ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΣΑΦΗ ΣΥΣΤΗΜΑΤΑ

8ο ΕΞΑΜΗΝΟ

ΕΡΓΑΣΙΑ #4

ΕΙΣΗΓΗΤΗΣ: ΘΕΟΧΑΡΗΣ Ι.

Όνομα : ΘΑΝΑΣΗΣ ΧΑΡΙΣΟΥΔΗΣ

A.E.M. : 9026

ΘΕΣΣΑΛΟΝΙΚΗ, 28 Φεβρουαρίου 2020

Πίνακας Περιεχομένων

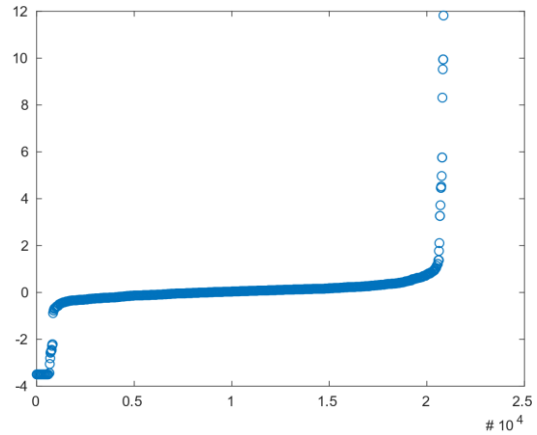
Πίνακας Εικόνων

Πίνακας Πινάκων

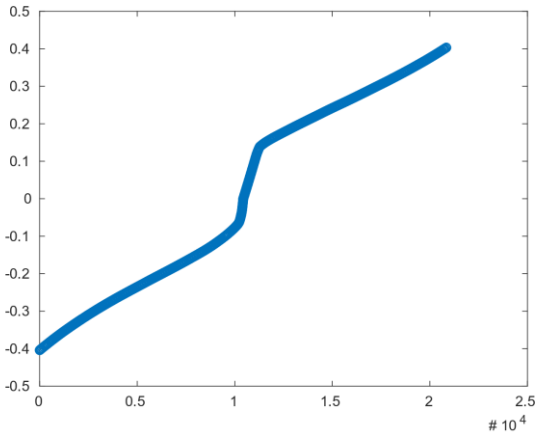
1. Εφαρμογή σε απλό dataset

1.1. Φόρτωση & Προ-επεξεργασία dataset

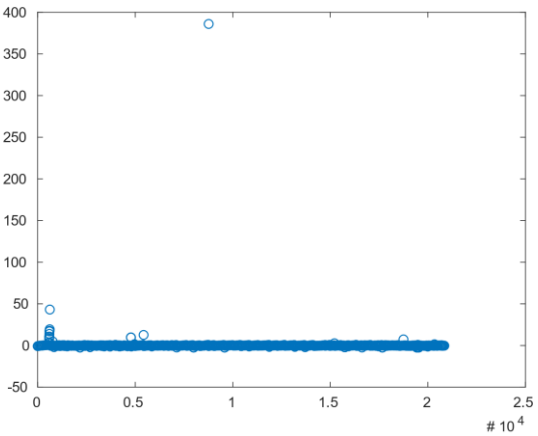
Το dataset του ερωτήματος, *avila*, αποδείχθηκε αρκετά “δύσκολο” dataset από την άποψη ότι έχει αρκετά σημεία τα οποία θα μπορούσαν να θεωρηθούν outliers. Για ισχυροποίηση του παραπάνω, παραθέτονται τα plots κάποιων features του dataset:



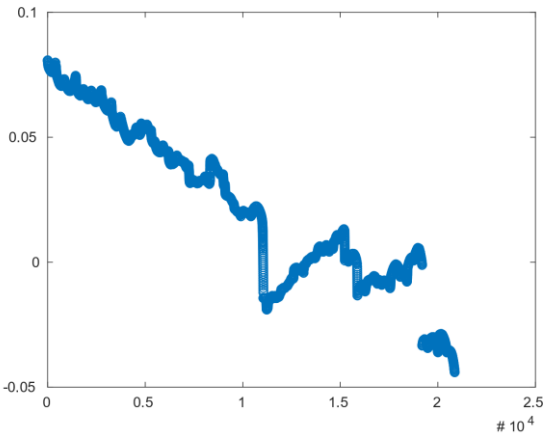
Εικόνα 1: Feature 1 (αρχικό)



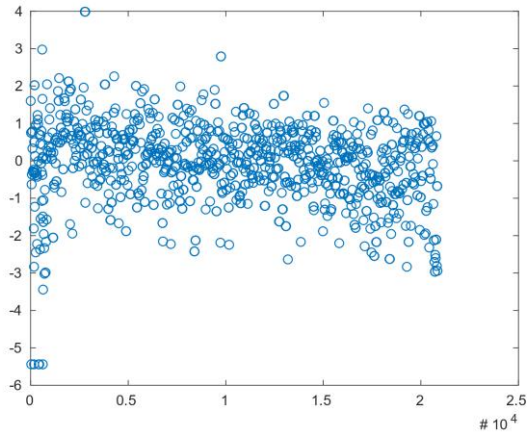
Εικόνα 2: Feature 1 (smoothed, SmoothingFactor=0.5)



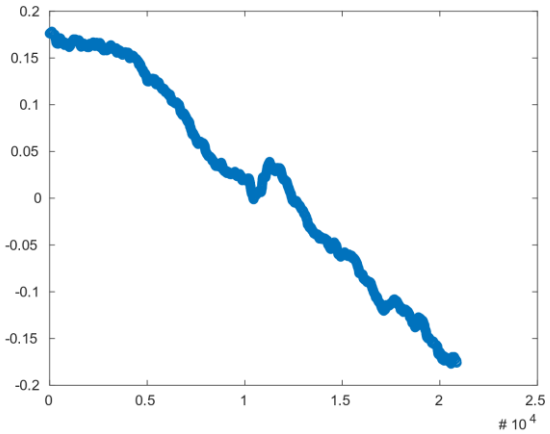
Εικόνα 3: Feature 2 (αρχικό)



Εικόνα 4: Feature 2 (smoothed, SmoothingFactor=0.75)



Εικόνα 5: Feature 4 (αρχικό)



Εικόνα 6: Feature 2 (smoothed, SmoothingFactor=0.5)

Στη δεξιά στήλη του παραπάνω πίνακα φαίνεται το αποτέλεσμα μίας από τις πιθανές μεθόδους προ-επεξεργασίας του dataset, το smoothing. Στην αρχή του matlab script που συνοδεύει το πρώτο μέρος της εργασίας (*main_4a.m*) δίνονται οι smoothing factors που χρησιμοποιήθηκαν για κάθε feature του dataset (βγήκαν με trial-and-error).

Μία άλλη μέθοδος προ-επεξεργασίας του dataset που δοκιμάστηκε είναι η ανίχνευση/αφαίρεση outliers. Για το σκοπό αυτό χρησιμοποιήθηκε η *rmoutliers()* του Matlab (R2019a). Πάλι με trial-and-error μεθοδολογία βρέθηκε ότι τα καλύτερα αποτελέσματα βγαίνουν για τη μέθοδο 'quartiles' και με *ThresholdFactor=20*.

Τέλος δοκιμάστηκε και το normalization του dataset ως προς το range (scaling όλων των data points στο εύρος [0,1]) ως μία ακόμη πιθανή μέθοδος προ-επεξεργασίας του dataset.

Διατηρώντας μία ακτίνα του SC (rad_{ii}) σταθερή, έγιναν πολλές δοκιμές για να διαπιστωθεί ποιος συνδυασμός μεθόδων προ-επεξεργασίας οδηγεί στα καλύτερα αποτελέσματα (μετρικές) στα παραγόμενα μοντέλα (συγκεκριμένα οι δοκιμές που έγιναν αφορούσαν το τελευταίο μοντέλο). Αξίζει να σημειωθεί, ωστόσο, ότι χωρίς smoothing ή rmoutliers στο dataset δεν θα ήταν δυνατή η εκτέλεση του πρώτου μέρους της εργασίας καθώς κατά την ομαδοποίηση (clustering) των training data points προκύπταν συνεχώς clusters με ένα στοιχείο, κάτι που απαγόρευε την εκπαίδευση του αντίστοιχου κανόνα του αρχικού FIS.

Ο βέλτιστος συνδυασμός μεθόδων, ο οποίος και χρησιμοποιήθηκε, είναι:

1. ***normalize()*** με παράμετρο "**range**"
2. ***rmoutliers()***, με βάση τα **quartiles** και με ***ThresholdFactor=20***

1.2. Διαχωρισμός του dataset

Για το διαχωρισμό (splitting) του dataset ακολουθήθηκε παρόμοια διαδικασία με την εργασία 3, με μία μικρή διαφοροποίηση. Επειδή έχουμε classification task καλό θα ήταν τα sub-sets που θα προκύψουν από το splitting να έχουν παρόμοιες συχνότητες εμφάνισης για κάθε κλάση του dataset. Για την επίτευξη του παραπάνω, πριν την κλήση της *AnfisWrapper::partition()* το dataset ταξινομείται ως προς την τελευταία στήλη που περιλαμβάνει τα indices των κλάσεων ταξινόμησης, έτσι ώστε κατά τη διαδοχική ανάθεση data points σε καθένα από τα τρία subsets να γίνει “δίκαια” (δηλ. να πάνε τόσα σημεία από κάθε κλάση όσος και ο λόγος των μεγεθών), το οποίο οδηγεί σε ίδιες ή κοντινές συχνότητες εμφάνισης κάθε κλάσης σε κάθε subset.

Εναλλακτικά, όπως και στην εργασία 3, χρησιμοποιήθηκε η *cvpartition()* με την επιλογή *Stratify=true* και με πρώτο όρισμα τα labels (τελευταία στήλη) του dataset. Οι συχνότητες εμφάνισης της κάθε κλάσης σε καθένα από τα τρία subsets, χρησιμοποιώντας οποιαδήποτε από τις παραπάνω μεθόδους (πολύ κοντινά αποτελέσματα), δίνεται παρακάτω:

Class	Dataset	Training	Validation	Testing
C_1	0.4023162135	0.4023162135	0.4023162135	0.4023162135
C_2	0.0081067472	0.0080563948	0.0083081571	0.0080563948
C_3	0.0352970796	0.0353306479	0.0352467271	0.0352467271
C_4	0.1048841893	0.1049009735	0.1047331319	0.1049848943
C_5	0.1938569990	0.1938569990	0.1938569990	0.1938569990
C_6	0.0449144008	0.0448976166	0.0450654582	0.0448136959
C_7	0.0514602216	0.0515273582	0.0513595166	0.0513595166
C_8	0.0800100705	0.0799765022	0.0800604230	0.0800604230
C_9	0.0029204431	0.0028533065	0.0030211480	0.0030211480
C_{10}	0.0510070493	0.0511077543	0.0508559919	0.0508559919
C_{11}	0.0252265861	0.0251762336	0.0251762336	0.0254279960
C_{12}	0.4023162135	0.4023162135	0.4023162135	0.4023162135

Πίνακας 1: Κατανομή πιθανότητας εμφάνισης κλάσης στο αρχικό dataset καθώς και στα παραχθέντα subsets χρησιμοποιώντας cvpartition() + stratify

Επίσης, παραθέτονται και οι αποστάσεις μεταξύ των τριών τελευταίων διανυσμάτων από τα παραπάνω (αυτών που αφορούν τα 3 subsets), οι οποίες αποτελούν μια ένδειξη της ομοιότητας των κατανομών των κλάσεων μεταξύ των subsets (με χρήση της *boxdist()*):

	Training	Validation	Testing
Training	-	0.0002518	0.0002518
Validation	0.0002518	-	0.0002518
Testing	0.0002518	0.0002518	-

Πίνακας 2: "Απόσταση" (μέτρο ομοιότητας) μεταξύ των κατανομών πιθανότητας εμφάνισης κλάσης στα 3 subsets

1.3. Εύρεση SC ακτινών από αριθμό κανόνων

Όπως και στο δεύτερο μέρος της τρίτης εργασίας, έτσι και εδώ, για κάθε τιμή αριθμού κανόνων του παραγόμενου TSK μοντέλου αναζητούμε μία ακτίνα του αλγορίθμου Subtractive Clustering (SC) που να δίνει τον αριθμό αυτό ως αριθμό clusters/κανόνων. Ακολουθώντας την ίδια, bisection-like, μέθοδο, καταλήγουμε στα ακόλουθα αποτελέσματα:

N_R	*4*	8	12	16	20
RAD_{II}	**0.26875000**	**0.18437500**	**0.15273438**	**0.12636719**	**0.11582031**

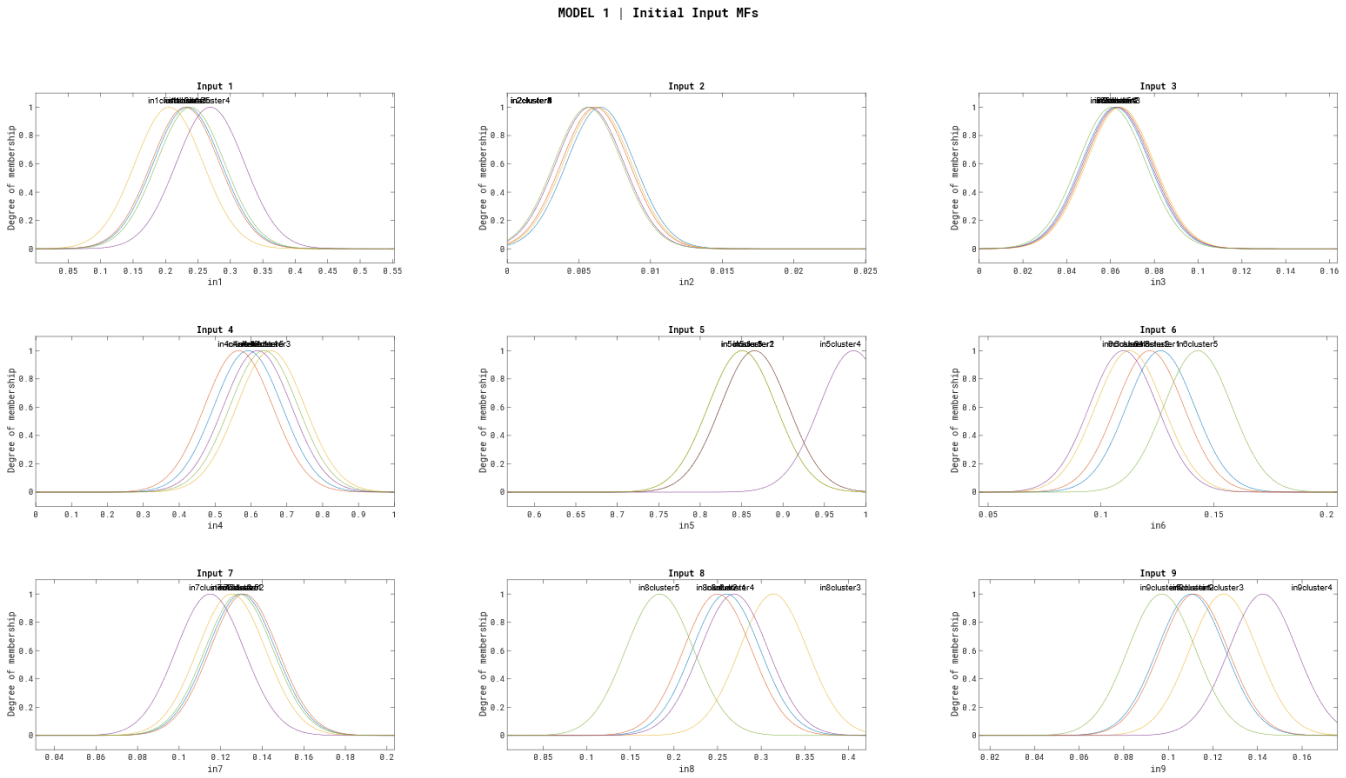
Πίνακας 3: Αντιστοίχιση ακτινών Subtractive Clustering στο ζητούμενο σετ αριθμών κανόνων των παραγόμενων TSK μοντέλων

1.4. Εκπαίδευση TSK μοντέλων

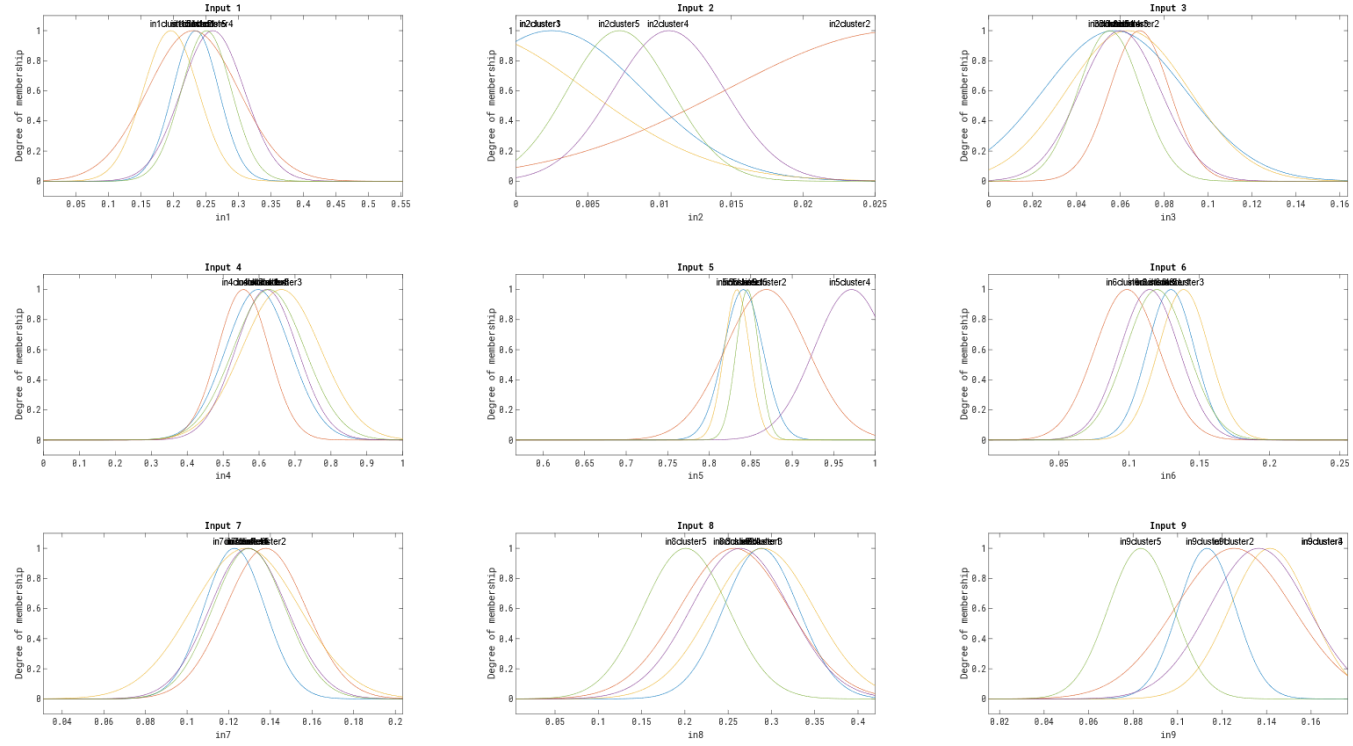
Εκπαιδεύτηκαν πέντε (5) TSK μοντέλα, ένα για κάθε πιθανό αριθμό κανόνων ή ακτινών όπως φαίνεται στον Πίνακα 3. Τα μοντέλα εκπαιδεύτηκαν με την υβριδική μέθοδο εκπαίδευσης με τη βοήθεια των συναρτήσεων *genfis()* και *anfis()* του MatLab. Παρακάτω παραθέτονται τα στοιχεία και οι μετρικές απόδοσης των εκπαιδευμένων μοντέλων.

1.4.1 Model 1: 5 clusters - fuzzy rules

Αρχικά, παρατίθενται οι αρχικές και τελικές μορφές των MFs για τις ασαφείς τιμές των ασαφών μεταβλητών εισόδου των κανόνων:

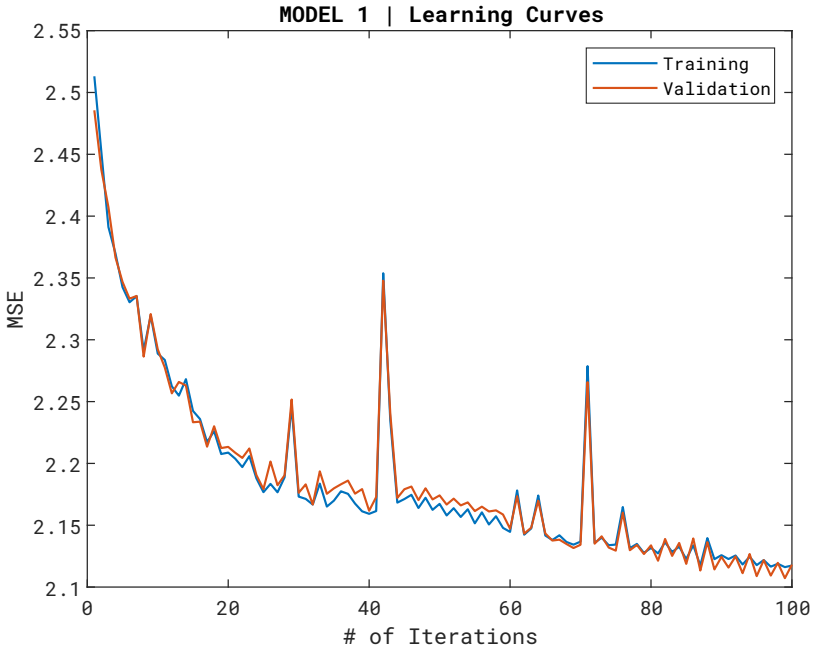


Εικόνα 7: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 1ου μοντέλου

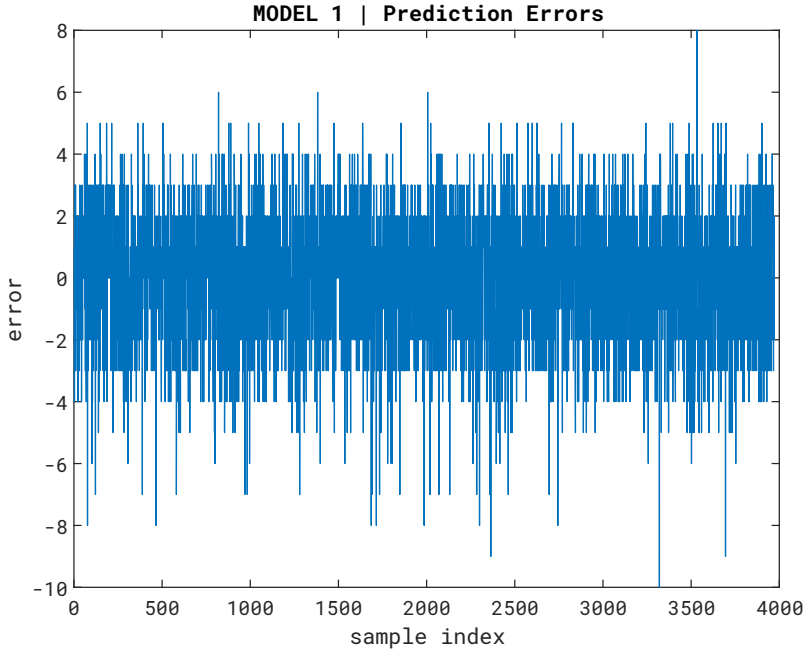


Εικόνα 8: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 1ου μοντέλου

Ακολούθως, δίνονται οι καμπύλες μάθησης (training) του μοντέλου καθώς και τα σφάλματα κατά την εφαρμογή του εκπαιδευμένου μοντέλου (testing) στο test set (*epochNumber=100*):



Εικόνα 9: Καμπύλες μάθησης 1ου μοντέλου (epochs=100)



Εικόνα 10: Σφάλματα πρόβλεψης κατά την εφαρμογή του 1ου μοντέλου στο test set

Τέλος, δίνεται ο confusion matrix του μοντέλου, ο οποίος επιτρέπει τον έλεγχο της απόδοσης του εκπαιδευμένου μοντέλου στο test set. Μαζί δίνονται και οι ζητούμενες μετρικές απόδοσης:

Pred. → ↓ Actual	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	238	439	473	306	103	35	3	0	1	0	0	0
C_2	0	0	0	0	0	0	0	0	0	0	0	0
C_3	0	0	1	11	7	9	4	0	0	0	0	0
C_4	1	4	33	55	33	11	3	0	0	0	0	0
C_5	3	9	18	91	107	121	57	9	2	0	0	0
C_6	3	64	224	273	141	55	4	5	0	1	0	0
C_7	0	1	12	50	53	44	14	1	2	1	0	0
C_8	1	4	20	49	46	50	32	1	1	0	0	0
C_9	1	0	1	0	0	1	3	13	241	56	2	0
C_{10}	0	2	1	1	4	2	2	0	0	0	0	0
C_{11}	1	2	1	11	6	17	8	16	16	30	48	46
C_{12}	0	0	0	3	3	3	0	2	2	6	29	53

Πίνακας 4: Confusion Matrix του 1ου μοντέλου

Class → ↓ Metric	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
OA	**0.2046 (20.46 %)**											
PA	0.149	NaN	0.031	0.393	0.257	0.071	0.079	0.005	0.758	0.000	0.238	0.525
UA	0.960	0.000	0.001	0.065	0.213	0.158	0.108	0.021	0.909	0.000	0.608	0.535
$\widehat{k}$	0.1414 (14.14 %)											

