



ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

Υπολογιστική Νοημοσύνη

4ο Παραδοτέο

Παναγιώτης Καρβουνάρης

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Contents

1	Aπλό dataset	3
1.1	Εκπαίδευση TSK Μοντέλων	3
1.2	Δείκτες απόδοσης	5
1.3	Διαγράμματα TSK Μοντέλων	5
1.4	TSK_model_1	6
1.5	TSK_model_2	8
1.6	TSK_model_3	10
1.7	TSK_model_4	12
1.8	Σχολιασμός αποτελεσμάτων	13
2	Dataset υψηλής διαστασιμότητας	15
2.1	Grid search	16
2.2	Εκπαίδευση και απόδοση TSK μοντέλου με βάση τις βέλτιστες τιμές	19
2.3	Δείκτες απόδοσης	24
2.4	Σχολιασμός αποτελεσμάτων	25

	Ακτίνα/μέγεθος clusters	Μέθοδος clustering
TSK_model_1	0.1	Class Independent
TSK_model_2	0.9	Class Independent
TSK_model_3	0.1	Class Dependent
TSK_model_4	0.9	Class Dependent

Table 1: Ταξινόμηση μοντέλων προς εκπαίδευση

1 Απλό dataset

Στην πρώτη φάση της εργασίας, επιλέγεται από το UCI repository το Haberman’s Survival dataset (βρίσκεται στο αρχείο haberman.data), το οποίο περιλαμβάνει 306 δείγματα (instances), από 3 χαρακτηριστικά (attributes) το καθένα. Θα χρησιμοποιηθεί για μια απλή διερεύνηση της διαδικασίας εκπαίδευσης και αξιολόγησης των TSK μοντέλων.

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 τέλευταίο 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 κάθισ 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 υπερχύβου.

1.1 Εκπαίδευση TSK Μοντέλων

Σε αυτό το στάδιο εξετάζονται διάφορα μοντέλα TSK όσον αφορά την απόδοσή τους στο σύνολο ελέγχου. Συγκεκριμένα, θα εκπαίδευτούν 4 TSK μοντέλα, στα οποία θα μεταβάλλεται το πλήθος των ασαφών IF-THEN κανόνων, με σκοπό να μελετηθεί η επίδραση της διαμέρισης του χώρου εισόδου – σε συνάρτηση με την πολυπλοκότητα που αυτή επιφέρει, στην απόδοση του ταξινομητή. Η διαμέριση του χώρου εισόδου γίνεται με τη μέθοδο του Subtractive Clustering (SC).

Τα TSK μοντέλα που θα προκύψουν θα διαφέρουν ως προς την παράμετρο που καθορίζει τον αριθμό των κανόνων και αρά το μέγεθος / ακτίνα των clusters. Επειδή όμως θέλουμε ο αριθμός των κανόνων στα δύο μοντέλα που θα προκύψουν να παρουσιάζει σημαντική διαφορά, η παράμετρος αυτή λαμβάνει δύο ωκραίες, μεταξύ τους, τιμές. Στην πρώτη περίπτωση (δύο πρώτα μοντέλα), το μέθοδος της διαμέρισης του χώρου εισόδου θα εκτελεστεί για όλα τα δεδομένα του συνόλου εκπαίδευσης (class independent), ενώ στη δεύτερη (επόμενα δύο), θα εξεταστεί ο διαμερισμός του χώρου εισόδου εφαρμόζοντας clustering στα δεδομένα του συνόλου εκπαίδευσης που ανήκουν στην εκάστοτε κλάση ζεχωριστά (class dependent). Αυτά περιγράφονται και στο Table 1.

Και τα 4 μοντέλα εκπαίδευονται 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).

Πιο συγκεκριμένα, για καθένα από τα δυο πρώτα μοντέλα του πίνακα 1 (δηλαδή τα class independent μοντέλα) περνάμε ως ρύθμιση στην genfis() συνάρτηση, μέσω της FISoptions = genfisOptions() με όρισμα 'SubtractiveClustering', την αντίστοιχη ακτίνα των clusters μέσω της ρύθμισης για τον βαθμό επιχάλυψης FISoptions.ClusterInfluenceRange = 0.1 (ή = 0.9) για να πάρουμε το «αρχικό» μοντέλο που θα περάσουμε ως επιλογή ANFISoptions.InitialFIS = initialFIS (όπου ANFISoptions = anfisOptions) και τέλος να πάρουμε το εκπαίδευμένο μοντέλο μέσω της anfis().

Για τα αλλά δυο μοντέλα (δηλαδή τα class dependent), εφαρμόζουμε clustering στα δεδομένα του συνόλου εκπαίδευσης που ανήκουν στην εκάστοτε κλάση ξεχωριστά μέσω της subclust(), ξεχωριστά δηλαδή για τα δεδομένα εκπαίδευσης που είναι ίσα με 1 και για αυτά που είναι ίσα με 2. Μετά δημιουργούμε βήμα-βήμα το ζητούμενο αρχικό μοντέλο (ορίζοντας αρχικά ένα Sugeno FIS μοντέλο sugfis), δηλαδή ξεχωριστά ορίζουμε τις μεταβλητές εισόδου (μέσω της addInput()), τις συναρτήσεις συμμετοχής εισόδου και εξόδου (μέσω της addMF()), την μεταβλητή εξόδου (μέσω της addOutput()) και την βάση κανόνων (μέσω της addrule()) και τέλος το περνάμε, όπως και τα άλλα, για εκπαίδευση και επικύρωση μέσω της anfis(). Εφόσον, η έξοδός μας αποτελείται από έναν ακέραιο αριθμό, ενδεικτικό της κλάσης στην οποία ανήκει το εκάστοτε δείγμα, έγινε σε κάθε μοντέλο η χειροκίνητη αλλαγή του τύπου συνάρτησης εξόδου από γραμμική σε singleton ορίζοντας την ρύθμιση στην addMF() της εξόδου outputMembershipFunctionType = 'constant' για τον ορισμό των συναρτήσεων συμμετοχής της μεταβλητής εξόδου. Κάθε μοντέλο εκπαίδεύτηκε για 100 iterations/epochs.

Όμως, επειδή η υλοποίηση μας των TSK ασαφών μοντέλων στο MATLAB είναι τέτοια ώστε η έξοδός τους να είναι πραγματική ενώ σε αυτό το κομμάτι της εργασίας, η μεταβλητή – στόχος είναι ακέραιος, λαμβάνοντας τιμές στο σύνολο ακεραίων 1, 2. Για να φέρουμε την έξοδο του μοντέλου ύπου στην ίδια μορφή στρογγυλοποιούμε κάθε στοιχείο στον πλησιέστερο ακέραιο μέσω της round() ενώ στην συνέχεια προβάλουμε τις τιμές εκείνες που βρίσκονται εκτός του συνόλου 1, 2, εντός αυτού κάνοντας την έξοδο y_hat ίση με min(max(1, y_hat), 2).

