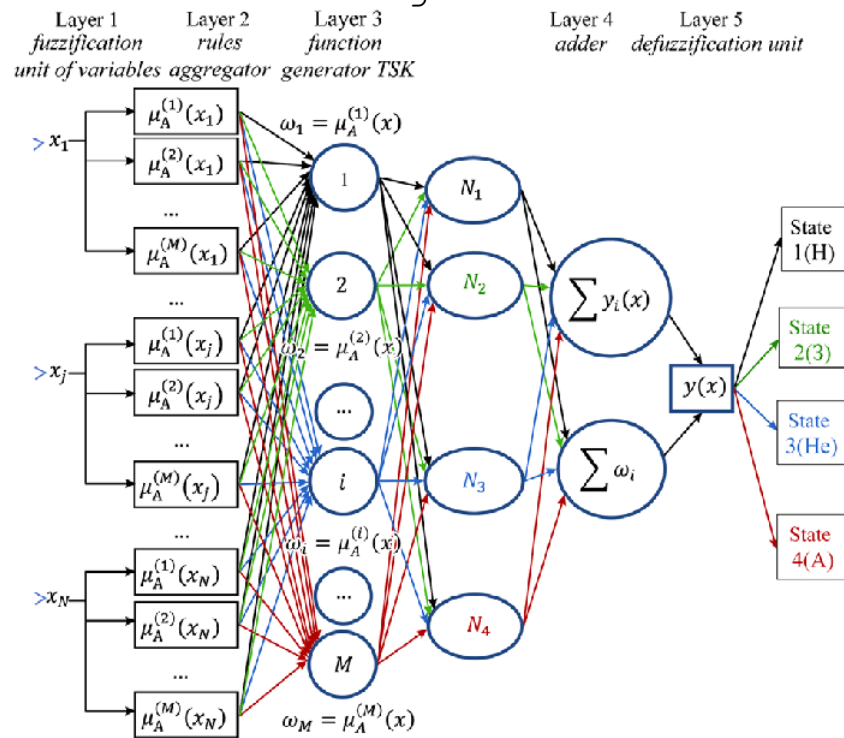


Επίλυση προβλημάτων με μοντέλα
Παλινδρόμησης TSK (Takagi-Sugeno-Kang)

TSK Regression



Χάρης Φίλης AEM: 9449
Github Repository Link : [here](#)

October 1, 2022

Contents

0.1	Εισαγωγή-Abstract	3
0.2	Εφαρμογή στο airfoil self-noise dataset	3
0.2.1	TSK model 1	4
0.2.2	TSK model 2	6
0.2.3	TSK model 3	8
0.2.4	TSK model 4	10
0.2.5	Τελική αξιολόγηση των μοντέλων - Δείκτες απόδοσης & Συμπεράσματα	12
0.3	Εφαρμογή σε μεγάλη διάστασης δεδομένα Superconductivity	13
0.3.1	Data Preprocessing	13
0.3.2	5-fold cross validation	13
0.3.3	Grid Search	13
0.3.4	Συμπεράσματα από το Grid Search	17
0.3.5	Optimal Model	17
0.4	Σύγκριση Grid Partitioning με Subtractive Clustering	20
0.5	Παραδοτέο	20

List of Figures

1	Γενική μορφή gbellmf τριών εισόδων-model 1 πριν την εκπαίδευση	4
2	Input Mfs after training	5
3	Learning Curve Model 1 TrnError,ValError	5
4	TSK model 1 Prediction Error	5
5	Γενική μορφή gbellmf τριών εισόδων-model 2 πριν την εκπαίδευση	6
6	Input Mfs after training	6
7	Learning Curve Model 2 TrnError,ValError	7
8	TSK model 2 Prediction Error	7
9	Γενική μορφή gbellmf τριών εισόδων-model 3 πριν την εκπαίδευση	8
10	Input Mfs after training	8
11	Learning Curve Model 3 TrnError,ValError	9
12	TSK model 3 Prediction Error	9
13	Γενική μορφή gbellmf τριών εισόδων-model 4 πριν την εκπαίδευση	10
14	Input Mfs after training	10
15	Learning Curve Model 4 TrnError,ValError	11
16	TSK model 4 Prediction Error	11
17	Error_Grid	15
18	Περίπτωση που το μοντέλο δεν μάθαινει	15
19	Περίπτωση που το μοντέλο μαθαίνει	16
20	Αριθμός κανόνων που δημιουργήθηκαν από διαφορετικές παραμέτρους	16
21	Kfold Prediction Errors with random params	17
22	Input Mfs after training	18
23	Learning Curve Model 4 TrnError,ValError	18
24	Σύγκριση Ground Truth με Predictions Μοντέλου	19
25	Optimal Model Prediction Error	19
26	Result Tables	19

0.1 Εισαγωγή-Abstract

Η παρούσα εργασία έχει ως αντικείμενο την εκπαίδευση και την εκτενή αξιολόγηση TSK μοντέλων με διαφορετικές παραμέτρους ως προς την επίλυση του προβλήματος παλινδρόμησης σε διαφορετικά dataset. Συγκεκριμένα χρησιμοποιούνται δύο dataset, για τα οποία γνωρίζοντας ορισμένα χαρακτηριστικά εκπαίδευσης θα πρέπει το μοντέλο να προσομοιώσει ουσιαστικά μια μη γραμμική συνάρτηση εκτίμησης ενός χαρακτηριστικού στόχος. Δηλαδή το νευρωνικό TSK που θα δημιουργηθεί θα πρέπει να προσεγγίζει όσο το δυνατόν καλύτερα την πραγματική απεικόνιση των χαρακτηριστικών εισόδου στο χαρακτηριστικό στόχος-έξοδος. Εφόσον αυτή είναι μια διαδικασία εκτίμηση δεδομένων και μάλιστα πραγματικού αριθμού το μοντέλο είναι παλινδρομικό.

