ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

Υπολογιστική Νοημοσύνη
3ο Παραδοτέο

Παναγιώτης Καρβουνάρης

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Contents

	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

Table 1: Ταξινόμηση μοντέλων προς εκπαίδευση

1 Απλό dataset

Στην πρώτη φάση της εργασίας, επιλέγεται από το UCI repository το Airfoil Self-Noise dataset (βρίσκεται στο αρχείο airfoil_self_noise.dat), το οποίο περιλαμβάνει 1503 δείγματα (instances) και 6 χαρακτηριστικά (features). Θα χρησιμοποιηθεί για μια απλή διερεύνηση της διαδικασίας εκπαίδευσης και αξιολόγησης μοντέλων αυτού του είδους, καθώς και για μια επίδειξη τρόπων ανάλυσης και ερμηνείας των αποτελεσμάτων.

Αρχικά, έγινε διαχωρισμός του dataset σε μη επικαλυπτόμενα υποσύνολα εκπαίδευσης επικύρωσης ελέγχου, από τα οποία το πρώτο χρησιμοποιήθηκε για εκπαίδευση, το δεύτερο για επικύρωση και αποφυγή του φαινομένου υπερεκπαίδευσης και το τελευταίο για τον έλεγχο της απόδοσης του τελικού μας μοντέλου. Έγινε, μάλιστα, χρήση του 60% των δειγμάτων για το υποσύνολο εκπαίδευσης και από 20% του συνόλου των δειγμάτων για κάθε ένα από τα δύο εναπομείναντα υποσύνολα. Ο διαχωρισμός και η απαιτούμενη προεπεξεργασία του dataset έγινε μέσω της MATLAB function split_scale.m (από eLearning μαθήματος) και χρησιμοποιήθηκε με όρισμα και $preproc = 1$, για κανονικοποιήση στην μονάδα του υπερκύβου.

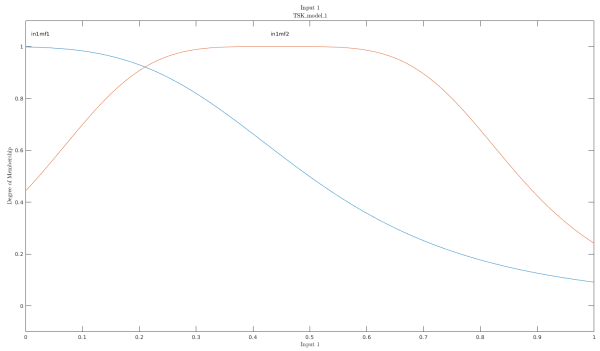
Σε αυτό το στάδιο εξετάζονται διάφορα μοντέλα TSK όσον αφορά την απόδοσή τους στο σύνολο ελέγχου. Συγκεκριμένα, θα εκπαιδευτούν 4 TSK μοντέλα (που περιγράφονται στο Table 1), στα οποία θα μεταβάλλονται η μορφή της εξόδου καθώς και το πλήθος των συναρτήσεων συμμετοχής για κάθε μεταβλητή εισόδου.

Και τα 4 μοντέλα εκπαιδεύονται με την υβριδική μέθοδο, σύμφωνα με την οποία οι παράμετροι των συναρτήσεων συμμετοχής βελτιστοποιούνται μέσω της μεθόδου της οπισθοδιάδοσης (backpropagation algorithm), ενώ οι παράμετροι της πολυωνυμικής συνάρτησης εξόδου βελτιστοποιούνται μέσω της μεθόδου των ελαχίστων τετραγώνων (Least Squares). Οι συναρτήσεις συμμετοχής θα είναι bell-shaped (γι' αυτό ορίζουμε ως ρύθμιση στην genfis() συνάρτηση να είναι InputMembershipFunctionType = 'gbellmf') και η αρχικοποίησή τους θα γίνει με τέτοιον τρόπο ώστε τα διαδοχικά ασαφή σύνολα να παρουσιάζουν σε κάθε είσοδο, βαθμό επικάλυψης περίπου 0.5 (είναι και η default τιμή στα options της genfis(), δηλαδή εννοείται ότι ClusterInfluenceRange = 0.5) (αυτό θα φανεί και πιο ποιοτικά σε διαγράμματα των συναρτήσεων συμμετοχής παρακάτω). Για τον ορισμό του πλήθους συναρτήσεων συμμετοχής κάνουμε χρήση της ρύθμισης NumMembershipFunctions και για την μορφή εξόδου της ρύθμισης OutputMembershipFunctionType (επιλέγοντας 'constant' για Singleton και 'linear' για Polynomial). Θα μελετήσουμε την κλασική περίπτωση του grid partitioning του χώρου εισό-

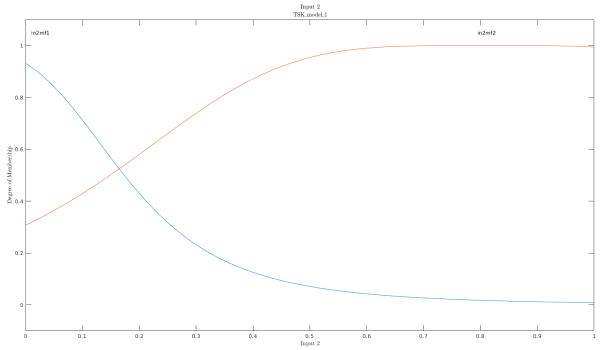
δου, με βάση ασαφών κανόνων με ένα κανόνα για κάθε συνδυασμό συνάρτησης συμμετοχής. Οπότε επιλέγουμε την ρύθμιση 'GridPartition' για όρισμα στην genfisOptions().

Κάθε μοντέλο εκπαιδεύεται για 100 iterations/epochs και για καθεμιά από τις περιπτώσεις των μοντέλων δίνουμε παρακάτω τα διαγράμματα στα οποία απεικονίζονται οι τελικές μορφές των ασαφών συνόλων που προέκυψαν μέσω της διαδικασίας εκπαίδευσης, τα διαγράμματα μάθησης όπου απεικονίζεται το σφάλμα του μοντέλου συναρτήσει του αριθμού των επαναλήψεων (iterations) και τέλος τα διαγράμματα όπου αποτυπώνονται τα σφάλματα πρόβλεψης.

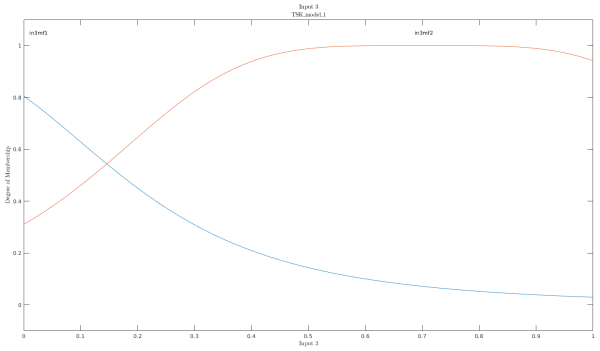
1.1 TSK Model 1



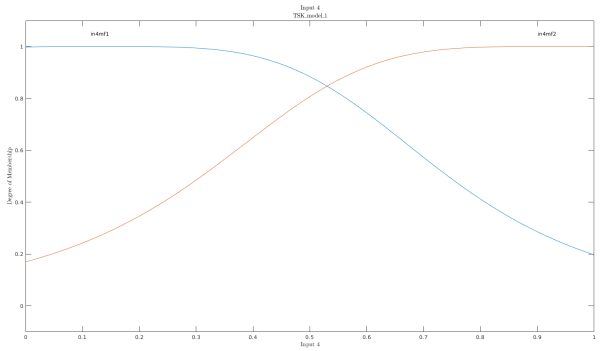
(a)



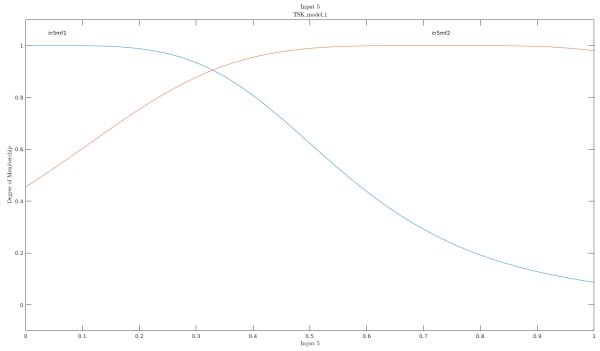
(b)



(c)



(a)



(b)

Figure 2: Διαγράμματα εισόδων στο TSK model 1.

