk <- err.chk[,2]

min(rmse_chk)
min(rmse_trn)
#which.min(rmse_chk) ## to check which epoch has the lowest rmse

#plot error curves
ep <- 1:epochs

par(mfrow=c(1,1))
##check curve plot
plot(ep, rmse_chk, type='l', col="steelblue4", xlab="Epochs",
      ylab="RMSE", main="Error curves" ,lwd=2,xaxt='n')
legend(x = "topright",
       col=c( "steelblue4"), lty = 1, lwd = 1,
       legend=c( "Check Error"))
axis(1, at=seq(0,epochs,by=50), labels=T)
##train curve plot
plot(ep, rmse_trn, type='l', col="violetred", xlab="Epochs",
      ylab="RMSE", main="Error curves" ,lwd=2,xaxt='n')
legend(x = "topright",
       col=c("violetred"), lty = 1, lwd = 1,
       legend=c( "Training Error"))
axis(1, at=seq(0,epochs,by=50), labels=T)

```

```

#####--CHECK--
#####

# #evaluate

#anfis_out_TRUE <- anfis.eval(anfis1, t1[ ,1:input.num])## ??NFIS OUTPUT ONLY FOR THE
TRUE=1 OUTPUTS

#anfis_out_FALSE <- anfis.eval(anfis1, f1[ ,1:input.num])## ANFIS OUTPUT ONLY FOR THE
FALSE=0 OUTPUTS

anfis_out_check <- anfis.eval(anfis1, dataCheck[,1:input.num])## ANFIS OUTPUT FOR THE
CHECK DATA

plot(dataCheck[,7])
plot(anfis_out_check)
for(i in 1 : nrow(anfis_out_check))
{
  if(anfis_out_check[i] >= 0.5)

    anfis_out_check[i] <- 1
  else
  {
    anfis_out_check[i] <- 0
  }
}

TP_matrix <- matrix(ncol = 3)
FN_matrix <- matrix(ncol = 3)
FP_matrix <- matrix(ncol = 3)
TN_matrix <- matrix(ncol = 3)
TP = 0 #True positive
FN = 0 #False negative
FP = 0 #False positive
TN = 0 #True negative
for(i in 1 : nrow(anfis_out_check))
{
  #True Possitive
  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 1)

```

```

{
  TP = TP + 1
  TP_matrix <- rbind(TP_matrix, c(i, 1, 1))
}
#False Negative
if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 1)
{
  FN = FN + 1
  FN_matrix <- rbind(FN_matrix, c(i, 0, 1))
}
#False Positive
if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 0)
{
  FP = FP + 1
  FP_matrix <- rbind(FP_matrix, c(i, 1, 0))
}

#True Negative
if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 0)
{
  TN = TN + 1
  TN_matrix <- rbind(TN_matrix, c(i, 0, 0))
}
}

#we delete the first row because is NA from the init
TP_matrix <- TP_matrix[-1,]
FN_matrix <- FN_matrix[-1,]
FP_matrix <- FP_matrix[-1,]
TN_matrix <- TN_matrix[-1,]

#### Precision ####
### Formula -> tp / tp + fp
# Precision is the number that were correctly picked,
# divided by all that the model picked.
precision = TP / (TP + FP)

```

```

### Recall #####
### Formula -> tp / tp + fn
# Recall is the number that were correctly picked,
# divided by the total number

recall = TP / (TP + FN)

##### f1-score #####
### Formula -> tp / ( tp + 1/2(fp + Fn) )

f1_score = TP / ( TP + 1/2 *(FP + FN))
#Second way
#f1_score2 = 2 * ((precision * recall) / (precision + recall))

### Accuracy ###
## Formula -> (tp + tn) / tp + tn + fp + fn
## Accuracy is defined as the ratio of correctly classified
## examples among all examples
accuracy = (TP + TN ) / (TP + TN + FP + FN)