0.2 Εφαρμογή στο airfoil self-noise dataset

To airfoil dataset απο το UCI repository περιέχει 1503 δείγματα 6 χαρακτηριστικών με πραγματικές τιμές. Οι 5 από αυτές, συγκεκριμένα Συχνότητα Δόνησης πτερυγίου, Γωνία κούσης αέριου ρεύματος, Μήκος χορδής, Ταχύτητα ρεύματος αέρα καθώς και το πάχος μετατόπισης πλευράς αναρρόφησης, αφορούν τα δεδομένα εκμάθησης του νευρωνικού δικτύου ενώ η 6 αποτελεί τον στόχο τον οποίο θέλουμε να εκτιμήσουμε που στην προκειμένη περίπτωση αναφέρεται στην κανονικοποιημένη πίεση θορύβου που παράγει το πτερύγιο σε dB.

Αρχικά γίνεται προεπεξεργασία δεδομένων δηλαδή κανονικοποίηση ώστε η νόρμα κάθε vector να είναι 1 ($x = \frac{x - x_{min}}{x_{max} - x_{min}}$) καθώς χωρισμός σε δεδομένα ελέγχου και εκπαίδευσης καθώς και επαλήθευσης σε τρία μη επικαλυπτόμενα υποσύνολα:

- 60% σύνολα εκπαίδευσης - training data (Dtrn)
- 20% σύνολο επαλήθευσης - validation data (Dval)
- 20% σύνολο ελέγχου - testData (Dtest)

Όσον αφορά την μέθοδο κανονικοποίησης αυτή εφαρμόζεται στο train κομμάτι του dataset και στην συνέχεια με βάση τα max και min values αυτού κανονικοποιούνται τα άλλα δυο κομμάτια του dataset.

Στο πρώτο κομμάτι της εργασίας προτείνονται 4 μοντέλα TSK τα οποία αρχικοποιούνται με ασαφές σύστημα διεπαφής το οποίο χρησιμοποιεί grid partitioning. Τα οποία φαίνονται παρακάτω.

#TSK	Πλήθος συναρτήσεων Συμμετοχής	Μορφή Εξόδου
TSK_model_1	2	Singleton
TSK_model_2	3	Singleton
TSK_model_3	2	Polynomial
TSK_model_4	3	Polynomial

Η εκπαίδευση των συναρτήσεων συμμετοχής βελτιστοποιούνται μέσω **adaptive fuzzy interference system** με τον αλγόριθμο **back propagation** και οι παράμετροι της πολυωνυμικής συνάρτησης εξόδου μέσω μεθόδου ελαχίστων τετραγώνων μια κυρτή μέθοδος εφόσον δεν ξέρουμε και τον βαθμό του πολυωνύμου εξόδου.

0.2.1 TSK model 1

Για όλα τα μοντέλα χρησιμοποιώ το **genfis** ως ασαφές σύστημα αρχικοποίησης του **anfis** και παραμέτρους στο **genfisOptions struct** για να χρησιμοποιεί **grid-Partitioning**, συνάρτηση συμμετοχής εισόδου **gbellmf** (καμπάνα) και ανάλογα τον ρυθμίζεται το πλήθος συναρτήσεων εισόδου και ο τύπος των συναρτήσεων εξόδου. Το μοντέλο 1 είναι μοντέλο **Singneton** δηλαδή η έξοδος είναι κάποια τιμή και όχι ασαφές σύνολο ή κάτι άλλο.

Τα μοντέλα **Singleton** περιγράφονται με ασαφείς κανόνες της μορφής;

[Singleton] $R(i) = IF ((x_1 \text{ is } A_1 \text{ AND...AND } x_m \text{ is } A_m \implies THEN y_{isw})$

Όπως αναφέρθηκε στο πρώτο μοντέλο **Tsk** οι συναρτήσεις συμμετοχής ήταν τύπου **gbellmf**. Παρακάτω φαίνεται η μορφή των **input** πριν το **training** του μοντέλου 1.

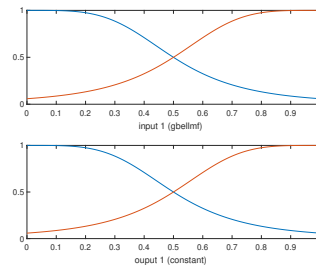
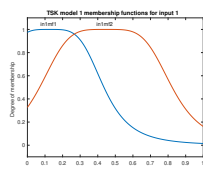


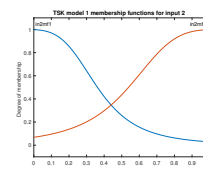
Figure 1: Γενική μορφή **gbellmf** τριών εισόδων-model 1 πριν την εκπαίδευση

Στην συνέχεια το σύστημα αυτορρυθμίζεται χρησιμοποιώντας τα δεδομένα εκπαίδευσης μέσω για την κατάλληλη επιλογή των παραμέτρων συναρτήσεων συμμετοχής μέσω αλγορίθμου **back-propagation** ο οποίο εκμεταλλεύεται τις παραγώγους των εξόδων και τις χρησιμοποιεί για να ανανεώσει τα βάρη των εισόδων των προηγούμενων **layer**. Αυτή η διαδικασία γίνεται μέσω του **anfis** συστήματος του **matlab** για 100 εποχές. Έτσι προκύπτουν οι παρακάτω ανανεωμένες εισοδοι συμμετοχής: Οι καμπύλες εκμάθησης που απεικονίζουν με πορτοκαλί την μεταβολή σφάλματος του συνόλου επαλήθευσης (**validation Data**) ενώ το με μπλε του συνόλου εκπαίδευσής (**training data**) φαίνονται στο παρακάτω διάγραμμα.

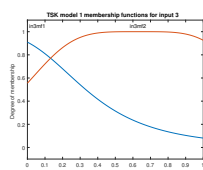
Τέλος παρουσιάζεται η κατανομή του σφάλματος πρόβλεψης του **target** από το **TSK model 1**.