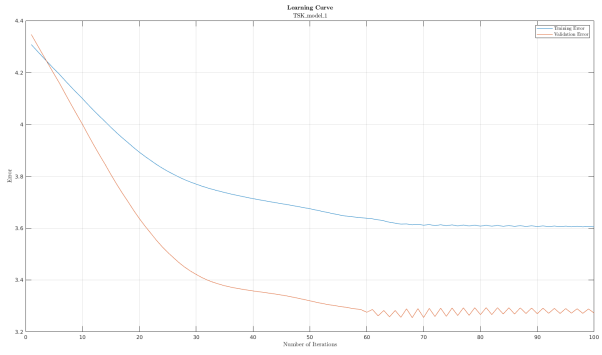


Figure 3: Διάγραμμα learnging curve στο TSK model 1.

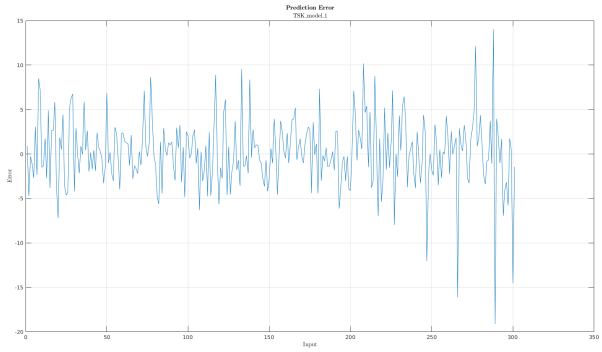
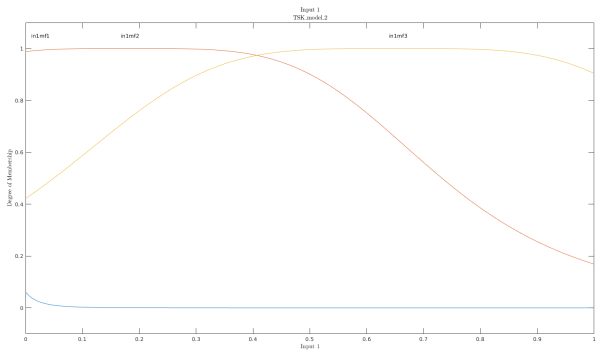
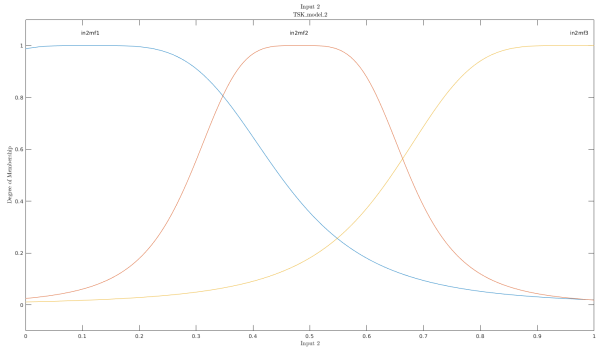


Figure 4: Διάγραμμα λάθος πρόβλεψης στο TSK model 1.

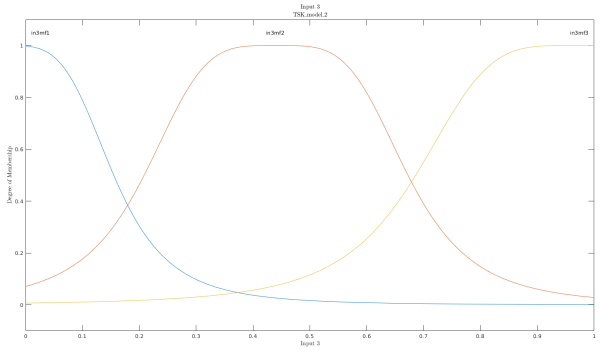
1.2 TSK Model 2



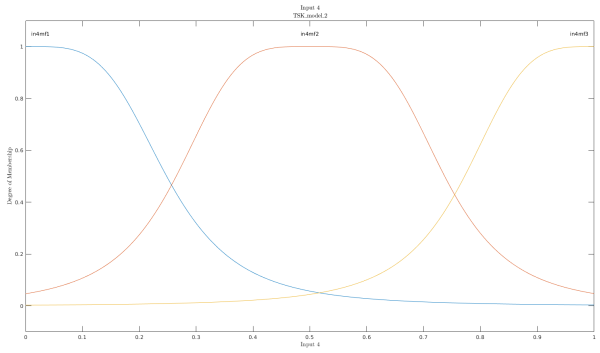
(a)



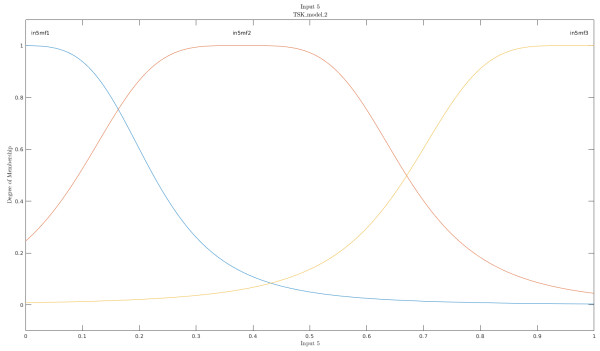
(b)



(c)



(a)



(b)

Figure 6: Διαγράμματα εισόδων στο TSK model 2.

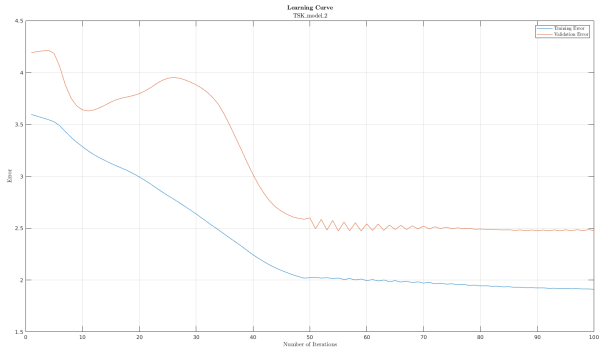


Figure 7: Διάγραμμα learnging curve στο TSK model 2.

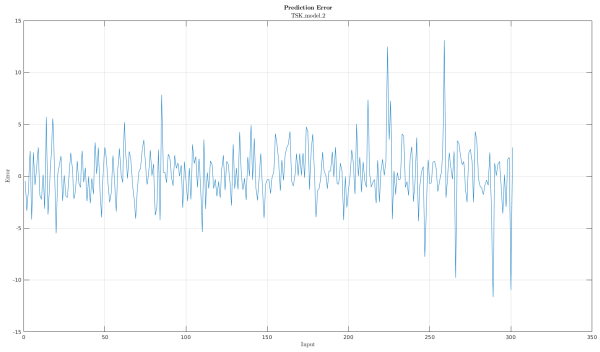
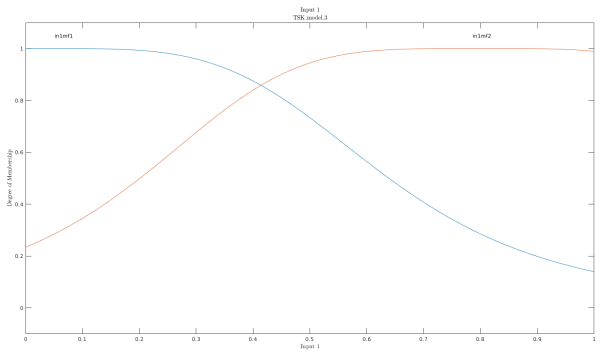
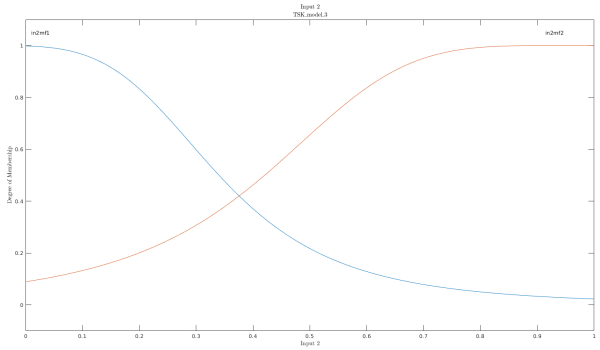


Figure 8: Διάγραμμα λάθος πρόβλεψης στο TSK model 2.