1.2 Δείκτες απόδοσης

	TSK_model_1	TSK_model_2	TSK_model_3	TSK_model_4
OA	0.59016	0.68852	0.60656	0.70492
PA(1)	0.70455	0.71429	0.72093	0.7451
PA(2)	0.29412	0.4	0.33333	0.5
UA(1)	0.72093	0.93023	0.72093	0.88372
UA(2)	0.27778	0.11111	0.33333	0.27778
K_hat	-0.0013132	0.05233	0.054264	0.18546

Table 2: Πίνακας δεικτών απόδοσης των TSK μοντέλων

Αρχικά, στο Table 2 παρουσιάζονται σε μορφή πίνακα και οι τιμές των ζητουμένων δεικτών απόδοσης OA(Overall Accuracy), PA(Producer's Accuracy), UA(User's Accuracy) και \bar{K} καθεμιά από τις τέσσερις περιπτώσεις εκπαίδευσης

Οι τύποι/σχέσεις για τους δείκτες απόδοσης δίνονται στην εκφώνηση της εργασίας και τις υλοποιούμε στα MATLAB αρχεία συναρτήσεων OA.m, PA.m, UA.m, K_hat.m.

1.3 Διαγράμματα TSK Μοντέλων

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 ο πίνακας των σφαλμάτων ταξινόμησης (Error Matrix) (με χρήση της συνάρτησης confusionmat(), η οποία δίνει αποτέλεσμα συμφώνα με την εκφώνηση δίνοντας ως άρισμα την έξοδο που δίνει το εκπαίδευμένο μοντέλο και το testing σύνολο του dataset), τόσο σε μορφή πίνακα όσο και σε μορφή εικόνας (με απλή χρήση της MATLAB συνάρτησης confusionchart()).

1.4 TSK_model_1

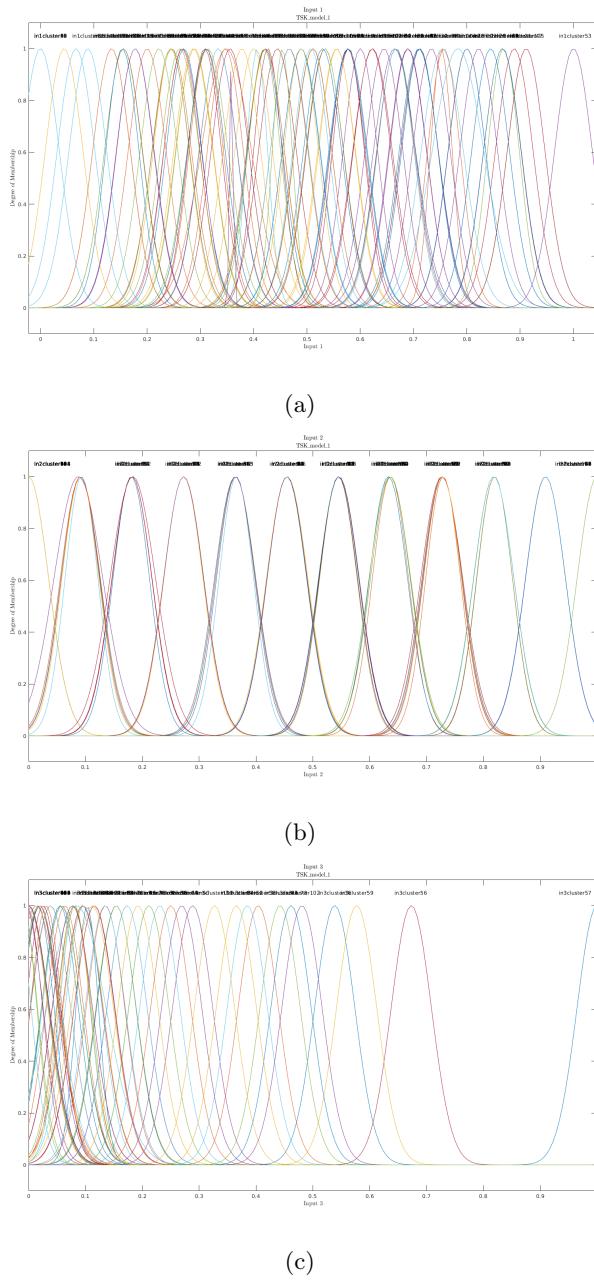


Figure 1: Τελική μορφές των ασαφών συνόλων που προέκυψαν μέσω της διαδικασίας εκπαίδευσης του TSK_model_1.

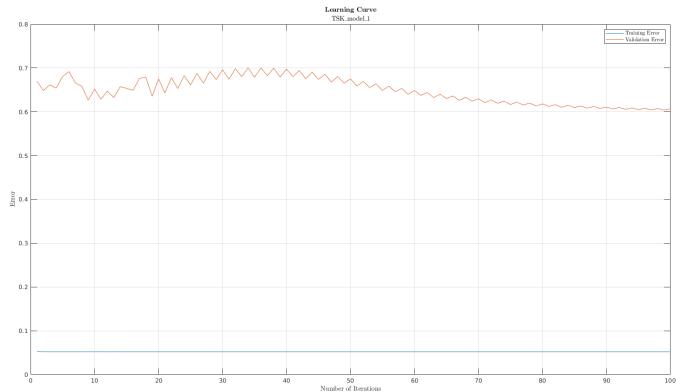


Figure 2: Καμπύλες μάθησης του TSK_model_1.

	Actual:1	Actual:2
Predicted:1	31	12
Predicted:2	13	5

Table 3: Πίνακας σφαλμάτων ταξινόμησης του TSK_model_1.