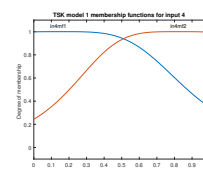
(a) gbellmf 1



(b) gbellmf 2



(c) gbellmf 3



(d) gbellmf 4

Figure 2: Input Mfs after training

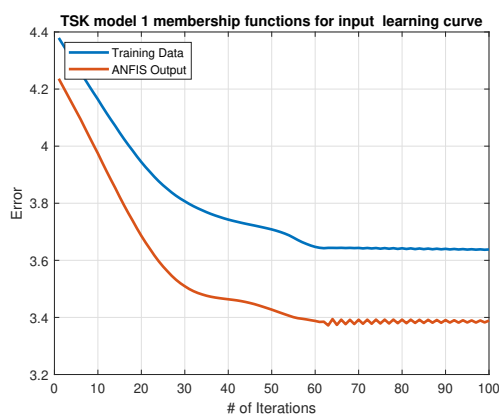


Figure 3: Learning Curve Model 1 TrnError,ValError

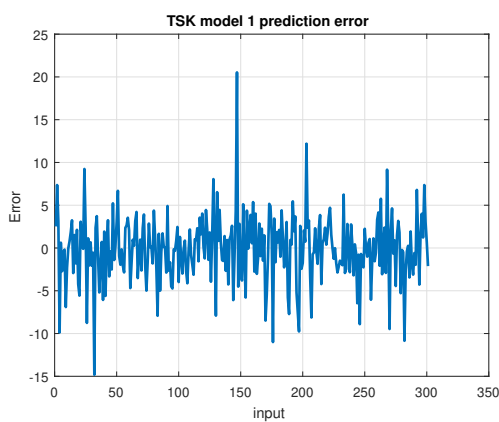


Figure 4: TSK model 1 Prediction Error

0.2.2 TSK model 2

Η ίδια διαδικασία εκτελείται και για το 2ο μοντέλο ασαφές νευρωνικού με 3 συναρτήσεις συμμετοχής στην είσοδο οι οποίες πριν την εκπαίδευση είναι ως εξής

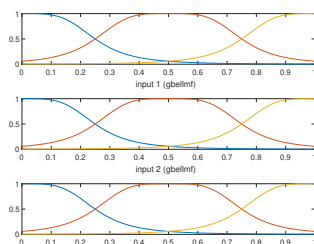


Figure 5: Γενική μορφή gbellmf τριών εισόδων-model 2 πριν την εκπαίδευση

Στην συνέχεια το anfis μετατρέπει αυτές κατά την διάρκεια εκπαίδευσης στις παρακάτω.

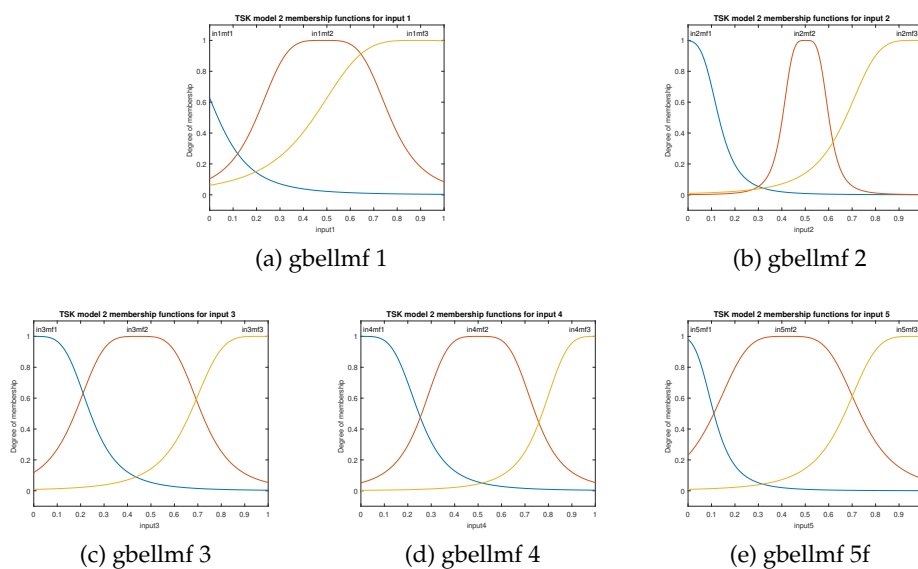


Figure 6: Input Mfs after training

Η καμπύλη εκμάθησης του TSK model 2 είναι η εξής:

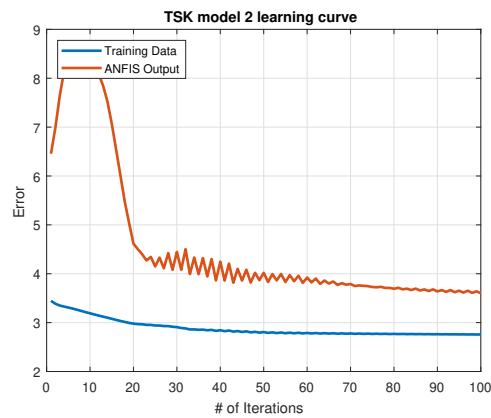


Figure 7: Learning Curve Model 2 TrnError,ValError

Ενώ η κατανομή του prediction error φαίνεται παρακάτω.

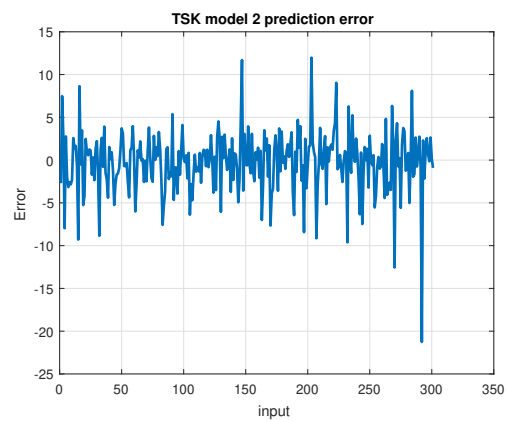


Figure 8: TSK model 2 Prediction Error

0.2.3 TSK model 3

Τα μοντέλα 3 και 4 έχουν πολυωνυμική έξοδο οπότε στο αρχικό ασφές σύστημα επιλέγεται ως έξοδος γραμμική έξοδος.