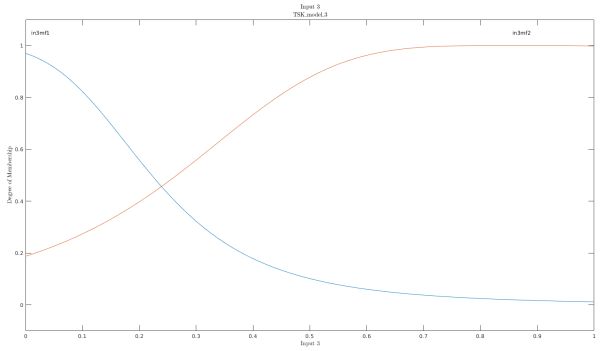
1.3 TSK Model 3



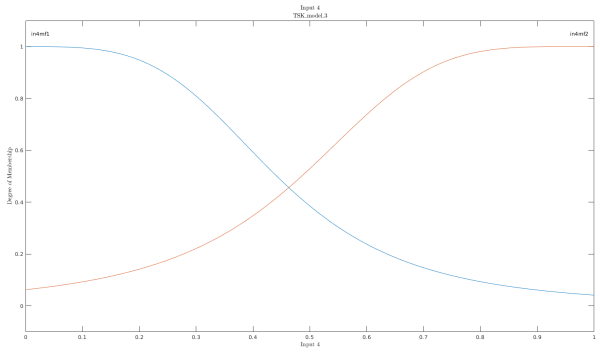
(a)



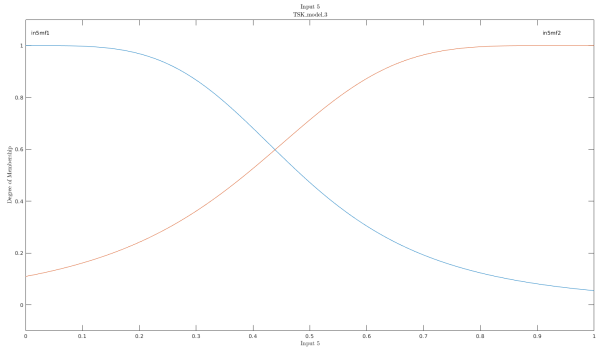
(b)



(c)



(a)



(b)

Figure 10: Διαγράμματα εισόδων στο TSK model 3.

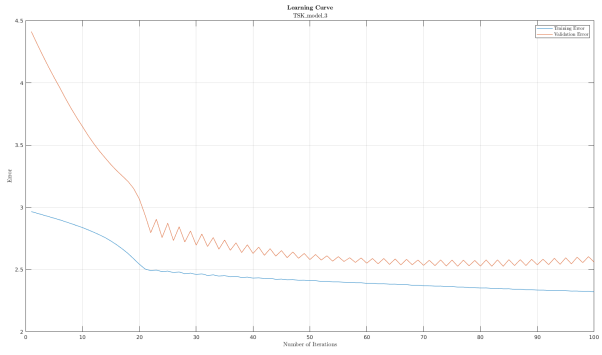


Figure 11: Διάγραμμα learnging curve στο TSK model 3.

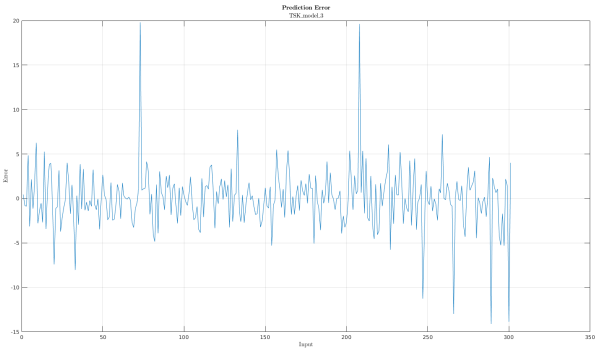
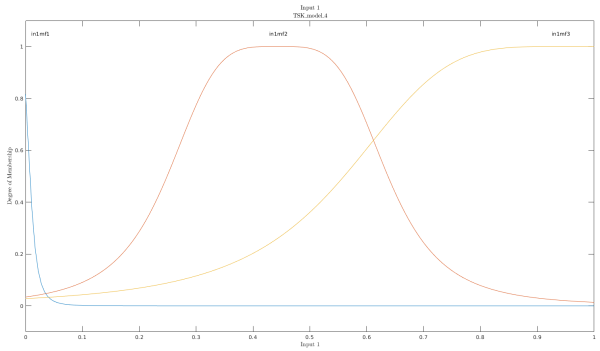
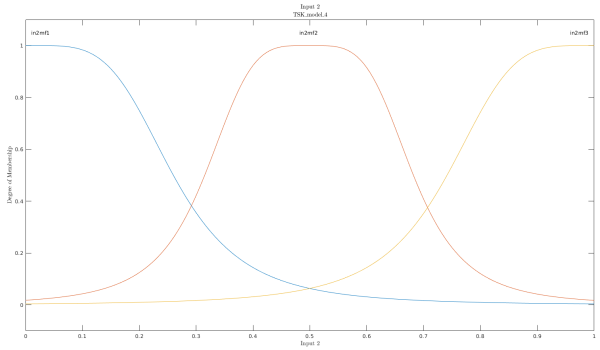


Figure 12: Διάγραμμα λάθος πρόβλεψης στο TSK model 3.

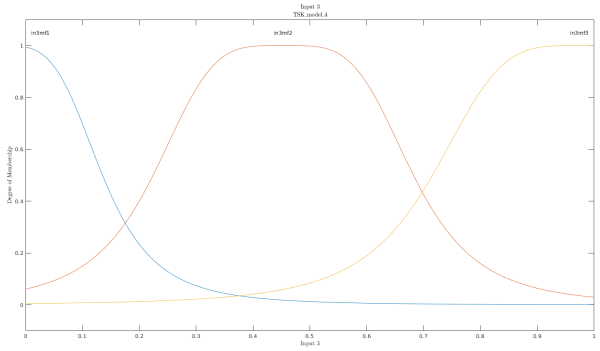
1.4 TSK Model 4



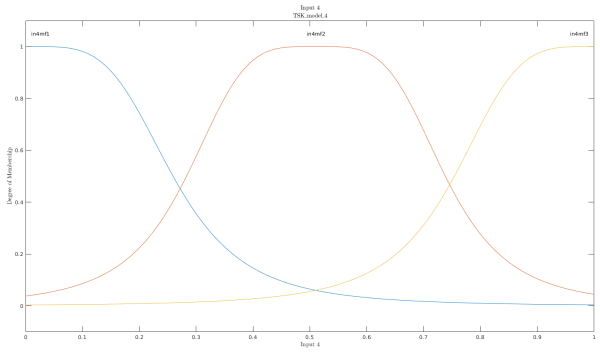
(a)



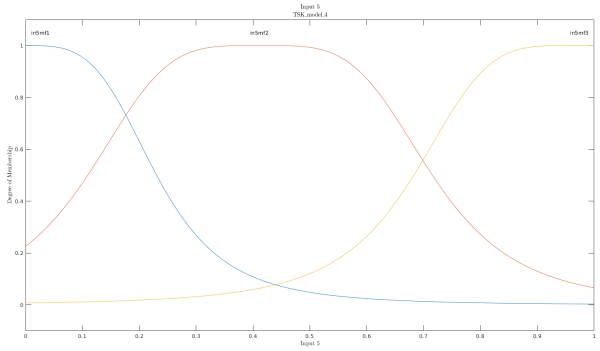
(b)



(c)



(a)



(b)

Figure 14: Διαγράμματα εισόδων στο TSK model 4.

