Ασαφή Συστήματα

Επίλυση προβλήματος παλινδρόμησης με μοντέλα TSK

Υπατία Δάμη
ΑΕΜ:8606

08/10/2019

Σκοπός εργασίας:

Σκοπός αυτής της εργασίας είναι η επίλυση προβλημάτων παλινδρόμισης με χρήση διαφόρων TSK μοντέλων. Στα προβλήματα παλινδρόμισης παρέχεται ένα σετ δεδομένων για την εκπαίδευση και την αξιολόγιση του μοντέλου, το οποίο περιέχει τιμές(εισόδους), για όλες τις μεταβλητές του προβλήματος, αλλά και την έξοδο του υπό μελέτη συστήματος για έναν ικανοποιητικό αριθμό εισόδων. Σκοπός είναι στο τέλος της διαδικασίας αυτής το μοντέλο να είναι ικανό να προβλέψει τις τιμές(εξόδους) του προβλήματος ενώ του παρέχονται πλέον τιμές μόνο των μεταβλητών(εισόδων) του συστήματος. Το μοντέλο ουσιαστικά "εκπαιδεύεται" στο να αναγνωρίζει κάποια πρότυπα συμπεριφοράς των δεδομένων και σύμφωνα με αυτά να μπορεί να προβλέψει την έξοδο του συστήματος για οποιεσδήποτε τιμές εισόδων. Στην εργασία αυτή μελετώνται δύο προβλήματα παλινδρόμισης. Στο πρώτο πρόβλημα χρησιμοποιείται το Cobined Cycle Power Plant (CCPP) dataset το οποίο περιέχει 5 χαρακτηριστικά και μία έξοδο . Λόγω του μικρού αριθμού των χαρακτηριστικών του το dataset μας επιτρέπει να ακολουθήσουμε μια απλή διαδικασία μοντελοποίησης .Αντιθέτως, στο δεύτερο πρόβλημα της εργασίας καλούμαστε να χρησιμοποιήσουμε ένα πολύ πιο σύνθετο dataset, που αποτελέιται από 81 χαρακτηριστικά και 1 έξοδο. Το μεγάλο πλήθος χαρακτηριστικών δεν επιτρέπει την επίλυση του προβλήματος με απλή μοντελοποίηση και γιαυτό καλούμαστε να χρησιμοποιήσουμε διάφορες τεχνικές και αλγορίθμους για την απλοποίηση του. Αφού ολοκληρωθεί η εκπαίδευση και η αξιολόγιση των μοντέλων, συνθέτουμε διαγράμματα που να απεικονίζουν την απόδοση και τη συμπεριφορά του εκάστοτε μοντέλου και συγκρίνουμε την απόδοση τους .

Επίλυση προβλήματος με απλό dataset:

Στο πρώτο πρόβλημα καλούμαστε να μοντελοποιήσουμε τη συμπεριφορά ενός κινητήρα χρησιμοποιώντας Cobined Cycle Power Plant (CCPP) dataset της UCI repository. Το dataset περιέχει 4 χαρακτηριστικά (Θερμοκρασία(T), Πίεση (RP), Σχετική Υγρασία (H) , Καυσαέρια(EV) και μία έξοδο , την ενέργεια που παράγεται. Για την μοντελοποίηση του συστήματος αυτού θα δημιουργήσουμε 4 TSK μοντέλα με διαφορετικά χαρακτηριστικά. Επειδή θέλουμε να αντιμετωπίσουμε το πρόβλημα μοντελοποίησης χρησιμοποιώντας την ασαφή λογική , θα δημιουργήσουμε για το κάθε μοντέλο ένα Fuzzy Inference System χρησιμοποιώντας την συνάρτηση " genfis " του matlab. Επομένως οι τιμές των χαρακτηριστικών του συστήματος θα αντιστοιχούν σε έναν αριθμό ασαφών συνόλων .Στο συγκεκριμένο πρόβλημα επειδή ο αριθμός των χαρακτηριστικών είναι μικρός , ο αριθμός κανόνων δεν θα είναι πολύ μεγάλος οπότε επιλέγουμε τη διαμέριση πλέγματος για τα ασαφή σύνολα. Με τη διαμέριση πλέγματος οι συναρτήσεις συμμετοχής των ασαφών συνόλων καταλαμβάνουν ομοιόμορφα όλο το πεδίο ορισμού των μεταβλητών και έχουν ίση επικάλυψη (0.5) .Η έξοδος του μοντέλου μπορεί να προσεγγιστεί από το ασαφές σύστημα είτε με ένα Singleton είτε με ένα πολυώνυμο .Τα TSK μοντέλα που θα δημιουργηθούν θα διαφέρουν στον αριθμό συναρτήσεων συμμετοχής των ασαφών συνόλων και στο είδος εξόδου του ασαφούς μοντέλου σύμφωνα με τον πίνακα στο Σχήμα 1.

	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

Έτσι θα μπορέσουμε να βγάλουμε συμπεράσματα για το πόσο επηρεάζουν οι δύο αυτές παράμετροι την εκπαίδευση του μοντέλου .

Υλοποίηση στο Matlab:

Για την υλοποίηση των μοντέλων στο Matlab δημιουργήθηκαν 4 scripts, ένα για το κάθε μοντέλο: TSK_model_1.m, TSK_model_2.m, TSK_model_3.m, TSK_model_4.m. Η διαδικασία που ακολουθήθηκε στο κάθε script είναι η ίδια ,αλλάζοντας τις τιμές των παραμέτρων που θέλουμε να εξετάσουμε στο εκάστοτε μοντέλο (αριθμός MF και output type),επομένως η περιγραφή αναφέρεται και στα 4 script.

Διαχωρισμός δεδομένων:

Αρχικά φορτώνεται στο Workspace το σετ δεδομένων και έπειτα μεταφέρεται σε έναν πίνακα για να υποστεί επεξεργασία. Το σύνολο των δεδομένων χωρίζεται έπειτα σε 3 πίνακες , 60% training _data, 20% evaluation_data και 20% testing_data. Τα δεδομένα του πίνακα training_data χρησιμοποιούνται για την εκπαίδευση του μοντέλου, του evaluation_data για την αξιολόγηση του κάθε κύκλου εκπαίδευσης και του testing_data για τον έλεγχο της εγκυρότητας του μοντέλου αφού ολοκληρωθεί η διαδικασία εκμάθησης.

Κανονικοποίηση δεδομένων:

Το επόμενο βήμα είναι η κανονικοποίηση των δεδομένων μας. Κατά την διαδικασία αυτή εντοπίζεται η μικρότερη και η μεγαλύτερη τιμή από όλα τα χαρακτηριστικά, και την έξοδο του σετ δεδομένων εκπαίδευσης .Έπειτα από όλα τα δεδομένα αφαιρείται η μικρότερη τιμή και διαιρούνται με τη διαφορά της μεγαλύτερης και της μικρότερης τιμής όπως φαίνεται παρακάτω:

```
training_data_min= min(training_data(:));
training_data_max=max(training_data(:));
training_data(:) = (training_data(:) - training_data_min) / (training_data_max -
training_data_min);
```

Παρόμοια εργαζόμαστε και για τα evaluation και testing data και τα δεδομένα κανονικοποιούνται στο διάστημα [0,1]. Η διαδικασία της κανονικοποίησης είναι σημαντική διότι φέρνει στην ίδια κλίμακα τις τιμές όλων των χαρακτηριστικών, και έτσι επηρεάζουν ισοδύναμα το σύστημα .Αν ένα από τα χαρακτηριστικά είχε πολύ μεγάλες τιμές σχετικά με τα υπόλοιπα και δεν υπέστησαν κανονικοποίηση, τότε θα επηρέαζε περισσότερο το σύστημα από τα άλλα χαρακτηριστικά και τα αποτελέσματα θα ήταν λανθασμένα.

Εκπαίδευση του μοντέλου:

Αφού κανονικοποιηθούν τα δεδομένα δημιουργούμε ένα αντικείμενο “fis_opt” μέσω της συνάρτησης genfisOptions(), στο οποίο θέτουμε τα χαρακτηριστικά του fuzzy inference system .Παραδείγματος χάριν για το πρώτο μοντέλο θέτουμε τον αριθμό των συναρτήσεων συμμετοχής σε “2”, και το είδος εξόδου σε “Constant”.Αφού κάνουμε τις απαραίτητες ρυθμίσεις δημιουργούμε το fuzzy inference system μέσω της συνάρτησης “genfis()”.Η συνάρτηση παίρνει

ως ορίσματα τα δεδομένα εκπαίδευσης εισόδου, τα δεδομένα εκπαίδευσης εξόδου και τα χαρακτηριστικά του συστήματος “fis_opt”.Το σύστημα που δημιουργείται αναπαρίσταται από το αντικείμενο “initial_fis”.’ Έπειτα μέσω της συνάρτησης “plot()”, δημιουργούμε διαγράμματα των συναρτήσεων συμμετοχής για κάθε είσοδο , για να τις συγκρίνουμε στο τέλος με την τελική τους μορφή μετά την εκπαίδευση του μοντέλου. Στη συνέχεια δημιουργούμε ένα αντικείμενο “anfis_opt” μέσω της συνάρτησης “anfisOptions()“ και θέτουμε τα χαρακτηριστικά της εκπαίδευσης του μοντέλου. Συγκεκριμένα θέτουμε τον αριθμό των εποχών εκπαίδευσης “400” που είναι ένας ικανοποιητικός αριθμός για να εξάγουμε συμπεράσματα και να μην είναι χρονοβόρα η διαδικασία εκπαίδευσης. Θέτουμε ως μοντέλο εκμάθησης το αντικείμενο “initial_fis” και εισάγουμε τα δεδομένα αξιολόγησης. Το μοντέλο πλέον είναι έτοιμο για εκπαίδευση , η οποία ξεκινάει καλώντας την συνάρτηση “anfis()” με ορίσματα τα δεδομένα εκπαίδευσης και τις ρυθμίσεις που κάναμε προηγουμένως. Η συνάρτηση επιστρέφει για κάθε εποχή το σφάλμα εκμάθησης(trainError), το σφάλμα αξιολόγησης(chkError) , το μοντέλο με το μικρότερο σφάλμα αξιολόγησης (chkFIS) και το βήμα διαφόρισης των παραμέτρων εισόδου και εξόδου. Οι διαδικασίες αυτές φαίνονται στον παρακάτω κώδικα:

```
anfis_opt=anfisOptions('InitialFIS',initial_fis);

anfis_opt.ValidationData=evaluation_data;

anfis_opt.EpochNumber=400;

[train_fis,trainError,stepSize,chkFIS,chkError] = anfis(training_data,anfis_opt);
```

Το “trainError” είναι το σφάλμα εκπαίδευσης σε κάθε εποχή εκπαίδευσης. Το “chkError” είναι το σφάλμα αξιολόγησης είναι το σφάλμα που προκύπτει όταν στο τέλος της κάθε εποχής εκπαίδευσης ελέγξουμε την απόδοση του μοντέλου με τα δεδομένα “evaluation_data”. Το μοντέλο “chkFIS” είναι το μοντέλο που έδωσε το μικρότερο σφάλμα αξιολόγησης , σε μία από τις 400 εποχές , και είναι το μοντέλο που επιλέγεται από το σύστημα με τις ανάλογες παραμέτρους εισόδου και εξόδου. Για την αξιολόγηση του τελικού μοντέλου καλούμε τη συνάρτηση “evalfis()” ,με ορίσματα το τελικό βέλτιστο μοντέλο και τα δεδομένα ελέγχου , “testing_data” και έτσι προκύπτει η τελική έξοδος του συστήματος που αποθηκεύεται στο διάνυσμα “system_output”. Με τη συνάρτηση “Plot()” δημιουργούμε διαγράμματα με την τελική μορφή των συναρτήσεων συμμετοχής του κάθε χαρακτηριστικού , για να τις συγκρίνουμε με τις αρχικές, πριν την εκπαίδευση.

