Εργασία 2 , Επίλυση προβλήματος παλινδρόμησης με χρήση μοντέλων TSK

Νίκος Λαδιάς

Αύγουστος 2021

1 Εισαγωγή

Στόχος της εργασίας αυτής είναι να διερευνηθεί η ικανότητα των μοντέλων TSK στη μοντελοποίηση πολυμεταβλητών, μη γραμμικών συναρτήσεων. Συγκεκριμένα, επιλέγονται δύο σύνολα δεδομένων από το UCI repository με σκοπό την εκτίμηση της μεταβλητής στόχου από τα διαθέσιμα δεδομένα, με χρήση ασαφών νευρωνικών μοντέλων. Το πρώτο σύνολο δεδομένων θα χρησιμοποιηθεί για μια απλή διερεύνηση της διαδικασίας εκπαίδευσης και αξιολόγησης μοντέλων αυτού του είδους, καθώς και για μια επίδεξη τρόπων ανάλυσης και ερμηνείας των αποτελεσμάτων. Το δεύτερο, πολυπλοκότερο σύνολο δεδομένων θα χρησιμοποιηθεί για μια πληρέστερη διαδικασία μοντελοποίησης, η οποία θα περιλαμβάνει μεταξύ άλλων προεπεξεργαστικά βήματα όπως επιλογή χαρακτηριστικών (feature selection), καθώς και μεθόδους βελτιστοποίησης των μοντέλων μέσω της διασταυρωμένης επικύρωσης (cross validation).

Γενικά για τα αρχεία κώδικα της εργασίας :Υπάρχουν δύο αρχεία για τα ερωτήματα της εργασίας Exe1 και Exe2 τα οποία τρέχουν το κύριο μέρος του ερωτήματος ένα και δύο αντίστοιχα. Τα υπόλοιπα είναι αρχεία .m τα οποία υλοποιούνε συναρτήσεις εκτός απο το bestmodel.m που .

Περιεχόμενα

2 Πρώτο μέρος εργασίας : Εκπαίδευση και αξιολόγηση μοντέλων TSK

2.1 Διαχωρισμός σε σύνολα εκπαίδευσης-επικύρωσης-ελέγχου

Αφού φορτωθούν τα ζητούμενα δεδομένα της άσκησης , γίνεται όπως ζητείται ένας διαχωρισμός των δεδομένων σε τρία μη επικαλυπτόμενα υποσύνολα . Το πρώτο χρησιμοποιείται για εκπαίδευση, το δεύτερο για επικύρωση και αποφυγή του φαινομένου υπερεκπαίδευσης και το τρίτο για τον έλεγχο της απόδοσης του τελικού μοντέλου. Αυτό γίνεται με την βοήθεια της συνάρτησης split_data. Η συνάρτηση αφού ανακατέψει τα δεδομένα , διαλέγει με τον τρόπο που προτείνεται στην εκφώνηση, απο το διάνυσμα τυχαία τοποθετημένων στοιχείων πλέον ,το πρώτο 60% των στοιχείων για το υποσύνολο εκπαίδευσης και απο 20% για τα υπόλοιπα υποσύνολα.

Αφού τα δεδομένα τεμαχιστούν καταλλήλως, ακολουθεί μια προεπεξεργασία των δεδομένων. Η συνάρτηση split_data δίνει δύο επιλογές ανάλογα την τιμή της μεταβλητής preprocess. Η πρώτη κανονικοποιεί τα δεδομένα σε μονάδα υπερκύβου. Η δεύτερη κανονικοποιεί τα δεδομένα γύρω απο κανονική κατανομή με μέση τιμή 0 και τυπική απόκλιση 1.

2.2 Εκπαίδευση TSK μοντέλων με διαφορετικές παραμέτρους

Δημιουργούνται τα 4 TSK μοντέλα της εκφώνησης,με τον αριθμό συναρτήσεων συμμετοχής και την αντίστοιχη μορφή εξόδου με την βοήθεια της συνάρτησης genfis1. Έπειτα ξεκινάει η διαδικασία εκπαίδευσης των μοντέλων , τα μοντέλα (αντικείμενα FIS) μαζί τα δεδομένα εκπαίδευσης εισάγονται σαν ορίσματα μαζί με τον πίνακα αρχικοποίησης και τα δεδομένα αξιολόγησης στην συνάρτηση anfis. Ο αλγόριθμος εκπαίδευσης χρησιμοποιεί έναν συνδυασμό των μεθόδων ελαχίστων τετραγώνων και οπισθοδιάδοσης βαθμωτής καθόδου για τη μοντελοποίηση του συνόλου δεδομένων εκπαίδευσης. Η εκπαίδευση γίνεται για 100 εποχές.