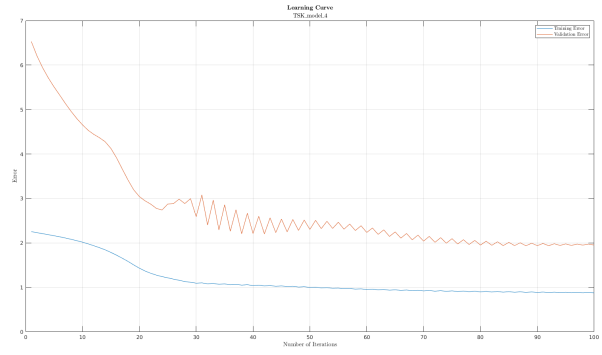


Figure 15: Διάγραμμα learnging curve στο TSK model 4.

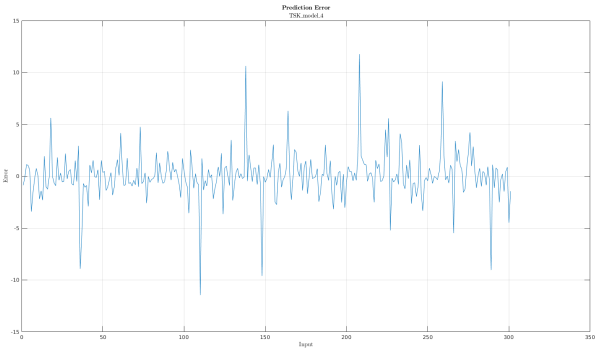


Figure 16: Διάγραμμα λάθος πρόβλεψης στο TSK model 4.

1.5 Δείκτες απόδοσης

	TSK_model_1	**TSK_model_2**	**TSK_model_3**	**TSK_model_4**
MSE	15.101	7.3951	10.914	5.0393
RMSE	3.8861	2.7194	3.3036	2.2448
NMSE	0.33852	0.16577	0.24465	0.11296
NDEI	0.58183	0.40715	0.49463	0.3361
R2	0.66148	0.83423	0.75535	0.88704

Table 2: Χαρακτηριστικά απόδοσης διαφορετικών TSK μοντέλων

Τέλος, παρακάτω παρουσιάζονται σε μορφή πίνακα και οι τιμές των ζητουμένων δεικτών απόδοσης RMSE (τετραγωνική ριζά του μέσου τετραγωνικού σφάλματος μεταξύ της εξόδου του μοντέλου και της πραγματικής εξόδου), NMSE (Normalized Mean Square Error), NDEI (Non-Dimensional Error Index), R^2 (συντελεστής προσδιορισμού) για καθεμιά από τις τέσσερις περιπτώσεις εκπαίδευσης, Table 2.

Οι τύποι/σχέσεις για τους δείκτες απόδοσης δίνονται στην εκφώνηση της εργασίας και τις υλοποιούμε μέσω MATLAB αρχεία συναρτήσεων με συμβολισμό .m (RMSE.m, NMSE.m, NDEI.m, R2.m).

Το MATLAB script που υλοποιεί την εκπαίδευση των παραπάνω 4 TSK μοντέλων, την δημιουργία των διαγραμμάτων και την εκτύπωση των απαραίτητων δεδομένων βρίσκονται στο αρχείο regression_simpleDataset.m . Μάλιστα, έχουμε αποθηκεύσει την έξοδο της κονσόλας του MATLAB της εκπαίδευσης στο αρχείο regression_simpleDataset_logs.txt ενώ τις μεταβλητές στο αρχείο μεταβλητών regression_simpleDataset_Variables.mat. Αξίζει να σημειωθεί πως η επανεκτέλεση του script θα δώσει διαφορετικά αριθμητικά αποτελέσματα του πίνακα 2 και διαφορετικά διαγράμματα λόγω της τυχαιότητας κατά την διαχώριση του dataset στην αρχή του script. Όμως, αυτά θα έχουν τα ιδιά χαρακτηριστικά με αυτά που

δίνουμε εδώ, δηλαδή θα υπακούν στα συγκριτικά συμπεράσματα που θα δώσουμε λίγο πιο μετά.

1.6 Αποτελέσματα

Παρατηρώντας ξεχωριστά τα παραπάνω διαγράμματα των τεσσάρων TSK μοντέλων βλέπουμε για το:

- **TSK_model_1**. Το εκπαιδευόμενο μοντέλο έκανε ένα μέτριο fitting στο dataset, αφού ο συντελεστής προσδιορισμού R^2 είναι 0.66148 (>0.5 και <0.8). Επίσης φαίνεται από το Figure 3 πως δεν έχουμε το φαινόμενο της υπερεκμάθησης (overfitting) (τουλάχιστον για τα 100 iterations που μελετάμε) καθώς όσο μειώνεται το σφάλμα το training error φαίνεται να μειώνεται και το validation error (πάρα την μεγάλη διαφορά τιμών τους και το σχετικά μεγάλο MSE με μεγάλη διακύμανση υποθέτουμε πως για περισσότερα iterations δεν θα παρατηρούσαμε μεγάλη διαφορά, επειδή φαίνεται στα τελευταία iterations μία μικρή ταλάντωση της τιμής γύρω από μία συγκεκριμένη τιμή.

- **TSK_model_2**. Το εκπαιδευόμενο μοντέλο έκανε πολύ καλό fitting στο dataset, αφού ο συντελεστής προσδιορισμού R^2 είναι 0.83423 (>0.8). Επίσης, φαίνεται από το Figure 7 πως εμφανίζεται το φαινόμενο της υπερεκμάθησης (overfitting) καθώς βλέπουμε πως ενώ το training error όλο και μειώνεται, το validation error έχει τάση αύξησης (ειδικά μετά τα 10 iterations, μέχρι το 25ο iteration), δηλαδή το μοντέλο αρχίζει να «ανταποκρίνεται» ικανοποιητικά μόνο στα δεδομένα εκμάθησης αλλά κακά σε άλλα. Αυτό στην συνέχεια βέβαια φαίνεται να βελτιώνεται δραστικά. Βλέπουμε το validation error να μειώνεται και στα τελευταία iterations να έχει μικρές ταλαντώσεις και ο ρυθμός πτώσης του να είναι ιδιαίτερα μικρός.

- **TSK_model_3**, έχουμε παρόμοια συμπεριφορά με το TSK_model_1, δηλαδή πάλι πως το εκπαιδευόμενο μοντέλο έκανε ένα μέτριο προς καλό fitting στο dataset, αφού ο συντελεστής προσδιορισμού R^2 είναι 0.75535 (>0.5 και <0.8). Πιο συγκεκριμένα, βλέπουμε στο Figure 11 πως τόσο το training όσο και το validation error μειώνονται διαρκώς. Μετά από κάποιο σημείο φαίνεται ότι έχουμε ταλαντώσεις στο validation error, καθώς σταθεροποιείται γύρω από μία τιμή.