Υπολογισμός μετρικών συστήματος:

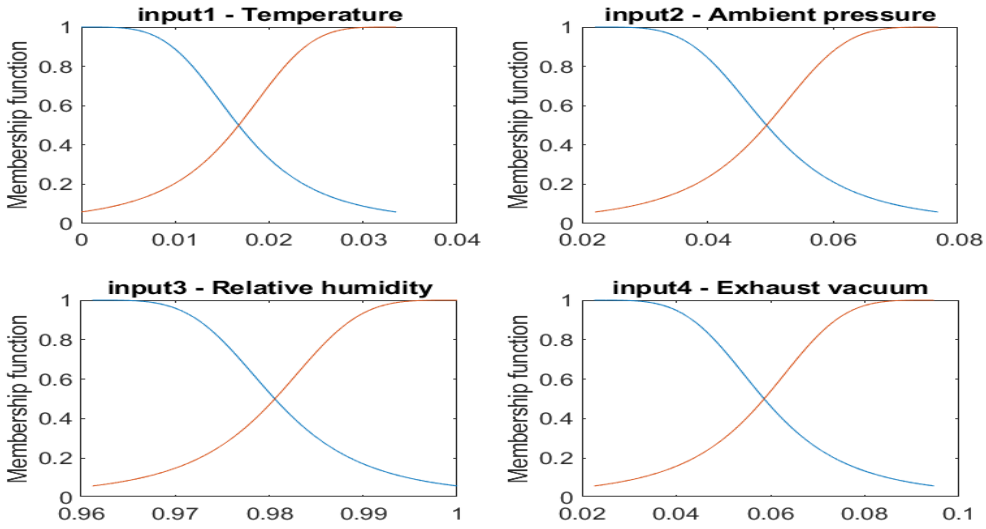
Τέλος , υπολογίζουμε ορισμένες χρήσιμες μετρικές για την αξιολόγηση του συστήματος οι οποίες είναι : το μέσο τετραγωνικό σφάλμα (MSE), ο συντελεστής προσδιορισμού(R2) , και οι δείκτες RMSE, NMSE και NDEI. Με τη συνάρτηση " plot()" δημιουργούμε διαγράμματα που απεικονίζουν τα διάφορα σφάλματα, την αρχική και τελική έξοδο του συστήματος

Αποτελέσματα και σύγκριση μοντέλων :

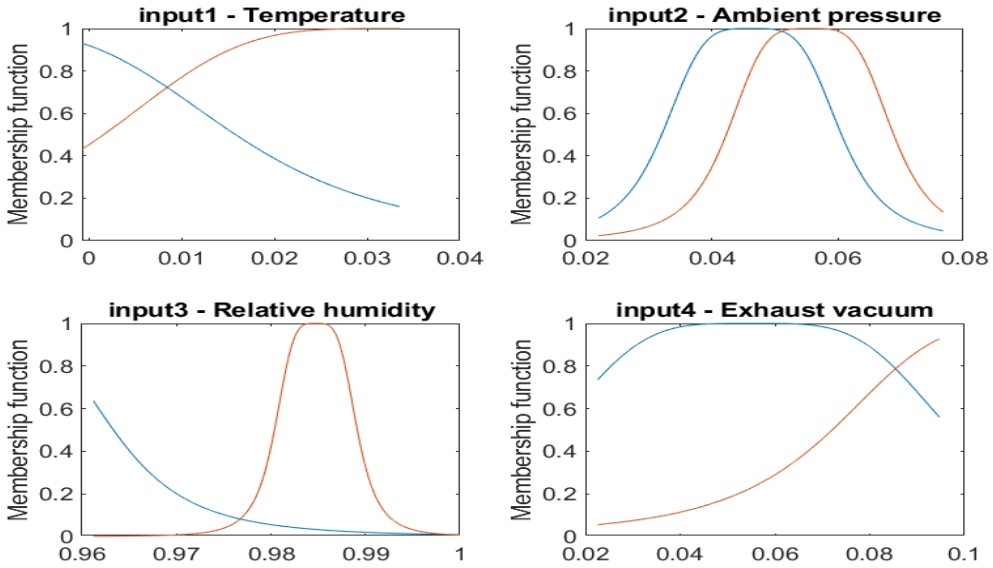
Στην ενότητα αυτή παρουσιάζονται τα διαγράμματα που προέκυψαν από την εκπαίδευση του κάθε μοντέλου , καθώς και σχολιασμός αυτών.

TSK_model_1:

Στην Εικόνα 1 απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητών για κάθε χαρακτηριστικό πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με grid partitioning με τα ασαφή σύνολα να έχουν ίση επικάλυψη . Στην εικόνα 2 απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητών των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης τουμοντέλου.

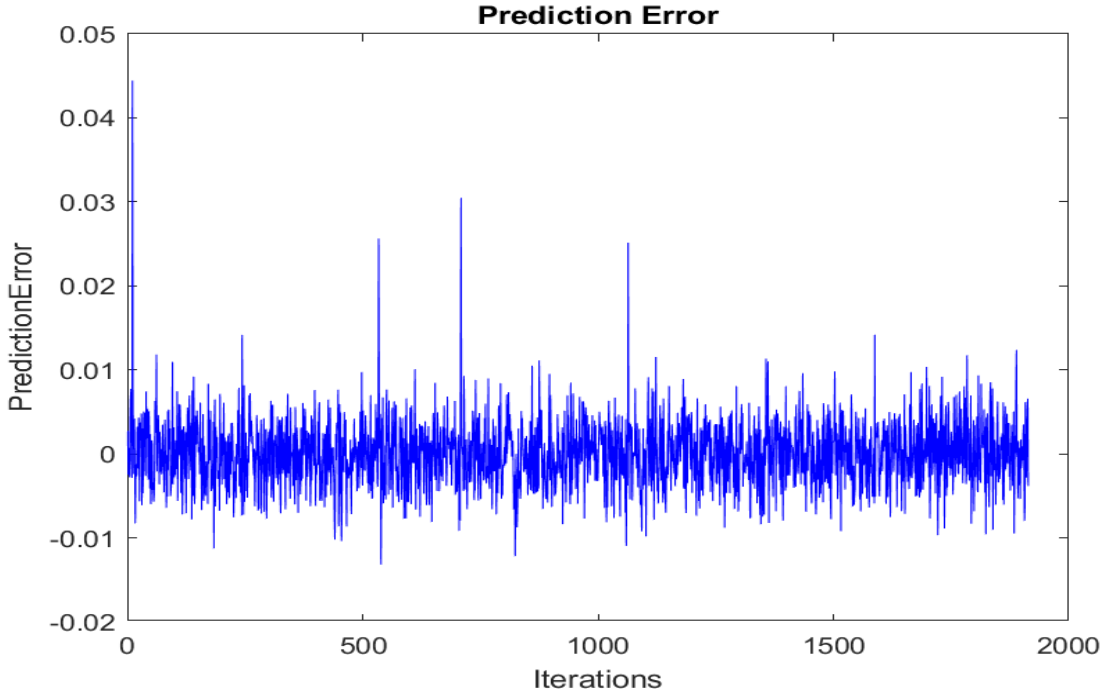


Εικόνα 1 :Αρχικές συναρτήσει συμμετοχής



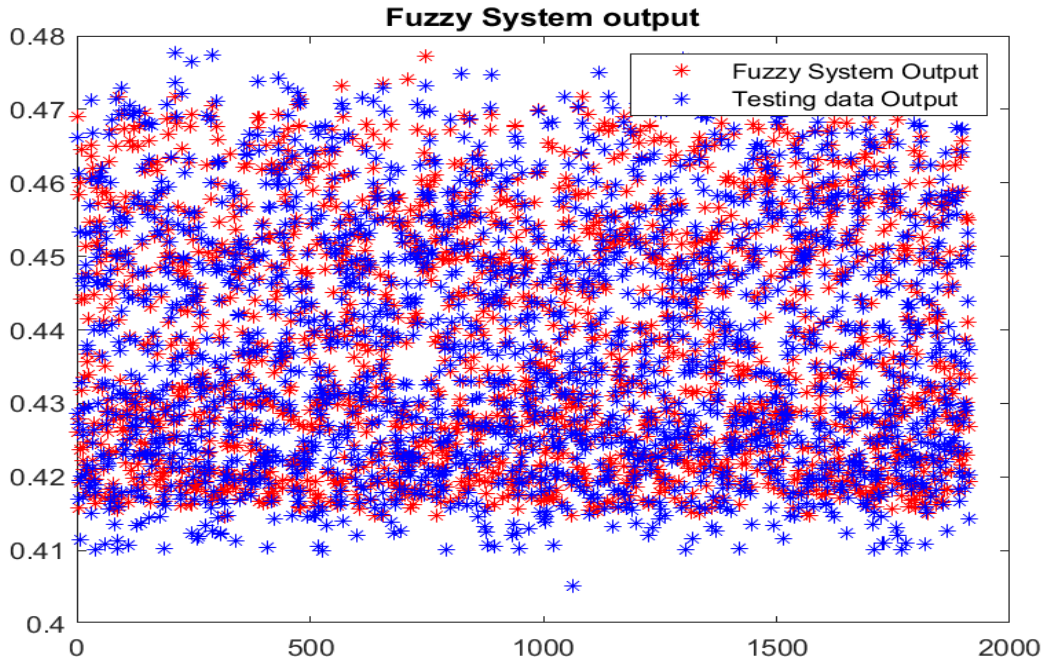
Εικόνα 2 :Τελικές συναρτήσεις συμμετοχής

Στην Εικόνα 3 απεικονίζεται το σφάλμα πρόβλεψης του μοντέλου , δηλαδή το σφάλμα του τελικού μοντέλου μετά την εκπαίδευση στο να προβλέψει σωστά την έξοδο του συστήματος. Το σφάλμα κινείται στο έυρος [-0.1,0.1] ,εκτός από ελάχιστες ακραίες τιμές σε μερικά σημεία.