Έπειτα υπολογίζονται όλοι οι δείκτες απόδοσης αφού με την βοήθεια της evalfis, υπολογίζεται η πραγματική έξοδος του Fuzzy Inference συστήματος μας με είσοδο τα δεδομένα ελέγχου απόδοσης του μοντέλου. Έτσι, δημιουργούνται όλα τα ζητούμενα διαγράμματα σε αριθμημένες εικόνες και παρουσιάζεται ο τελικός πίνακας των αποτελεσμάτων με τους δείκτες απόδοσης για κάθε μοντέλο.

Παραθέτονται για κάθε μοντέλο:

- Τα διαγράμματα με τις τελικές μορφές των ασαφών συνόλων
- Τα διαγράμματα του learning curve για training και validation error
- Τα διαγράμματα των σφαλμάτων πρόβλεψης

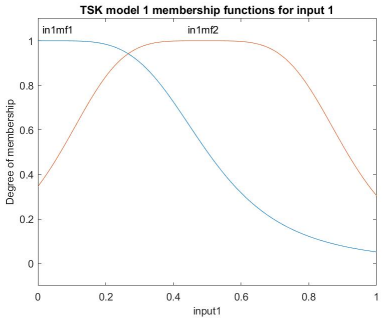
Παραθέτουμε και τον πίνακα των μετρικών των μοντέλων που ζητούνται:

	1ο μοντέλο TSK	2ο μοντέλο TSK	3ο μοντέλο TSK	4ο μοντέλο TSK
R^2	0.61198	-0.2682	0.79814	0.72821
RMSE	4.2979	7.7701	3.1	3.5971
NMSE	0.38802	1.2682	0.20186	0.27179
NDEI	0.62291	1.1261	0.44929	0.52134

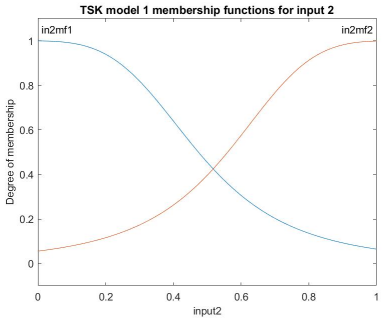
2.3 Συμπεράσματα

Απο τα παραπάνω έτσι , βγαίνουν τα παρακάτω συμπεράσματα:

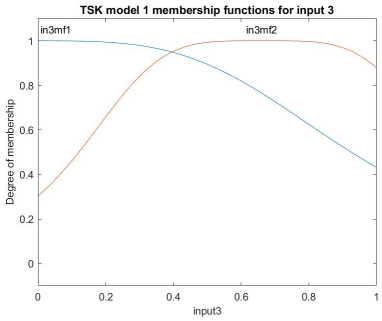
1. Το καλύτερο μοντέλο είναι το 3ο
2. Το χειρότερο μοντέλο είναι το 4ο (με μικρή διαφορά απο το 1ο)
3. Τα μοντέλο 3, με πολυωνυμική έξοδο παρουσιάζει καλύτερη συμπεριφορά, με μικρότερο RMSE, αλλά και καλύτερο R^2.
4. Το μοντέλο 2 παρουσιάζει μεμονωμένα την πιο ακραία τιμή prediction error.
5. Αν και στα πρώτα τρία μοντέλα υπάρχει μια παλινδρόμηση απο ένα iteration της εκπαίδευσης γύρω απο κάποια τιμή, όσο αφορά το validation error, σε κανένα μοντέλο δεν φαίνεται να υπάρχει απο ένα σημείο και μετά υπερεκπαίδευση. Στο 4ο όμως μοντέλο απο το 60 iteration και μετά φαίνεται ξεκάθαρα το overfitting παρατηρώντας το validation error.
6. Γενικά, στα μοντέλα με δύο Membership functions έχουμε μικρότερη απόκλιση μεταξύ του Training Error και του Validation Error, όπως φαίνεται ξεκάθαρα στα διαγράμματα των Learning curves. Για παράδειγμα, στο 1ο μοντέλο, έχουμε απόκλιση περίπου $3.6 - 3.5 = 0.1$ ενώ στο 4ο μοντέλο, έχουμε απόκλιση περίπου $6 - 0.9 = 5.1$ (τελικές τιμές των errors).
7. Τα μοντέλα με τρεις Membership functions είναι πιο επιρρεπή σε υπερεκπαίδευση, μιας και επειδή έχουν περισσότερες συναρτήσεις συμμετοχής έχουν την δυνατότητα να 'μάθουν' καλύτερα το training dataset. Το γεγονός αυτό, φάνηκε όταν τρέξαμε το σκριπτάκι στο Matlab για αρκετές φορές, κάτι το οποίο έπρεπε να γίνει μιας και η όλη παραπάνω διαδικασία έχει στοχαστικά χαρακτηριστικά.