- **TSK_model_4**, σε σχέση με τα παραπάνω 3 TSK μοντέλα, έκανε το καλύτερο fitting στο dataset, αφού ο συντελεστής προσδιορισμού R^2 είναι 0.88704 (>0.8). Επίσης, εύκολα διακρίνουμε από το Figure 15 πως δεν έχουμε εμφάνιση του φαινομένου της υπερεκμάθησης, καθώς όσο συνεχώς μειώνεται το training error, τόσο μειώνεται συνεχώς και το validation error. Άρα φαίνεται να είχαμε καλό split στο dataset και καλή εφαρμογή του regression μοντέλου. Μάλιστα, έχουμε και το μικρότερο MSE (5.0393). Άρα φαίνεται να δίνει και τα καλύτερα αποτελέσματα, αν και στα τελευταία βήματα δίνει μία συμπεριφορά ανόλογη του TSK_model_4 με μικρές ταλαντώσεις γύρω από μία τιμή που μειώνεται με μικρό ρυθμό.

1.7 Συμπεράσματα

Για την σωστή σύγκριση, όμως, των αποδόσεων των τεσσάρων TSK μοντέλων μεταξύ τους θα χρησιμοποιήσουμε ως μέτρο σύγκρισης τους κανονικοποιημένους δείκτες R^2 για το fit και το NMSE για το σφάλμα εκπαίδευσης.

Συγκρίνοντας, τώρα, ανά δυο τα μοντέλα που είναι όμοια ως προς την μορφή εξόδου, δηλαδή συγκρίνοντας μεταξύ τους το TSK_model_1 με το TSK_model_2 (Singleton) και το TSK_model_3 με το TSK_model_4 (Polynomial), βλέπουμε πως στην 1 περίπτωση (Singleton) το TSK_model_1 (με 2 συναρτήσεις συμμετοχής) έχει NMSE = 0.33852 και R^2 = 0.66148, ενώ το TSK_model_2 (με 3 συναρτήσεις συμμετοχής) έχει μικρότερο NMSE = 0.16577 και μεγαλύτερο R^2 = 0.83423 . Στην άλλη περίπτωση (Polynomial) το TSK_model_3 (με 2 συναρτήσεις συμμετοχής) έχει NMSE = 0.24465 και R^2 = 0.75535, ενώ το TSK_model_4 (με 3 συναρτήσεις συμμετοχής) έχει πάλι μικρότερο NMSE = 0.11296 και πάλι μεγαλύτερο R^2 = 0.88704. Άρα, γενικά η αύξηση του πλήθους των συναρτήσεων συμμετοχής (κρατώντας σταθερή την μορφή εξόδου) αυξάνει το R^2 και ταυτόχρονα μειώνει και το μέσο κανονικοποιημένο σφάλμα εκπαίδευσης NMSE. Αξίζει βέβαια εδώ να λάβουμε και υπόψη μας την μεγάλη αύξηση του αριθμού των IF-THEN κανόνων από την αύξηση των συναρτήσεων συμμετοχής, η λεγόμενη «έκρηξη» του πλήθους των IF-THEN κανόνων (rule explosion). Στην περίπτωση μας εδώ δεν αποτελεί μεγάλο πρόβλημα, καθώς έχουμε dataset μικρής διαστασιμότητας. Άρα γενικά οδηγούμαστε σε καλύτερα αποτελέσματα με περισσότερες συναρτήσεις συμμετοχής, αφού έτσι παρέχουμε καλύτερη κάλυψη του χώρου εισόδου και το TSK μοντέλο προσαρμόζεται στις διακυμάνσεις των δεδομένων.

Συγκρίνοντας, τώρα, ανά δυο τα μοντέλα που είναι όμοια ως προς τον αριθμό των συναρτήσεων συμμετοχής, δηλαδή συγκρίνοντας μεταξύ τους το TSK_model_1 με το TSK_model_3 (με 2 συναρτήσεις συμμετοχής) και το TSK_model_2 με το TSK_model_4 (με 3 συναρτήσεις συμμετοχής), βλέπουμε πως η στην 1 περίπτωση (2 συναρτήσεις συμμετοχής) το TSK_model_1 (Singleton) έχει NMSE = 0.33852 και R^2 = 0.66148, ενώ το TSK_model_3 (Polynomial) έχει μικρότερο NMSE = 0.24465 και μεγαλύτερο R^2 = 0.75535. Στην άλλη περίπτωση (3 συναρτήσεις συμμετοχής) το TSK_model_2 (Singleton) έχει NMSE = 0.16577 και R^2 = 0.83423 ενώ το TSK_model_4 (Polynomial) έχει πάλι μικρότερο NMSE = 0.11296 και πάλι μεγαλύτερο R^2 = 0.88704. Άρα γενικά διατήρηση περισσότερων όρων στην έξοδο του κάθε κανόνα του μοντέλου (κρατώντας σταθερό τον αριθμό των συναρτήσεων συμμετοχής) αυξάνει το R^2 και ταυτόχρονα μειώνει και το μέσο κανονικοποιημένο σφάλμα εκπαίδευσης NMSE. Άρα γενικά οδηγεί σε καλύτερα αποτελέσματα η επιλογή Polynomial ως μορφή εξόδου, αφού επιτρέπει πιο σύνθετες (και άρα αποτελεσματικές ως προς το αποτέλεσμα) σχέσεις με την εισαγωγή πολυωνυμικών όρων.

Τα παραπάνω ερμηνεύουν και τα σχόλια που κάναμε παραπάνω για τις αποδόσεις των τεσσάρων μοντέλων και το TSK_model_4 δίνει άρα και δικαιολογημένα τα καλύτερα αποτελέσματα αφού έχει τις περισσότερες συναρτήσεις συμμέτοχης αλλά και μορφή εξόδου Polynomial.

Αξίζει, τελικά, να σημειώσουμε πως, στο TSK_model_4 (σε αντίθεση με το TSK_model_2 που έχει επίσης το ίδιο πλήθος ασαφών συνόλων ανά είσοδο) το

μεγαλύτερο πλήθος ασαφών συνόλων ανά είσοδο στην περίπτωση των αντίστοιχων TSK μοντέλων δεν οδήγησε σε υπερεκπαίδευση (overfitting) του μοντέλου. Οπότε δεν προκύπτει πειραματικά εδώ κάποια συσχέτιση μεταξύ του πλήθους των ασαφών συνόλων ανά είσοδο και του φαινομένου της υπερεκπαίδευσης.

2 Dataset υψηλής διαστασιμότητας

Στη δεύτερη φάση της εργασίας θα ακολουθηθεί μια πιο συστηματική προσέγγιση στο πρόβλημα μοντελοποίησης μιας άγνωστης συνάρτησης. Για το σκοπό αυτό θα επιλεχθεί ένα dataset με υψηλότερο βαθμό διαστασιμότητας. Το dataset που θα επιλεγεί για την επίδειξη των παραπάνω μεθόδων είναι το Superconductivty dataset από το UCI Repository, το οποίο περιλαμβάνει 21263 δείγματα καθένα από τα οποία περιγράφεται από 81 μεταβλητές/χαρακτηριστικά (βρίσκεται στο αρχείο superconduct.csv).

Ένα προφανές πρόβλημα που ανακύπτει από την επιλογή αυτή, είναι η λεγόμενη «έκρηξη»του πλήθους των IF-THEN κανόνων (rule explosion). Όπως είναι γνωστό από τη θεωρία, για την κλασική περίπτωση του grid partitioning του χώρου εισόδου, ο αριθμός των κανόνων αυξάνεται εκθετικά σε σχέση με το πλήθος των εισόδων, γεγονός που καθιστά πολύ δύσκολη την μοντελοποίηση μέσω ενός TSK μοντέλου ακόμα και για datasets μεσαίας κλίμακας (εδώ π.χ. με 81 μεταβλητές/predictors, αν διαμερίζαμε το χώρο εισόδου κάθε μεταβλητής με δύο ασαφή σύνολα, θα καταλήγαμε με 2^{81} κανόνες).