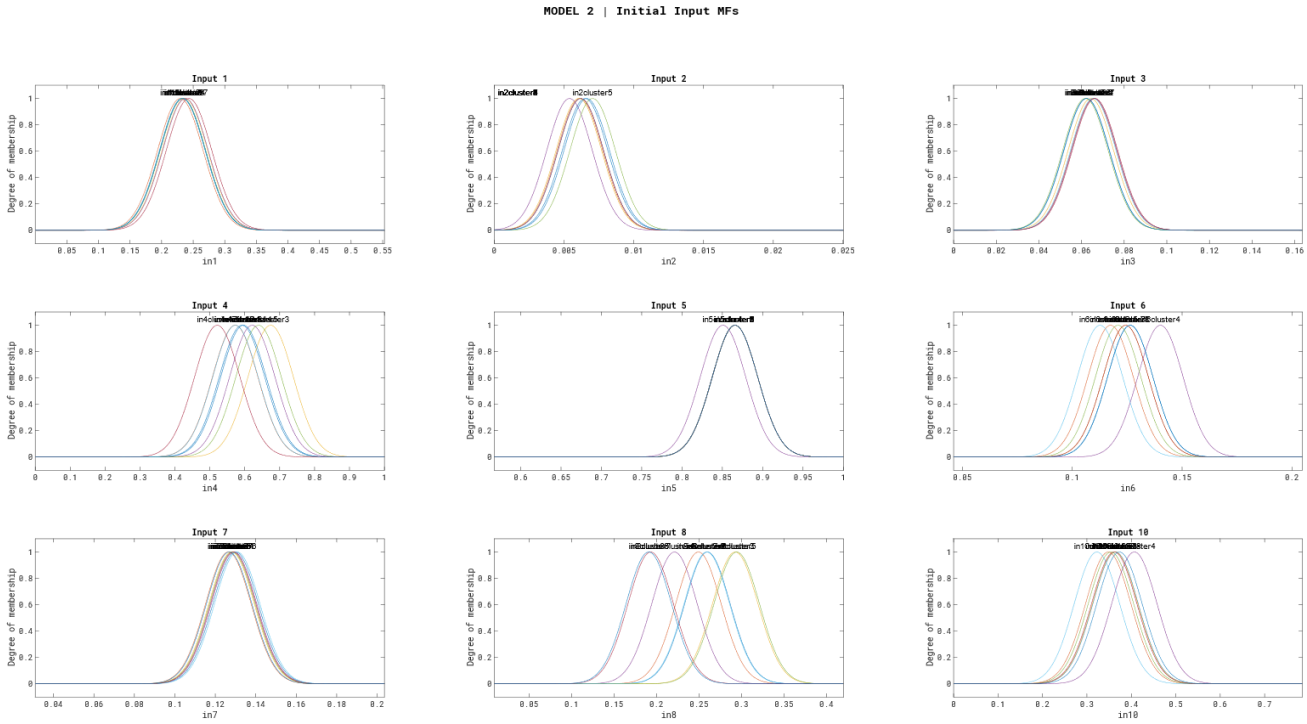
Πίνακας 5: Μετρικές Απόδοσης του 1ου μοντέλου

Μια πρώτη ανάλυση:

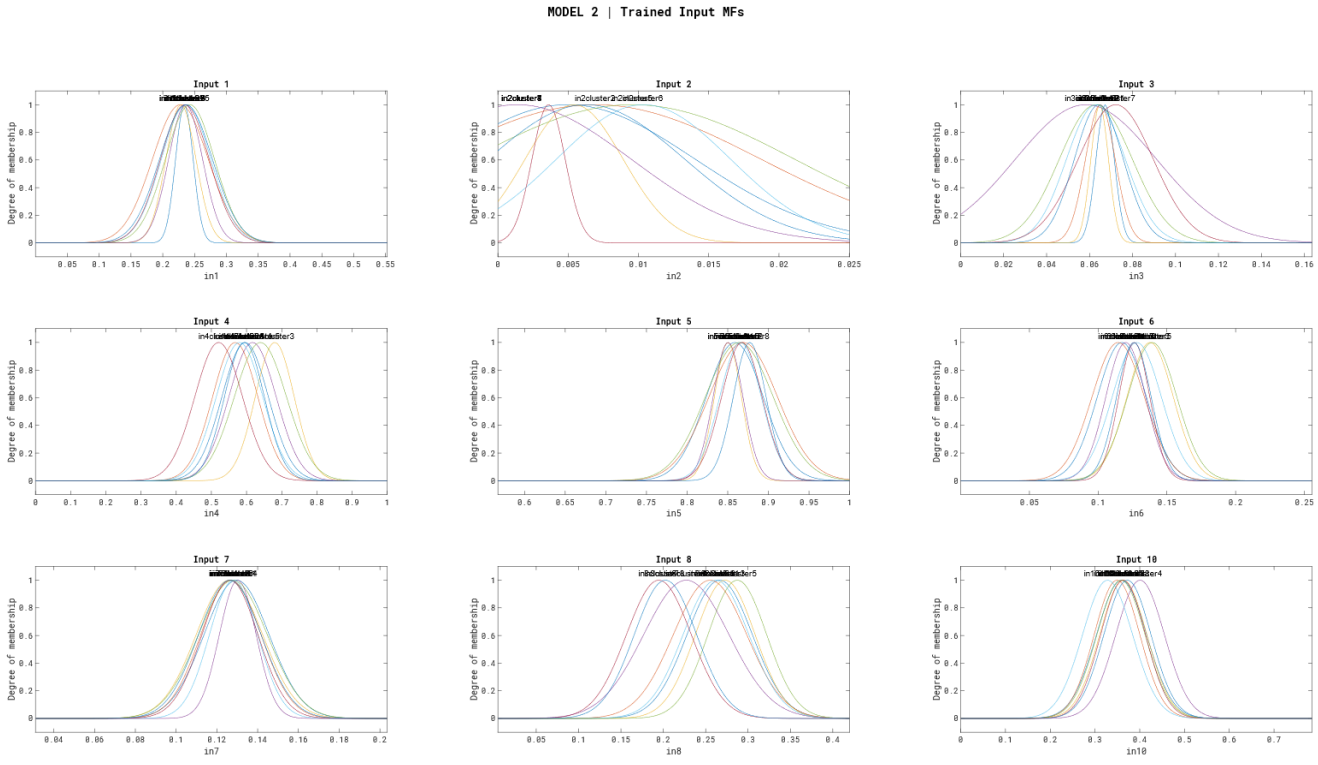
Από τα παραπάνω γίνεται προφανές ότι το 1ο μοντέλο δεν αποδίδει καλά στην ταξινόμηση των σημείων του δοθέντος test set με 20.46% Overall Accuracy. Όπως αναλύεται και στη συνέχεια, αυτό έχει σίγουρα να κάνει με τον μικρό αριθμό clusters / ασαφών κανόνων που χρησιμοποιεί αλλά κυρίως έχει να κάνει με την φύση του αρχικού dataset. Στο αρχικό dataset εκτός από την ύπαρξη πολλών ακραίων σημείων (outliers) που δυσχεραίνουν την αποτελεσματική ομαδοποίηση, η κατανομή των κλάσεων είναι εντελώς ανομοιόμορφη. Χαρακτηριστικό παράδειγμα αποτελεί ότι για την κλάση *2 (C_2)* υπάρχουν μόλις 10 δείγματα από το σύνολο των ~21K δειγμάτων. Παρόμοια είναι η κατάσταση και με την κλάση *10*. Μετά την παρουσίαση και των υπόλοιπων μοντέλων που εκπαιδεύτηκαν ακολουθεί μια πιο ενδελεχής ανάλυση σχετικά με τις αποδόσεις των TSK μοντέλων που αναπτύχθηκαν.

1.4.2 Model 2: 8 fuzzy rules

Αρχικά, παρατίθενται οι αρχικές και τελικές μορφές των MFs για τις ασαφείς τιμές των ασαφών μεταβλητών εισόδου των κανόνων του 2ου TSK μοντέλου που αναπτύχθηκε:

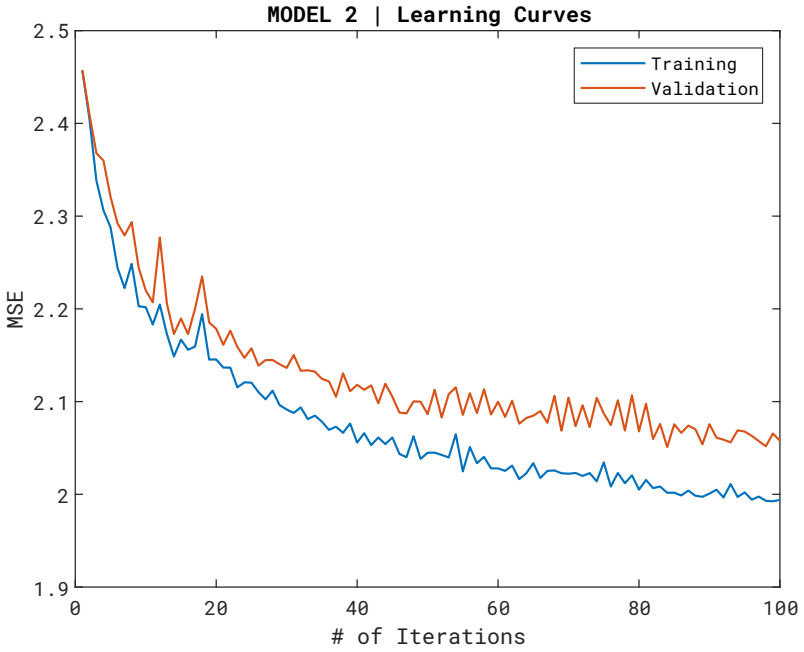


Εικόνα 11: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 2ου μοντέλου

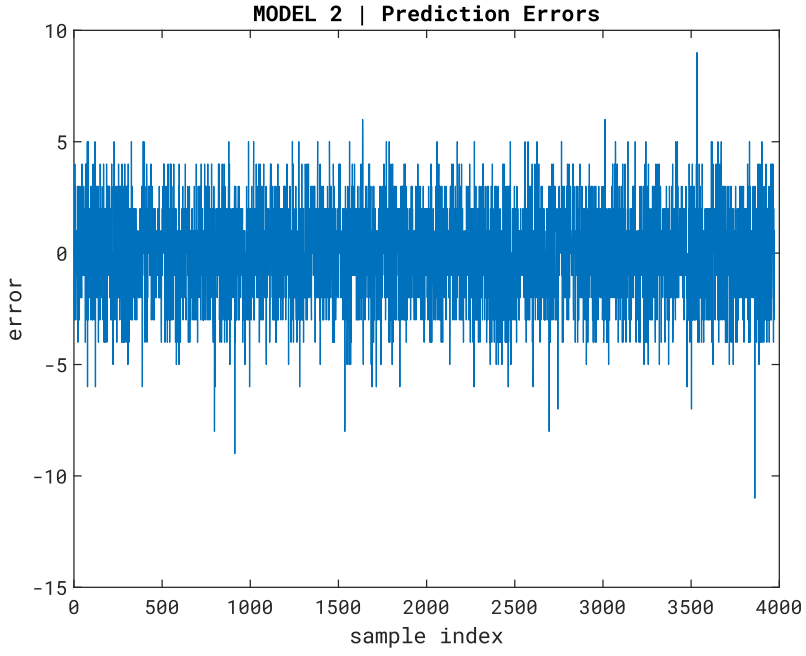


Εικόνα 12: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 2ου μοντέλου

Ακολούθως, δίνονται οι καμπύλες μάθησης (training) του μοντέλου καθώς και τα σφάλματα κατά την εφαρμογή του εκπαιδευμένου μοντέλου (testing) στο test set (*epochNumber=100*):



Εικόνα 13: Καμπύλες μάθησης 2ου μοντέλου (epochs=100)



Εικόνα 14: Σφάλματα πρόβλεψης κατά την εφαρμογή του 2ου μοντέλου στο test set

Τέλος, δίνεται ο confusion matrix του μοντέλου, ο οποίος επιτρέπει τον έλεγχο της απόδοσης του εκπαιδευμένου μοντέλου στο test set. Μαζί δίνονται και οι ζητούμενες μετρικές απόδοσης:

Pred. → ↓ Actual	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	354	387	413	291	112	38	2	0	0	1	0	0
C_2	0	0	0	0	0	0	0	0	0	0	0	0
C_3	0	0	9	13	7	2	1	0	0	0	0	0
C_4	2	4	23	54	35	17	4	1	0	0	0	0
C_5	2	10	29	85	127	91	55	13	5	0	0	0
C_6	10	78	169	242	187	67	12	4	1	0	0	0
C_7	0	0	8	40	43	52	20	12	3	0	0	0
C_8	1	3	9	33	51	53	41	9	3	1	0	0
C_9	0	0	0	0	1	8	11	59	190	44	5	0
C_{10}	0	0	0	1	1	6	3	1	0	0	0	0
C_{11}	0	0	2	1	4	9	11	22	40	45	35	33
C_{12}	0	0	0	0	1	4	4	2	1	11	25	53