1.5 TSK_model_2

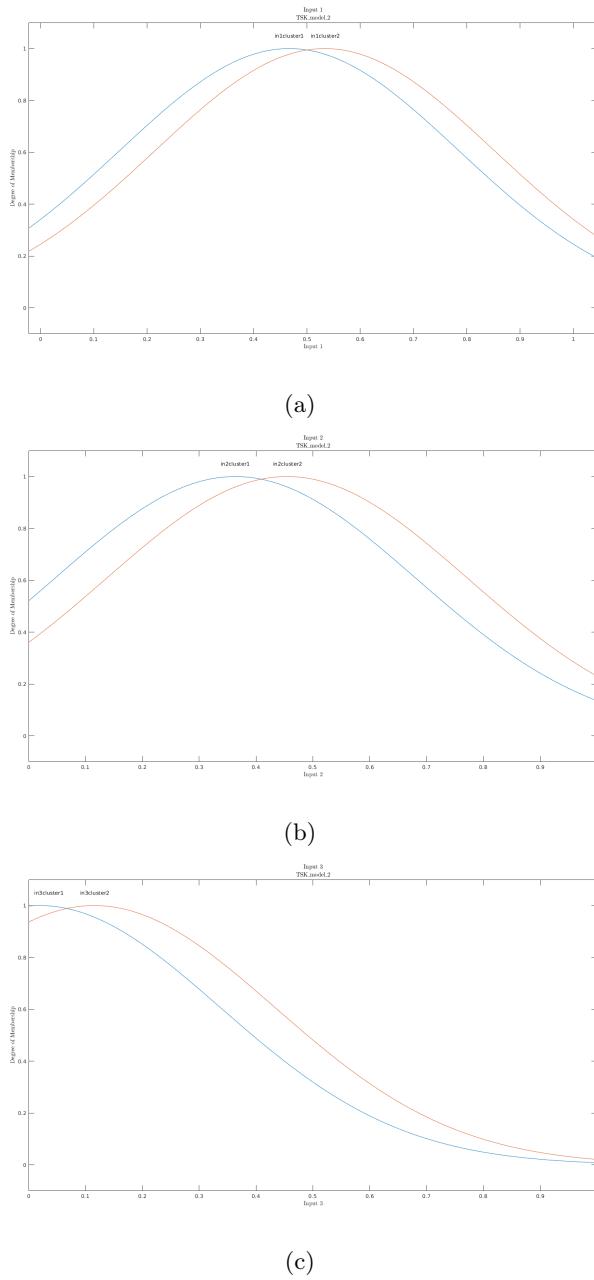


Figure 3: Τελική μορφές των ασαφών συνόλων που προέκυψαν μέσω της δι-αδικασίας εκπαίδευσης του TSK_model_2.

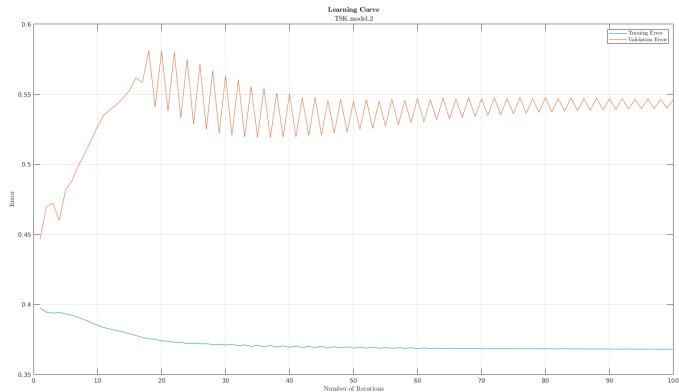


Figure 4: Καμπύλες μάθησης του TSK_model_2.

	Actual:1	Actual:2
Predicted:1	40	3
Predicted:2	16	2

Table 4: Πίνακας σφαλμάτων ταξινόμησης του TSK_model_2.

1.6 TSK_model_3

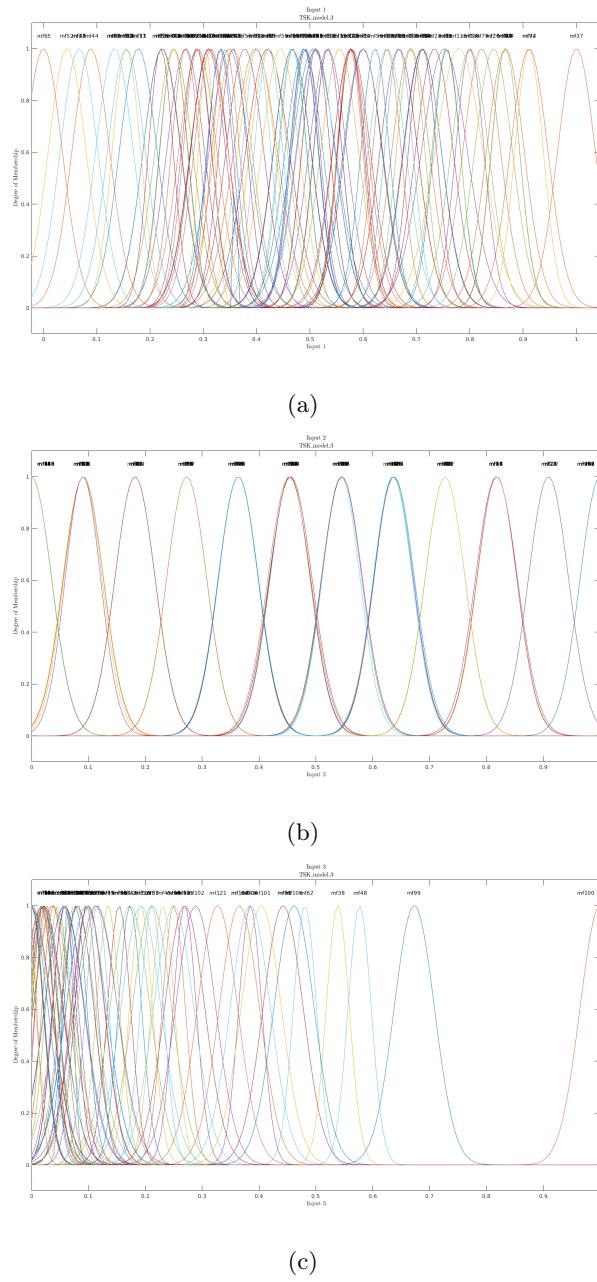


Figure 5: Τελική μορφές των ασαφών συνόλων που προέκυψαν μέσω της δι-
αδικασίας εκπαίδευσης του TSK_model_3.

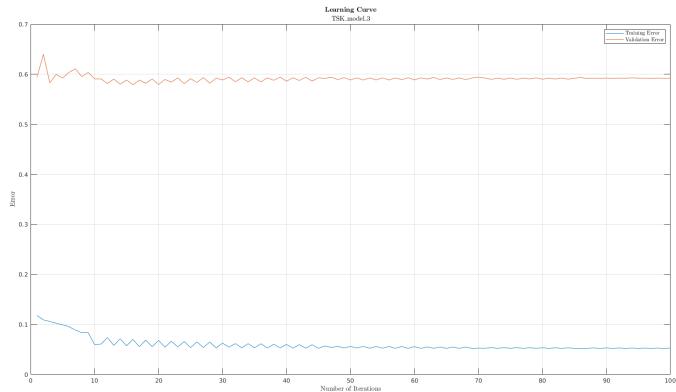


Figure 6: Καμπύλες μάθησης του TSK_model_3.

	Actual:1	Actual:2
Predicted:1	31	12
Predicted:2	12	6

Table 5: Πίνακας σφαλμάτων ταξινόμησης του TSK_model_3.