Επομένως, καθίσταται αναγκαία η χρήση μεθόδων μείωσης της διαστασιμότητας καθώς και του αριθμού των IF-THEN κανόνων. Οι δύο αυτές μέθοδοι όμως, παρά τη ελάττωση της πολυπλοκότητας που επιφέρουν, εισάγουν στο πρόβλημα δύο ελεύθερες παραμέτρους, συγκεκριμένα, τον αριθμό των χαρακτηριστικών προς επιλογή και τον αριθμό των ομάδων που θα δημιουργηθούν. Στην παρούσα εργασία, υλοποιήθηκε η μέθοδος αναζήτησης πλέγματος (grid search) για την εύρεση των βέλτιστων τιμών των παραμέτρων αυτών.

2.1 Grid search

Αρχικά, έγινε πάλι διαχωρισμός του dataset σε μη επικαλυπτόμενα υποσύνολα εκπαίδευσης-επικύρωσης-ελέγχου, από τα οποία το πρώτο χρησιμοποιήθηκε για εκπαίδευση, το δεύτερο για επικύρωση και αποφυγή της υπερεκπαίδευσης και το τελευταίο για τον έλεγχο της απόδοσης του τελικού μας μοντέλου. Έγινε, μάλιστα, χρήση του 60% των δειγμάτων για το υποσύνολο εκπαίδευσης και από 20% του συνόλου των δειγμάτων για κάθε ένα από τα δύο εναπομείναντα υποσύνολα. Πάλι, ο διαχωρισμός και η απαιτούμενη προεπεξεργασία του dataset έγινε μέσω πάλι της MATLAB function split_scale.m και χρησιμοποιήθηκε με όρισμα και preproc = 1, για κανονικοποιήση στην μονάδα του υπερκύβου.

Όπως αναφέρθηκε παραπάνω, το σύστημά μας περιλαμβάνει δύο ελεύθερες παραμέτρους την τιμή των οποίων πρέπει να επιλέξουμε εμείς. Για τους σκοπούς της εργασίας, ορίζουμε τις εξής παραμέτρους:

- **Αριθμός χαρακτηριστικών:** Το πλήθος των χαρακτηριστικών που θα χρησιμοποιηθούν στην εκπαίδευση των μοντέλων (μεταβλητή/διάνυσμα τιμών με όνομα numOfFeatures).

- **Ακτίνα των clusters** r_a: Η παράμετρος που καθορίζει την ακτίνα επιρροής των clusters και κατ' επέκταση το πλήθος των κανόνων που θα προκύψουν (μεταβλητή/διάνυσμα τιμών με όνομα clusterRadius).

Εφόσον ο καθορισμός των τιμών των παραμέτρων μπορούν να επιλεχθούν ελεύθερα, επιλέξαμε τον έλεγχο για τις τιμές:

$$numOfFeatures = \begin{bmatrix} 4 & 6 & 8 & 10 \end{bmatrix}$$

$$clusterRadius = \begin{bmatrix} 0.25 & 0.5 & 0.75 & 1 \end{bmatrix}$$

Η σκέψη για την επιλογή των τιμών προς μελέτη των ακτινών των clusters και των αριθμών των features έγινε αυθαίρετα αφού θέλουμε να μελετήσουμε ένα μοντέλο με διαφορετικές τάξεις μεγέθους ακτινών καθώς και να μην έχουμε μια ιδιαίτερα χρονοβόρα διαδικασία διασταυρωμένης επικύρωσης, λόγω προσωπικής μικρής υπολογιστικής δύναμης (αρά θέλουμε σχετικά μικρό αριθμό features).

Στην συνέχεια δημιουργούμε ένα 2-διάστατο πλέγμα όπου κάθε σημείο του πλέγματος αντιστοιχεί σε μια 2-άδα τιμών για τις εν λόγω παραμέτρους. Σε αυτά τα σημεία χρησιμοποιούμε την μέθοδο αξιολόγησης διασταυρωμένης επικύρωσης (cross validation) για να ελέγξουμε την ορθότητα των συγκεκριμένων τιμών. Μάλιστα, θα γίνει αξιολόγηση μέσω 5-πτυχης διασταυρωμένης επικύρωσης (5-fold cross validation) (ορίζουμε numOfFolds = 5) για την επιλογή των βέλτιστων τιμών των παραμέτρων. Σύμφωνα με τη μέθοδο αυτή, και για τις παραπάνω επιλεγμένες τιμές των παραμέτρων, χωρίζουμε το σύνολο εκπαίδευσης σε δύο υποσύνολα, από τα οποία το ένα χρησιμοποιείται για την εκπαίδευση ενός μοντέλου και το δεύτερο για την αξιολόγησή του. Η διαδικασία αυτή επαναλαμβάνεται πέντε φορές όπου κάθε φορά χρησιμοποιείται διαφορετικός διαχωρισμός του συνόλου εκπαίδευσης (αυτό το επιτυγχάνουμε ευκολά με την χρήση της MATLAB συνάρτησης cvpartition() με όρισμα το εκάστοτε numOfFolds κάθε φορά), και στο τέλος λαμβάνουμε τον μέσο όρο του σφάλματος του μοντέλου. Ο διαχωρισμός των δεδομένων γίνεται έτσι ώστε σε κάθε επανάληψη, το 80% των δεδομένων να χρησιμοποιείται για εκπαίδευση και το υπόλοιπο 20% για επικύρωση (ως είσοδοι στη συνάρτηση anfis() του MATLAB, κάνοντας ταυτόχρονα και τους απαραίτητους ελέγχους για την σωστή λειτουργία του script). Η λογική πίσω από τις πολλαπλές εκπαιδεύσεις και ελέγχους έγκειται στο ότι με αυτό τον τρόπο, αποκτούμε μια αρκετά καλή εκτίμηση της απόδοσης του μοντέλου, και έμμεσα των τιμών των παραμέτρων με βάση τις οποίες χτίστηκε το μοντέλο.

Ως μέθοδος ομαδοποίησης για τη δημιουργία των IF-THEN κανόνων ζητήθηκε να είναι ο αλγόριθμος του Subtractive Clustering (SC). Περνάμε γι' αυτό ως όρισμα στην genfisOptions(), για την genfis() και την δημιουργία του FIS μοντέλου, την επιλογή ifThenRules_teamMethod = 'SubtractiveClustering'.

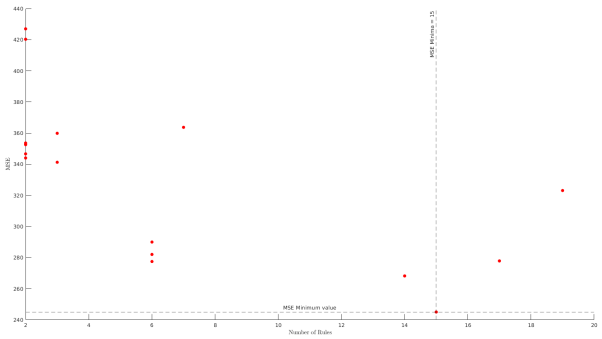


Figure 17: MSE συναρτήσει του πλήθους κανόνων.

Η επιλογή χαρακτηριστικών ζητήθηκε να εκτελεστεί με τον αλγόριθμο Relief (μέσω της MATLAB συνάρτησης relieff()) για k-nearest neighbors, με k = 10, επιλέγοντας μάλιστα και ως 'method' προφανώς την επιλογή 'regression'.

Επίσης, για κάθε cross validation, επιλέξαμε τα μοντέλα να εκπαιδευτούν, πάλι, για 100 iterations.

Συνοπτικά, αρά, η λογική του MATLAB script είναι πως για κάθε grid point, δηλαδή για κάθε τριάδα (fold, numOfFeatures, clusterRadius), εκπαιδεύουμε με βάση το training dataset (για το fold που έχουμε) ένα TSK μοντέλο με SC στο χώρο των εισόδων με παράμετρο την παραπάνω ακτίνα αναζήτησης και παίρνουμε το τελικό validation error του μοντέλου στο test dataset (για το fold που έχουμε) με την απλή χρήση των συναρτήσεων genfis() και anfis() του MATLAB και το αποθηκεύουμε στα αντίστοιχα βοηθητικά διανύσματα. Τέλος, για κάθε δυάδα (numOfFeatures, clusterRadius) βρίσκουμε την mean() τιμή των MSE των 5 folds και το αποθηκεύουμε εκ νέου.