Πίνακας 6: Confusion Matrix του 2ου μοντέλου

Class → ↓ Metric	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
OA	**0.2311 (23.11 %)**											
PA	0.222	NaN	0.281	0.386	0.305	0.087	0.112	0.044	0.597	0.000	0.173	0.525
UA	0.959	0.000	0.014	0.071	0.223	0.193	0.122	0.073	0.782	0.000	0.538	0.616
$\hat{k}$	0.1576 (15.76 %)											

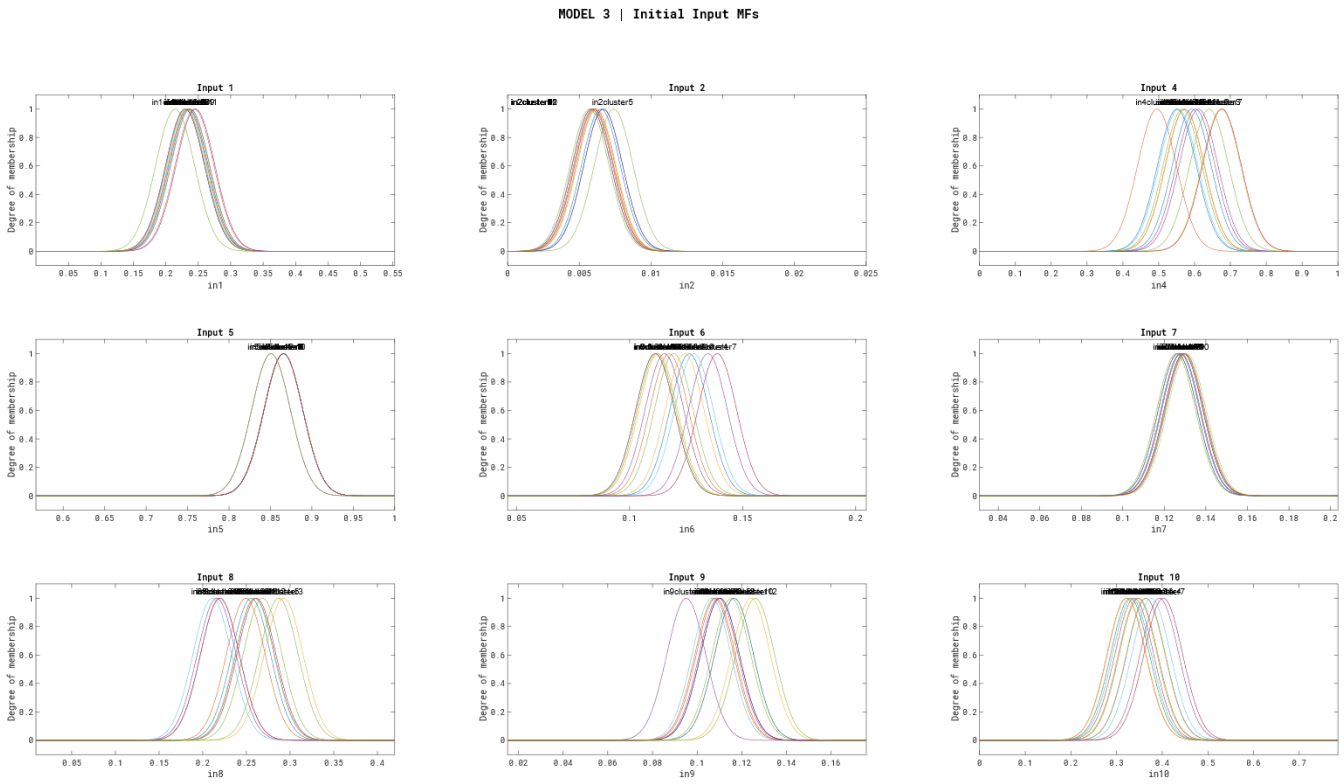
Πίνακας 7: Μετρικές Απόδοσης του 2ου μοντέλου

Μια πρώτη ανάλυση:

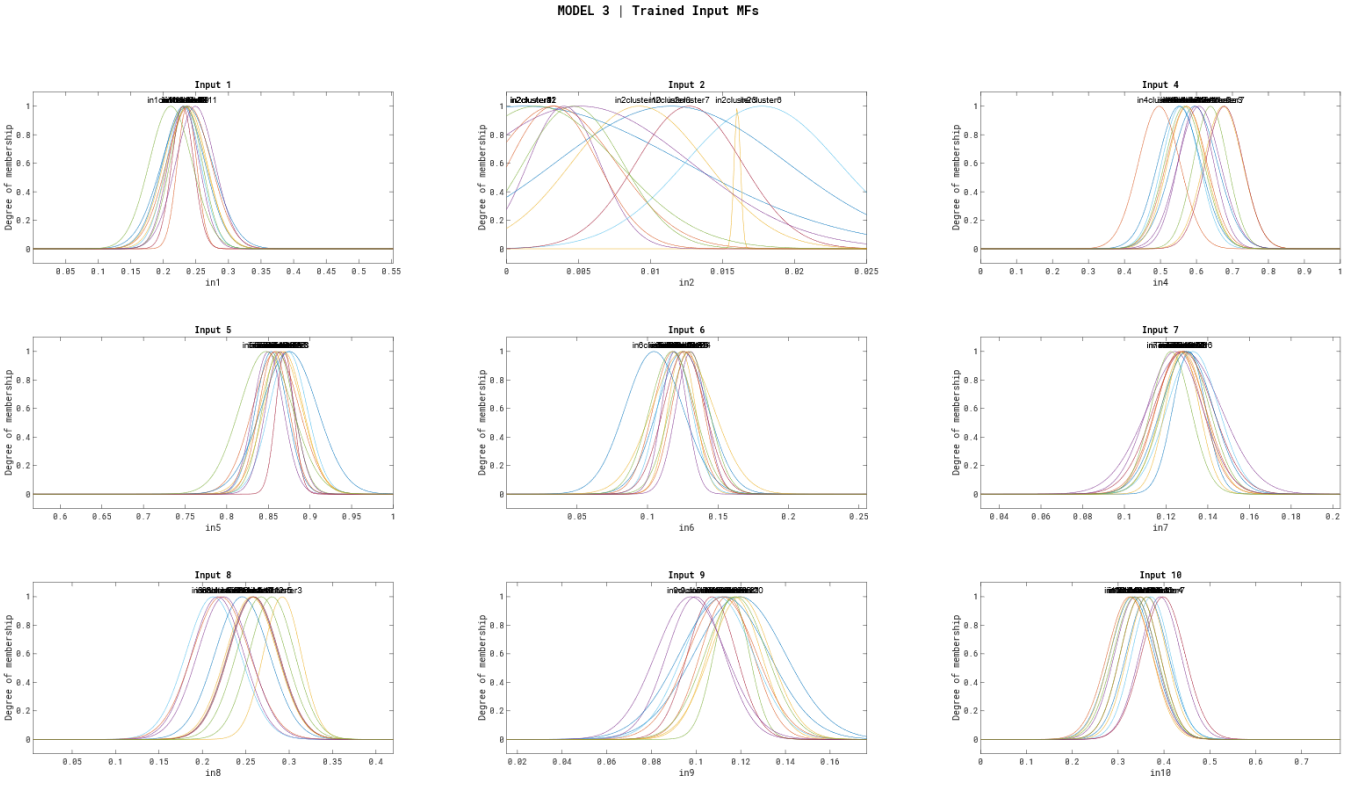
Από τα παραπάνω φαίνεται ότι αν και το 2ο μοντέλο αποδίδει καλύτερα από το 1ο, ούτε το 2ο μοντέλο αποδίδει αρκετά καλά στην ταξινόμηση των σημείων του δοθέντος test set, με 23.11% Overall Accuracy. Μετά την παρουσίαση και των υπόλοιπων μοντέλων που εκπαιδεύτηκαν ακολουθεί μια πιο ενδελεχής ανάλυση σχετικά με τις αποδόσεις των TSK μοντέλων που αναπτύχθηκαν.

1.4.3 Model 3: 12 fuzzy rules

Αρχικά, παρατίθενται οι αρχικές και τελικές μορφές των MFs για τις ασαφείς τιμές των ασαφών μεταβλητών εισόδου των κανόνων του 3ου TSK μοντέλου που αναπτύχθηκε:

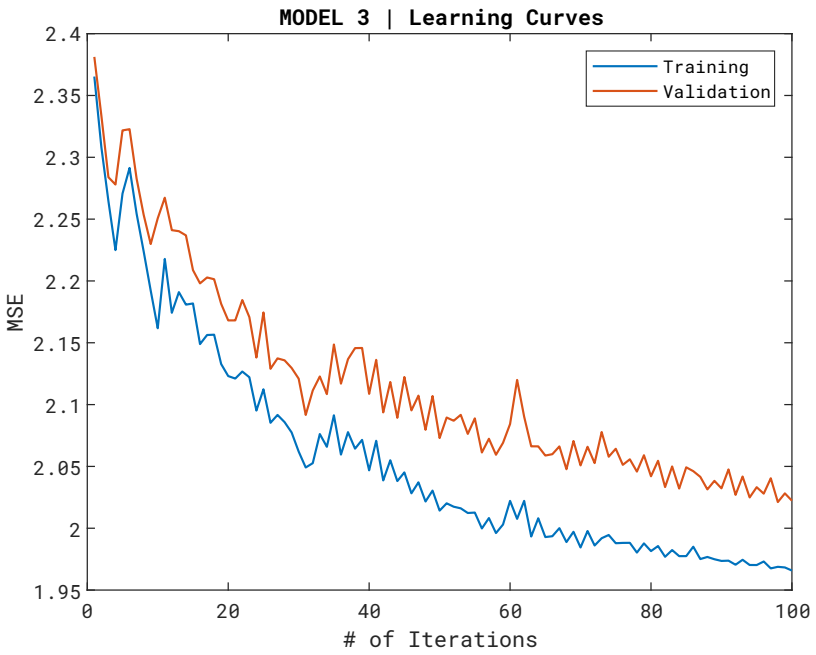


Εικόνα 15: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 3ου μοντέλου

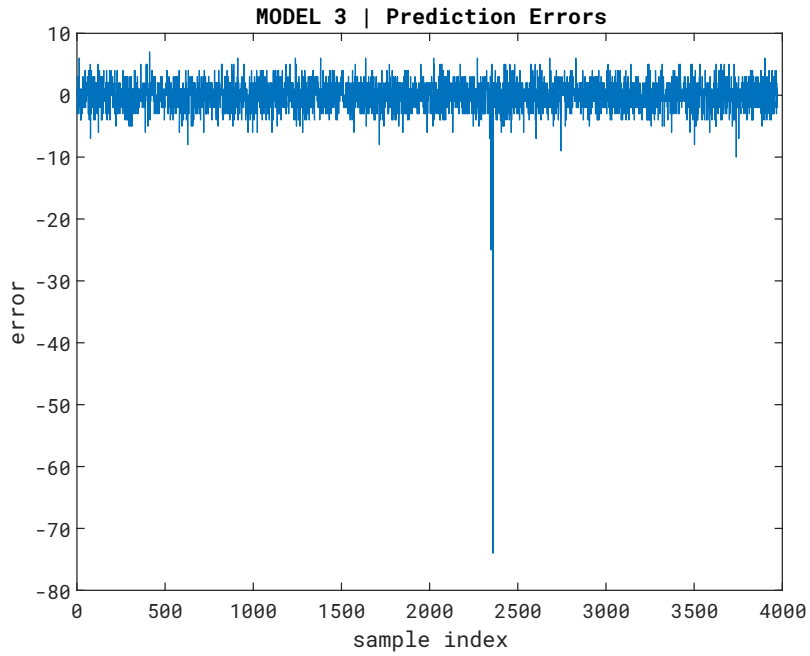


Εικόνα 16: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 3ου μοντέλου

Ακολούθως, δίνονται οι καμπύλες μάθησης (training) του μοντέλου καθώς και τα σφάλματα κατά την εφαρμογή του εκπαιδευμένου μοντέλου (testing) στο test set (*epochNumber=100*):



Εικόνα 17: Καμπύλες μάθησης 3ου μοντέλου (epochs=100)



Εικόνα 18: Σφάλματα πρόβλεψης κατά την εφαρμογή του 3ου μοντέλου στο test set

Τέλος, δίνεται ο confusion matrix του μοντέλου, ο οποίος επιτρέπει τον έλεγχο της απόδοσης του εκπαιδευμένου μοντέλου στο test set. Μαζί δίνονται και οι ζητούμενες μετρικές απόδοσης:

Pred. → ↓ Actual	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	361	415	432	231	106	42	10	1	0	0	0	0
C_2	0	0	0	0	0	0	0	0	0	0	0	0
C_3	1	6	3	11	5	6	0	0	0	0	0	0
C_4	2	8	31	50	32	14	3	0	0	0	0	0
C_5	0	8	26	103	134	89	44	7	4	2	0	0
C_6	16	73	175	215	165	93	26	3	2	1	1	0
C_7	0	0	9	41	38	71	11	4	3	0	1	0
C_8	1	4	9	36	49	55	41	8	1	0	0	0
C_9	0	0	2	0	1	5	3	37	229	40	1	0
C_{10}	0	0	0	0	2	3	1	3	1	0	2	0
C_{11}	1	0	1	2	8	6	15	22	30	37	47	33
C_{12}	0	0	1	1	2	3	0	3	4	9	20	58

Πίνακας 8: Confusion Matrix του 3ου μοντέλου

Class → ↓ Metric	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
OA	**0.2502 (25.02 %)**											
PA	0.226	NaN	0.094	0.357	0.321	0.121	0.062	0.039	0.720	0.000	0.233	0.574
UA	0.945	0.000	0.004	0.072	0.247	0.240	0.071	0.091	0.836	0.000	0.653	0.637
$\widehat{k}$	0.1766 (17.66 %)											

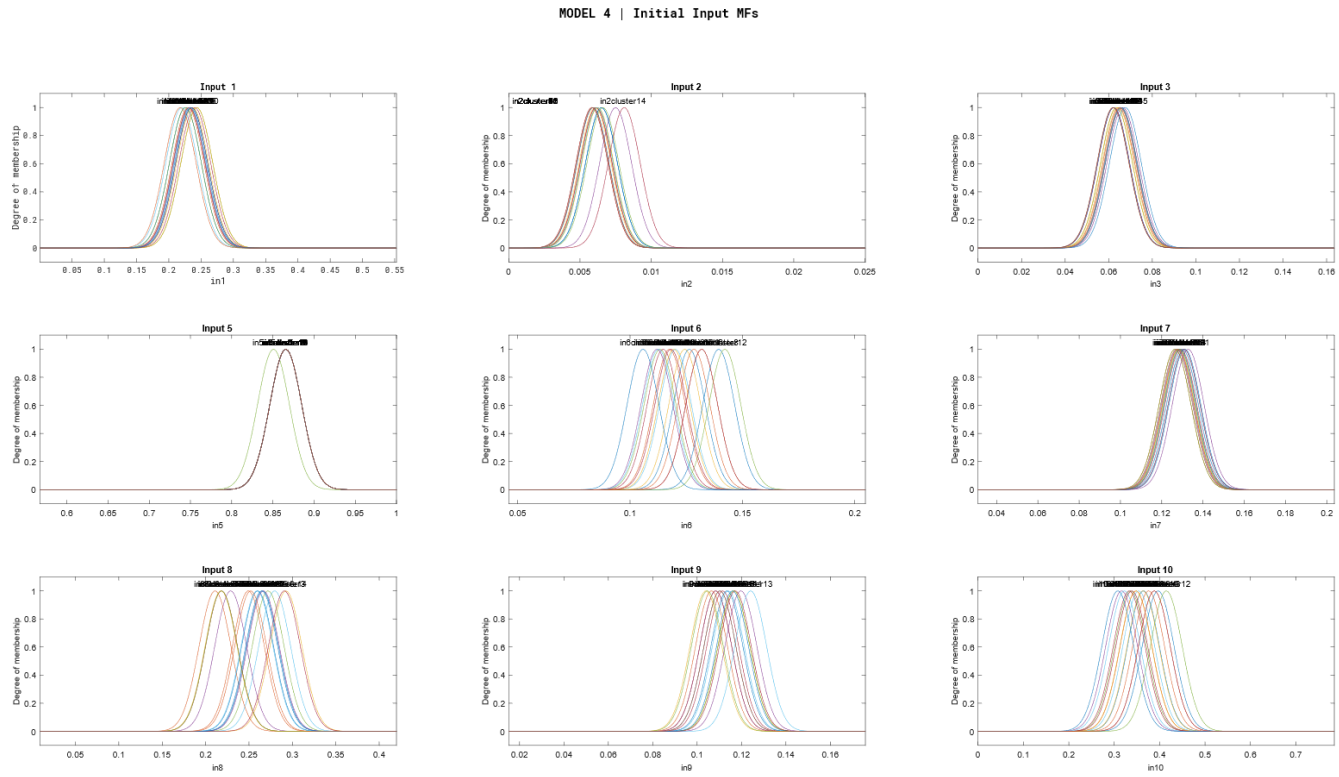
Πίνακας 9: Μετρικές Απόδοσης του 3ου μοντέλου

Μια πρώτη ανάλυση:

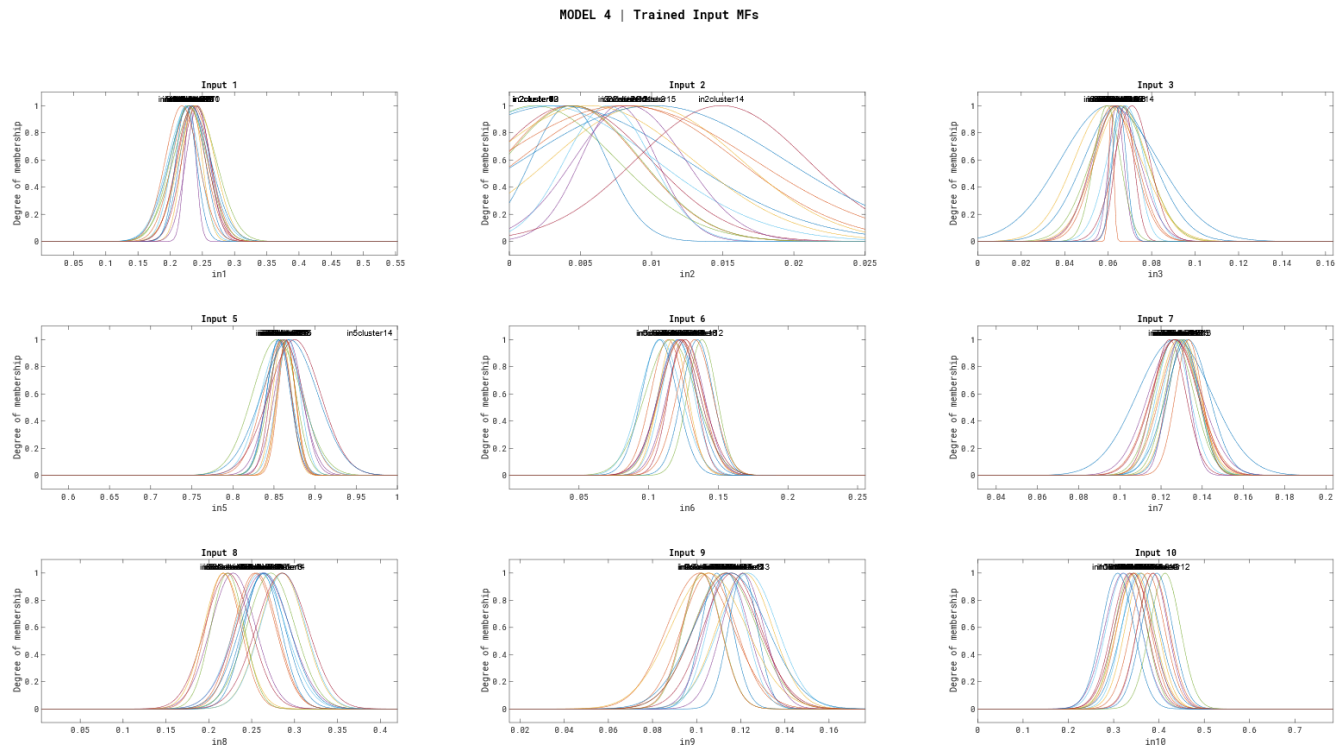
Από τα παραπάνω φαίνεται ότι αν και το 3ο μοντέλο αποδίδει λίγο καλύτερα από τα προηγούμενα δύο, ούτε το 3ο μοντέλο αποδίδει αρκετά καλά στην ταξινόμηση των σημείων του δοθέντος test set με μόλις 25.02% Overall Accuracy. Εδώ παρατηρούμε ότι το μοντέλο αναγνωρίζει καλύτερα τις κλάσεις *6* και *9*, κάτι που πιθανώς να οφείλεται στο ότι νέα clusters δημιουργήθηκαν γύρω από data points αυτών των κλάσεων και έτσι το ασαφές μοντέλο κάλυπτε πιο λεπτομερώς τη περιοχή εκείνη (στον 10-διάστατο χώρο).

1.4.4 Model 4: 16 fuzzy rules

Αρχικά, παρατίθενται οι αρχικές και τελικές μορφές των MFs για τις ασαφείς τιμές των ασαφών μεταβλητών εισόδου των κανόνων του 4ου TSK μοντέλου που αναπτύχθηκε:

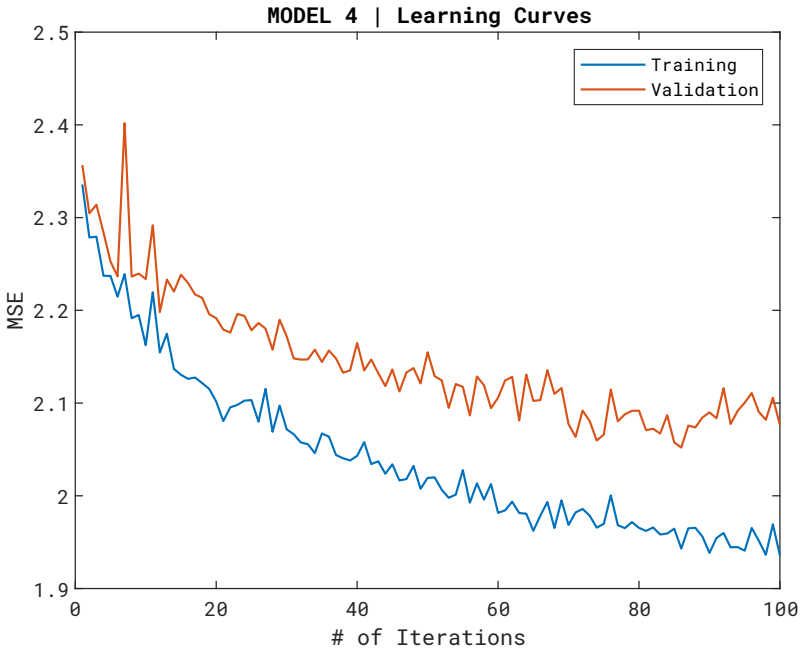


Εικόνα 19: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 4ου μοντέλου

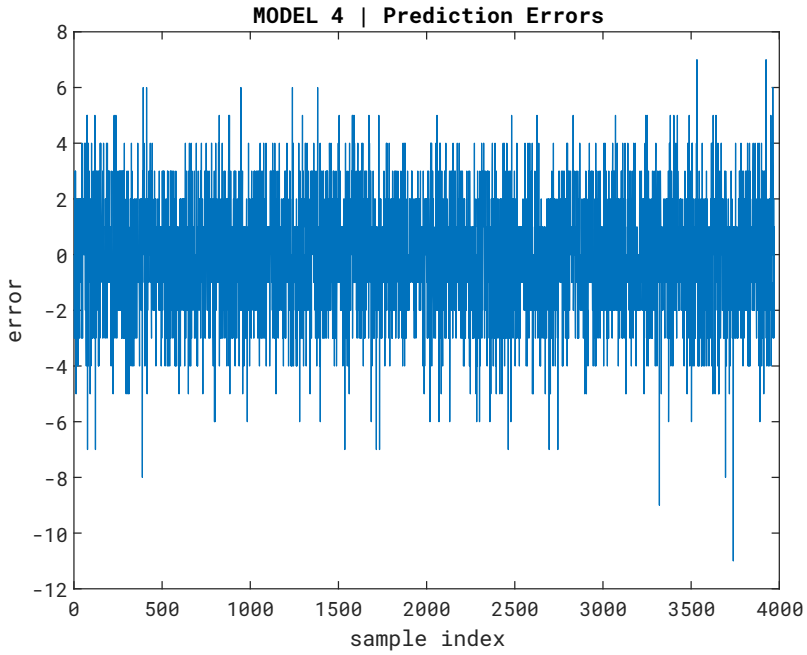


Εικόνα 20: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 4ου μοντέλου

Ακολούθως, δίνονται οι καμπύλες μάθησης (training) του μοντέλου καθώς και τα σφάλματα κατά την εφαρμογή του εκπαιδευμένου μοντέλου (testing) στο test set (*epochNumber=100*):



Εικόνα 21: Καμπύλες μάθησης 4ου μοντέλου (epochs=100)