1.7 TSK_model_4

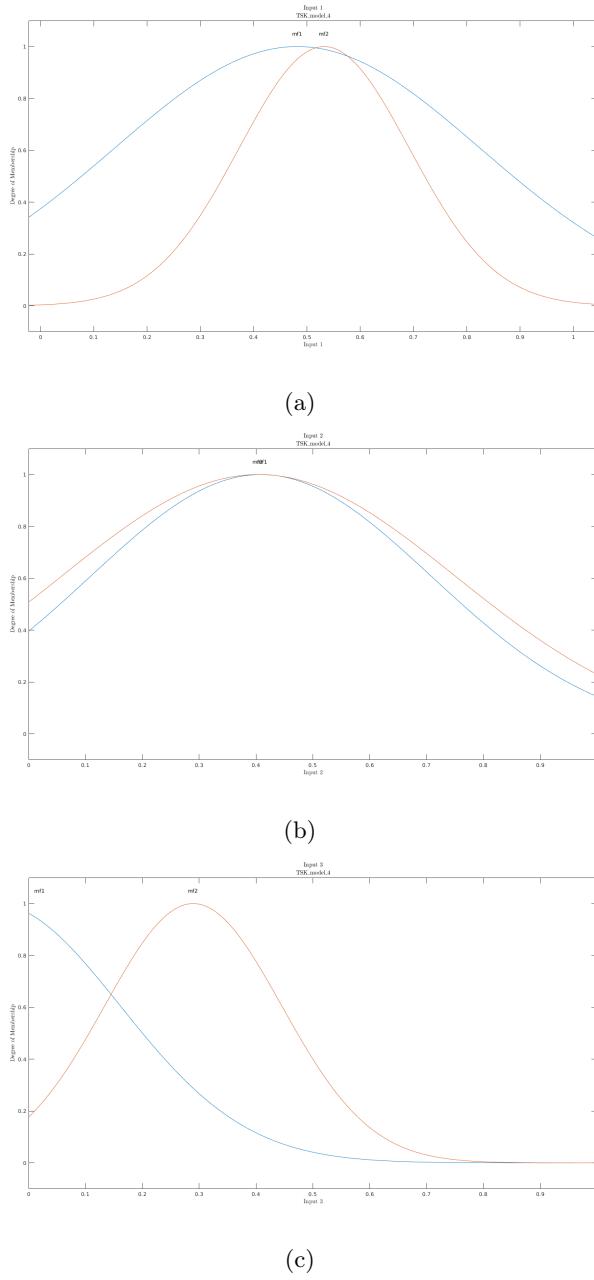


Figure 7: Τελική μορφές των ασαφών συνόλων που προέκυψαν μέσω της διαδικασίας εκπαίδευσης του TSK_model_4.

Παρακάτω φαίνε

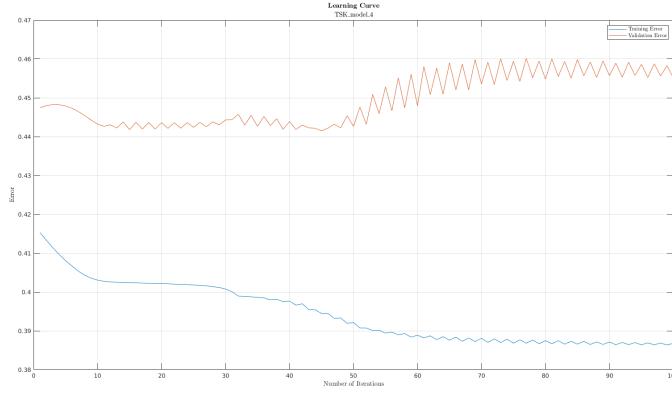


Figure 8: Καμπύλες μάθησης του TSK_model_4.

	Actual:1	Actual:2
Predicted:1	38	5
Predicted:2	13	5

Table 6: Πίνακας σφαλμάτων ταξινόμησης του TSK_model_4.

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 classification_simpleDataset.m. Έχουμε αποθηκεύσει την έξοδο της κονσόλας του MATLAB της εκπαίδευσης στο αρχείο classification_simpleDataset_logs.txt ενώ τις μεταβλητές που δημιουργήθηκαν (λόγω μεγάλου απαιτούμενου χρόνου εκτέλεσης) στην συνεδρία της εκπαίδευσης στο MATLAB αρχείο μεταβλητών classification_simpleDataset_Variables.mat. Αξίζει να σημειωθεί πως η επανεκτέλεση του script θα δώσει διαφορετικά αριθμητικά αποτελέσματα δεικτών απόδοσης και διαφορετικά διαγράμματα λόγω της τυχαιότητας κατά την διαχώριση του dataset στην αρχή του script. Όμως, αυτά θα έχουν τα ίδια χαρακτηριστικά με αυτά που δίνουμε εδώ, δηλαδή θα υπακούν στα συγκριτικά συμπεράσματα που θα δώσουμε στην συνέχεια.

1.8 Σχολιασμός αποτελεσμάτων

Παρατηρώντας τα παραπάνω διαγράμματα των τεσσάρων TSK μοντέλων βλέπουμε για το TSK_model_2 πως έχουμε το φαινόμενο της υπερεκπαίδευσης (overfitting) καθώς βλέπουμε πως ενώ το training error όλο και μειώνεται, το validation error έχει τάση αύξησης. Για το TSK_model_1 και TSK_model_3 δεν μπορούμε να αποφανθούμε σχετικά με την ύπαρξη του φαινομένου της υπερεκπαίδευσης καθώς βλέπουμε πως από κάποια iterations και μετά πως τα σφάλματα

	TSK_1	TSK_2	TSK_3	TSK_4
Number of Rules	117	2	126	2

Table 7: Πίνακας αριθμών κανόνων των TSK μοντέλων.

παραμένουν σταθερά (για το TSK_model_1 ίσως να μπορούμε να πούμε πως δεν έχουμε overfitting καθώς μέχρι τα περίπου 50 iterations μειώνονται το training error αλλά και το validation error). Για το TSK_model_4 βλέπουμε πως ενώ μέχρι περίπου τα 50 iterations δεν έχουμε overfitting και μειώνονται με ικανοποιητικό βαθμό το training error αλλά και το validation error, από εκεί και μετά ξεκινάει μια ταλάντωση (με πλάτος το οποίο φαίνεται να αρχίζει να αυξάνεται) του validation error. Επομένως, μπορούμε να πούμε πως ίσως να έχουμε σημάδια εμφάνισης του overfitting.

Όσον αφορά την ακρίβεια ταξινόμησης των τεσσάρων μοντέλων, βλέπουμε πως όλα κάνουν μια μέτρια δουλειά καθώς όλα έχουν OA μεγαλύτερο από 0.5. Οπότε έχουμε γενικά πιθανότητα λίγο μεγαλύτερη από 1 στις 2 τιμές κλάσεων να ταξινομήσουν σωστά. Μάλιστα, όπως φαίνεται και από τα Error Matrixes και στα τέσσερα μοντέλα βλέπουμε πως η σωστότερη ταξινόμηση γίνεται για αυτές τις τιμές που ανήκουν στην κλάση 1 (αυτό φαίνεται από τις τιμές των UA(1) που είναι κοντά στην μονάδα και για τα τέσσερα μοντέλα). Για τις τιμές που ανήκουν στην κλάση 2 δεν βλέπουμε να έχουμε αντίστοιχα καλά αποτελέσματα.

Για να αποφανθούμε, όμως, ως προς το πιο μοντέλο είναι καταλληλότερο και δίνει καλύτερα αποτελέσματα για το dataset αυτό, θα κάνουμε ένα συγχριτικό σχολιασμό όλων των μοντέλων με βάση την μετρική OA.

Αρχικά, μέσω του ίδιου παραπάνω MATLAB script, πάρνουμε και τον αριθμό των κανόνων κάθε TSK classification μοντέλου, Table 7.