#####--CHECK--
#####

#####--TRAIN--
#####

# #evaluate

anfis_out_train <- anfis.eval(anfis1, dataTrain[,1:input.num])## ANFIS OUTPUT FOR THE
CHECK DATA

plot(dataTrain[,7])

plot(anfis_out_train)
for(i in 1 : nrow(anfis_out_train))
{
  if(anfis_out_train[i] >= 0.5)

    anfis_out_train[i] <- 1
  else

```

```

    {
        anfis_out_train[i] <- 0
    }
}

TP_train = 0 #True positive
FN_train = 0 #False negative
FP_train = 0 #False positive
TN_train = 0 #True negative
for(i in 1 : nrow(anfis_out_train))
{
    #True Possitive
    if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 1)
        TP_train = TP_train + 1

    #False Negative
    if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 1)
        FN_train = FN_train + 1

    #False Positive
    if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 0)
        FP_train = FP_train + 1

    #True Negative
    if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 0)
        TN_train = TN_train + 1
}

#### Precision ####
precision_train = TP_train / (TP_train + FP_train)

### Recall #####
recall_train = TP_train / (TP_train + FN_train)

##### f1-score #####
f1_score_train = TP_train / ( TP_train + 1/2 *(FP_train + FN_train))

```



```

### Accuracy ###
accuracy_train = (TP_train + TN_train ) / (TP_train + TN_train + FP_train + FN_train)

#####--TRAIN--
#####

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#V1 = Position, V2 = Prediction, V3 = Real
#TP_FN_FP_check_matrices <- rbind(TP_matrix,
# FN_matrix,
#FP_matrix)
#Sort by Position
#TP_FN_FP_check_matrices
TP_FN_FP_check_matrices[order(TP_FN_FP_check_matrices[,1],decreasing=FALSE),] <-

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN, FP, TN, TP)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Check Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN_train, FP_train, TN_train, TP_train)
df <- data.frame(Real, Predicted, Y)

```

```

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3",trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Train Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#
# # num_anfis_out = 1:nrow(anfis_out)
#
# par(mfrow=c(1,1))
# plot(num_anfis_out, anfis_out, type='l', main='Anfis Output-Real Output', xlab='Time
(625:1124)', ylab='Output')
# lines(time(index), x[index], col='red')

## Comparing between simulation and real data
#op <- par(mfrow = c(2, 1))
##x1 <- seq(from = 1, to = nrow(res.fit))
#result.fit <- cbind(data.train[, 5], res.fit)
#plot(x1, result.fit[, 1], col="red", main = "Fitting phase (Training Data(red) Vs
Result(blue))", type = "l", ylab = "MG")
#lines(x1, result.fit[, 2], col="blue")

```

- PCA Mammography DENFIS

```
library(frbs)
require(ggplot2)

dataTrain      <-      read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/PCA
transformed      datasets_4      inputs/Training
dataset/pca_transformed_m_mammo_trn_binary01.dat",
                                quote="\\"", comment.char="")

dataCheck      <-      read.table("C:/Users/thoma/Desktop/THESIS/DATA/mammography/PCA
transformed      datasets_4      inputs/Testing
dataset/pca_transformed_m_mammo_tst_binary01.dat",
                                quote="\\"", comment.char="")

my_data_train <- dataTrain
my_data_check <- dataCheck

## Define interval of data min max
range.data_train <- matrix(apply(my_data_train, 2, range), nrow = 2)

data.train <- my_data_train
data.fit <- data.train[,1:(ncol(data.train)-1)]
real.train <- matrix(my_data_train[, (ncol(my_data_train))], ncol=1)

data.check <- my_data_check[,1:(ncol(my_data_check)-1)]
real.val <- matrix(my_data_check[, (ncol(my_data_check))], ncol=1)

##-epochs-##
epochs <- 50
#####

#####-- Set Denfis-for a specific epoch--####
control.DENFIS <- list(Dthr= 0.1,
                      max.iter = epochs,
                      step.size = 0.01,
                      d = 2,
                      name = "Mammography")
```

```

method.type <- "DENFIS"
g <- rnorm(100000)
h <- rep(NA, 100000)