Εικόνα 22: Σφάλματα πρόβλεψης κατά την εφαρμογή του 4ου μοντέλου στο test set

Τέλος, δίνεται ο confusion matrix του μοντέλου, ο οποίος επιτρέπει τον έλεγχο της απόδοσης του εκπαιδευμένου μοντέλου στο test set. Μαζί δίνονται και οι ζητούμενες μετρικές απόδοσης:

Pred. → ↓ Actual	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	375	400	398	273	119	26	5	2	0	0	0	0
C_2	0	0	0	0	0	0	0	0	0	0	0	0
C_3	1	1	7	11	7	5	0	0	0	0	0	0
C_4	5	4	27	40	49	15	0	0	0	0	0	0
C_5	4	7	29	99	125	112	25	8	6	2	0	0
C_6	18	70	152	244	179	86	19	1	0	0	0	1
C_7	1	3	9	50	37	43	22	10	3	0	0	0
C_8	0	4	10	30	50	59	31	17	2	1	0	0
C_9	0	0	0	1	1	0	11	48	209	42	5	1
C_{10}	0	0	0	2	1	3	3	3	0	0	0	0
C_{11}	1	1	2	3	7	5	14	15	30	33	44	47
C_{12}	0	0	0	0	5	2	2	0	1	9	26	56

Πίνακας 10: Confusion Matrix του 4ου μοντέλου

Class → ↓ Metric	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
OA	**0.2470 (24.70 %)**											
PA	0.235	NaN	0.219	0.286	0.300	0.112	0.124	0.083	0.657	0.000	0.218	0.554
UA	0.926	0.000	0.011	0.053	0.216	0.242	0.167	0.163	0.833	0.000	0.587	0.533
$\widehat{k}$	0.1713 (17.13 %)											

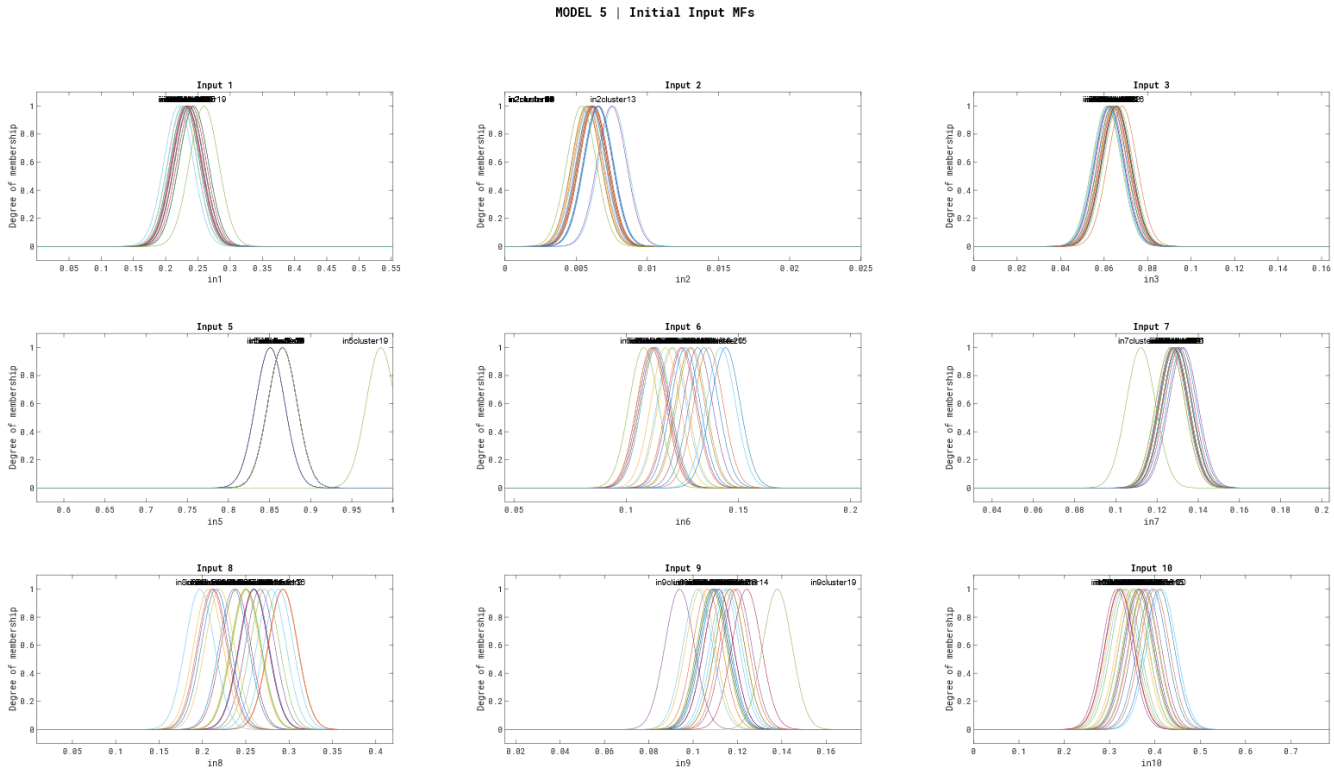
Πίνακας 11: Μετρικές Απόδοσης του 4ου μοντέλου

Μια πρώτη ανάλυση:

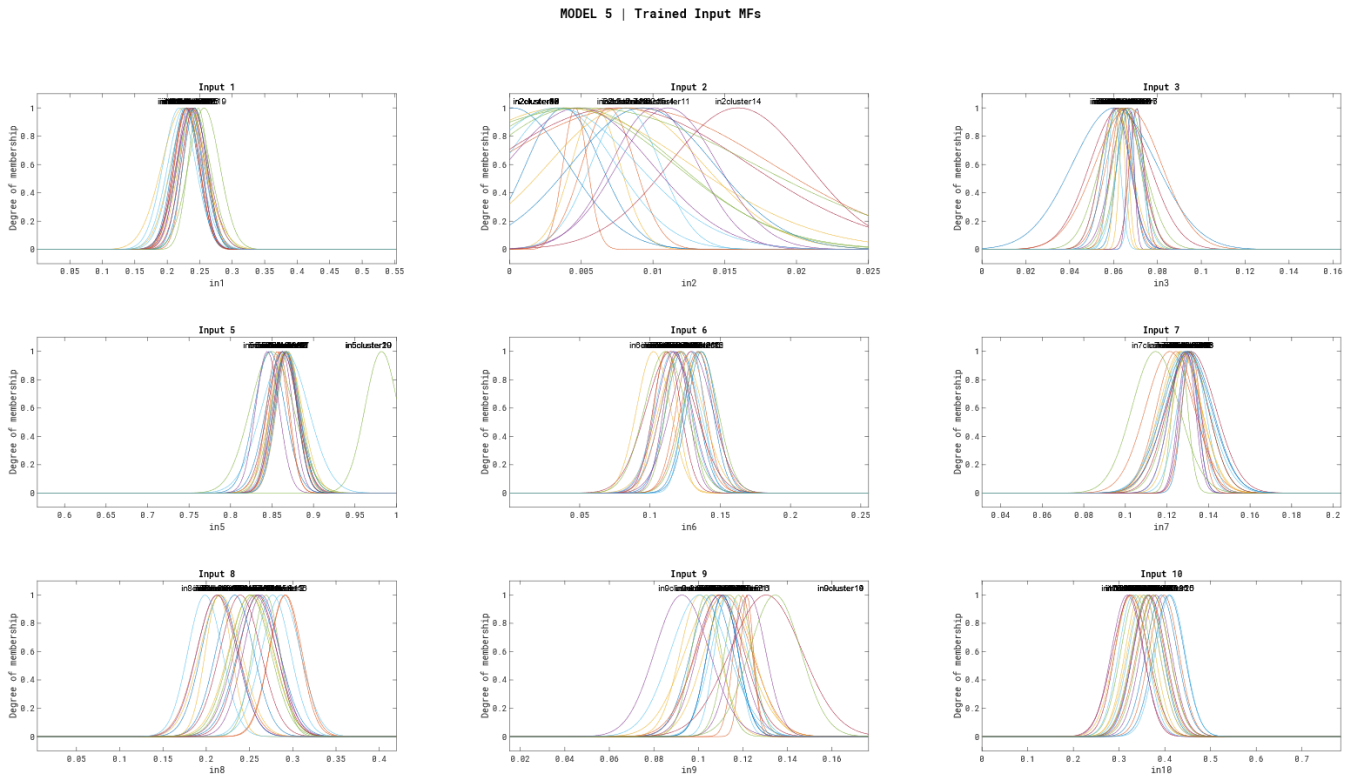
Από τα παραπάνω φαίνεται ότι το 4ο μοντέλο (για 16 clusters/κανόνες – rad_{ii}≈0.126) αποδίδει λίγο χειρότερα από το προηγούμενο κάτι το οποίο φαίνεται και στον confusion matrix. Παρατηρείται μια μικρή μείωση στην αποδοτικότητα εύρεσης σχεδόν όλων των κλάσεων πλην των 10 και μετά. Ακολουθεί το 5ο και τελευταίο μοντέλο που εκπαιδεύτηκε, για 20 κανόνες/clusters.

1.4.5 Model 5: 20 fuzzy rules

Αρχικά, παρατίθενται οι αρχικές και τελικές μορφές των MFs για τις ασαφείς τιμές των ασαφών μεταβλητών εισόδου των κανόνων του 5ου TSK μοντέλου που αναπτύχθηκε:

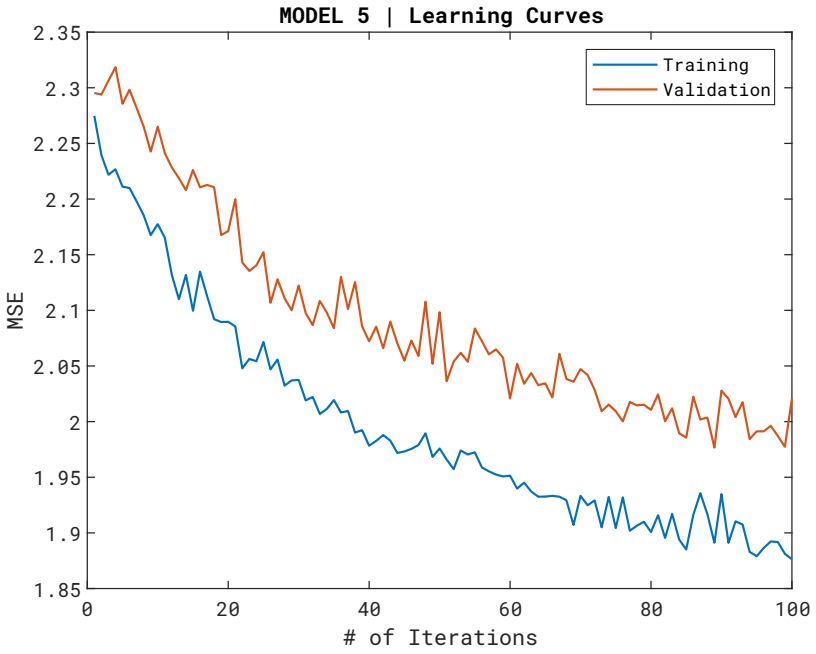


Εικόνα 23: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 5ου μοντέλου

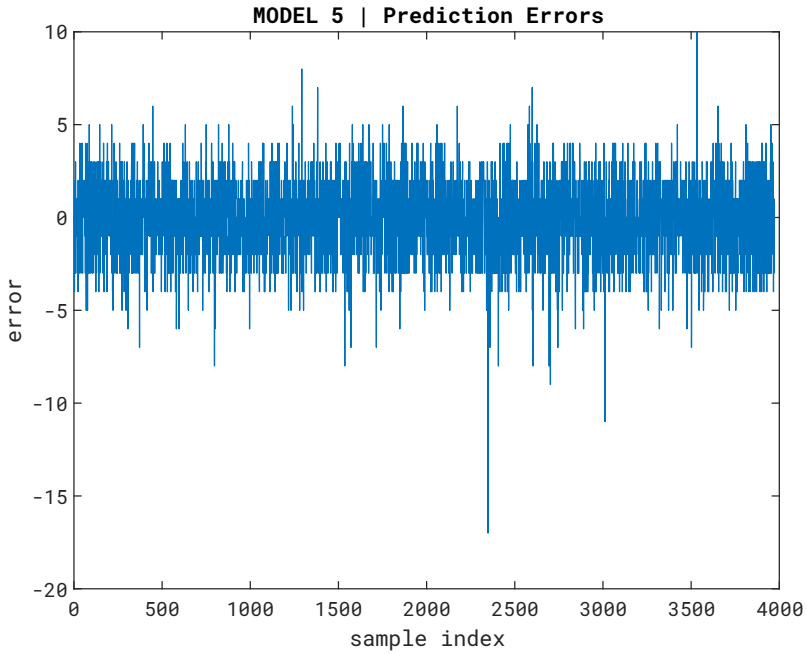


Εικόνα 24: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου 5ου μοντέλου

Ακολούθως, δίνονται οι καμπύλες μάθησης (training) του μοντέλου καθώς και τα σφάλματα κατά την εφαρμογή του εκπαιδευμένου μοντέλου (testing) στο test set (*epochNumber=100*):