Συγκρίνοντας, τώρα, ανά δύο τα μοντέλα που είναι όμοια ως προς το μέγεθος των clusters, δηλαδή συγκρίνοντας μεταξύ τους το TSK_model_1 (Class Independent) με το TSK_model_3 (Class Dependent) (μέγεθος ίσο με 0.1) και το TSK_model_2 (Class Independent) με το TSK_model_4 (Class η Dependent) (μέγεθος ίσο με 0.9). Στην 1η περίπτωση βλέπουμε το TSK_model_1 έχει OA = 0.59016 ενώ το TSK_model_3 έχει επίσης OA = 0.60656 ενώ όπως βλέπουμε και από το Table 7 αυτά τα δύο έχουν περίπου και τον ίδιο αριθμό κανόνων. Στην 2η περίπτωση το TSK_model_2 έχει OA = 0.68852 και το TSK_model_4 έχει OA = 0.70492 ενώ όπως βλέπουμε και από το Table 7 αυτά τα δύο έχουν και τον ίδιο αριθμό κανόνων.

Άρα, βλέπουμε γενικά τα class dependent μοντέλα δίνουν καλυτέρα αποτελέσματα και έχουν καλύτερη απόδοση με μια μικρή αύξηση (αν υπάρξει) του αριθμού των κανόνων, σε σχέση με τα class independent.

Συγκρίνοντας, τώρα, ανά δύο τα μοντέλα που είναι όμοια ως προς τη μέθοδο SC, δηλαδή αν είναι class dependent ή όχι, δηλαδή συγκρίνοντας μεταξύ τους το TSK_model_1 (μέγεθος ίσο με 0.1) με το TSK_model_2 (μέγεθος ίσο με 0.9) (Class Independent) και το TSK_model_3 (μέγεθος ίσο με 0.1) με το TSK_model_4 (μέγεθος ίσο με 0.9) (Class Dependent). Στην 1η περίπτωση βλέπουμε πως το TSK_model_1 έχει OA = 0.59016 ενώ το TSK_model_2

έχει OA = 0.68852 ενώ όπως φαίνεται και από το Table 7 το TSK_model_1 έχει 117 κανόνες στο μοντέλο του ενώ το TSK_model_2 έχει μόλις μόνο 2 κανόνες στο μοντέλο του. Στην 2η περίπτωση βλέπουμε πως το TSK_model_3 έχει OA = 0.60656 ενώ το TSK_model_4 έχει OA = 0.70492 ενώ όπως φαίνεται και από το Table 7 το TSK_model_3 έχει 126 κανόνες στο μοντέλο του ενώ το TSK_model_4 έχει μόλις μόνο 2 κανόνες στο μοντέλο του. Επομένως, βλέπουμε πως η μεγάλη/ραγδαία αύξηση της ακτίνας των clusters τόσο στα class independent όσο και στα class dependent TSK classification μοντέλα δίνει γενικά καλυτέρα αποτελέσματα ακρίβειας αυξάνοντας ελάχιστα την μετρική OA ενώ χρησιμοποιεί και πολύ λιγότερους κανόνες (2 < 127). Επομένως, μπορούμε ευκολά να αντιληφθούμε πως αυτό γίνεται χρήσιμο σε μοντέλα υψηλής διαστασιμότητας αποφεύγοντας έτσι φαινόμενα «έκρηξης» αριθμού κανόνων, το λεγόμενο rule explosion.

Η αύξηση όμως του αριθμού των κανόνων σημαίνει και αύξηση του πλήθους των συναρτήσεων συμμετοχής εισόδους. Όπως φαίνεται και από τα σχήματα παραπάνω των τελικών μορφών των ασαφών συνόλων του κάθε TSK μοντέλου ταξινόμησης, παρατηρούμε πως όσο μικρότερη ακτίνα clusters έχουμε, τόσο περισσότερες συναρτήσεις συμμετοχής έχουμε και συνεπώς είναι και λογικό να υπάρχει τόσο μεγαλύτερη επικάλυψη υπάρχει μεταξύ αυτών. Αυτό φαίνεται πιο έντονα στα διαγράμματα των μοντέλων TSK_model_1 και TSK_model_3 που είπαμε ότι γενικά έχουν μικρότερο OA από τα υπόλοιπα. Επομένως, η επικάλυψη των ασαφών συνόλων μειώνει την τιμή της μετρικής OA του μοντέλου ενώ φαίνεται και να κάνει το μοντέλο να αναγνωρίζει καλυτέρα συγκεκριμένες τιμές κλάσεων (εδώ την 1) και να μην τα καταφέρνει τόσο καλά στην γενική περίπτωση (όπως π.χ. και με την 2) (μικρή δυνατότητα γενίκευσης (Generalization)).

Συνεπώς, το TSK_model_4 δίνει δικαιολογημένα τα καλύτερα αποτελέσματα αφού έχει την μεγαλύτερη ακτίνα clusters αλλά και είναι Class Dependent κάνοντας χρήση μόλις 2 κανόνων αποφεύγοντας έτσι και τα αρνητικά αποτελέσματα που προκαλεί η μεγάλη επικάλυψη των ασαφών συνόλων.

Για τη βελτίωση της σχεδίασης του τμήματος υπόθεσης, μια μέθοδος θα ήταν η διαγραφή όλων ή κάποιων ασαφών συνόλων κάθε cluster που έχουν μεγάλο βαθμό επικάλυψης με στόχο να κάνουμε το μοντέλο μας ίσως λίγο πιο απλό με στόχο να δίνει αποτελέσματα καλυτέρα στην γενική περίπτωση και όχι μόνο για την κλάση με τιμή 1. Τότε, θα αυξανόταν το OA (καθώς θα ταξινομούσαμε καλυτέρα στην κλάση με τιμή 2 ρίχνοντας λίγο τις φορές που ταξινομούμε σωστά μια τιμή που ανήκει στην κλάση 1 αλλά γενικά θα παίρναμε καλύτερο αποτέλεσμα) και ίσως να μειώναμε και την εμφάνιση των φαινομένων overfitting, που είδαμε ότι εμφανίζονται συχνά παραπάνω.

2 Dataset υψηλής διαστασιμότητας

Στη δεύτερη φάση της εργασίας θα ακολουθηθεί μια πιο συστηματική προσέγγιση στο πρόβλημα της χρήσης ασαφών νευρωνικών μοντέλων σε προβλήματα ταξινόμησης. 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 Epileptic Seizure Recognition dataset από το UCI repository. Το συγκεκριμένο dataset, περιλαμβάνει 11500 δείγματα, καθένα από τα οποία περιγράφεται από 179 μεταβλητές/χαρακτηριστικά (βρίσκεται στο αρχείο epileptic_seizure_data.csv και λαμβάνουμε το .data κομμάτι στο MATLAB).