# Start the clock!
ptm <- proc.time()

## Generate fuzzy model
object.DENFIS <- frbs.learn(data.train,
                            range.data_train,
                            method.type,
                            control.DENFIS)

# Stop the clock
proc.time() - ptm

## Predicting step
res.check <- predict(object.DENFIS, data.check)
res.train <- predict(object.DENFIS, data.train[,1:4])

## error calculation

## check
y.pred_check <- res.check
y.real_check <- real.val
residuals_check <- (y.real_check - y.pred_check)
RMSE_check_root_epoch <- sqrt(mean(residuals_check^2))

##train
y.pred_train <- res.train
y.real_train <- real.train
residuals_train <- (y.real_train - y.pred_train)
RMSE_train_root_epoch <- sqrt(mean(residuals_train^2))

min(RMSE_check_root_epoch)
min(RMSE_train_root_epoch)

```

```

#####-- Set Denfis-for a specific epoch--#####
RMSE_check_root <- matrix()
RMSE_train_root <- matrix()
for(i in 1:epochs)
{
  ## Set the method and its parameters
  print(i)
  control.DENFIS <- list(Dthr= 0.1,
                        max.iter = i,
                        step.size = 0.01,
                        d = 2,
                        name = "Mammography")

  method.type <- "DENFIS"
  ## Generate fuzzy model
  object.DENFIS <- frbs.learn(data.train,
                              range.data_train,
                              method.type,
                              control.DENFIS)

  ## Predicting step
  res.check <- predict(object.DENFIS, data.check)
  res.train <- predict(object.DENFIS, data.train[,1:4])

  ## error calculation
  ## check
  y.pred_check <- res.check
  y.real_check <- real.val
  residuals_check <- (y.real_check - y.pred_check)
  RMSE_check_root[i] <- sqrt(mean(residuals_check^2))

  ##train
  y.pred_train <- res.train
  y.real_train <- real.train
  residuals_train <- (y.real_train - y.pred_train)
  RMSE_train_root[i] <- sqrt(mean(residuals_train^2))
}

```

```

}
#####--plot error curves--#####
ep <- 1:epochs

plot(ep, RMSE_check_root, type='l', col="steelblue4", xlab="Epochs",
      ylab="RMSE", main="Error curves", lwd=2,xaxt='n')

axis(1, at=seq(0,epochs,by=50), labels=T)
lines(ep, RMSE_train_root, lwd=2, col="violetred")
legend(x = "topright",
       col=c("violetred", "steelblue4"), lty = 1, lwd = 1,
       legend=c("Training Error", "Check Error"))
min(RMSE_check_root)
min(RMSE_train_root)
#####--plot error curves--#####

#####--Check--
#####

## Predicting step
anfis_out_check <- predict(object.DENFIS, data.check[,1:4])
check <- my_data_check
for(i in 1 : nrow(anfis_out_check))
{
  if(anfis_out_check[i] > 0.5)

    anfis_out_check[i] <- 1
  else
  {
    anfis_out_check[i] <- 0
  }
}

TP = 0 #True positive
FN = 0 #False negative
FP = 0 #False positive
TN = 0 #True negative

```

```

for(i in 1 : nrow(anfis_out_check))
{
  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 1)
    TP = TP + 1

  if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 1)
    FN = FN + 1

  if((anfis_out_check[i] == 1 ) & check[i, ncol(check)] == 0)
    FP = FP + 1

  if((anfis_out_check[i] == 0 ) & check[i, ncol(check)] == 0)
    TN = TN + 1
}

#### Precision ####
### Formula -> tp / tp + fp
# Precision is the number that were correctly picked,
# divided by all that the model picked.
precision = TP / (TP + FP)

### Recall #####
### Formula -> tp / tp + fn
# Recall is the number that were correctly picked,
# divided by the total number
recall = TP / (TP + FN)