Πριν-εκαπαίδευση συναρτήσεις συμμετοχής

. Στην συνέχεια με back-propagation,

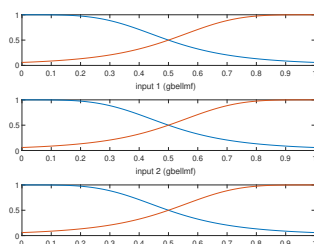


Figure 9: Γενική μορφή gbellmf τριών εισόδων-model 3 πριν την εκπαίδευση

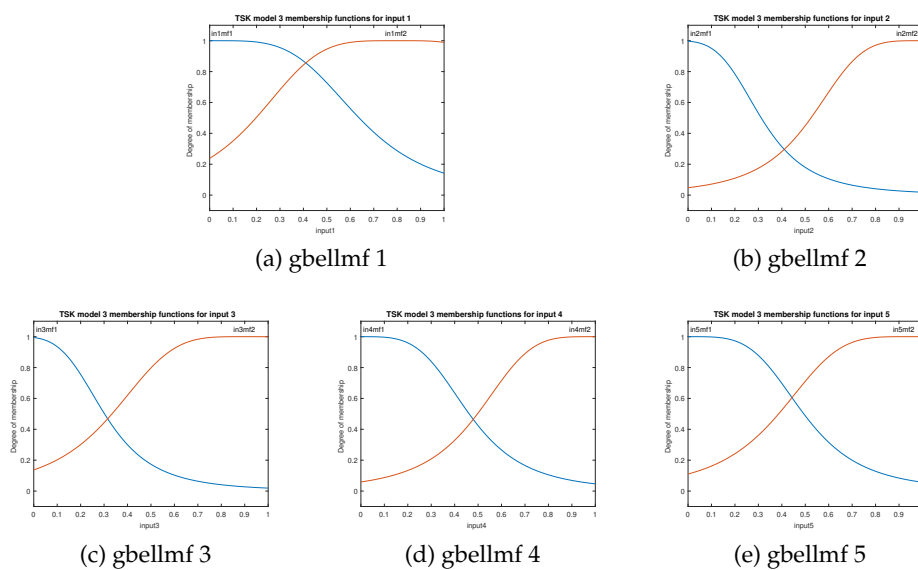


Figure 10: Input Mfs after training

Learning Curve

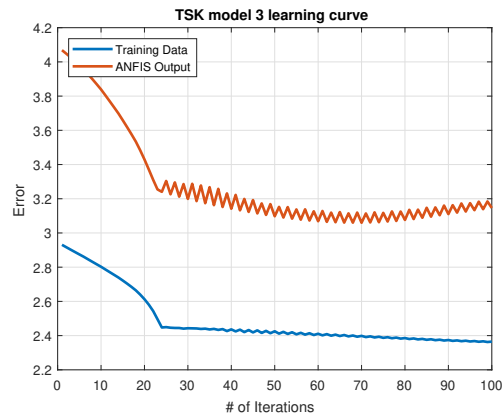


Figure 11: Learning Curve Model 3 TrnError,ValError

Prediction Error

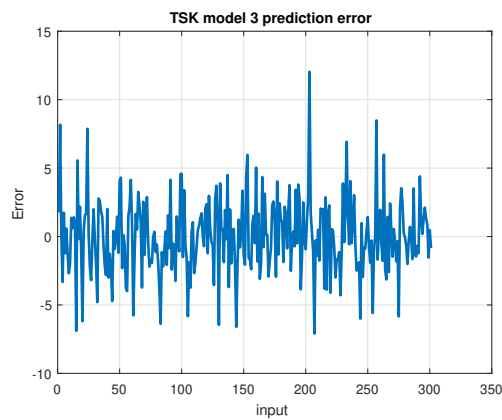


Figure 12: TSK model 3 Prediction Error

0.2.4 TSK model 4

Προ-εκαπαιδευμένες συναρτήσεις συμμετοχής

. Στην συνέχεια με back-propagation,

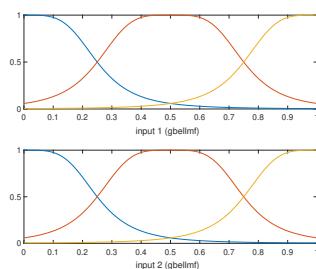
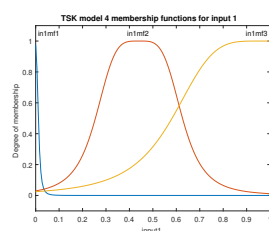
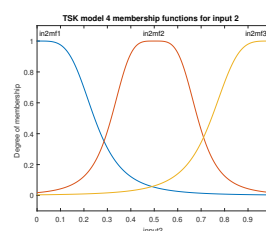


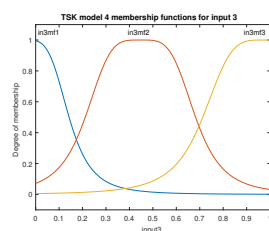
Figure 13: Γενική μορφή gbellmf τριών εισόδων-model 4 πριν την εκπαίδευση



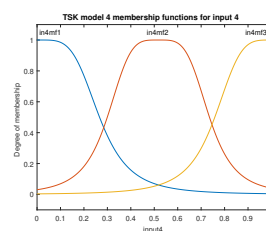
(a) gbellmf 1



(b) gbellmf 2



(c) gbellmf 3



(d) gbellmf 4

Figure 14: Input Mfs after training

Learning Curve

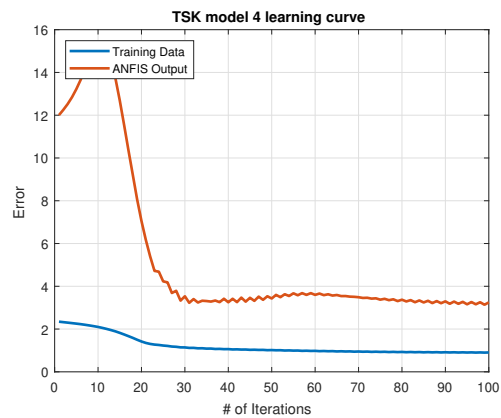


Figure 15: Learning Curve Model 4 TrnError, ValError

Prediction Error

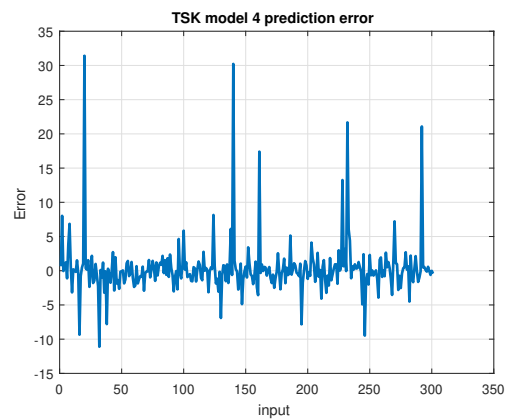


Figure 16: TSK model 4 Prediction Error

0.2.5 Τελική αξιολόγηση των μοντέλων - Δείκτες απόδοσης & Συμπεράσματα

#TSK	# Of In. MFs	Out.Type	RMSE	NMSE	NDEI	R^2
TSK_1	2	Singleton	0,71916	0,83377	0,63402	0,66211
TSK_2	3	Singleton	3,5005	2,6932	3,9961	3,8397
TSK_3	2	Polynomial	0,28084	0,16623	0,36598	0,33789
TSK_4	3	Polynomial	0,52994	0,40771	0,60496	0,58129