Ένα προφανές πρόβλημα που ανακύπτει από την επιλογή αυτή, είναι η λεγόμενη «έκρηξη» του πλήθους των IF-THEN κανόνων (rule explosion). Ο μεγάλος αριθμός μεταβλητών καθιστά αναγκαία τη χρήση μεθόδων μείωσης της διαστασιμότητας καθώς και του αριθμού των IF-THEN κανόνων. Ο στόχος αυτός θα επιτευχθεί μέσω της επιλογής χαρακτηριστικών και της χρήσης ασφαφούς ομαδοποίησης. Οι δύο αυτές μέθοδοι ήμως, παρά την ελάττωση της πολυπλοκότητας που επιφέρουν, εισάγουν στο πρόβλημα δύο ελεύθερες παραμέτρους, 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 αποψυγή 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. 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 κανονικοποίηση στην μονάδα του υπερκύβου ενώ κάνουμε και την δική μας κατάλληλη προεπεξεργασία του testing και validation μέρος του dataset.

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 που θα εξεταστούν μπορούν να επιλεχθούν ελεύθερα, επιλέξαμε τον έλεγχο για τις τιμές:

$$numOfFeatures = [4 \quad 6 \quad 8 \quad 10]$$

$$clusterRadius = [0.25 \quad 0.5 \quad 0.75 \quad 1]$$

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 (αφάντηση 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, όταν γίνεται 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 άλλο 20% για επικύρωση.

Με τα δεδομένα αυτά, η διαδικασία της μεθόδου διαμέρισης, Subtractive Clustering, εφαρμόζεται ξεχωριστά για κάθε κλάση (με την ίδια τιμή στην παράμετρο που καθορίζει την ακτίνα). Μετά τον αρχικό διαχωρισμό των δεδομένων, τα δεδομένα εκπαίδευσης διαμερίζονται επιπλέον ανάλογα με την κλάση στην οποία ανήκουν και η συνολική βάση κανόνων προκύπτει ως η ένωση των επιμέρους κανόνων που προκύπτουν από τις class-specific ομαδοποιήσεις. Δηλαδή, αρχικά, εφαρμόζουμε clustering στα δεδομένα του συνόλου εκπαίδευσης που ανήκουν στην εκάστοτε κλάση ξεχωριστά μέσω της subclust(), ξεχωριστά δηλαδή για τα δεδομένα εκπαίδευσης που είναι ίσα με 1 και για αυτά που είναι ίσα με 2, ίσα με 3, ίσα με 4 και ίσα με 5, αφού η μεταβλητή – στόχος είναι ακέραιος, λαμβάνοντας τιμές στο σύνολο ακεραίων {1, 2, 3, 4, 5}.

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

Συνοπτικά, αφάντηση της εργασίας, έγινε η χειροκίνητη αλλαγή του δηλαδή για κάθε τριάδα (fold, numOffeatures, clusterRadius), εκπαίδευση 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 χρήση της υλοποίησης που κάναμε στο πρώτο κομμάτι της εργασίας για το class dependent μοντέλο. Τέλος, για κάθε δυάδα (numOffeatures, clusterRadius) βρίσκουμε την mean() τιμή των MSE των 5 folds και το αποθηκεύουμε εκ νέου.

Όπως και στο πρώτο μέρος της εργασίας, έγινε η χειροκίνητη αλλαγή του τύπου συνάρτησης εξόδου από γραμμική σε singleton, ορίζοντας την ρύθμιση

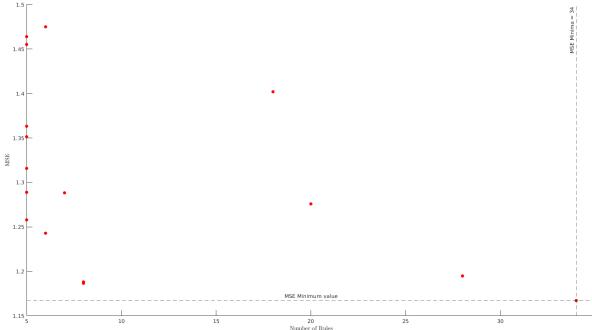


Figure 9: MSE συναρτήσει του πλήθους κανόνων.

στην addMF() της εξόδου outputMembershipFunctionType = 'constant' για τον ορισμό των συναρτήσεων συμμετοχής της μεταβλητής εξόδου.

Όμως, επειδή η υλοποίηση μας των TSK ασαφών μοντέλων στο MATLAB είναι τέτοια ώστε η έξοδός τους να είναι πραγματική ενώ σε αυτό το κομμάτι της εργασίας, η μεταβλητή – στόχος είναι ακέραιος, λαμβάνοντας τιμές στο σύνολο ακεραίων {1, 2, 3, 4, 5}. Για να φέρουμε την έξοδο του μοντέλου ύ στην ίδια μορφή στρογγυλοποιούμε κάθε στοιχείο στον πλησιέστερο ακέραιο μέσω της round() ενώ στην συνέχεια προβάλουμε τις τιμές εκείνες που βρίσκονται εκτός του συνόλου 1, 2, 3, 4, 5, εντός αυτού κάνοντας την έξοδο y_{hat} ίση με $\min(\max(1, y_{\text{hat}}), 5)$.

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

Στο Figure 9 φαίνεται το ζητούμενο διάγραμμα που απεικονίζει το μέσο σφάλμα σε σχέση με τον αριθμό των κανόνων. Παρατηρούμε ακόμη, 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 καλύτερα 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 classification.

Επίσης, στο Figure 10 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 3-D (surface plot) μορφή για μια πιο πλήρη απεικόνιση. Παρατηρούμε ακόμη, πως η επιφάνεια του MSE είναι γενικά φυλίνουσα 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 και μπορεί να ανταπεξέλθει καλύτερα στις ταξινομήσεις του αργότερα ενώ όσο μειώνουμε την ακτίνα των clusters οι συναρτήσεις συμμετοχής γίνονται πιο «στενές» και άρα στα σημεία δεδομένων εκχωρούνται μεγαλύτερα degrees of

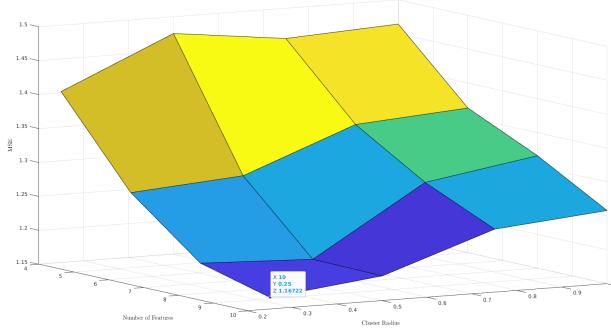


Figure 10: MSE συναρτήσει τιμών όλων των παραμέτρων του grid search.

membership σε λιγότερα ασαφή σύνολα και άρα κατά συνέπεια, κάθε κανόνας έχει μια μικρότερη περιοχή επιφροής στο χώρο εισόδου.