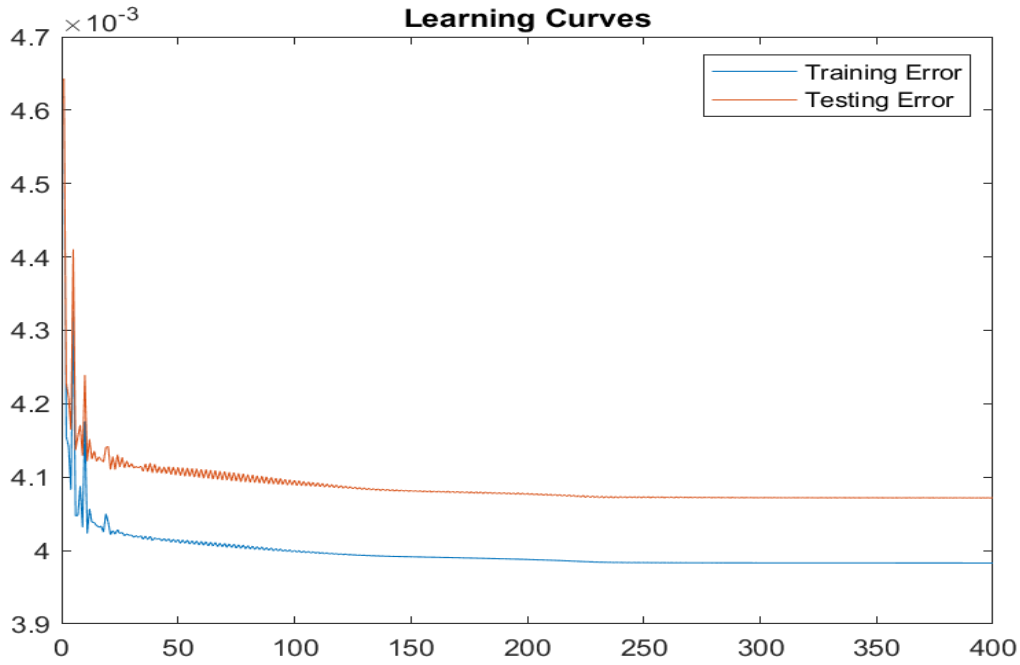
Εικόνα 3

Στην Εικόνα 4 απεικονίζεται η έξοδος του ασαφούς μοντέλου (κοκκινο χρώμα) , σε σχέση με την έξοδ των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγίζει αρκετά καλα τις τιμές της εξόδου χωρίς μεγάλο σφάλμα. Αυτο συμβαίνει κυρίως επειδή ο αριθμός χαρακτηριστικών είναι μικρός και κάνει εύκολη την εκπαίδευση του μοντέλου .



Εικόνα 4

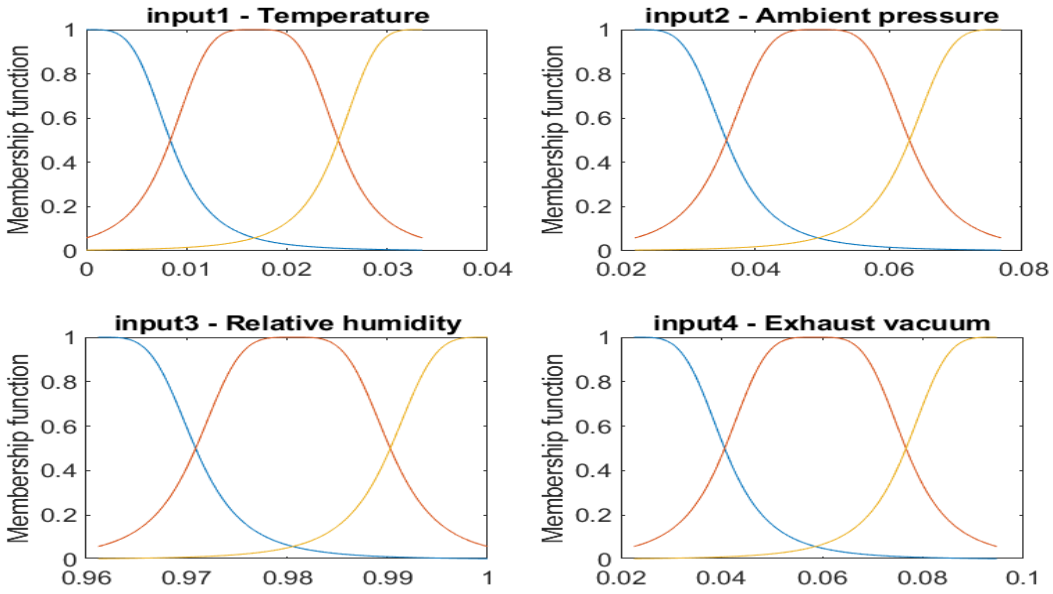
Στην Εικόνα 5 φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και εμ κόκκινο οτο σφάλμα αξιολόγισης . Το σφάλμα αξιολόγισης είναι λογικά πιο μεγάλο απο της εκπαίδευσης , αλλά εντός των επιτρεπτών ορίων και με λογικές τιμές. Λόγω του ότι έχουμε 2 συναρτήσεις συμμετοχής και Singleton έξοδο η εκμάθηση του μοντέλου είναι πιο απλή καθώς περιέχει λίγες πιθανές τιμές για κάθε χαρακτηριστικό , άρα και λιγότερες παραμέτους προς βελτιστοποίηση στο fuzzy inference system . Επομένως οι βέλτιστες τιμές παραμέτρων εντοπίζονται γρήγορα από το σύστημα και λόγω αυτού δεν υπάρχει μεγάλη διακύμανση στις τιμές του σφάλματος .



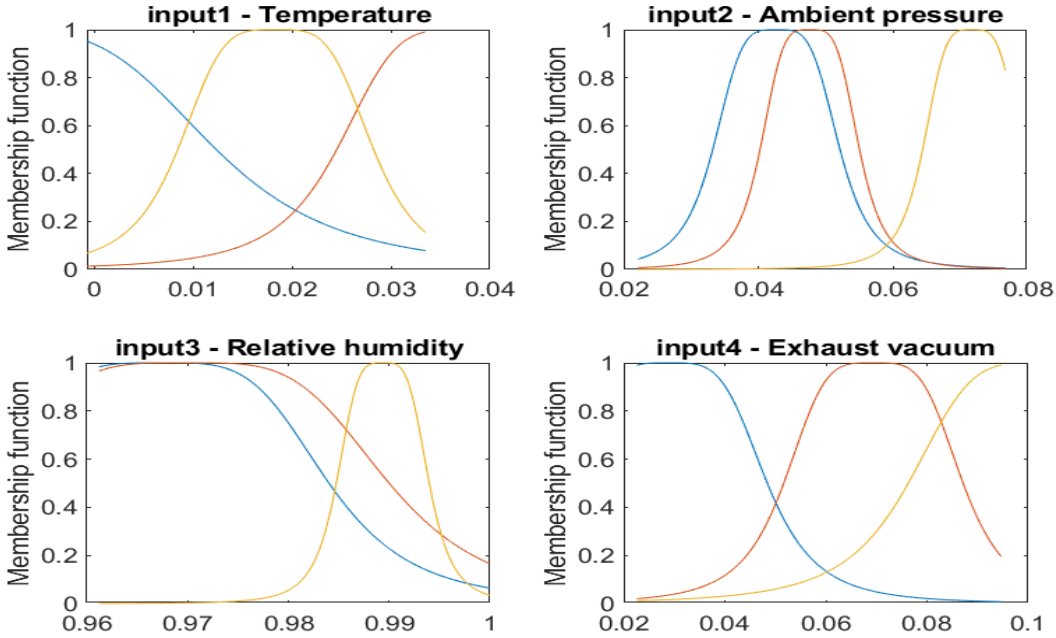
Εικόνα 5

TSK_model_2:

Στην εικόνα 6 απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητών για κάθε χαρακτηριστικό πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με grid partitioning με τα ασαφή σύνολα να έχουν ίση επικάλυψη . Στην εικόνα 7 απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητών των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του

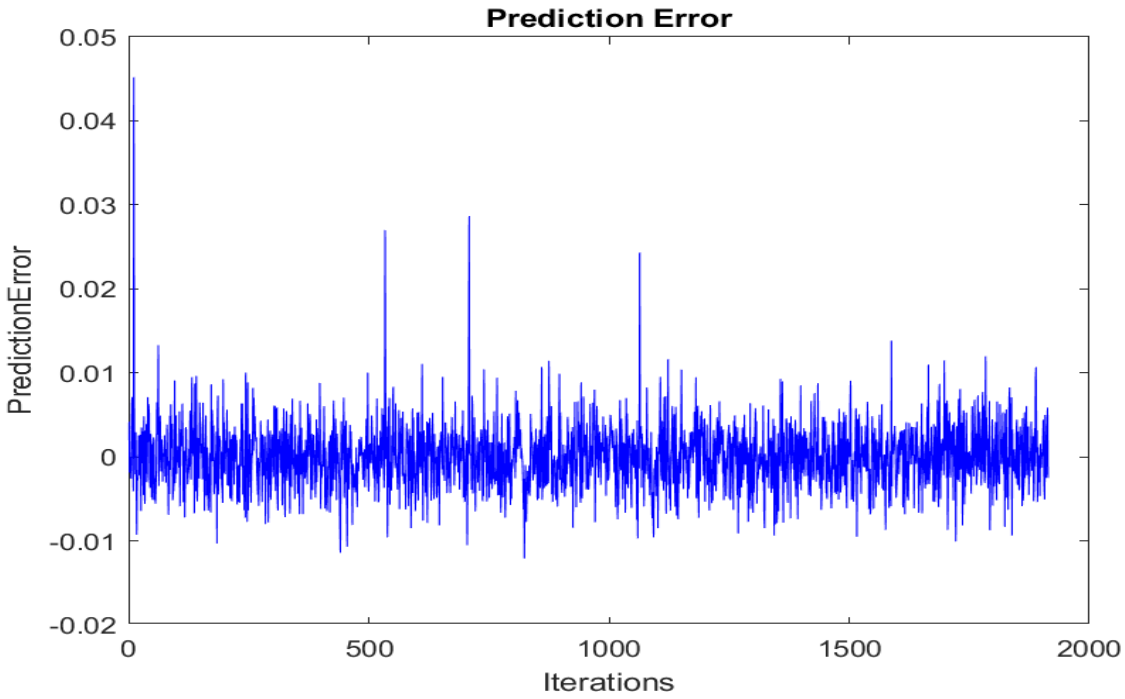


Εικόνα 6



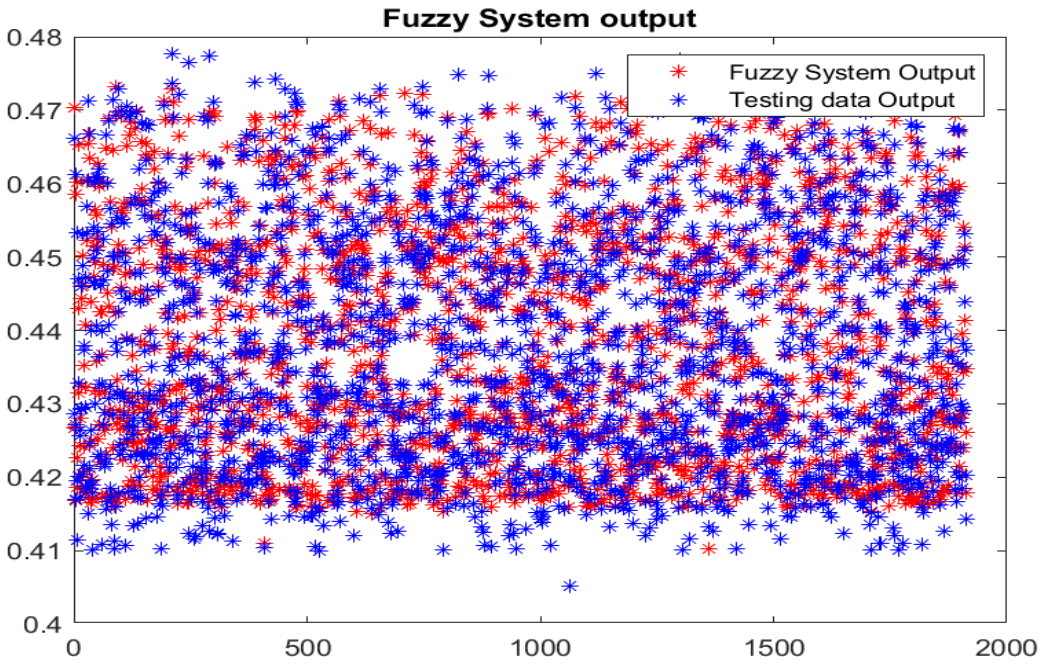
Εικόνα 7

Στην εικόνα 8 απεικονίζεται το σφάλμα πρόβλεψης του μοντέλου , δηλαδή το σφάλμα του τελικού μοντέλου μετά την εκπαίδευση στο να προβλέψει σωστά την έξοδο του συστήματος. Το σφάλμα κινείται στο έυρος [-0.1,0.1] ,εκτός από ελάχιστες ακραίες τιμές σε μερικά σημεία.