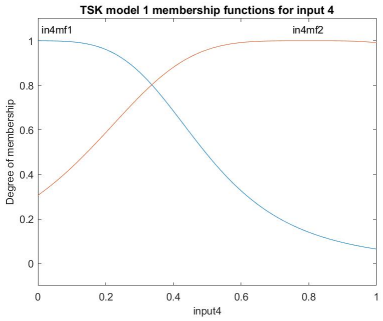
(a) Input 1



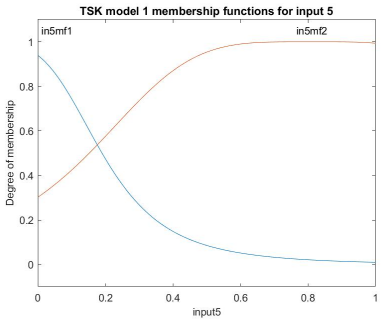
(b) Input 2



(c) Input 3

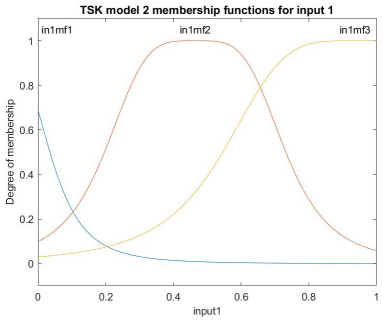


(d) Input 4

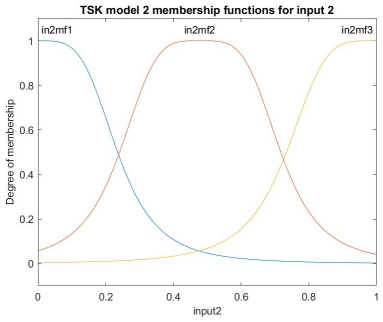


(e) Input 5

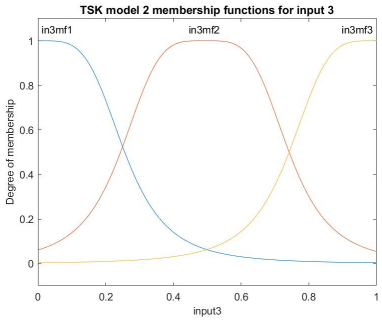
Εικόνα 1: TSK model 1 membership functions for all 5 inputs



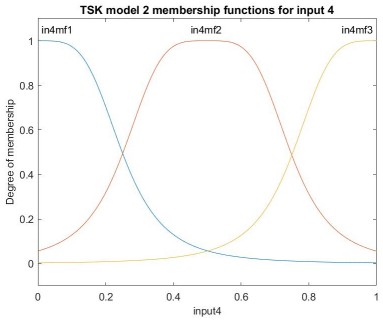
(a) Input 1



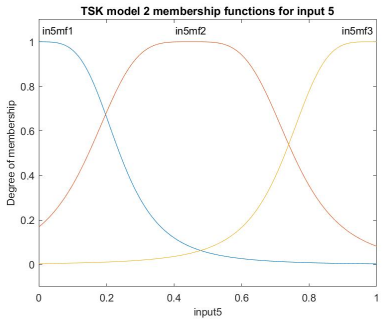
(b) Input 2



(c) Input 3

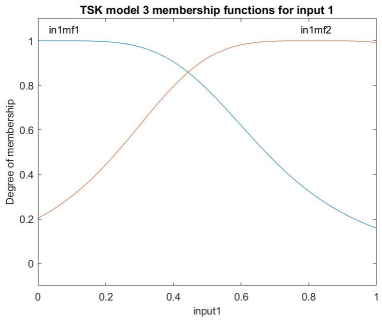


(d) Input 4

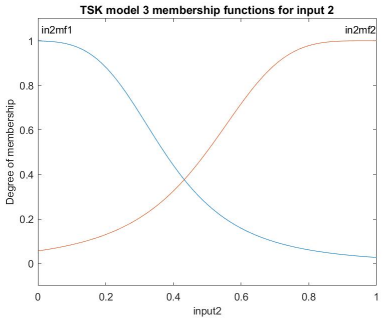


(e) Input 5

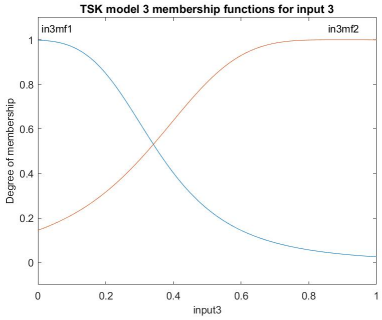
Εικόνα 2: TSK model 2 membership functions for all 5 inputs