Παρατηρούμε ότι τα μοντέλα 1,3,4 είναι πολύ κοντά μεταξύ τους σε αντίθεση με το μοντέλο 2 που φαίνεται να υστερεί κατά πολύ στην εκτίμηση της τιμής του χαρακτηριστικού στόχος. Αυτό μάλλον καταλογίζεται στο γεγονός ότι μάλλον 3 συναρτήσεις συμμετοχής στην έξοδο μάλλον δεν μπορούν καλά να προβλέψουν την έξοδο μιας και στην πολυωνυμική έξοδο το μοντέλο με τις 2 συναρτήσεις συμμετοχής εισόδου αποδίδει πολύ καλύτερα συγκριτικά με το αντίστοιχο μοντέλο με τις τρεις συναρτήσεις συμμετοχής εισόδου. Μάλιστα αυτό το μοντέλο αποδίδει καλύτερα από όλα με $RMSE = 0.28$ που είναι μια σχετικά καλή τιμή απόκλισης από τις εκτιμώμενες από την πραγματική τιμή μιας και υπήρχαν μοντέλα με πολύ χειρότερα αποτελέσματα.

Βέβαια πρέπει να γίνει αντιληπτό πώς γενικότερα, η χρήση γραμμικής πολυωνυμικής εξόδου βελτιώνει το αποτέλεσμα, ανεξάρτητα από τον αριθμό των συναρτήσεων συμμετοχής, το οποίο είναι λογικό, καθώς δίνει τη δυνατότητα να χρησιμοποιούνται πιο ακριβή αποτελέσματα στην έξοδο του μοντέλου μας.

Τέλος, στο μοντέλο 4 φαίνεται η χαμπύλη εκμάθησης να συγκλίνει σχεδόν αμέσως. Ενώ σε ορισμένα άλλα μοντέλα εμφανίζεται ένα μεταβατικό φαινόμενο που μοιάζει με αδυναμία σύγκλισης του μοντέλου μιας και παρατηρούνται φαινόμενα ταλάντωσης αν και στο τέλος όλα τα μοντέλα συγκλίνουν.

Η γρήγορη σύγκλιση ενός μοντέλου σε χαμηλό σφάλμα εκτίμησης σημαίνει 2 πράγματα πρώτον ότι η περαιτέρω εκπαίδευση είναι περιττή και δεύτερον ότι χρειαζόμαστε ένα άλλο **dataset** με το οποίο συγκρίνουμε τις εκτιμήσεις και στην συγκεκριμένη περίπτωση το **validation set** το οποίο θα μας βοηθά να συνεχίζουμε την εκπαίδευση χωρίς όμως να φτάνουμε σε σημείο **overfitting** μηδενίζοντας το σφάλμα και οδηγώντας το μοντέλο ουσιαστικά να μιμείται τα δεδομένα που έχει. Σε αυτή την περίπτωση του **overfitting** νέα διαφορετικά δεδομένα θα οδηγούν σε απρόβλεπτα και συνήθως μεγάλα σφάλματα.

0.3 Εφαρμογή σε μεγάλη διάστασης δεδομένα | **Superconductivity**

Το δεύτερο μέρος της εργασίας είναι μια πιο εκτενής προσέγγιση του προβλήματος της παλινδρόμησης με χρήση TSK-Nnets. Έτσι το **dataset** που χρησιμοποιείται είναι το **Superconductivity** από το **UCI repository** και συγκεκριμένα το **train.csv** αρχείο. Το συγκεκριμένο αρχείο περιέχει 21263 δείγματα με 81 χαρακτηριστικά εκπαίδευσης καθώς και 1 χαρακτηριστικό στόχου η κρίσιμη θερμοκρασία των **superconductors** η οποία και είναι το χαρακτηριστικό προς εκτίμηση στην παρούσα εργασία.

Έτσι κρίνεται απαραίτητο να γίνει επιλογή των πιο σημαντικών χαρακτηριστικών από όλα του **dataset** για να έχουμε λειτουργικό και λιγότερης υπολογιστικής πολυπλοκότητας TSK μοντέλο. Για αυτό γίνεται η χρήση του αλγορίθμου **relief** ο οποίος προσδίδει βάρος στα δεδομένα και δημιουργεί με αυτό ένα **ranking** στα **indices**.

0.3.1 Data Preprocessing

Για αυτό το κομμάτι υλοποιήθηκε η συναρτηση **normalize data** η οποία όμως δεν χρησιμοποιήθηκε καθώς η συνάρτηση **split_scale** η οποία κάνει και αρχικό **shuffling** στα δεδομένα με χρήση **randperm** στην επιλογή δεικτών και κανονικοποίηση με βάση τα δεδομένα του **trainset** αλλά και χωρισμό του **dataset** στα παρακάτω υποσύνολα:

- 60% σύνολο εκπαίδευσης - **training data (trainData)**
- 20% σύνολο επαλήθευσης - **validation data (validationData)**
- 20% σύνολο ελέγχου - **testData (testData)**

Στην συνέχεια εφαρμόζεται ο αλγόριθμος **Relief** μέσω της προυπάρχουσας συνάρτησης του **MATLAB relief** και επιλέγεται να λειτουργεί με αριθμό γειτόνων 100. Η εντολή αυτή εκτελείται μία φορά στην αρχή μιας και είναι χρονοβόρα και στην συνέχεια χρησιμοποιούνται τα **ranking indices** στο κομμάτι του **cross validation data split** της εκκίνησης και της εκπαίδευσης του νευρωνικού.