Εικόνα 25: Καμπύλες μάθησης 5ου μοντέλου (epochs=100)



Εικόνα 26: Σφάλματα πρόβλεψης κατά την εφαρμογή του 5ου μοντέλου στο test set

Τέλος, δίνεται ο confusion matrix του μοντέλου, ο οποίος επιτρέπει τον έλεγχο της απόδοσης του εκπαιδευμένου μοντέλου στο test set. Μαζί δίνονται και οι ζητούμενες μετρικές απόδοσης:

Pred. → ↓ Actual	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
C_1	406	431	419	217	96	21	4	2	1	0	1	0
C_2	0	0	0	0	0	0	0	0	0	0	0	0
C_3	1	11	5	5	10	0	0	0	0	0	0	0
C_4	2	5	17	50	38	18	8	2	0	0	0	0
C_5	2	9	31	90	143	96	31	8	5	0	2	0
C_6	13	69	183	212	179	98	12	4	0	0	0	0
C_7	1	4	7	35	42	52	25	9	2	1	0	0
C_8	2	5	10	30	46	51	35	21	3	1	0	0
C_9	0	1	0	1	1	1	4	21	238	48	2	1
C_{10}	0	0	0	0	0	3	4	2	2	1	0	0
C_{11}	0	0	4	1	3	3	9	22	32	42	47	39
C_{12}	0	0	0	0	2	0	4	3	4	3	38	47

Πίνακας 12: Confusion Matrix του 5ου μοντέλου

Class → ↓ Metric	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
OA	**0.2721 (27.21 %)**											
PA	0.254	NaN	0.156	0.357	0.343	0.127	0.140	0.103	0.748	0.083	0.233	0.465
UA	0.951	0.000	0.007	0.078	0.255	0.286	0.184	0.223	0.829	0.010	0.522	0.540
$\hat{k}$	0.1983 (19.83 %)											

Πίνακας 13: Μετρικές Απόδοσης του 5ου μοντέλου

Μια πρώτη ανάλυση:

Όπως ήταν αναμενόμενο το 5ο μοντέλο, έχοντας τους περισσότερους ασαφείς κανόνες σε σχέση με τα προηγούμενα τέσσερα, πετυχαίνει καλύτερη Overall Accuracy στο test set. Η απόδοση του μοντέλου όμως και πάλι είναι λίγο ικανοποιητική καθώς είναι μόλις πάνω από 27%. Παρακάτω παραθέτονται μερικά συμπερασματικά σχόλια για το πρώτο μέρος της τέταρτης εργασίας.

1.5. Συμπερασματική ανάλυση

Αρχικά παραθέτονται συγκεντρωτικά οι μετρικές απόδοσης όλων των μοντέλων:

Metric → ↓ Model	**OA**	$\widehat{k}$
Model 1	0.2046 (**20.46%**)	0.1414
Model 2	0.2311 (**23.11%**)	0.1576
Model 3	0.2502 (**25.02%**)	0.1766
Model 4	0.2470 (**24.70%**)	0.1713
Model 5	0.2721 (**27.21%**)	0.1983

Πίνακας 14: Συγκεντρωτικές Μετρικές Απόδοσης όλων των μοντέλων

Βάσει των παραπάνω μετρικών απόδοσης φαίνεται ότι για το συγκεκριμένο dataset αύξηση του αριθμού των clusters του scatter partitioning στις εισόδους των TSK μοντέλων, οδηγεί εν γένει σε καλύτερες αποδόσεις classification. Έτσι, όπως φαίνεται και από τις μετρικές των 5 μοντέλων που εκπαιδεύτηκαν, στον Πίνακα 14 παραπάνω, τη **καλύτερη απόδοση** (δείκτης Overall Accuracy - OA) παρουσιάζει το **5ο μοντέλο (20 clusters) με 27.21%**.

Επίσης, σημαντικό ρόλο τόσο στην ολοκλήρωση του training όσο και στις μετρικές των εκπαιδευμένων μοντέλων παίζει η προ-επεξεργασία του dataset και συγκεκριμένα η αφαίρεση των outliers. Δοκιμές που έγιναν έδειξαν πως αυτή η μέθοδος πριν την εκπαίδευση οδηγεί αφενός σε «καλύτερες» ακτίνες rad_{ii} για το SC (δηλαδή όχι πολύ μικρές ακτίνες) και αφετέρου σε πολύ καλύτερες μετρικές απόδοσης των εκπαιδευμένων μοντέλων TSK.

Όπως έχει αναφερθεί και παραπάνω, οι κλάσεις είναι εντελώς ανομοιόμορφα διανεμημένες στο dataset με χαρακτηριστικά παραδείγματα τις κλάσεις 2 και 10 με μόλις 10 και 89 στοιχεία (data points) αντίστοιχα. Αυτό επιδεινώνεται ακόμα περισσότερο καθώς μετα την κλήση της *rmoutliers()* στο στάδιο της προ-επεξεργασίας του dataset πολλά από τα στοιχεία των δύο αυτών κλάσεων αφαιρούνται. Αυτός είναι και ο λόγος που για παράδειγμα το User Accuracy (UA) της κλάσης 2 και στα 5 μοντέλα που εκπαιδεύτηκαν είναι 0.

2. Εφαρμογή σε high-dimensional dataset

2.1. Κώδικας 2ου μέρους εργασίας

Ο κώδικας σε MATLAB που υλοποιεί το μέρος β' της εργασίας βρίσκεται στο αρχείο */4/matlab/main_b.m*. Εκεί υπάρχει μόνο η λογική της εκτέλεσης, ενώ βοηθητικές κλάσεις και συναρτήσεις υπάρχουν στο φάκελο */Matlab Helpers/**.

2.2. Φόρτωση & Προ-επεξεργασία dataset

Το dataset του ερωτήματος, *isolet*, αποτελείται από 7797 δείγματα (data points) με 617 features και μία ετικέτα (label) το καθένα. Υπάρχουν συνολικά 26 labels, από το «1» έως το «26».

Σε ότι αφορά τη προ-επεξεργασία του dataset, αρχικά, γίνεται έλεγχος για διπλότυπα δείγματα (~66 δείγματα αφαιρέθηκαν). Κατόπιν, ακολουθούμε παρόμοιο pre-processing του dataset όπως και στο μέρος α' της εργασίας. Έτσι, πάλι με trial-and-error καταλήγουμε στις εξής μεθόδους pre-processing (με τη σειρά που δίνονται):

1. ***unique()*** με παράμετρο "**rows**" και "**stable**"
2. ***normalize()*** με παράμετρο "**range**"
3. ***smoothdata()*** με παράμετρο *SmoothingFactor*=0.5

2.3. Διαχωρισμός Dataset

Το splitting του dataset σε training, validation & testing subsets γίνεται με βάση τη λογική και τις συναρτήσεις (μια ιδία και μία του Matlab – *cvpartition()*) που αναφέρονται στο μέρος α' του παρόντος (βλ. παράγραφος 1.3). Οι δύο συναρτήσεις παράγουν πολύ κοντινά αποτελέσματα. Τελικά, χρησιμοποιήθηκε η *cvpartition()* με την επιλογή *Stratify=true* και με πρώτο όρισμα τα labels (τελευταία στήλη) του dataset. Οι συχνότητες εμφάνισης της κάθε κλάσης σε καθένα από τα τρία subsets, χρησιμοποιώντας οποιαδήποτε από τις παραπάνω μεθόδους (πολύ κοντινά αποτελέσματα), δίνεται παρακάτω:

Class	Dataset	Training	Validation	Testing
C_1	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_2	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_3	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_4	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_5	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_6	0.0382198281	0.0382642155	0.0384615385	0.0378447723
C_7	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_8	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_9	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_{10}	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_{11}	0.0384763371	0.0384779820	0.0384615385	0.0384862091
C_{12}	0.038476337	0.038477982	0.038461538	0.038486209
C_{13}	0.038348083	0.038264215	0.038461538	0.038486209
C_{14}	0.038476337	0.038477982	0.038461538	0.038486209
C_{15}	0.03847634	0.03847798	0.03846154	0.03848621
C_{16}	0.038476337	0.038477982	0.038461538	0.038486209
C_{17}	0.038476337	0.038477982	0.038461538	0.038486209
C_{18}	0.038476337	0.038477982	0.038461538	0.038486209
C_{19}	0.038476337	0.038477982	0.038461538	0.038486209
C_{20}	0.038476337	0.038477982	0.038461538	0.038486209
C_{21}	0.038476337	0.038477982	0.038461538	0.038486209
C_{22}	0.038476337	0.038477982	0.038461538	0.038486209
C_{23}	0.038476337	0.038477982	0.038461538	0.038486209
C_{24}	0.038476337	0.038477982	0.038461538	0.038486209
C_{25}	0.038476337	0.038477982	0.038461538	0.038486209
C_{26}	0.038476337	0.038477982	0.038461538	0.038486209

Πίνακας 15: Κατανομή πιθανότητας εμφάνισης κλάσης στο αρχικό dataset καθώς και στα παραχθέντα subsets χρησιμοποιώντας cvpartition() + stratify

Επίσης, παραθέτονται και οι αποστάσεις μεταξύ των τριών τελευταίων διανυσμάτων από τα παραπάνω (αυτών που αφορούν τα 3 subsets), οι οποίες αποτελούν μια ένδειξη της ομοιότητας των κατανομών των κλάσεων μεταξύ των subsets (με χρήση της *boxdist()*):

	Training	Validation	Testing
Training	-	0.0001973	0.0004194
Validation	0.0001973	-	0.0006168
Testing	0.0004194	0.0006168	-

Πίνακας 16: "Απόσταση" (μέτρο ομοιότητας) μεταξύ των κατανομών πιθανότητας εμφάνισης κλάσης στα 3 subsets

2.4. Μείωση Διαστασιμότητας Dataset

Επειδή το δοσμένο dataset έχει high-dimensionality, ο αριθμός των απαιτούμενων κανόνων με βάση το grid partitioning στο χώρο των εισόδων – features του μοντέλου θα είναι τεράστιος. Για το λόγο αυτό, όπως αναφέρεται και στην εκφώνηση του 2^{ου} μέρους της παρούσας εργασίας, θα χρησιμοποιήσουμε τεχνικές μείωσης της διαστασιμότητας και τεχνικές ομαδοποίησης για το διαχωρισμό του χώρου των εισόδων με σκοπό τη περαιτέρω μείωση του απαιτούμενου αριθμού ασαφών κανόνων (καθώς αυτοί θα λάβουν σαν είσοδο όχι τα features αλλά τα ομαδοποιημένα features). Οι δύο τεχνικές που θα χρησιμοποιηθούν είναι:

- Ο αλγόριθμός ReliefF για feature subset selection
- Ο αλγόριθμος Fuzzy C-Means (**FCM**) για ομαδοποίηση των features (scatter partitioning) πριν την δημιουργία των κανόνων

2.5. Grid Search: Εύρεση Βέλτιστου Συνδυασμού Αριθμού Features – Αριθμού Κανόνων

Το σετ των πιθανών χαρακτηριστικών που θα κρατήσουμε κατά το feature subset selection με τον ReliefF είναι NF={5,10,15,20} ενώ το σετ των πιθανών αριθμών κανόνων (clusters εισόδων) του μοντέλου θα είναι NR={5,10,15,20,25}. Για κάθε ένα συνδυασμό αριθμού χαρακτηριστικών – αριθμού κανόνων στο grid, θα αναζητήσουμε το μέσο σφάλμα εκπαίδευσης στο validation set (validation error) και μέσω αυτού θα καταλήξουμε στο βέλτιστο συνδυασμό ή βέλτιστο σημείο στο grid (optimum grid point).

Για την ακριβέστερη εύρεση του optimum grid point θα εφαρμόσουμε 5-fold cross-validation κατά την εκπαίδευση. Έτσι, για κάθε grid point, το validation error θα προκύψει ως ο μέσος όρος των (τελικών) validation errors για κάθε ένα από τα 5 folds.

2.5.1 Cross Validation

Ακολούθως, θα ξεκινήσουμε την αναζήτηση πλέγματος (grid search) υλοποιώντας 5-fold cross validation στο dataset. Έτσι, για κάθε ένα από τα πέντε folds του dataset και για κάθε ένα από τους 20 συνδυασμούς, grid points, θα

εκπαιδεύσουμε ένα TSK μοντέλο χρησιμοποιώντας την υβριδική μέθοδο και συγκεκριμένα τις συναρτήσεις *genfis()* και *anfis()* του MatLab, όπου ως training και validation set σε κάθε fold θα δίνονται **subsets του αρχικού training set** που προκύπτουν με τη βοήθεια της συνάρτησης *cvpartition()* του MatLab.