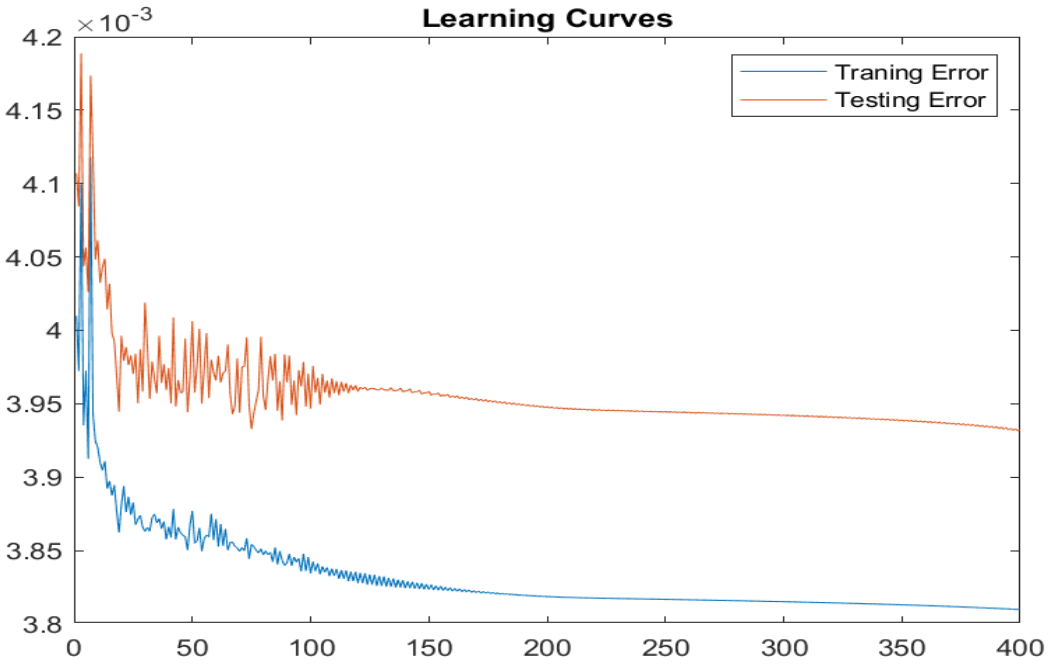
Εικόνα 8

Στο σχήμα απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει αρκετά καλά τις τιμές της εξόδου χωρίς μεγάλο σφάλμα. Αυτό συμβαίνει κυρίως επειδή ο αριθμός χαρακτηριστικών είναι μικρός και κάνει εύκολη την εκπαίδευση του μοντέλου .



Εικόνα 9

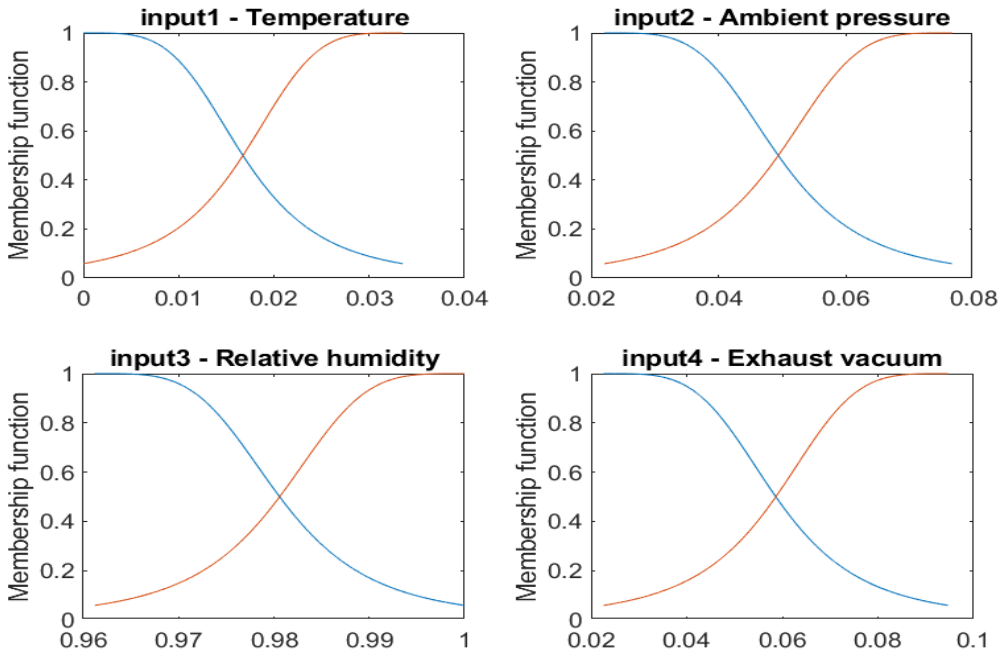
Στο σχήμα φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και εμ κόκκινο οτο σφάλμα αξιολόγισης . Το σφάλμα αξιολόγισης είναι λογικά πιο μεγάλο απο της εκπαίδευσης , αλλά εντός των επιτρεπτών ορίων και με λογικές τιμές. Λόγω του ότι έχουμε 3 συναρτήσεις συμμετοχής η εκμάθηση του μοντέλου είναι πιο σύνθετη καθώς περιέχει πιο πολλες πιθανές τιμές για κάθε χαρακτηριστικό , άρα και περισσότερες παραμέτρους εισόδου για βελτιστοποίηση.Επομένως υπάρχει μεγάλη διακύμανση στις τιμές του σφάλματος στην αρχή μέχρι το σύστημα να σταθεροποιηθεί βρίσκοντας τον κατάλληλο συνδυασμό παραμέτρων.Παρατηρούμε πως σε σχέση με το προηγούμενο μοντέλο το σφάλμα αξιολόγισης δεν μειώνεται τόσο αναλογικά με το σφάλμα εκπαίδευσης , άρα μπορούμε να συμπεράνουμε πως η αύξηση των χαρακτηριστικών εισόδου οδήγησε σε ενα μικρο βαθμο σε υπερεκπαίδευση.



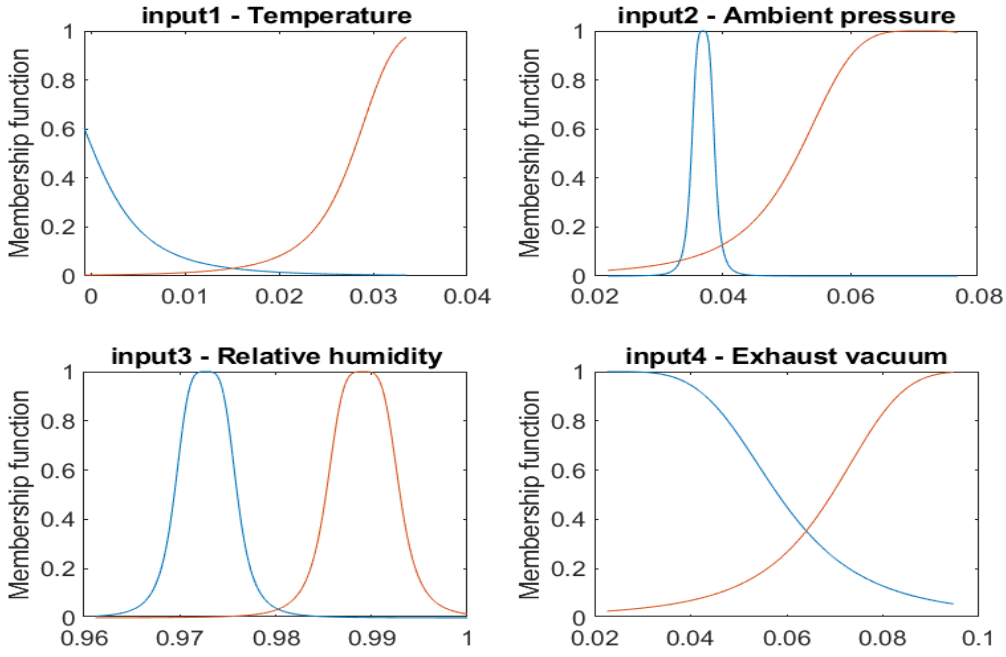
Εικόνα 10

TSK_model_3:

Στην εικόνα 11 απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητων για κάθε χαρακτηριστικό πρίν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με grid partitioning με τα ασαφή σύνολα να έχουν ίση επικάλυψη . Στην εικόνα απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητών των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του