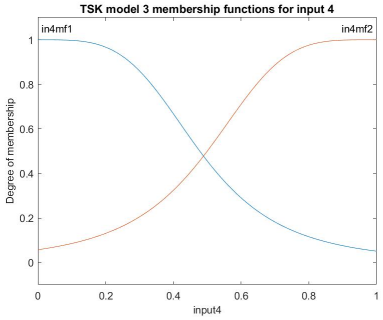
(a) Input 1



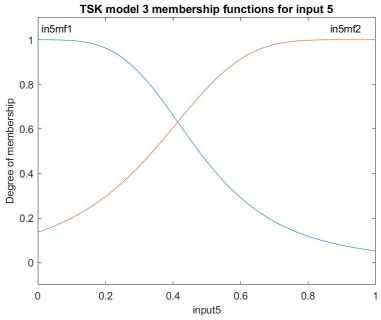
(b) Input2 2



(c) Input 3

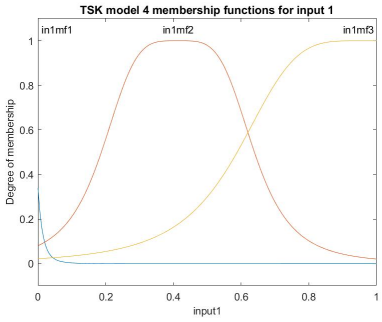


(d) Input 4

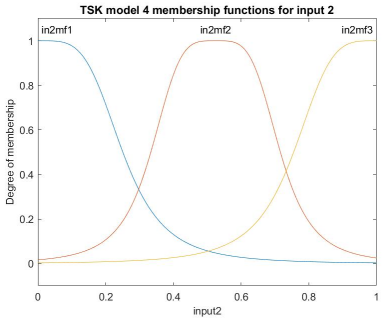


(e) Input 5

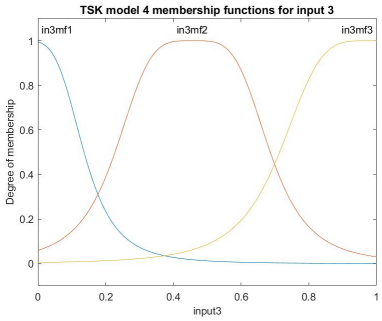
Εικόνα 3: TSK model 3 membership functions for all 5 inputs



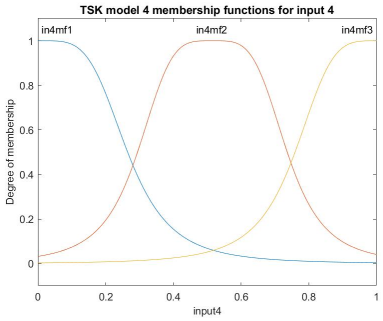
(a) Input 1



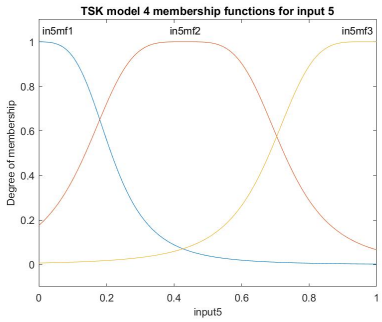
(b) Input2 2



(c) Input 3

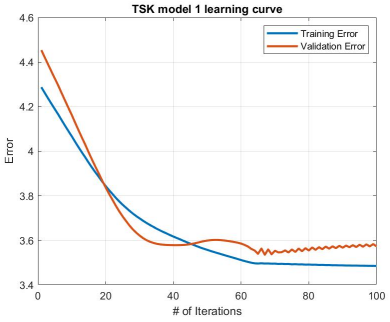


(d) Input 4

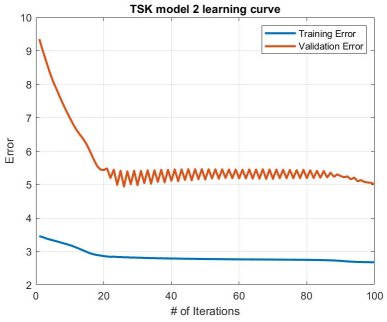


(e) Input 5

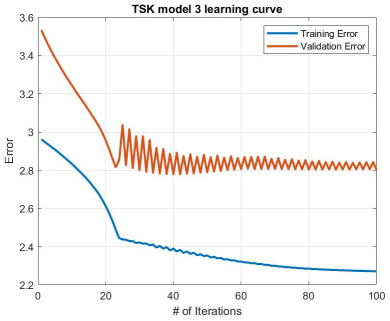
Εικόνα 4: TSK model 4 membership functions for all 5 inputs