Στο Figure 17 φαίνεται το ζητούμενο διάγραμμα που απεικονίζει το μέσο σφάλμα σε σχέση με τον αριθμό των κανόνων. Παρατηρούμε, πως καθώς αυξάνεται το πλήθος των κανόνων, μειώνεται το MSE. Αυτό είναι και το λογικό καθώς όσο αυξάνουμε των αριθμό των κανόνων, το μοντέλο είναι λογικό να ανταπεξέλθει καλυτέρα στην είσοδο του καθώς αυτή περνάει από περισσοτέρους IF-THEN ελέγχους και η πιθανότητα ή η τιμή του λάθους μειώνεται. Όμως, όπως είναι λογικό, έτσι αυξάνουμε την πολυπλοκότητα του μοντέλου και άρα τους χρόνους εκτέλεσης του regression.

Επίσης, στο Figure 18 δίνουμε το ζητούμενο διάγραμμα που απεικονίζει το μέσο σφάλμα σε σχέση με όλες τις τιμές των παραμέτρων, δηλαδή και με τον αριθμό των επιλεχθέντων χαρακτηριστικών αλλά και με των ακτινών των clusters. Απεικονίζουμε το διάγραμμα αυτό σε 3D (surface plot) μορφή για μια πιο πλήρη απεικόνιση. Παρατηρούμε, πως η επιφάνεια του MSE είναι γενικά φθίνουσα αρχικά όσο μειώνεται η τιμή της ακτίνας των clusters αλλά και όσο αυξάνεται ο αριθμός των χαρακτηριστικών. Αυτό γίνεται αφού όσο αυξάνουμε τον αριθμό των χαρακτηριστικών, το μοντέλο εκπαιδεύεται σε μεγαλύτερης διάστασης δεδομένα

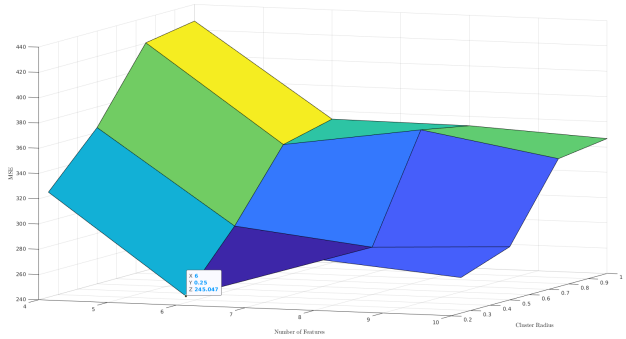


Figure 18: MSE συναρτήσει τιμών όλων των παραμέτρων του grid search.

και μπορεί να ανταπεξέλθει καλύτερα στις εισόδους του, αργότερα όμως αυτό φαίνεται να αντιστρέφεται, δημιουργώντας έτσι ένα ολικό ελάχιστο στο σημείο όπου NumbeOfFeatures = 6, ClustersRadius = 0.25.

Το MATLAB script που υλοποιεί το grid search process και την δημιουργία των διαγραμμάτων βρίσκονται στο αρχείο regression_highDimDataset_gridSearch.m . Μάλιστα, έχουμε αποθηκεύσει την έξοδο της κονσόλας του MATLAB της συνεδρίας στο αρχείο regression_highDimDataset_gridSearch_logs.txt ενώ ολες οι μεταβλητές που δημιουργήθηκαν (λόγω μεγάλου απαιτούμενου χρόνου εκτέλεσης) στο MATLAB αρχείο regression_highDimDataset_gridSearch_Variables.mat .

2.2 Εκπαίδευση και απόδοση TSK μοντέλου με βάση τις βέλτιστες τιμές

Τώρα που εκτελέστηκε η παραπάνω διαδικασία για κάθε σημείο του πλέγματος, λαμβάνουμε ως βέλτιστες τιμές των παραμέτρων, τις τιμές που αντιστοιχούν στο μοντέλο που παρουσίασε το ελάχιστο μέσο σφάλμα. Όπως εύκολα διακρίνουμε και από το σημείο ελαχίστου στο Figure 18, οι τιμές αυτές είναι NumbeOfFeatures = 6, ClustersRadius = 0.25.

Τρέχοντας τώρα άλλη μια φορά την ίδια παραπάνω μέθοδο και με τις ίδιες προδιαγραφές όπως και προηγουμένως (αλλά αυτή την φορά για τις παραπάνω συγκεκριμένες τιμές) εκπαιδεύουμε το τελικό TSK μοντέλο. Έτσι, δίνονται τα διαγράμματα εκμάθησης όπου να απεικονίζεται το σφάλμα συναρτήσει του αριθμού επαναλήψεων, Figure 19, οι προβλέψεις του τελικού μοντέλου καθώς και οι πραγματικές τιμές Figure 20 και το διάγραμμα της διαφοράς τους Figure 21.

Επίσης, στα Figure 22 και Figure 22 δίνουμε όλα τα ασαφή σύνολα στην αρχική, δηλαδή πριν την διαδικασία της εκπαίδευσης, και τελική τους μορφή, δηλαδή του εκπαιδευμένου μοντέλου, αντίστοιχα.

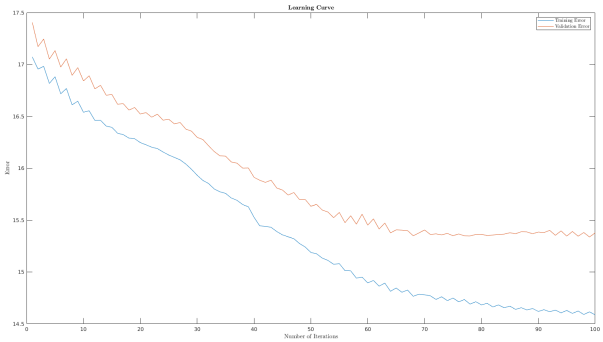


Figure 19: Πραγματικές τιμές.

(a) Πραγματικές τιμές.

(b) Προβλέψεις τιμών.

Figure 20: Πραγματικές τιμές και τιμές πρόβλεψης μοντέλου.