##### f1-score #####
### Formula -> tp / ( tp + 1/2(fp + Fn) )

f1_score = TP / ( TP + 1/2 *(FP + FN))
#Second way
#f1_score2 = 2 * ((precision * recall) / (precision + recall))

### Accuracy ###
## Formula -> (tp + tn) / tp + tn + fp + fn

```

```

## Accuracy is defined as the ratio of correctly classified
## examples among all examples
accuracy = (TP + TN ) / (TP + TN + FP + FN)
#####--Check--
#####

#####--TRAIN--
#####

# #evaluate
anfis_out_train <- predict(object.DENFIS, data.train[, 1:4])
train <- my_data_train

for(i in 1 : nrow(anfis_out_train))
{
  if(anfis_out_train[i] >= 0.5)

    anfis_out_train[i] <- 1
  else
  {
    anfis_out_train[i] <- 0
  }
}

TP_train = 0 #True positive
FN_train = 0 #False negative
FP_train = 0 #False positive
TN_train = 0 #True negative
for(i in 1 : nrow(anfis_out_train))
{
  #True Possitive
  if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 1)
    TP_train = TP_train + 1

  #False Negative
  if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 1)

```



```

    FN_train = FN_train + 1

#False Positive
if((anfis_out_train[i] == 1 ) & train[i, ncol(train)] == 0)
    FP_train = FP_train + 1

#True Negative
if((anfis_out_train[i] == 0 ) & train[i, ncol(train)] == 0)
    TN_train = TN_train + 1
}

#### Precision ####
precision_train = TP_train / (TP_train + FP_train)

### Recall #####
recall_train = TP_train / (TP_train + FN_train)

##### f1-score #####
f1_score_train = TP_train / ( TP_train + 1/2 *(FP_train + FN_train))

### Accuracy ###
accuracy_train = (TP_train + TN_train ) / (TP_train + TN_train + FP_train + FN_train)

#####--TRAIN--
#####

#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#V1 = Position, V2 = Prediction, V3 = Real
#TP_FN_FP_check_matrices <- rbind(TP_matrix,
# FN_matrix,
#FP_matrix)
#Sort by Position

#TP_FN_FP_check_matrices
TP_FN_FP_check_matrices[order(TP_FN_FP_check_matrices[,1],decreasing=FALSE),] <-

```

```

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN, FP, TN, TP)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3", trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Check Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title
#####-CHECK-BIND-MATRICES-
FP/TP/FP_for_ploting#####

#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

Real <- factor(c(0, 0, 1, 1))
Predicted <- factor(c(0, 1, 0, 1))
Y <- c(FN_train, FP_train, TN_train, TP_train)
df <- data.frame(Real, Predicted, Y)

ggplot(data = df, mapping = aes(x = Real, y = Predicted)) +
  geom_tile(aes(fill = Y), colour = "white") +
  geom_text(aes(label = sprintf("%1.0f", Y)), vjust = 1) +
  scale_fill_gradient(low = "lightskyblue1", high = "seagreen3", trans = "log") +
  theme_bw() + theme(legend.position = "none") +
  ggtitle("Train Data") +
  theme(plot.title = element_text(hjust = 0.5))#Center the Title
#####-Train-BIND-MATRICES-
FP/TP/FP_for_ploting#####

```

```

#
# ## error calculation
# y.pred <- res.test
# y.real <- real.val
# bench <- cbind(y.pred, y.real)
# colnames(bench) <- c("pred. val.", "real. val.")
# print("Comparison DENFIS Vs Real Value on Mackey Glass Data Set")
# print(bench)
#
# residuals <- (y.real - y.pred)
# MSE <- mean(residuals^2)
# RMSE <- sqrt(mean(residuals^2))
# SMAPE <- mean(abs(residuals)/(abs(y.real) + abs(y.pred))/2)*100
#
# err <- c(MSE, RMSE, SMAPE)
# names(err) <- c("MSE", "RMSE", "SMAPE")
# print("DENFIS: Error Measurement: ")
# print(err)
#
#
#
# plot( type = "l", xlab="", residuals)
summary(object.DENFIS)

```