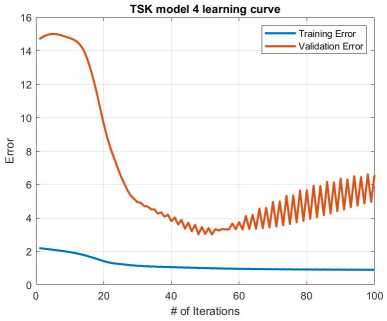
(a) Learning curves for model 1



(b) Learning curves for model 2

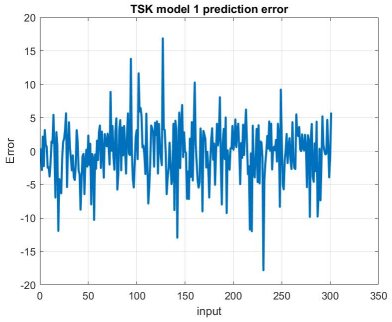


(c) Learning curves for model 3

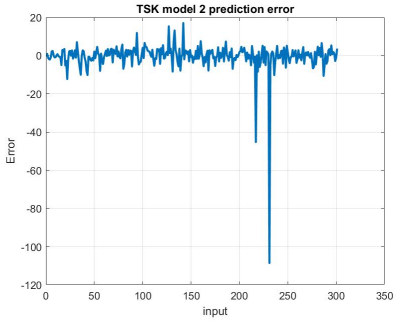


(d) Learning curves for model 4

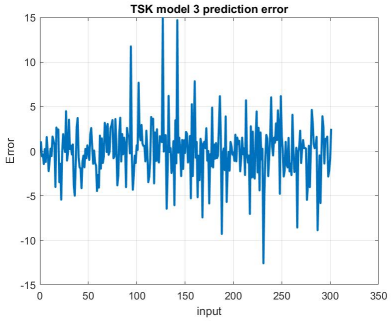
Εικόνα 5: Learning curves for all 4 models



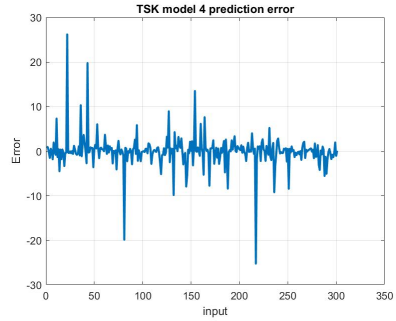
(a) Prediction errors for model 1



(b) Prediction errors for model 2



(c) Prediction errors for model 3



(d) Prediction errors for model 4

Εικόνα 6: Prediction errors for all 4 models

3 Εφαρμογή σε dataset με υψηλή διαστασιμότητα

3.1 Διαδικασία εύρεσης βέλτιστου μοντέλου TSK

Στο δεύτερο μέρος της εργασίας, θα χρησιμοποιήσουμε ένα dataset με υψηλότερο βαθμό διαστασιμότητας. Το dataset που χρησιμοποιήθηκε είναι το Superconductivty dataset από το UCI Repository, το οποίο περιέχει 21263 δείγματα και 81 μεταβλητές / features. Είναι προφανές ότι πρέπει να διαλέξουμε έναν μικρό αριθμό features που θα κρατήσουμε, λόγω της έκρηξης των κανόνων IF - THEN , ώστε το μοντέλο TSK να μπορεί να είναι λειτουργικό. Έτσι η πρώτη μας κίνηση είναι να τρέξουμε ένα script (Exe2.m) ώστε να μπορέσουμε να βρούμε τις βέλτιστες τιμές, των κρατημένων features και της ακτίνας r των clusters. Για την παραπάνω διαδικασία χρησιμοποιούμε τη μέθοδο της αναζήτησης πλέγματος.

Επίσης, στα features των data κάναμε κανονικοποίηση με την customμας συνάρτηση normaliseData(), η οποία κανονικοποιεί τα δεδομένα σε όλες τις στήλες εκτός απο την target, δηλαδή την τελευταία. Η μέθοδος ομαδοποίησης για τη δημιουργία των IF – THEN κανόνων επιλέχθηκε να είναι ο αλγόριθμος Subtractive Clustering (SC) και η επιλογή των χαρακτηριστικών έγινε με τον αλγόριθμο Relief.

Η λογική του κώδικα έχει ως εξής:

✓ Ορίζεται ο τρισδιάστατος πίνακας που έχει όλες τις τιμές των κρατημένων features και των ακτινών r των clusters. Χειροκίνητα γεμίζουμε τις τιμές του με τις δοκιμαστικές τιμές που επιλέξαμε .

✓ Λούπα στον τρισδιάστατο πίνακα , διαβάζουμε τις δύο παραμέτρους του πίνακα και έπειτα περνάμε στη διαδικασία cross validation με λούπα μέχρι την παράμετρο k. Η custom συνάρτηση μας crossValidationDatasets, ουσιαστικά επιστρέφει τα σύνολα δεδομένων που απαιτούνται για τη k-fold cross validation διαδικασία, τα δεδομένα χωρίζονται κατά 60%, 20% ,20% σε trainData , checkData, testData αντίστοιχα .