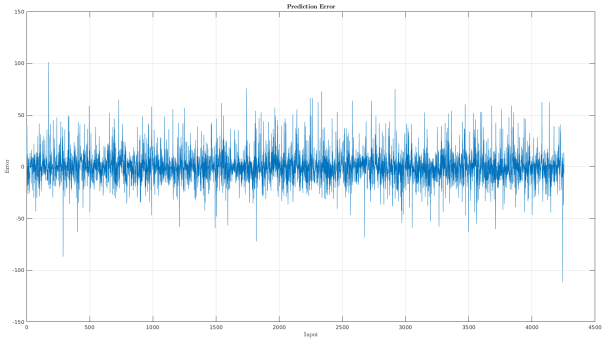
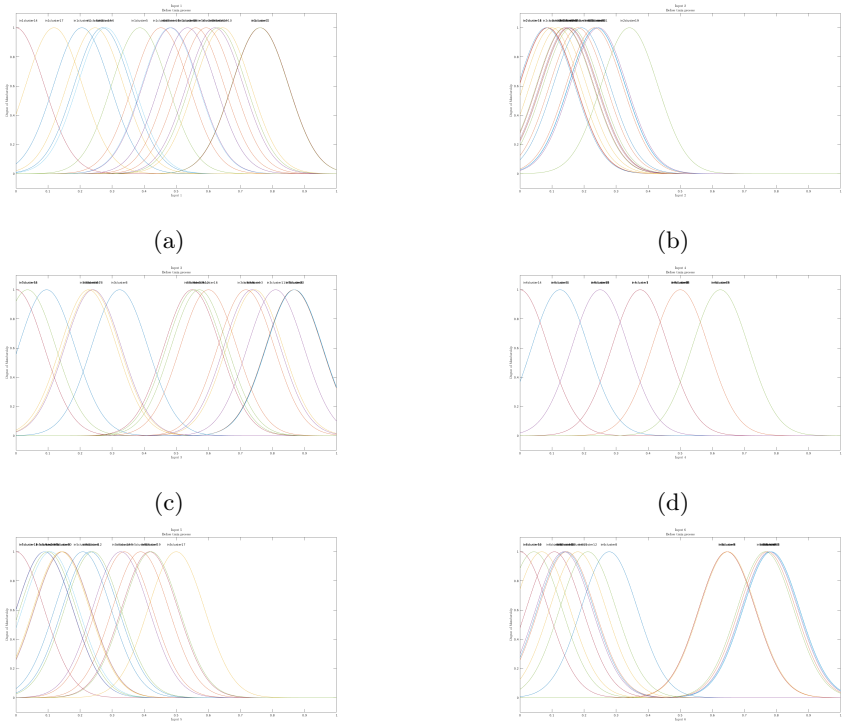
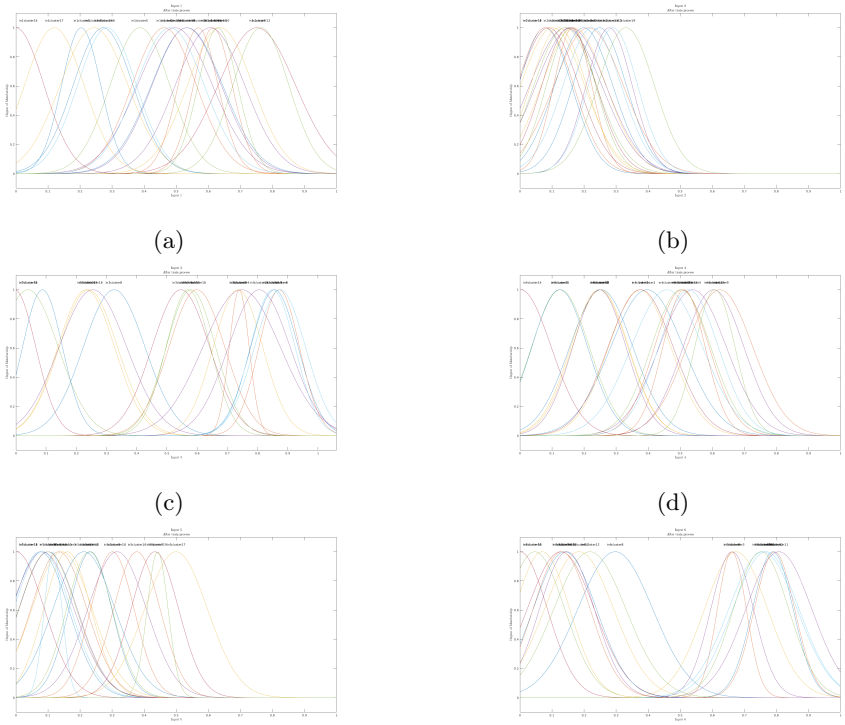


Figure 21: Σφάλματα πρόβλεψης κατά την εφαρμογή του μοντέλου βέλτιστων τιμών.



(a) (b) (c) (d) (e) (f)

Figure 22: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου μοντέλου βέλτιστων τιμών.



(a) (b)

(c) (d)

(e) (f)

Figure 23: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου μοντέλου βέλτιστων τιμών.

	Optimal Model
MSE	225.79
RMSE	15.026
NMSE	0.19591
NDEI	0.44262
R2	0.80409

Table 3: Performance metrics for the optimal model

2.3 Δείκτες απόδοσης

Τέλος, στο Table 3 παρουσιάζονται σε μορφή πίνακα και οι τιμές των ζητουμένων δεικτών απόδοσης RMSE, NMSE, NDEI, R^2 για το τελικό TSK μοντέλο με τις βέλτιστες τιμές των παραμέτρων, κάνοντας χρήση των ιδίων συναρτήσεων που χρησιμοποιήσαμε στο πρώτο μέρος της εργασίας

Το MATLAB script που υλοποιεί την εκπαίδευση του TSK μοντέλου με βάση τις βέλτιστες τιμές που βρήκαμε, την δημιουργία των διαγραμμάτων του και εκτύπωση των ζητούμενων τιμών στην κονσόλα του κομματιού αυτού βρίσκονται στο αρχείο regression_highDimDataset_optimalVals.m (γίνεται των προηγούμενων μεταβλητών, regression_highDimDataset_gridSearch_Variables.mat). Μάλιστα, έχουμε αποθηκεύσει την έξοδο της κονσόλας του MATLAB της συνεδρίας στο αρχείο regression_highDimDataset_optimalVals_logs.txt ενώ τις μεταβλητές που δημιουργήθηκαν (λόγω μεγάλου απαιτούμενου χρόνου εκτέλεσης) στην συνεδρία της εκπαίδευσης του βέλτιστου μοντέλου στο MATLAB αρχείο μεταβλητών regression_highDimDataset_optimalVals_Variables.mat .

2.4 Σχολιασμός αποτελεσμάτων

Παρατηρούμε πως πάρα την χρήση μόνο των 6 χαρακτηριστικών (από τα 81 του dataset) για την εκπαίδευση του ασαφούς νευρωνικού δικτύου, δημιουργήσαμε ένα πολύ καλό regression model και παίρνουμε πολύ καλά ικανοποιητικά στο τέλος. Πιο συγκεκριμένα, βλέπουμε από το Figure 19 πως δεν έχουμε εμφάνιση του φαινομένου της υπερεκπαίδευσης (overfitting), αφού όσο μειώνεται το σφάλμα για το training dataset, τόσο μειώνεται και το σφάλμα για το validation dataset. Επίσης, βλέπουμε πως έχουμε και ένα πάρα πολύ καλό fitting του TSK μοντέλου μας στα δεδομένα παίρνοντας $R^2 = 0.80409$ (>0.8) ενώ από την τιμή του NMSE = 0.19591 βλέπουμε και ένα πολύ μικρής τάξης μέσο σφάλμα καθώς και πολύ μικρή διακύμανση. Αυτό φαίνεται και από τα Figure 20 και Figure 21.

Όπως φαίνεται από το Figure 17, το ελάχιστο MSE (δηλαδή της περίπτωσης που μελετάμε στο κομμάτι αυτό) το παίρνουμε για μόνο 15 IF-THEN κανόνες. Αν για το ίδιο πλήθος χαρακτηριστικών, είχαμε επιλέξει την μέθοδο του Grid Partitioning με δύο (ή τρία) ασαφή σύνολα ανά είσοδο τότε προφανώς θα είχαμε συνολικά 2^{19} (ή 3^{19}) κανόνες. Αυτό προφανώς θα εκτόξευε εκθετικά την απαιτούμενη υπολογιστική δύναμη ή και χρόνο για την εκτέλεση ή και training του regression model, αφού η είσοδος στο μοντέλο θα έπρεπε να περάσει από όλους αυτούς τους

κανόνες (2^{19} » 19). Αξίζει, όμως, να σημειωθεί πως σε εκείνη την περίπτωση θα είχαμε καλυτέρα αποτελέσματα ως προς το σφάλμα. Οπότε δημιουργούμε έτσι ένα trade-off σφαλμάτων – χρόνων εκτέλεσης/εκπαίδευσης για τις μεθόδους Grid Partitioning και SC αντίστοιχα. Άρα, σε κάθε περίπτωση πρέπει να δούμε τι μας ενδιαφέρει περισσότερο από τα δύο κομμάτια του trade-off και να επιλέγουμε την κατάλληλη μέθοδο για dataset μεγάλης διαστασιμότητας.