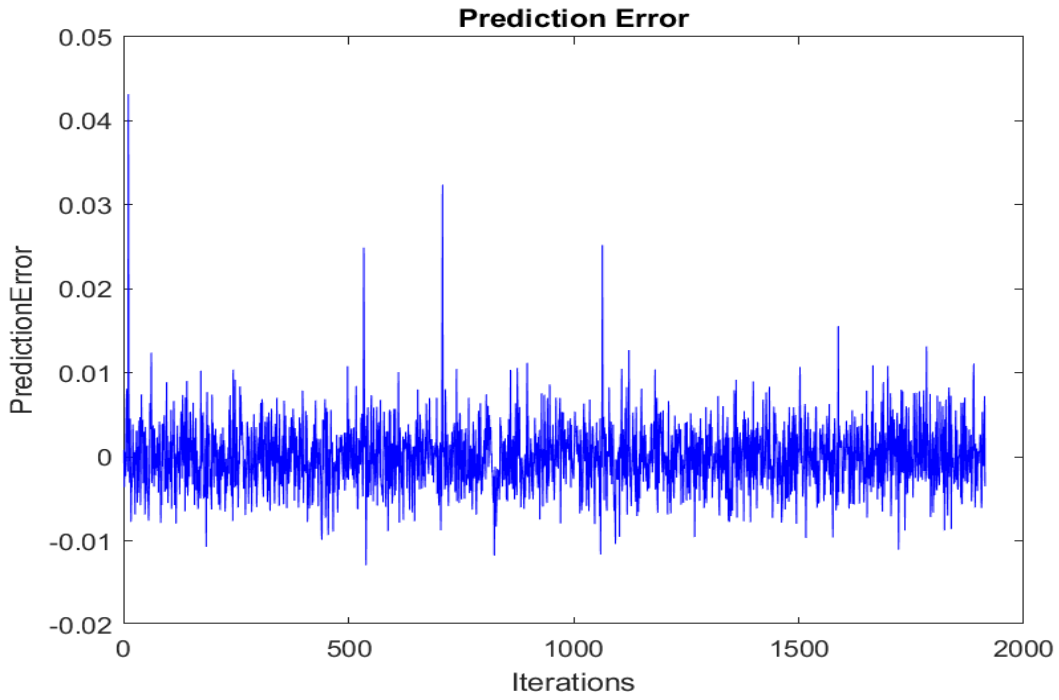


Εικόνα 11



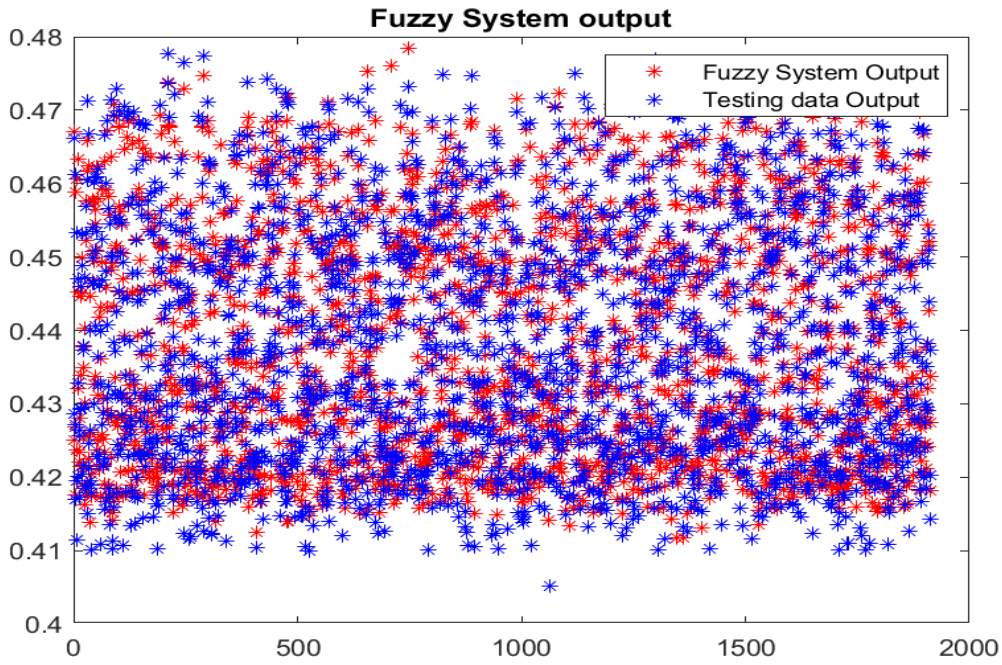
Εικόνα 12

Στην εικόνα 13 απεικονίζεται το σφάλμα πρόβλεψης του μοντέλου , δηλαδή το σφάλμα του τελικού μοντέλου μετά την εκπαίδευση στο να προβλέψει σωστά την έξοδο του συστήματος. Το σφάλμα κινείται στο έυρος [-0.1,0.1] ,εκτός από ελάχιστες ακραίες τιμές σε μερικά σημεία.



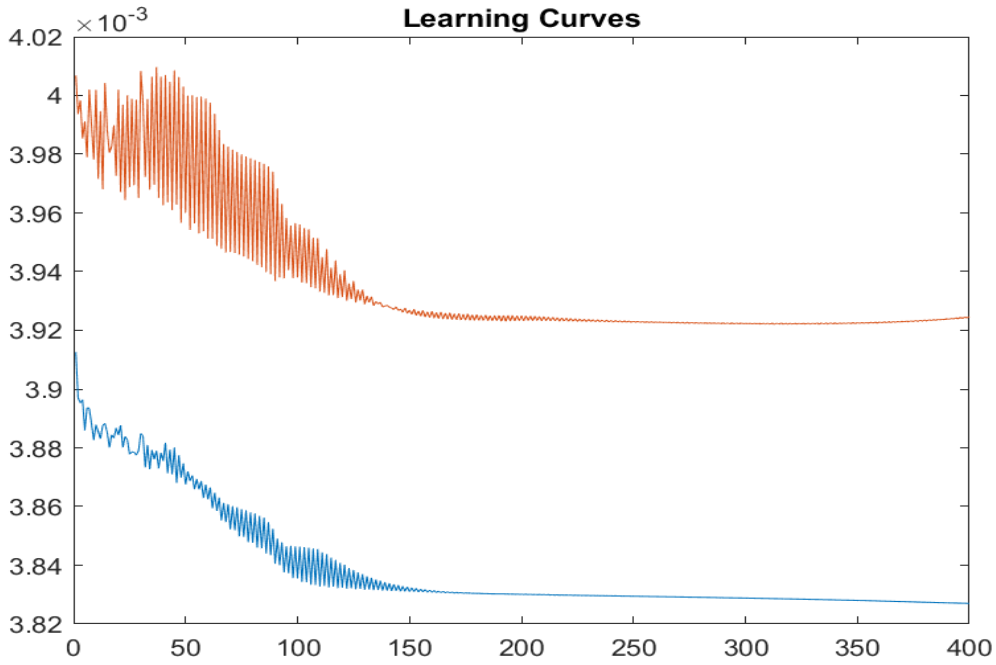
Εικόνα 13

Στο σχήμα απεικονίζεται η έξοδος του ασαφούς μοντέλου (κοκκινο χρώμα) , σε σχέση με την έξοδ των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγίζει αρκετά καλα τις τιμές της εξόδου χωρίς μεγάλο σφάλμα. Αυτο συμβαίνει κυρίως επειδή ο αριθμός χαρακτηριστικών είναι μικρός και κάνει εύκολη την εκπαίδευση του μοντέλου .



Εικόνα 14

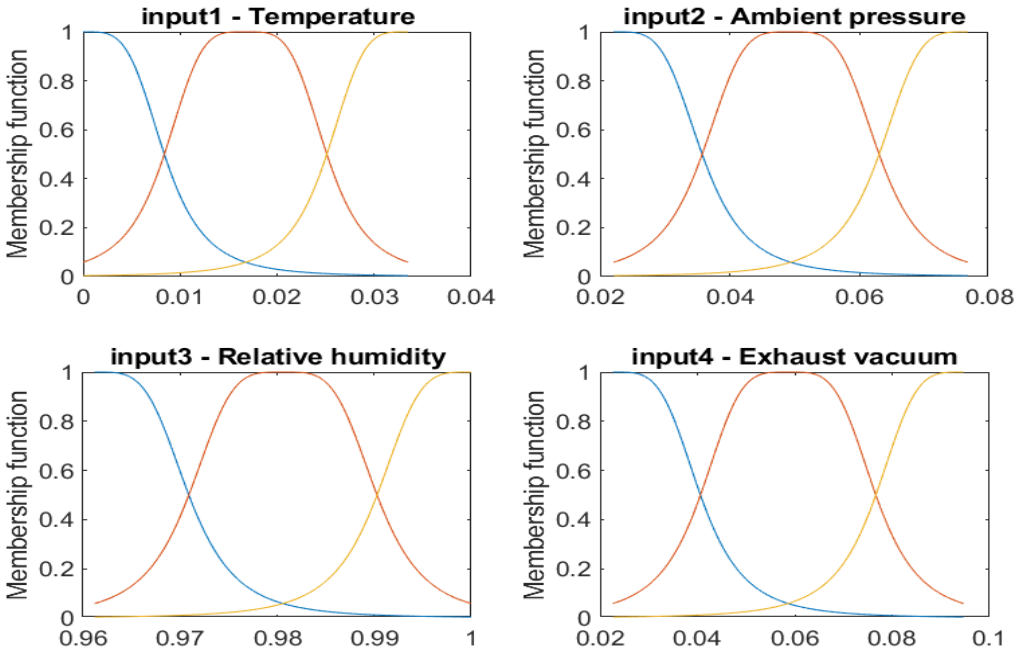
Στο σχήμα φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και εμ κόκκινο οτο σφάλμα αξιολόγισης . Το σφάλμα αξιολόγισης είναι λογικά πιο μεγάλο απο της εκπαίδευσης , αλλά εντός των επιτρεπτών ορίων και με λογικές τιμές. Λόγω του ότι έχουμε Linear έξοδο η εκμάθηση του μοντέλου είναι πιο συνθετη καθώς περιέχει περισσότερες παραμέτρους εξόδου, επομένως υπάρχει μεγάλη διακύμανση στις τιμές του σφάλματος έως ώτου σταθεροποιηθεί , όταν το μοντέλο έχει αντιληφθεί το pattern εξόδου .



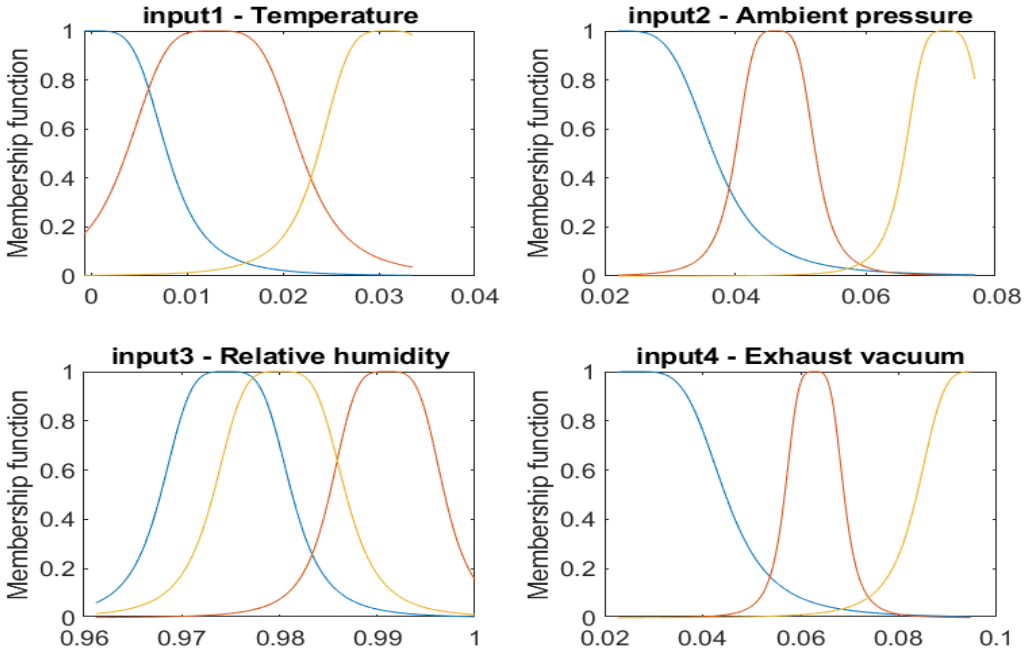
Εικόνα 15

TSK_model_4:

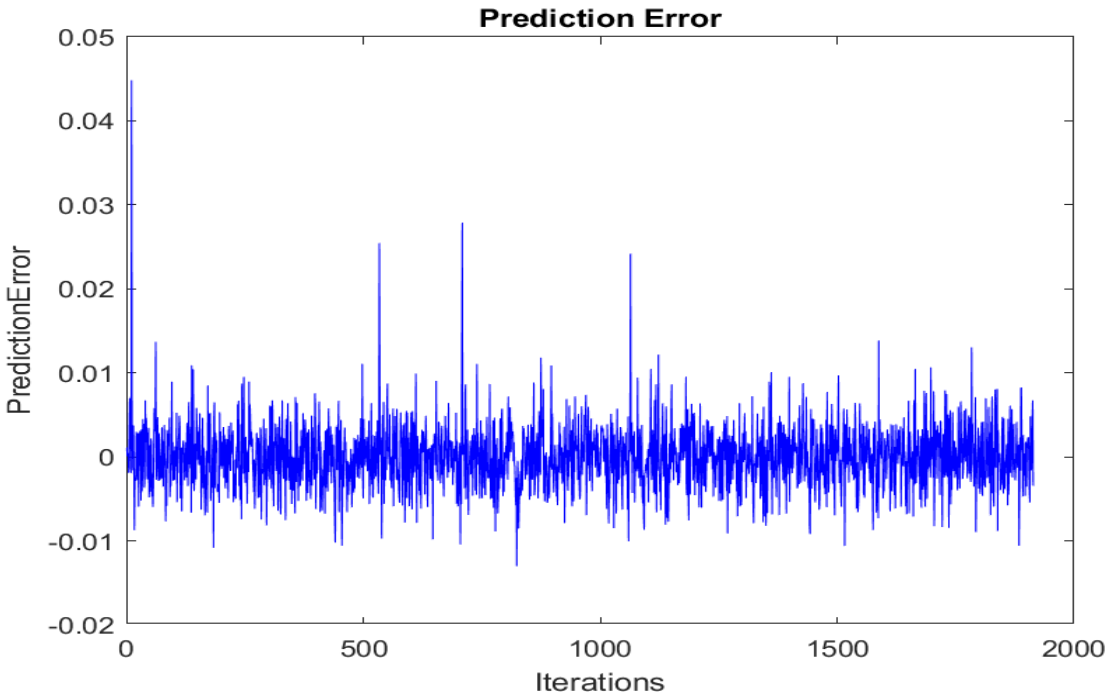
Στο σχήμα απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητων για κάθε χαρακτηριστικό πρίν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με grid partitioning με τα ασαφή σύνολα να έχουν ίση επικάλυψη . Στο σχήμα απεικονίζονται οι συναρτήσεις συμμετοχής των μεταβλητών των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του



Εικόνα 16

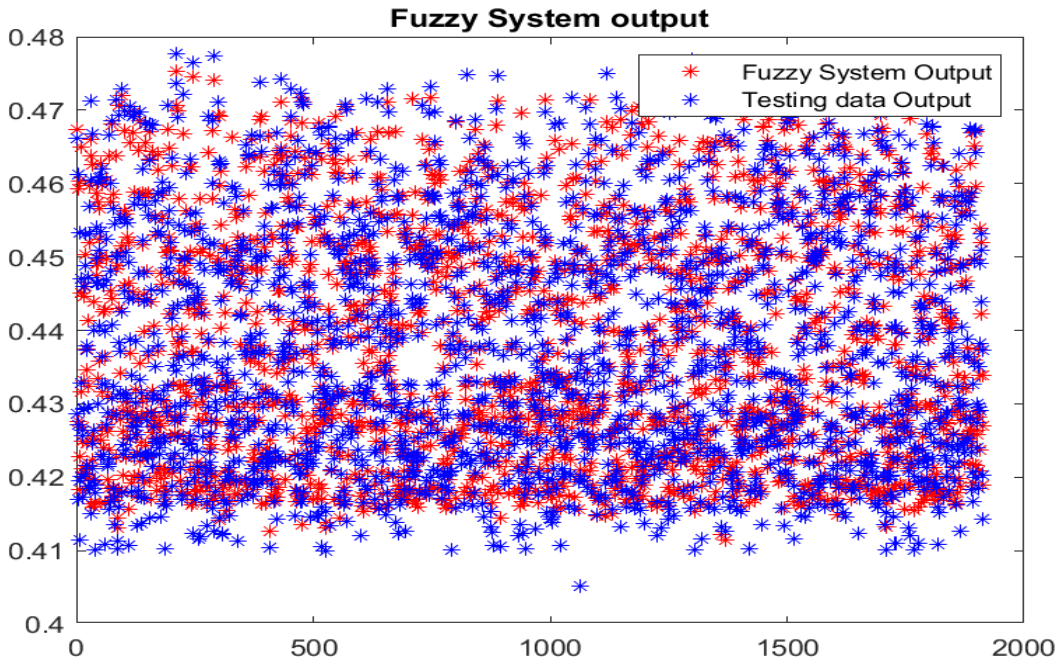


Εικόνα 17



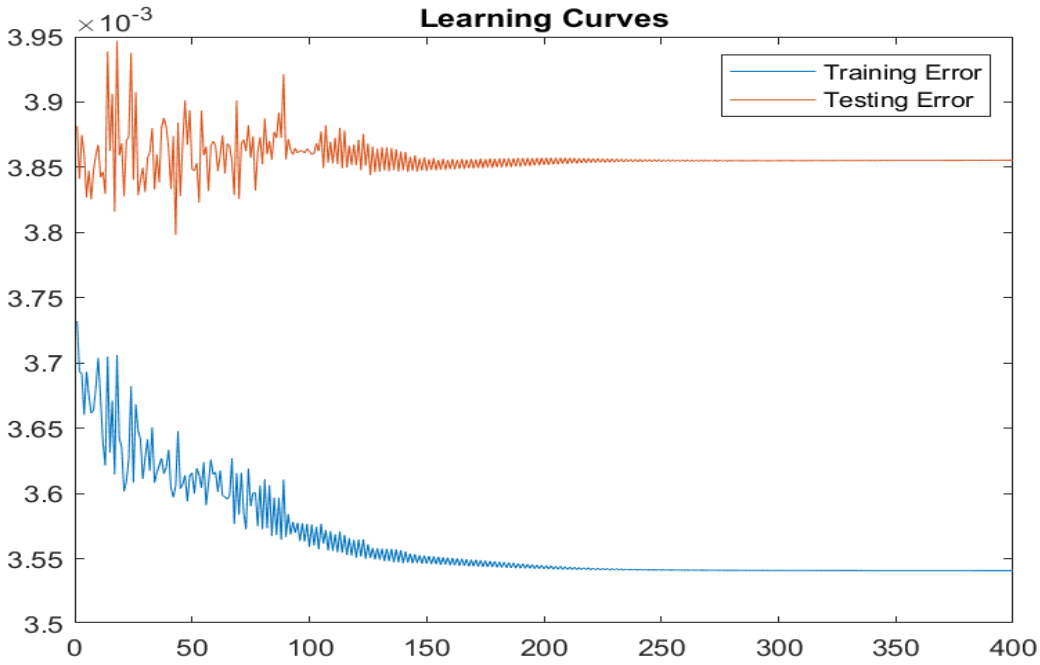
Εικόνα 18

Στο σχήμα απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει αρκετά καλά τις τιμές της εξόδου χωρίς μεγάλο σφάλμα. Αυτο συμβαίνει κυρίως επειδή ο αριθμός χαρακτηριστικών είναι μικρός και κάνει εύκολη την εκπαίδευση του μοντέλου .



Εικόνα 19

Στο σχήμα φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο οτο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο απο της εκπαίδευσης , αλλά εντός των επιτρεπτών ορίων και με λογικές τιμές. Λόγω του ότι έχουμε 3 συναρτήσεις συμμετοχής και Linear έξοδο η εκμάθηση του μοντέλου είναι πιο σύνθετη καθώς περιέχει αρκετές πιθανές τιμές παραμέτρων για κάθε χαρακτηριστικό και έξοδο , επομένως υπάρχει μεγάλη διακύμανση στις τιμές του σφάλματος μέχρι να σταθεροποιηθεί. Παρατηρούμε επίσης πως η μείωση του σφάλματος ελέγχου δεν είναι ανάλογη με τη μείωση του σφάλματος εκπαίδευσης , υπάρχει δηλαδή ελαφρώς υπερεκπαίδευση του μοντέλου. Αυτό συμβαίνει λόγω του μεγαλύτερου αριθμού συμμετοχής και της σύνθετης εξόδου , που κάνει το μοντέλο να αποκτά πιο συγκεκριμένη και όχι τόσο γενική γνώση. Παρόλα αυτά η υπερεκπαίδευση δεν είναι τόσο μεγάλη , καθώς το σφάλμα αξιολόγησης μετά από κάποιο σημείο παραμένει σε σχεδόν σταθερή τιμή.



Εικόνα 20

Στο σχήμα φαίνονται οι τιμές για τις διάφορες μετρικές για τα 4 μοντέλα που εκπαιδεύσαμε. Το μοντέλο TSK_4 έχει τις καλύτερες τιμές για όλες τις μετρικές .Παρά την υπερεκπαίδευση που παρατηρήθηκε στο συγκεκριμένο μοντέλο(το σφάλμα αξιολόγησης έπαψε να μειώνεται αναλογικά με το σφάλμα εκπαίδευσης) , το συνολικό σφάλμα σε σχέση με την πραγματική έξοδο είναι το μικρότερο. Οι τιμές των μετρικών είναι αρκετά κοντά και στα υπόλοιπα, διακρίνοντας μια καλύτερη συμπεριφορά από το μοντέλο TSK_2. Συνολικά παρατηρείται λοιπόν ελαφρώς καλύτερη συμπεριφορά στα μοντέλα με 3 συναρτήσεις εισόδου , παρά σε αυτά με 2 συναρτήσεις εισόδου , και τη διαφορά στο 4ο μοντέλο κάνει η πολυώνυμική έξοδος που προσθέτει ακρίβεια στο μοντέλο.

	TSK_1	TSK_2	TSK_3	TSK_4
MSE	0.000017	0.000016	0.000016	0.000016
RMSE	0.004150	0.004016	0.004046	0.003940
R2	0.999939	0.999934	0.999934	0.999937
NMSE	0.63336	0.059313	0.060188	0.057091
NDEI	0.251667	0.243543	0.245331	0.238937

Εικόνα 21

Επίλυση προβλήματος με dataset μεγάλης διαστασιμότητας :

Στο δεύτερο πρόβλημα της εργασίας καλούμαστε να μοντελοποιήσουμε ένα πρόβλημα χρησιμοποιώντας το Superconductivty dataset από το UCI Repository το οποίο αποτελείται από 81 χαρακτηριστικά (εισόδους) και 1 έξοδο (άγνωστη συνάρτηση). Λόγω του μεγάλου αριθμού εισόδων του μοντέλου αυξάνεται κατά πολύ ο αριθμός των κανόνων του ασαφούς συστήματος , πράγμα που κάνει πολύ δύσκολη την μοντελοποίηση του συστήματος. Στην εργασία αυτή αντιμετωπίζουμε το πρόβλημα της πολυδιαστασιμότητας του σετ δεδομένων με το να επιλέξουμε κάποια από τα χαρακτηριστικά εισόδου (τα πιο σημαντικά), μέσω του αλγορίθμου του Relief , και με το να διαμερίσουμε τον χώρο εισόδου χρησιμοποιώντας τον αλγόριθμο fuzzy C-means (FCM).Με αυτόν τον τρόπο επιλέγουμε μόνο τα χαρακτηριστικά που προσφέρουν σημαντική πληροφορία για την εκπαίδευση του μοντέλου και και δημιουργούμε Clusters(ομάδες) με αυτά μειώνοντας έτσι σημαντικά τον αριθμό των κανόνων. Για να πραγματοποιήσουμε αυτό το στόχο αρκεί να ορίσουμε δύο παραμέτρους, τον αριθμό χαρακτηριστικών (NF) και τον αριθμό των κανόνων(NR) του συστήματος που ταυτίζεται με τον αριθμό των clusters που θα δημιουργηθούν από τον FCM. Πιο συγκεκριμένα για τη συγκεκριμένη εργασία η μεταβλητή NR παίρνει τιμές [3, 5 ,7 ,10] και η μεταβλητή NR παίρνει τιμές [3, 5 ,7 ,9 .12]. Οι τιμές της εκφώνησης αλλάχτηκαν σε μικρότερες λόγω της χρονοβόρας εκτέλεσης του προγράμματος για μεγάλες τιμές των NF και NR.Καλούμαστε να επιλέξουμε για αυτές τις τιμές τον συνδυασμό των παραμέτρων που θα δίνει το μικρότερο σφάλμα στο μοντέλο μας. Για ακόμα πιο έγκυρα αποτελέσματα χρησιμοποιούμε την μέθοδο Cross Validation κατά την οποία ,για έναν αριθμό επαναλήψεων, για κάθε πιθανό συνδυασμό τιμών των παραμέτρων , το dataset εκπαίδευσης χωρίζεται στα 2 σε διαφορετικά

σημεία σε κάθε επανάληψη . Στο τέλος της εκπαίδευσης κρατάμε τον μέσο όρο των σφαλμάτων για όλους τους πιθανούς διαχωρισμούς του σετ .