0.3.2 5-fold cross validation

Το κομμάτι του **cross-validation** είναι μια τεχνική που εφαρμόζεται στα δεδομένα κατά το κομμάτι του αλγορίθμου **grid Search** που υλοποιείται με σκοπό την επιλογή βέλτιστων παραμέτρων για το **fis** μοντέλο.

0.3.3 Grid Search

Το πρόβλημα που υπάρχει στην αντιμετώπιση προβλημάτων με δεδομένα πολλών διαστάσεων με την χρήση ασαφών νευρωνικών δικτύων είναι το λεγόμενο **rule-**

explosion. Συγκεκριμένα η έκρηξη των IF-THEN κανονών απο του οποίους απαρτίζεται το δίκτυο με σκοπό να μπορέσουν να δημιουργήσουν όλες τις συσχετίσεις με τα δεδομένα και να οδηγήσουν σε μια μορφή εξόδου. Έτσι στην συγκεκριμένη περίπτωση η αρχικοποίηση του ασαφούς νευρωνικού συστήματος δεν γίνεται με την χρήση **grid Partitioning** (περίπτωση όπου ο αριθμός κανόνων αυξάνεται εκθετικά σε σχέση με το πλήθος εισόδων) αλλά με την χρήση **Substarctive clustering** με σκοπό την ομαδοποίηση δεδομένων για δημιουργηθούν εν τέλει οι IF-THEN κανόνες.

Σε αυτό το κομμάτι υπεισέρχεται η έννοια του **cross validation data processing (splitting)**. Συγκεκριμένα εκτελώ **5-fold cross validation** στα δεδομένα για διαφορετικές τιμές της ακτίνας ομαδοποίησης αλλά και διαφορετικό αριθμό επιλεγμένων χαρακτηριστικών, δύο μεταβλητές οι οποίες είναι ο στόχος εύρεσης του κομμάτι **grid Search**. Ουσιαστικά αυτή η αναζήτησ πλέγματος έχει ως στόχο την επιλογή του καλύτερου αντιπροσωπευτικού αριθμού χαρακτηριστικών του **dataset** καθώς και της καλύτερης ακτίνας ομαδοποίησης. Και ως μέθοδο αξιολόγησης χρησιμοποιείται η διασταυρώμενη επικύρωση (**cross-validation**) με την οποία σπάμε το **dataset** με διαφορετικό τρόπο σε κάθε **fold** και έτσι παίρνουμε όλους τους πιθανούς συνδυασμούς δεδομένων εκπαίδευσης για συγκεκριμένο αριθμό χαρακτηριστικών κάθε φορά και για συγκεκριμένη ακτίνα ομαδοποίησης κάθε φορά.

Πιο συγκεκριμένα στο κομμάτι της αναζήτησης πλέγματος χρησιμοποιήθηκαν όλοι οι παρακάτω πιθανοί συνδυασμοί:

- **Number of** χαρακτηριστικών : 10, 15, 20, 25
- **r_a** : 0.3, 0.4, 0.5, 0.6 ... 0.8,0.9

Με βάση την βιβλιογραφία προτιμήθηκαν ακτίνες ομαδοποίησης που να είναι μεγαλύτερες του 0,3 αλλά για μεγάλο αριθμό χαρακτηριστικών μικρές ακτίνες ομαδοποίησης παρουσίαζαν κακά αποτελέσματα. Γενικά ο αριθμός κανόνων επηρεάζει σε μεγάλο βαθμό τους κανόνες που δημιουργούνται και πρέπει να πηγαίνει αναλογικά με την ακτίνα ομαδοποίησης δεδομένων για να μην έχουμε τεράστιο αριθμό ασαφών κανόνων.

Τέλος ο αριθμός των χαρακτηριστικών επιλέχθηκε με το σκεπτικό να έχω το πολύ το 20% των χαρακτηριστικών.

Η διαδικασία του αλγορίθμου αναζήτησης πλέγματος περιλαμβάνει τα εξής:

- Δημιουργία ενός μητοειδούς (**error_grid**) για την αποθήκευση του σφάλματος σε κάθε επανάληψη του αλγορίθμου.
- Διαχωρισμός του **training set** με την χρήση του **cvpartition** σε 5 folds. Και έτσι χωρισμός του σε **chunk** (κομμάτι) εκπαίδευσης και **chunk** επαλήθευσης.
- Δημιουργία του **fis**(Ασαφές σύστημα διεπαφής) ώστε να χρησιμοποιεί **substarctive clustering** με την επιλογή απο το μητροειδές **Params** της ακτίνας ομαδοποίησης **r_a** καθώς και με είσοδο δεδομένα από το **training**

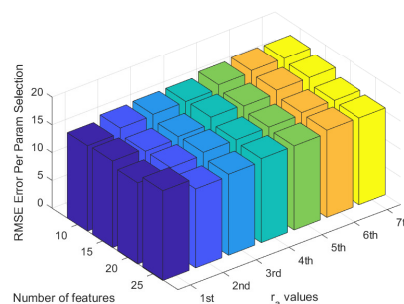
chunk αλλά με επιλεγμένο αριθμό features (number of Features) και αξιολογημένα από την **relief**. Γίνεται έλεγχος αν είμαστε εντός ορίου 1 και 100 ασαφών κανόνων.