✓ Γίνεται επιλογή των features με τον αλγόριθμο Relief. Έπειτα επιβάλλεται να κρατήσουμε τα feautures που αναλογούν στην κάθε επανάληψη στο κάθε σέτ δεδομένων που έχουμε.

✓ Ορισμός του fis struct object όπου παιρνάμε καταλλήλως τα δεδομένα εκπαίδευσης.

✓ Εκτέλεση της εκπαίδευσης με χρήση της anfis για 250 εποχές. Παραγωγή γραφήματος καμπυλών σφάλματος εκπαίδευσης και επικύρωσης, υπολογισμός ζητούμενων μετρικών.

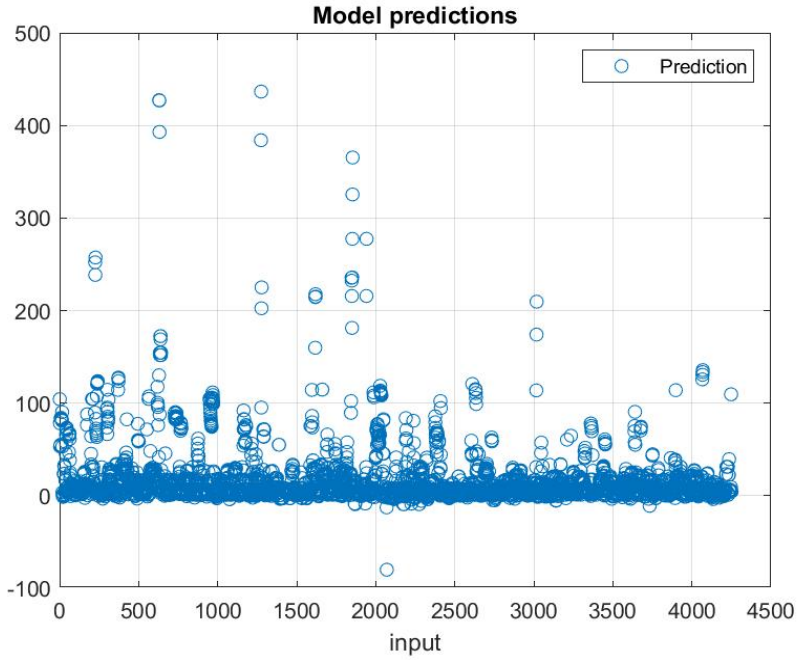
3.2 Βέλτιστο μοντέλο

Έχοντας τρέξει το script για την εύρεση του βέλτιστου μοντέλου, βρέθηκε πως αυτό παρουσιάζεται για 25 ως αριθμό χαρακτηριστικών και 0.7 ακτίνα τω clusters. Αφού υπολογιστεί έτσι το καλύτερο μοντέλο (βέλτιστος αριθμός features και ακτίνας r), δημιουργήθηκε ένα τελικό TSK μοντέλο όπου ελέγχουμε και την απόδοση του στο σύνολο ελέγχου. Η διαδικασία που τρέχει το script (optimalModelScript.m) είναι παρόμοια με αυτήν του Exe2.m , με μόνη διαφορά πως τώρα είναι γνωστό το μοντέλο και άρα δεν τρέχει κάποια διαδικασία εκπαίδευσης για όλα τα πιθανά μοντέλα. Έτσι, απλά υπολογίζονται τα ζητούμενα γραφήματα και δημιουργούνται οι αντίστοιχες εικόνες για το βέλτιστο μοντέλο που βρέθηκε. Ακολουθούν τα αποτελέσματα της εκπαίδευσης παρακάτω:

- Πίνακας δεικτών απόδοσης

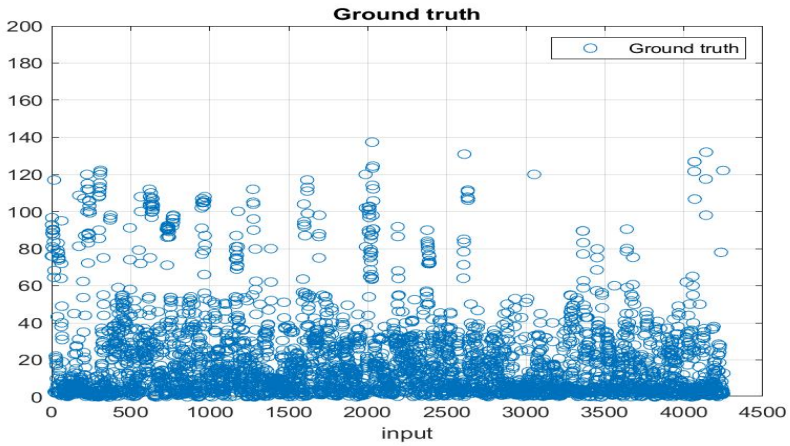
Δείκτης απόδοσης	R^2	RMSE	NMSE	NDEI
Απόδοση τιμής βέλτιστου μοντέλου TSK	0.5020	16.6347	0.4980	0.7057