Υλοποίηση στο Matlab:

Για την υλοποίηση της εκπαίδευσης του μοντέλου στο Matlab , δημιουργήθηκαν τα scripts high_dim_data.m και final_model.m .Στο high_dim_data.m πραγματοποιείται η επιλογή δεδομένων με τον αλγόριθμο Relief, η επαναληπτική διαδικασία της εκπαίδευσης με cross Vallidation για κάθε σετ παραμέτρων RF και RL, και η εξαγωγή συμπερασμάτων για το βέλτιστο μοντέλο. Στο final_model.m το βέλτιστο μοντέλο που επιλέχθηκε εκπαιδεύεται και διεξάγονται οι μετρικές και τα διαγράμματα σφάλματος του τελικού μοντέλου.

High_dim_data.m:

Αρχικά φορτώνουμε τα δεδομένα από το excel data sheet τα διαμερίζουμε και τα κανονικοποιούμε όπως ακριβώς και στα TSK μοντέλα για το απλό dataset. Στη συνέχεια εφαρμόζουμε στο training set τον αλγόριθμο Reflief για την επιλογή των βέλτιστων-χρήσιμων χαρακτηριστικών. Ο αλγόριθμος επιστρέφει έναν πίνακα με τα βάρη σημαντικότητας των χαρακτηριστικών και τον δείκτη του καθενός . Ταξινομούμε τα χαρακτηριστικά σε έναν νέο πίνακα " relief_array" από το πιο σημαντικό στο πιο ασήμαντο .Η διαδικασία αυτή φαίνεται στον παρακάτω κώδικα:

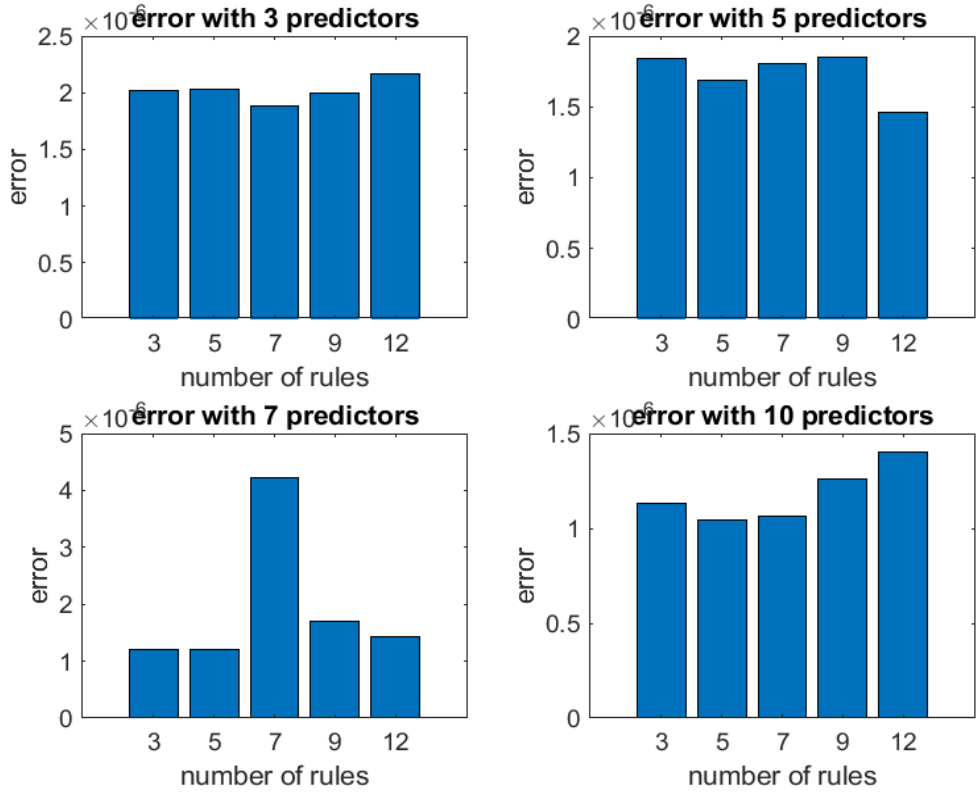
```
[idx,weights] =relieff(training_data(:,1:81),training_data(:,82),100);
relief_array=zeros(length(idx),2);
relief_array(:,1)=idx;
relief_array(:,2)=weights;
[relief_array,index] = sortrows(relief_array,2,'descend');
```

Έπειτα δημιουργούμε τα διανύσματα με τις τιμές των παραμέτρων NF και NR και έναν πίνακα errors_array διαστάσεων NF*NR για να αποθηκεύουμε το error του μοντέλου για κάθε ζεύγος παραμέτρων. Στη συνέχεια ξεκινάει η επαναληπτική διαδικασία, για NF*NR επαναλήψεις .Με την συνάρτηση "cvpartition()" δημιουργούμε έναν αριθμό από διαμερισμού του τεστ εκπαίδευσης δεδομένων που αποθηκεύεται στον πίνακα Sets. Για έναν αριθμό επαναλήψεων που ορίζουμε εμείς το μοντέλο εκπαιδεύεται κάθε φορά με διαφορετικά διαμερισμένο σετ δεδομένων και

αποθηκεύεται για κάθε σετ το τετραγωνικό σφάλμα στον πίνακα cvsetError. Η διαδικασία φαίνεται στον παρακάτω κώδικα:

```
for k=1:Sets.NumTestSets
        training_set=Sets.training(k);
        testing_set=Sets.test(k);
        .
        .
        .
         anfis_opt = anfisOptions('InitialFIS', initial_fis, 'EpochNumber', 40, 'ValidationData', [testing_input testing_output]);
         [train_fis,trainError,stepSize,chkFIS,chkError] = anfis([training_input training_output],anfis_opt);
         system_output = evalfis(chkFIS,testing_data(:,relief_array(1:NF(i),1)));
         cvSetError(k)=sum(system_output - testing_data(:, 82)).^2;
 End
```

Στο τέλος της κάθε επανάληψης (για συγκεκριμένο σετ παραμέτρων) αποθηκεύεται ο μέσος όρος των σφαλμάτων όλων των σετ για την εκπαίδευση του μοντέλου στον πίνακα errors_array.Τέλος , με τη συνάρτηση “plot()” δημιουργούμε ένα διάγραμμα που απεικονίζει το σφάλμα του μοντέλου για κάθε συνδυασμό παραμέτρων. Το διάγραμμα φαίνεται στο παρακάτω σχήμα:



.

Εικόνα 22

Η ελάχιστη τιμή σφάλματος παρατηρήθηκε για NF=10 και για NR=5 .Παρατηρούμε πως για τον μεγαλύτερο αριθμό χαρακτηριστικών (10) το σφάλμα είναι αρκετά μικρότερο από ότι για μικρότερες τιμές χαρακτηριστικών αλλά αυξάνει για μεγαλύτερο αριθμό κανόνων . Αυτό συμβαίνει διότι για μεγάλο αριθμό κανόνων η εκπαίδευση του μοντέλου γίνεται αρκετά συγκεκριμένη και χάνει τη δυνατότητα να αντιμετωπίσει με γενικότητα άγνωστες εισόδους. Επομένως επιλέγουμε έναν επαρκή αριθμό χαρακτηριστικών και κανόνων που να μπορεί να εκπαιδεύσει σωστά το μοντέλο, χωρίς όμως υπερβολή για να αποφευχθεί η υπερεκπαίδευση.

Final_model.m:

Έχοντας επιλέξει το τελικό μας μοντέλο για NF=10 και NR=5, στη συνάρτηση αυτή το εκπαιδεύουμε ακολουθώντας παρόμοια διαδικασία με τα μοντέλα TSK για το απλό dataset. Πριν το διαχωρισμό των δεδομένων σε training, evaluation και testing , ανακατεύουμε τα δεδομένα για να έχουμε καλύτερα αποτελέσματα , και εφαρμόζουμε τον αλγόριθμο του Relief όπως περιγράφηκε προηγουμένως για την επιλογή των βέλτιστων χαρακτηριστικών. Στη συνέχεια

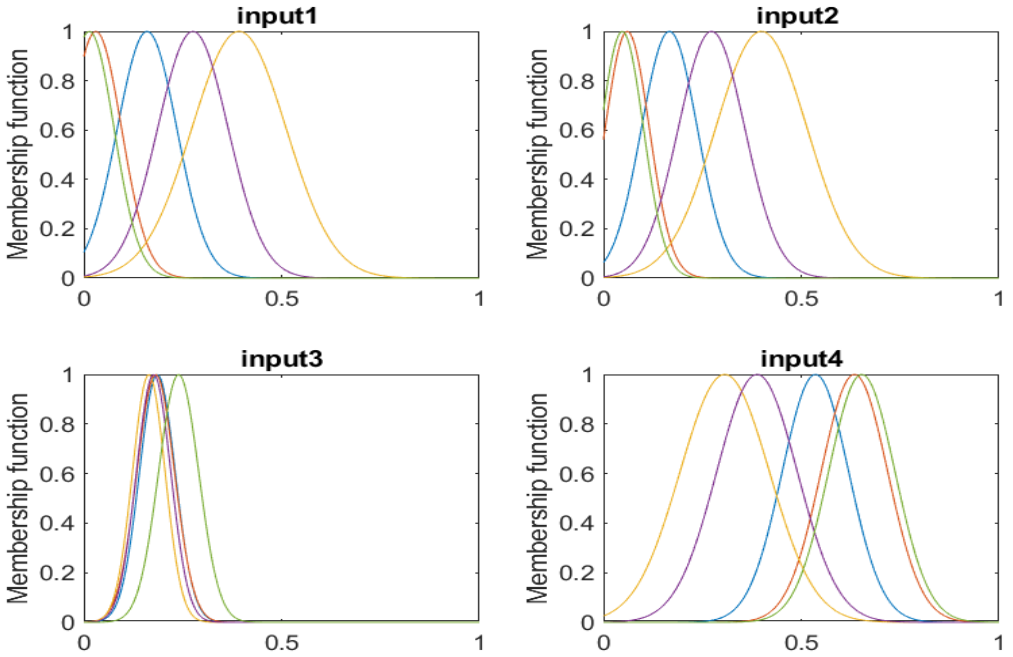
χωρίζουμε και κανονικοποιούμε τα δεδομένα , καθώς και επιλέγουμε μόνο τα χαρακτηριστικά που επέστρεψε ο αλγόριθμος Relief .Εφόσον το βέλτιστο μοντέλο έχει 10 χαρακτηριστικά επιλέγουμε τα 10 πρώτα από τον "relief_array".Η διαδικασία φαίνεται στον παρακάτω κώδικα:

```
input_training_data=training_data(:,relief_array(1:numofPred,1));
output_training_data=training_data(:,82);
```