To MATLAB script που υλοποιεί το grid search και την δημιουργία των διαγραμμάτων βρίσκονται στο αρχείο `classification_highDimDataset_gridSearch.m`. Μάλιστα, έχουμε αποθηκεύσει την έξοδο της κονσόλας του MATLAB στο αρχείο `classification_highDimDataset_gridSearch_logs.txt` ενώ όλες οι μεταβλητές που δημιουργήθηκαν (λόγω μεγάλου απαιτούμενου χρόνου εκτέλεσης) στο MATLAB αρχείο μεταβλητών `classification_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 διαχρίνουμε και από το σημείο ελαχίστου στο σχήμα 14, οι τιμές αυτές είναι για $numOfFeatures = 10$ και $clusterRadius = 0.25$ (υπάρχει ελάχιστο αφού δείξαμε ότι φύλνουσα ως προς τις παραμέτρους). 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, στο Figure 11 τα διαγράμματα εκμάθησης όπου να απεικονίζεται το σφάλμα συναρτήσει του αριθμού επαναλήψεων.

Επίσης, στο Figure 12 δίνονται τα ζητούμενα διαγράμματα όπου να αποτυπώνονται οι προβλέψεις του τελικού μοντέλου καθώς και οι πραγματικές τιμές (δίνονται δίπλα δίπλα για σύγκριση).

Στο Figure 13 δίνουμε και τον ζητούμενο πίνακας σφαλμάτων ταξινόμησης (Error Matrix), στο Figure 8 μορφή πίνακα αλλά και σε μορφή εικόνας.

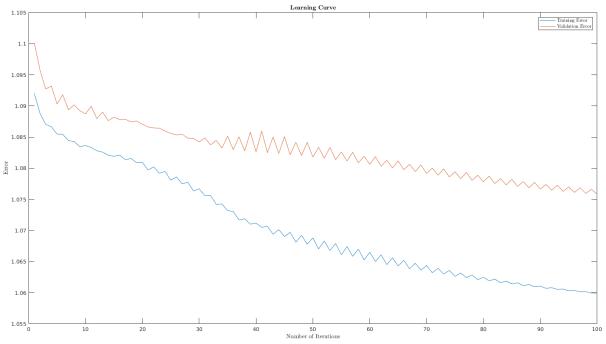


Figure 11: Καμπύλες μάθησης του μοντέλου βέλτιστων τιμών.

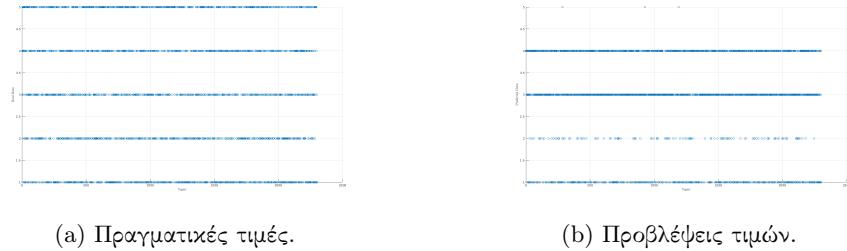


Figure 12: Πραγματικές τιμές και τιμές πρόβλεψης μοντέλου.

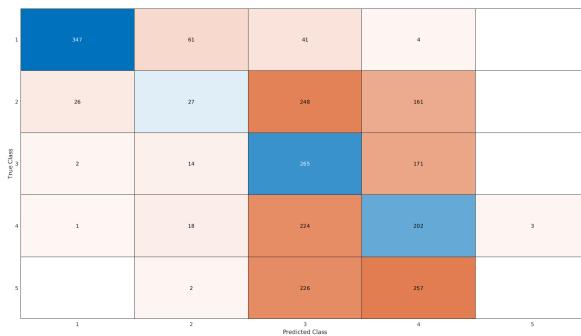


Figure 13: Πίνακας σφαλμάτων ταξινόμησης του μοντέλου βέλτιστων τιμών σε μορφή εικόνας.

	Actual: 1	Actual: 2	Actual: 3	Actual: 4	Actual: 5
Predicted: 1	347	26	2	1	0
Predicted: 2	61	27	14	18	2
Predicted: 3	41	248	265	224	226
Predicted: 4	4	161	171	202	257
Predicted: 5	0	0	0	3	0

Table 8: Πίνακας σφαλμάτων ταξινόμησης του μοντέλου βέλτιστων τιμών.

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

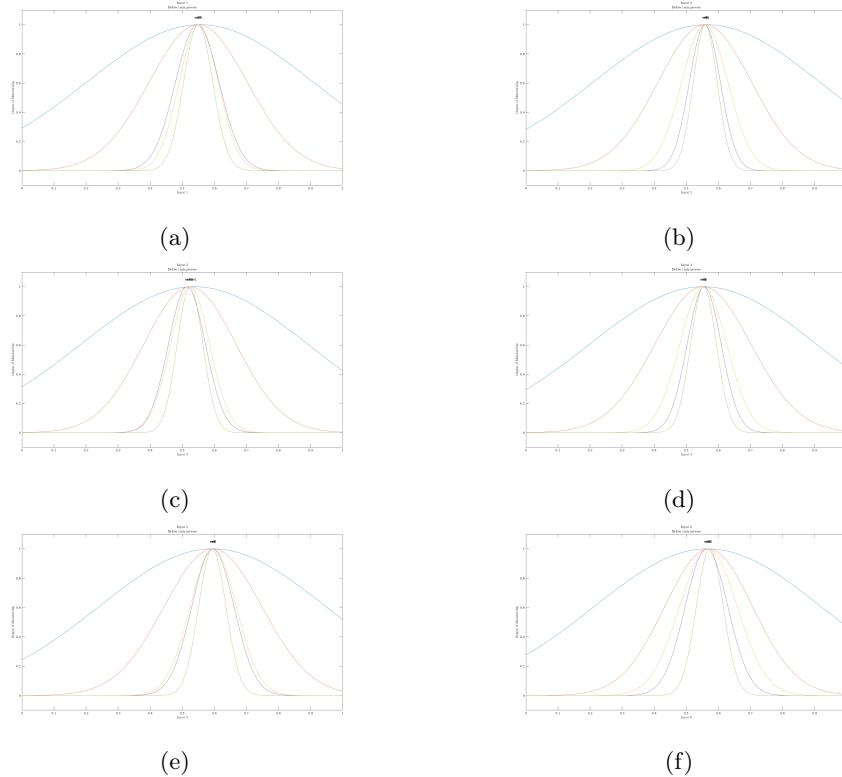


Figure 14: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου μοντέλου βέλτιστων τιμών.

Figure 15: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου μοντέλου βέλτιστων τιμών.

Figure 16: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου μοντέλου βέλτιστων τιμών.

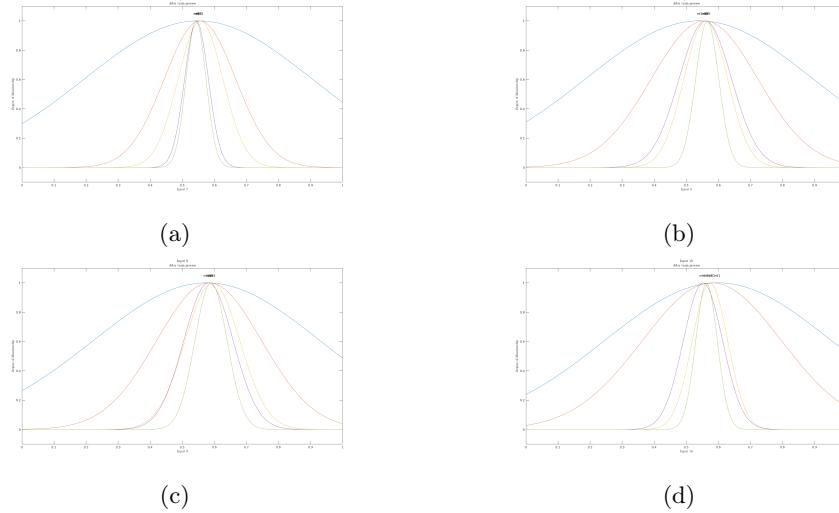


Figure 17: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου μοντέλου βέλτιστων τιμών.