- Για κάθε fold εκτελείται εκπαίδευση μέσω **anfis** για 40 εποχές και ελέγχεται το σφάλμα μεταξύ των εκτιμήσεων του μοντέλου και των δεδομένων επαλήθευσης του συνολικού αρχικού κανονικοποιημένου **dataset**.
- υπολογίζονται οι μετρικές που βρέθηκαν στο προηγούμενο κομμάτι και συμπληρώνεται το **error_grid** μητροειδές με τον μέσω όρο των R^2 μετρικών και των **RMSE** μετρικών

Η συνολική διάρκεια του αλγορίθμου έφτασε τα 10.0000 δευτερόλεπτα. Και το πλέγμα των σφαλμάτων για κάθε τιμή αριθμού επιλεγμένων χαρακτηριστικών καθώς και ακτίνες ομαδοποίησης φαίνεται παρακάτω. Στο παραπάνω οι 7 κλά-

```
val(:, :, 1) =
    0.7939    0.7687    0.7634    0.7619    0.7395    0.7301    0.7318
    0.7879    0.8019    0.7922    0.7580    0.7656    0.7482    0.7477
    0.8137    0.8012    0.7888    0.7891    0.7889    0.7794    0.7814
    0.7704    0.8237    0.8145    0.8004    0.8002    0.7851    0.7873

val(:, :, 2) =
    15.3765    16.2887    16.4763    16.5278    17.2882    17.5963    17.5413
    15.5963    15.0756    15.4414    16.6641    16.3997    16.9975    17.0135
    14.6190    15.1017    15.5670    15.5537    15.5637    15.9100    15.8369
    16.1693    14.2227    14.5869    15.1320    15.1412    15.7025    15.6198
```



(a) val1-Error_grid_ R^2
val-2-Error_grid_RMSE

(b) Error_grid_RMSE-Bar3d

Figure 17: Error_Grid

σεις ακτίνων είναι οι προαναφερθείσες. Φυσικά τυπώνονται σε κάθε επανάληψη **learning curves** και **prediction errors**. Και εμφανίζονται και ενδιαφέρουσες περιπτώσεις όπου το μοντέλο δεν μαθαίνει όπως η παρακάτω: Ωστόσο κατά την

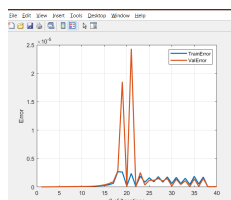


Figure 18: Περίπτωση που το μοντέλο δεν μαθαίνει

πλειοψηφία των περιπτώσεων το μοντέλο μαθαίνει και είχαμε σωστές καμπύλες. Βέβαια επειδή πρόκειται για 200 γραφήματα παραλείπονται τα περισσότερα. Το σχόλιο στο κομμάτι που το μοντέλο δεν μαθαίνει είναι πως είτε έχει γίνει μείωση

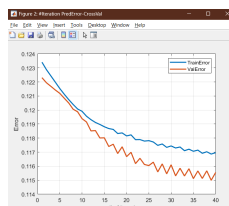


Figure 19: Περίπτωση που το μοντέλο μαθαίνει

του σφάλματος σε επίπεδα που μιλάμε για **overfitting** ή κάποιο πρόβλημα υπήρξε με τους κανόνες ή ίσως με τα δεδομένα που πιθανόν σε εκείνο το run να μην είχαν κανονικοποιηθεί σωστά.

Τέλος δημιουργήθηκε και πίνακας κανόνων που δημιουργούνται για κάθε fold ο οποίος παρουσιάζεται στο παρακάτω 3d Barchart. RadII είναι οι κλάσεις της ακτίνας ομαδοποίησης(ορος που χρησιμοποιείται στο genfis2 στον κώδικα χρησιμοποιήσα genfis μιας και η matlab δεν υποστηρίζει πλέον το genfis2 ή μάλλον το χρησιμοποιεί εντός του genfis).

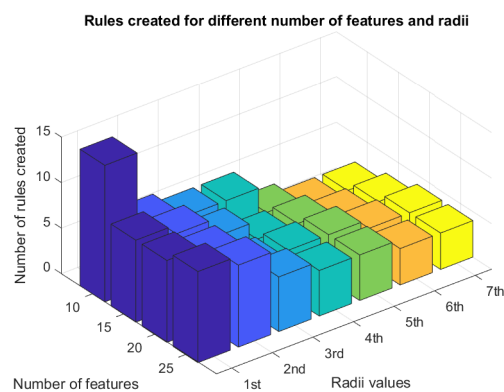


Figure 20: Αριθμός κανόνων που δημιουργήθηκαν από διαφορετικές παραμέτρους

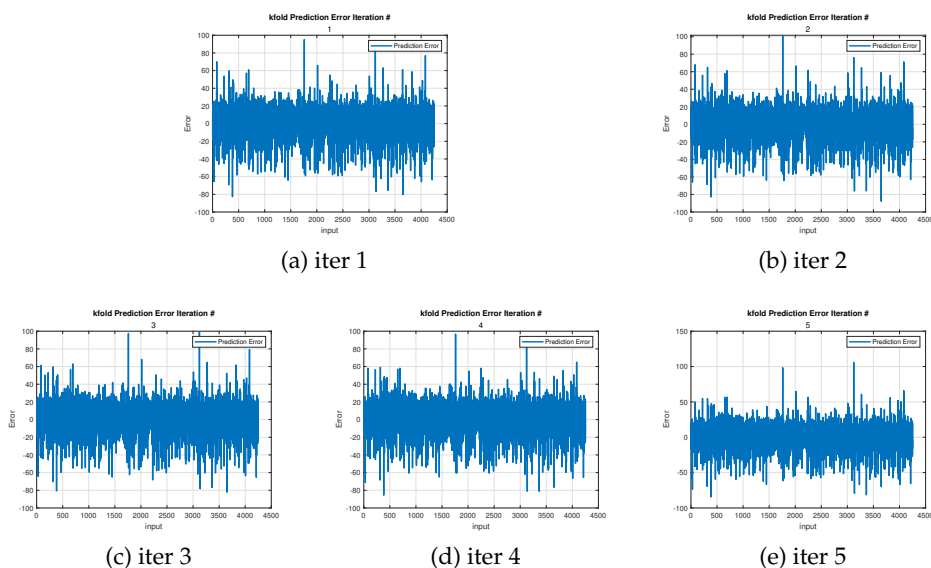


Figure 21: Kfold Prediction Errors with random params

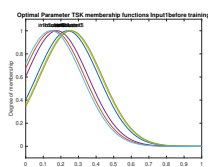
0.3.4 Συμπεράσματα από το **Grid Search**

Λαμβάνοντας υπόψιν το πλέγμα σφαλμάτων καθώς και ότι το μεγάλο πλήθος επιλεγμένων χαρακτηριστικών απαιτεί μεγαλύτερη ακτίνα ομαδοποίησης δεδομένου ότι δεν χρησιμοποιήθηκε πάνω από το $1/3$ των χαρακτηριστικών του **dataset**. Η επιλογή έγινε με 25 features και 0.5 clustering radius μιας και αυτές οι επιλογές είχαν τις καλύτερες μετρικές RMSE, R^2 .