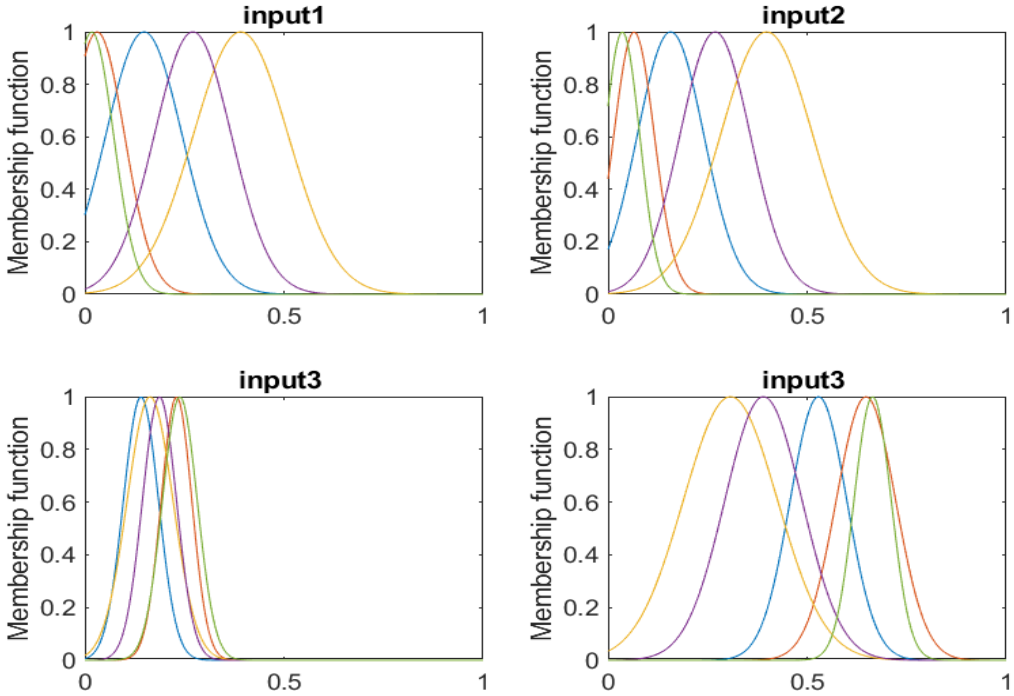
Παρόμοια εργαζόμαστε και για τα evaluation και testing data.Έπειτα επιλέγουμε τα χαρακτηριστικά της συνάρτησης genfis() για τη δημιουργία του fuzzy inference system , θέτοντας αριθμό clusters το 5 (NR=5) και ως μέθοδο διαχωρισμού του χώρου εισόδου τον FCM. Το μοντέλο εκπαιδεύεται με την συνάρτηση anfis() παρόμοια με τα προηγούμενα μοντέλα και τέλος υπολογίζουμε τις μετρικές και εξάγουμε τα διαγράμματα εξόδου συστήματος και σφαλμάτων.

Αποτελέσματα και σύγκριση μοντέλων :

Στο σχήμα απεικονίζονται οι συναρτήσεις συμμετοχής για 4 από τα χαρακτηριστικά πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με scatter partitioning με τα ασαφή σύνολα να μην έχουν ίση επικάλυψη . Στο σχήμα απεικονίζονται οι συναρτήσεις συμμετοχής των των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

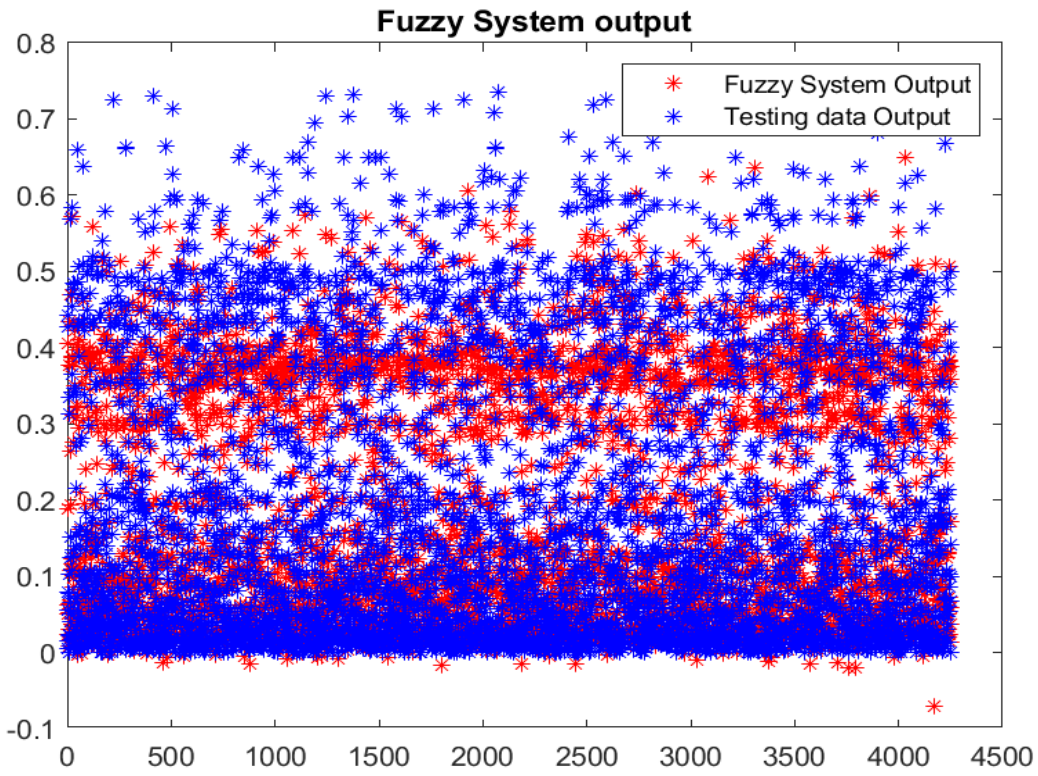


Εικόνα 23



Εικόνα 24

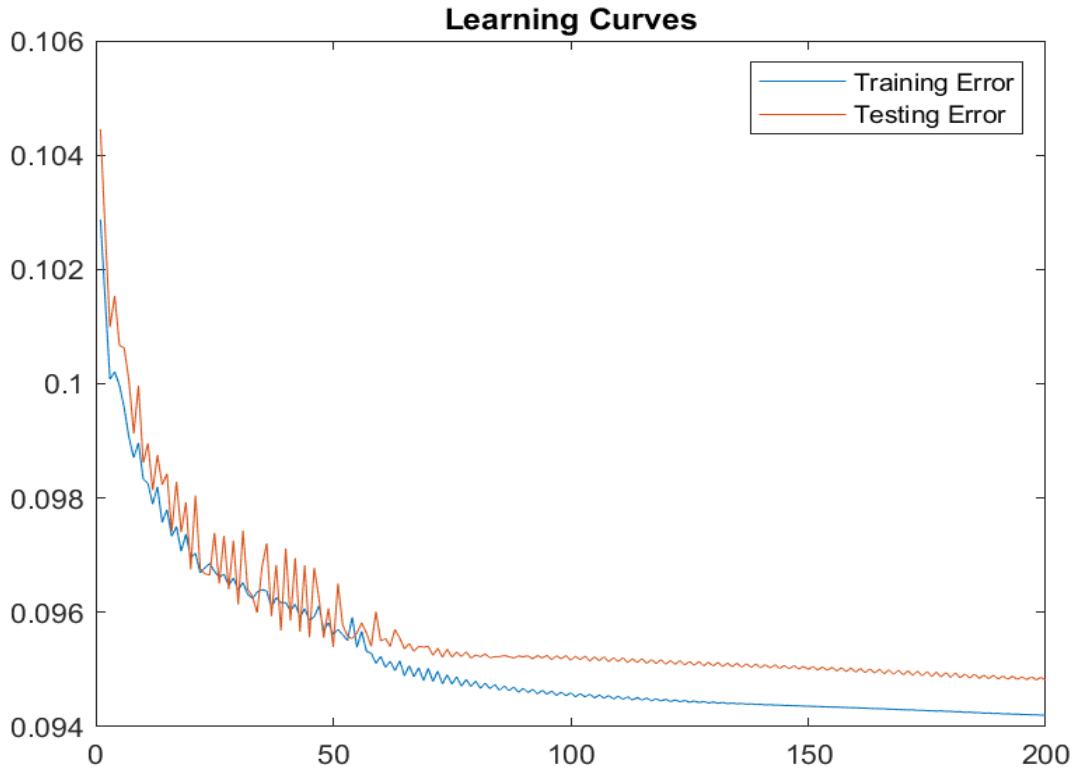
Στο σχήμα απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει τις τιμές της εξόδου αλλά με μεγαλύτερο σφάλμα απότι τα μοντέλα του απλού dataset. Αυτό είναι λογικό καθώς όσο μεγαλώνει η διαστασιμότητα του dataset τόσο πιο δύσκολη είναι η προσέγγιση της εξόδου λόγω της αύξησης παραμέτρων προς βελτιστοποίηση και της διασπασιμότητας του προβλήματος.



Εικόνα 25

Στο σχήμα φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο στο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο από της εκπαίδευσης , αλλά εντός των επιτρεπτών ορίων και με λογικές τιμές. Λόγω του ότι έχουμε μεγάλο αριθμό χαρακτηριστικών και κανόνων η εκμάθηση του μοντέλου είναι πιο συνθέτη , επομένως υπάρχει κάποια διακύμανση στις τιμές του σφάλματος έως ότου σταθεροποιηθεί , .Η τιμή του σφάλματος παρατηρούμε πως είναι μεγαλύτερη από τα μοντέλα που εκπαιδεύτηκαν με απλό dataset κατά περίπου 1.5 τάξη μεγέθους. Αυτό είναι λογικό καθώς η διασπασιμότητα αυτού

του προβλήματος ήταν πολύ μεγαλύτερη και επομένως η διαδικασία εκμάθησης του μοντέλου δυσκολότερη.



Εικόνα 26

Στο σχήμα φαίνονται οι τιμές για τις διάφορες μετρικές για το τελικό μοντέλο.

	mse	rmse	R2	nmse	ndei
final_model	0.008717	0.093365	0.942806	0.251929	0.501926

Εικόνα 27