- Προβλέψεις βέλτιστου μοντέλου



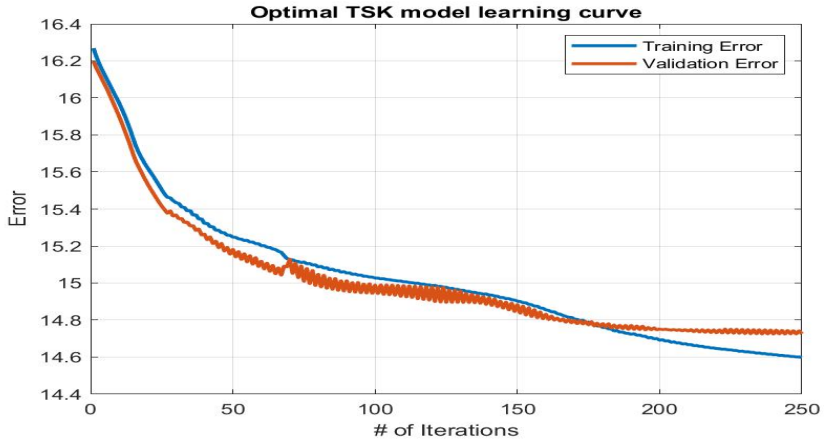
Εικόνα 7: Προβλέψεις μοντέλου

- Πραγματικές τιμές



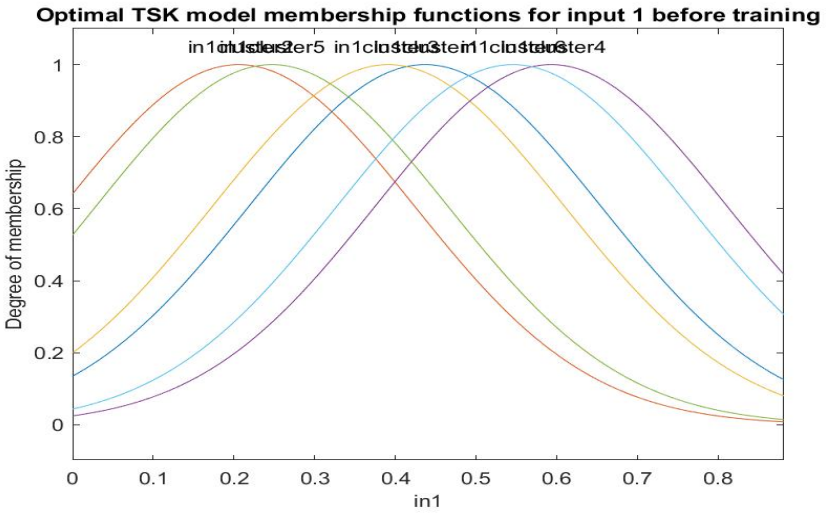
Εικόνα 8: Πραγματικές τιμές

- Καμπύλες εκμάθησης για training και validation sets

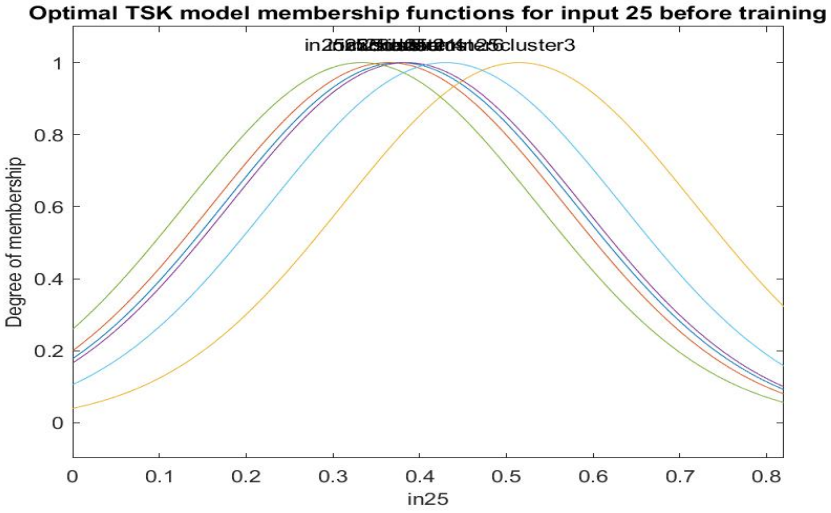


Εικόνα 9: Καμπύλες εκμάθησης βέλτιστου μοντέλου

- Ασαφή σύνολα στην αρχική τους μορφή

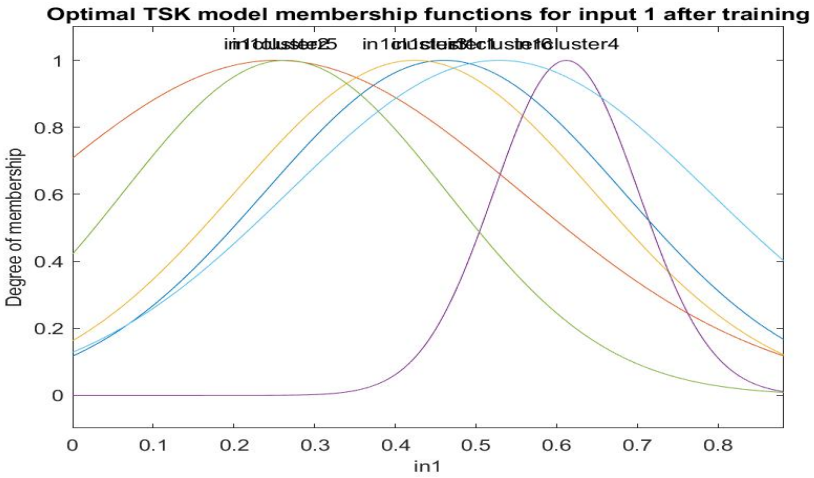


Εικόνα 10: Ασαφές μοντέλο για πρώτη είσοδο στην αρχική του μορφή

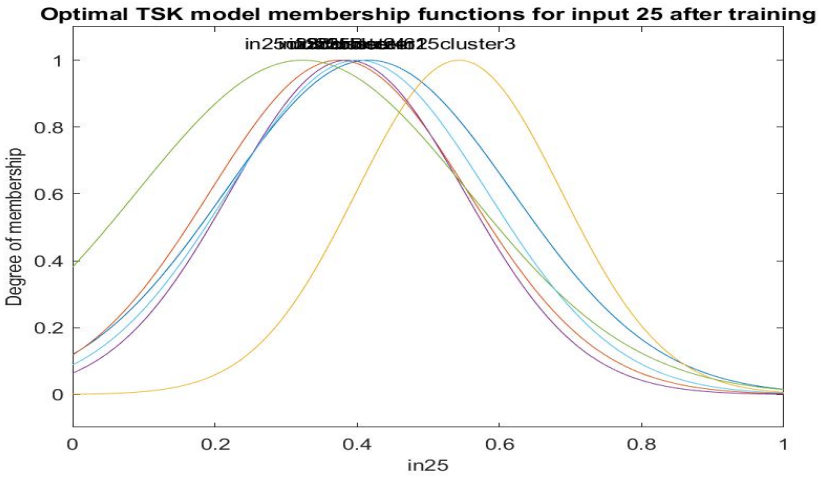


Εικόνα 11: Ασαφές μοντέλο για 25η είσοδο στην αρχική του μορφή

- Ασαφή σύνολα στην τελική τους μορφή



Εικόνα 12: Ασαφές μοντέλο για πρώτη είσοδο στην τελική του μορφή



Εικόνα 13: Ασαφές μοντέλο για 25η είσοδο στην τελική του μορφή

Κάποια τελικά συμπεράσματα για το βέλτιστο μοντέλο

Βλέποντας, τα διαγράμματα των Predicted Values και των Real Values, βλέπουμε ότι το μοντέλο προβλέπει αρκετά καλά το testing dataset. Επιπλέον, το πλήθος των κανόνων του βέλτιστου μοντέλου είναι 6 κανόνες και αυτό το βλέπουμε από το valFis.rule (αφού τρέξουμε το optimalModelScript.m στο workspace του Matlab βλέπουμε το αντικείμενο fis). Για τον ίδιο αριθμό κρατημένων χαρακτηριστικών με την μέθοδο του grid partitioning, αν για κάθε είσοδο είχαμε 2 ή 3 ασαφή σύνολα, θα είχαμε συνολικά 2^{25} ή 3^{25} κανόνες αντίστοιχα. Τα δύο τελευταία νούμερα, σε σχέση με το 6 έχουν τεράστια διαφορά (πολλές τάξεις μεγέθους κιόλας) και μας επιβεβαιώνουν πόσο μη πρακτικό και χρονοβόρο στη φάση του training, θα ήταν το μοντέλο μας, αν αντί για την μέθοδο του subtractive clustering χρησιμοποιούσαμε τη μέθοδο του grid partitioning.