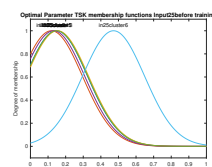
0.3.5 Optimal Model

Στο συγκεκριμένο κομμάτι παρουσιάζω τα αποτελέσματα του βέλτιστου μοντέλου για το δεύτερο **dataset** με την όπως παρουσιάστηκαν και τα μοντέλα στο πρώτο κομμάτι της εργασίας.

Πριν-εκπαίδευση συναρτήσεων συμμετοχής



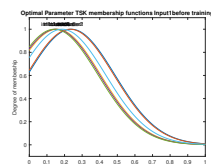
(a) gbellmf 1



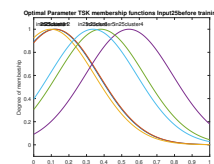
(b) gbellmf 2

Μετα-εκπαίδευση συναρτήσεων συμμετοχής

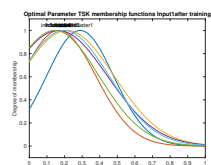
.



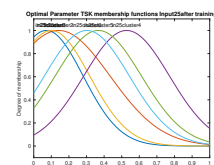
(a) gbellmf 1



(b) gbellmf 2



(c) gbellmf 3



(d) gbellmf 4

Figure 22: Input Mfs after training

Learning Curve

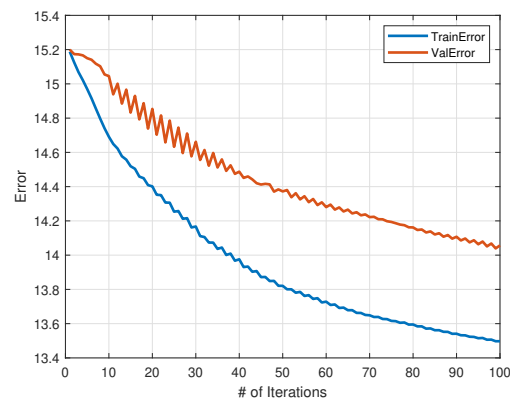
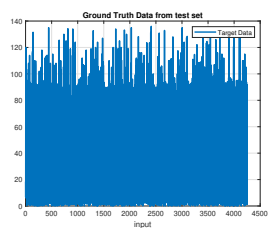
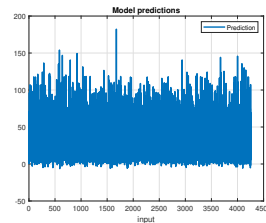


Figure 23: Learning Curve Model 4 TrnError, ValError



(a) Ground Truth Data



(b) Predictions

Figure 24: Σύγκριση Ground Truth με Predictions Μοντέλου

Prediction Error

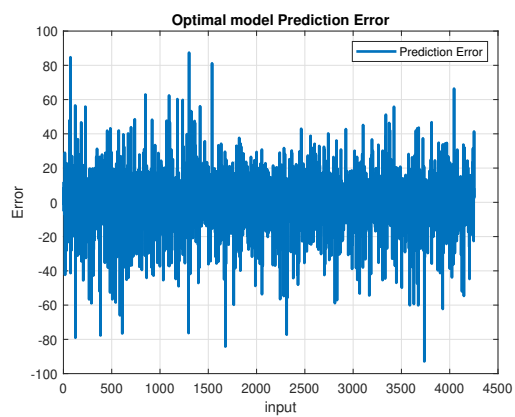


Figure 25: Optimal Model Prediction Error

```
End of training 5 iteration.....
Perd =
  0.8136 + 0.0000i
 14.7592 + 0.0000i
-13.7592 + 0.0000i
 0.0000 + 3.7093i
Elapsed time is 1397.767649 seconds.
```

Figure 26: Result Tables

0.4 Σύγκριση **Grid Partitioning** με **Subtractive Clustering**

Είναι εμφανές απο τα αποτελέσματα πως η μία τεχνική είναι καλύτερη για μικρότερο αριθμό δεδομένων μια και με το **grid partitioning** είχαμε πολύ μικρό σφάλμα ωστόσο τα ασαφή νευρωνικά δίκτυα εμφανίζουν υψηλή πολυπλοκότητα όταν αυξάνονται πολύ οι συναρτήσεις συμμετοχής που τα μοντελοποιούν. Έτσι με 25 χαρακτηριστικά που ήταν το βέλτιστο τελικό μοντέλο για το 2 κομμάτι της εργασίας θα είχαμε με 2 συναρτήσεις συμμετοχής 2^{15} ασαφής κανόνες και με τρεις 3^{15} νούμερα αρκετά μεγάλα στο να αποτρέπουν την εκπαίδευση του δικτύου. Έτσι το **subtractive clustering** σαν τεχνική παρόλο που τελικά είχε σχετικά μη ικανοποιητικό RMSE είναι καλή πρακτική για δεδομένα μεγάλης διαστασιμότητας.

0.5 Παραδοτέο

Στο κομμάτι του παραδοτέου περιλαμβάνονται τα εξής

- **Report9449.pdf** : Αρχείο Αναφοράς εργασίας
- **img** : αρχείο με όλες τις εικόνες που παρήχθησαν κατά την εκπόνηση της εργασίας.
- **datasets** : φάκελος με τα δεδομένα
- **Testing** : αρχείο με **Testing scripts** που δεν είναι για βαθμολόγηση
- **ex-PDF** η εκφώνηση της εργασίας
- **normalize.m** , **split_scale.m** : αρχεία προεπεξεργασίας δεδομένων
- **TSK_Regression.m** : script για το πρώτο κομμάτι της εργασίας
- **TskReg2_GridSearch.m**, **OptimalParameterModelTskReg.m** : scripts για το δεύτερο κομμάτι της εργασίας
- **rule_grid_bar_chart.png**: εικόνα αποτύπωσης πλήθους κανόνων ανά παραμέτρους.
- **WorkspaceResultValues.m**: Αποθηκευμένες οι τελικές τιμές στο **workspace** μετά το **GridSearch** (γιατι ήταν χρονοβόρα διαδικασία - ίσως να μην αποθηκεύτηκε σωστα).