2.3 Δείκτες απόδοσης

Τέλος, στο Table 9 παρουσιάζονται σε μορφή πίνακα και οι τιμές των ζητουμένων δεικτών απόδοσης OA, για το τελικό TSK μοντέλο με τις βέλτιστες τιμές των παραμέτρων, κάνοντας χρήση των ιδίων PA, UA, K MATLAB συναρτήσεων που χρησιμοποιήσαμε στο πρώτο μέρος της εργασίας

Optimal values model	
OA	0.36565
PA(1)	0.92287
PA(2)	0.22131
PA(3)	0.26394
PA(4)	0.25409
PA(5)	0
Optimal values model	
UA(1)	0.766
UA(2)	0.058442
UA(3)	0.58628
UA(4)	0.45089
UA(5)	0
Optimal values model	
K_hat	0.21077

Table 9: Πίνακας δεικτών απόδοσης του TSK μοντέλου με βέλτιστες τιμές παραμέτρων.

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 `classification_highDimDataset_optimalVals.m` (γίνεται χρήση των μεταβλητών `classification_highDimDataset_gridSearch_Variables.mat`). Έχουμε αποθηκεύσει την έξοδο της κονσόλας του MATLAB της συνεδρίας στο αρχείο `classification_highDimDataset_optimalVals_logs.txt` ενώ για τις μεταβλητές που δημιουργήθηκαν, λόγω μεγάλου απαιτούμενου χρόνου εκτέλεσης, στην συνεδρία της εκπαίδευσης του βέλτιστου μοντέλου στο MATLAB αρχείο μεταβλητών `classification_highDimDataset_optimalVals_Variables.mat`.

2.4 Σχολιασμός αποτελεσμάτων

Παρατηρούμε πως πάρα την χρήση μόνο των 10 χαρακτηριστικών (από τα συνολικά 179 του dataset) για την εκπαίδευση του ασαφούς νευρωνικού δικτύου, δημιουργήσαμε ένα classification model το οποίο δίνει αποτελέσματα που χρίζουν σχολιασμό για το ποσό καλά είναι ή όχι, καθώς με μια πρώτη ματιά στα αποτελέσματα παρατηρούμε κάποια προβλήματα.

Αρχικά, βλέπουμε από το Figure 11 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 και αρχίζει το δεύτερο να συγκλίνει στο πρώτο.

Ρίχνοντας μια ματιά στον πίνακα 9 των μετρικών αξιολόγησης του μοντέλου εύκολα παρατηρούμε προβλήματα να δημιουργούνται με το μοντέλο ταξινόμησης. Γενικά βλέπουμε ο classifier ότι ταξινομεί γενικά σωστά περίπου 3 στις 10 τιμές κλάσεων σωστά ενώ στις υπόλοιπες αστοχεί (αυτό το βλέπουμε από την μετρική OA). Επίσης από την μετρική PA βλέπουμε πως το μοντέλο ταξινομεί σχεδόν πάντα σωστά ένα τυχαίο δείγμα στην τιμή κλάσης 1 που όντως ανήκει σε αυτήν ($PA(1) = 0.92287$, δηλαδή κοντά στην μονάδα). Αρκετά καλά αποτελέσματα ταξινόμησης βλέπουμε και για τις τιμές κλάσεων 3 και 4 (Βλέπουμε τις μετρικές UA(3) και UA(4) είναι περίπου 0.5, δηλαδή σχεδόν τις μισές φορές έχει γίνει ορθή ταξινόμηση ενός τυχαίου δείγματος που ανήκει στην εκάστοτε σωστή του κλάση). Όμως, δεν φαίνεται να έχουμε παρόμοια καλά αποτελέσματα για τις υπόλοιπες τιμές κλάσεων, όπως επιβεβαίνεται άλλωστε και από το Figure 12, όπου για τις τιμές κλάσεων 2 και 5 βλέπουμε να υπάρχει μεγάλη αντίθεση στα διαγράμματα πραγματικών τιμών και προβλεπόμενων. Μάλιστα, στην τιμή κλάσης 5 βλέπουμε ότι το μοντέλο αποτυγχάνει τελείως ($PA(5) = 0$, $UA(5) = 0$). Συμπερασματικά, οι τιμές των μετρικών είναι κακές.

Επίσης, παρατηρούμε στα Figure 14 και Figure 15 πως υπάρχει μεγάλης τάξης επικάλυψη προβολών των ασαφών συνόλων στο χώρο των μεταβλητών εισόδου. Έτσι, το μοντέλο ταξινόμησης που φτιάξαμε να έχει μικρή διακριτική ικανότητα και έχοντας και μικρό αριθμό κανόνων καταλαβαίνουμε πως χρειαζόμαστε σίγουρα πολλά παραπάνω από τα 10 σημαντικότερα χαρακτηριστικά του dataset για την αποτελεσματική εκπαίδευση ενός χρήσιμου TSK μοντέλου ταξινόμησης, για το dataset υψηλής διαστασιμότητας που μελετάμε.

Παρόλο αυτά, παρατηρούμε από το Figure 9 πως το βέλτιστο μοντέλο ασαφούς

συμπερασμού περιέχει μόνο 35 IF-THEN κανόνες. Οι λίγοι κανόνες φάίνεται πως σίγουρα αποτελούν ένας από τους λογούς που πήραμε κακά αποτελέσματα ταξινόμησης (βλέποντας τον πίνακα σφαλμάτων ταξινόμησης στο Figure 13). 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^{25} (ή 3^{25}) κανόνες. Αυτό προφανώς θα εκτόξευε εκθετικά την απαιτούμενη υπολογιστική δύναμη ή και χρόνο για την εκτέλεση ή και training του classification model, αφού η είσοδος στο μοντέλο θα έπρεπε να περάσει από όλους αυτούς τους κανόνες 35 (2 » 35). Αξίζει, όμως, να σημειωθεί πως σε εκείνη την περίπτωση θα είχαμε καλυτέρα αποτελέσματα ως προς το σφάλμα ενώ θα είχαμε και λιγότερες συναρτήσεις συμμετοχής για τις μεταβλητές εισόδου. Οπότε δημιουργούμε έτσι ένα tradeoff σφαλμάτων – χρόνων εκτέλεσης/εκπαίδευσης για τις μεθόδους Grid Partitioning και SC αντίστοιχα. Άφα 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.