2.5.2 Αποτελέσματα – Optimum Grid Point

Αρχικά δίνουμε ένα index σε κάθε grid point από το 1 έως το 20, σκανιάροντας τα grid points κατά τα NF, δηλ. 1 ↦ (NF(1),NR(1)), 2 ↦ (NF(1),NR(2)), κ.ο.κ . Για κάθε ένα από τα cross-validation runs κάνουμε τα εξής:

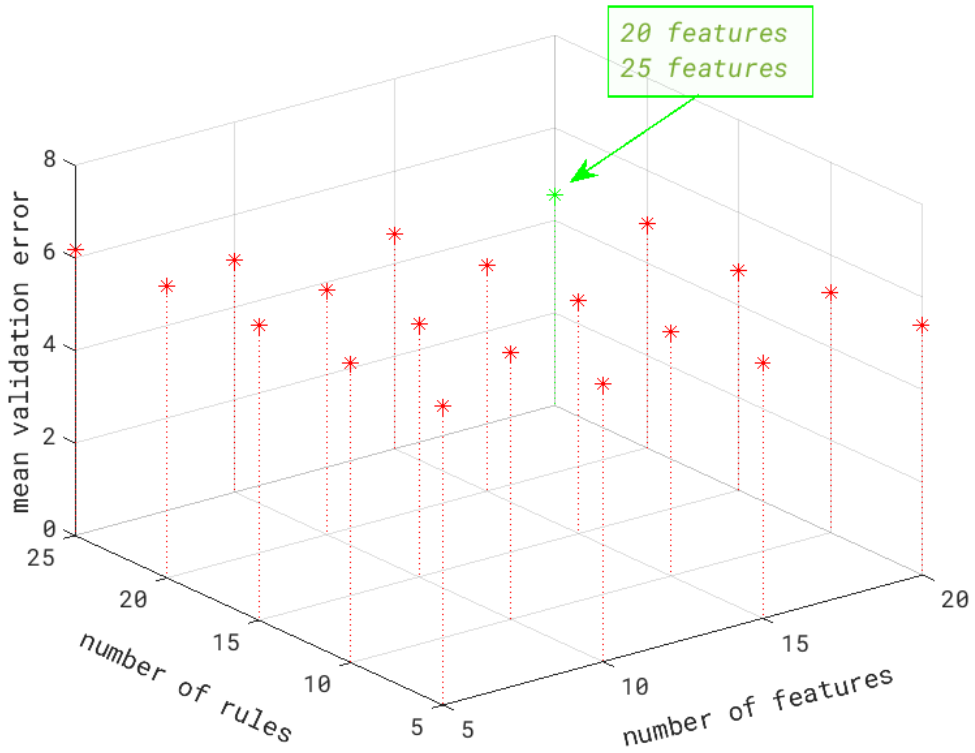
1. Εξάγουμε από το αρχικό training set, το training set και το test set που θα χρησιμοποιηθούν σε αυτό το fold, με βάση τα indices της *cvpartition()* με πρώτο όρισμα τις labels και με την επιλογή *Stratify=true*
2. Για κάθε grid point, κάνουμε τα εξής:
    - Εκπαιδεύουμε με βάση το training set του fold ένα TSK μοντέλο με FCM στο χώρο των εισόδων με παράμετρο τον ζητούμενο αριθμό κανόνων/clusters
    - Λαμβάνουμε το τελικό (ως προς τα epochs) validation error του μοντέλου στο test set του fold

Έχοντας τα τελικά validation errors των grid points για κάθε cross-validation run, υπολογίζουμε το μέσο όρο αυτών και κατόπιν επιλέγουμε σαν βέλτιστο το grid point του οποίου ο μέσος όρος των (τελικών) validation errors των folds είναι ελάχιστος. Παρακάτω, παρατίθεται ο μέσος όρος των τελικών validation errors για κάθε grid point:

NR → ↓ NF	**5**	**10**	**15**	**20**	**25**
5	6.4750	6.4682	6.3830	6.3129	6.1821
10	5.9954	5.7600	5.4761	5.2710	5.0127
15	5.5090	5.2915	5.0364	4.8799	4.6554
20	5.4105	5.1961	4.7593	4.8519	4.5412

Πίνακας 17: Μέσο (τελικό) validation error για κάθε grid point ως προς τα 5 folds

Ενώ παρακάτω δίνουμε ένα 3D plot των μέσων όρων των τελικών validation errors ως προς τα runs (folds) του cross validation για κάθε grid point:



Εικόνα 27: 3D plot του μέσου (τελικού) validation error για κάθε grid point ως προς τα 5 folds

Όπως φαίνεται από τον Πίνακα 16 καθώς και από την Εικόνα 27, παραπάνω, ο βέλτιστος συνδυασμός αριθμού χαρακτηριστικών – αριθμού κανόνων (grid point) είναι το (20,25) και άρα:

$$\mathbf{nf_{opt} = 20\ features}$$
$$\mathbf{nr_{opt} = 25\ rules}\ \text{(clusters)}$$

Επίσης, από τα παραπάνω, φαίνεται ότι μείωση του αριθμού των χρησιμοποιούμενων χαρακτηριστικών (features) οδηγεί σε μεγαλύτερα μέσα validation errors, κάτι απολύτως αναμενόμενο εάν λάβουμε υπόψη την πολύ υψηλή διαστασιμότητα του dataset. Είναι επομένως λογικό ότι ο βέλτιστος συνδυασμός θα περιέχει το μεγαλύτερο από τους πιθανούς αριθμούς χαρακτηριστικών, δηλ. 20, κάτι που συμβαίνει. Αντίστοιχα, όπως επίσης φαίνεται στον Πίνακα 16 παραπάνω, για σταθερό αριθμό features, αύξηση του αριθμού των clusters/κανόνων οδηγεί σε μείωση του (mean) validation error. Έτσι, όπως αναμένεται, θα επιλεγεί ο μέγιστος αριθμός κανόνων, δηλ. 25.

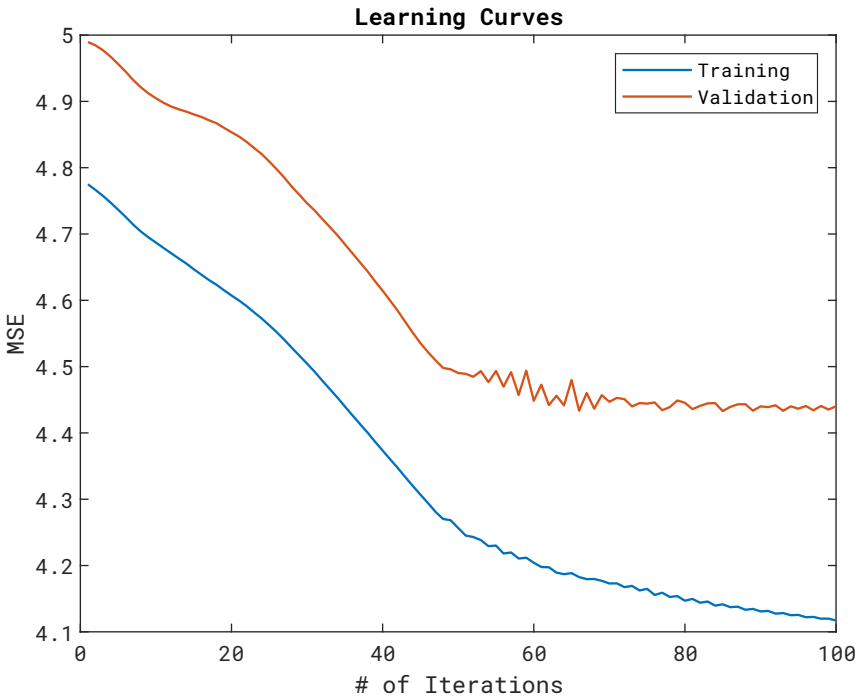
2.6. Τελικό Μοντέλο με βάση το Optimum Grid Point

Έχοντας καταλήξει στο βέλτιστο συνδυασμό αριθμού χαρακτηριστικών – αριθμού κανόνων (grid point), προχωράμε στην επιλογή των 20 χαρακτηριστικών με βάση τον ReliefF και κατόπιν στην εκπαίδευση του τελικού μοντέλου στο οποίο θα θέλαμε να υπάρχουν 25 clusters στον Fuzzy C-Means (FCM). Με παράμετρο

25 λοιπόν θα εκπαιδεύσουμε το τελικό μοντέλο, ενώ θα κρατήσουμε μόνο τα 20 πιο σημαντικά features στο dataset και άρα και στα παραχθέντα subsets.

2.6.1 Απόδοση Τελικού Μοντέλου

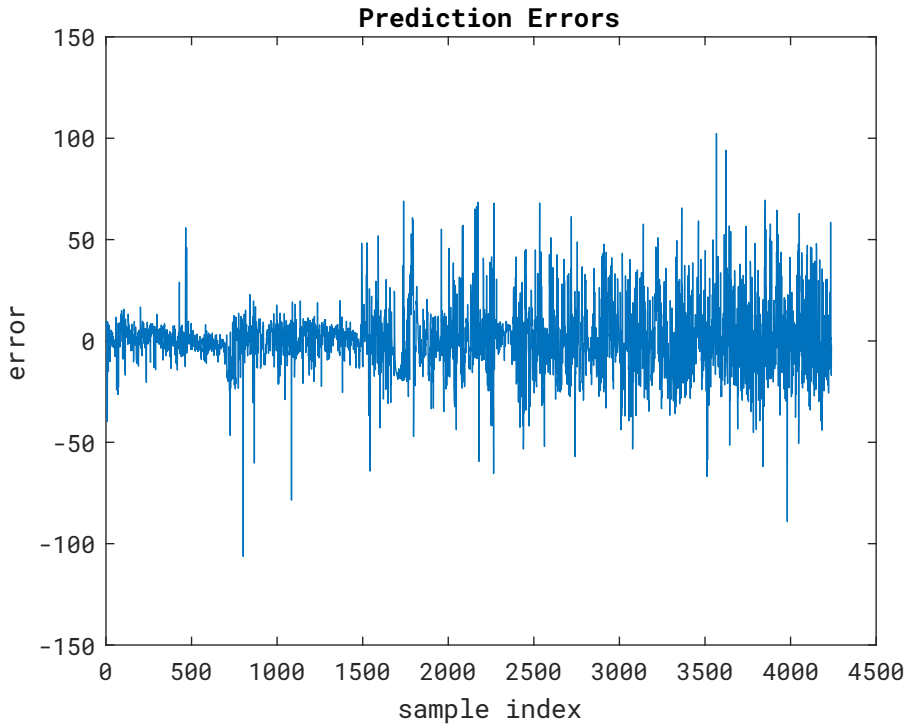
Παρακάτω, δίνονται οι καμπύλες μάθησης (learning curves) του τελικού TSK μοντέλου με τον βέλτιστο συνδυασμό χαρακτηριστικών – κανόνων:



Εικόνα 28: Καμπύλες μάθησης τελικού TSK μοντέλου με overfitting (epochs=100)

όπου όπως φαίνεται, για τον χρησιμοποιούμενο αριθμό epochs (100) δεν υπάρχει το φαινόμενο της υπερεκπαίδευσης (overfitting).

Ακολούθως, σε ότι αφορά την απόδοση του τελικού μοντέλου, δίνεται το διάγραμμα με τα σφάλματα πρόβλεψης του μοντέλου ως προς τις πραγματικές τιμές του test set:



Εικόνα 29: Σφάλματα πρόβλεψης κατά την εφαρμογή του τελικού TSK μοντέλου στο test set

Τέλος δίνονται ο confusion matrix και οι μετρικές απόδοσης του τελικού μοντέλου στο classification task στο testing subset:

Pr → ↓ Act	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}
C_1	12	3	4	2	0	3	3	2	6	9	1	3	1	1	1	2	1	2	0	1	2	0	1	0	0	0
C_2	4	3	1	11	6	10	5	6	2	5	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_3	9	12	15	9	8	5	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_4	9	21	14	7	0	2	3	1	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
C_5	3	4	2	4	3	5	6	4	6	8	5	2	3	3	1	0	0	0	0	0	0	0	0	0	0	1
C_6	0	0	1	0	1	3	6	4	6	7	10	6	2	4	3	1	1	1	2	0	1	0	0	0	0	0
C_7	0	0	0	0	1	3	3	7	9	9	8	5	2	9	2	0	0	1	1	0	0	0	0	0	0	0
C_8	0	0	0	0	1	2	7	15	6	8	5	4	6	2	0	1	0	0	2	0	0	0	1	0	0	0
C_9	0	0	0	0	0	1	6	7	6	10	13	6	2	2	4	1	0	1	0	1	0	0	0	0	0	0
C_{10}	0	0	0	0	0	1	0	3	5	14	9	12	6	3	1	1	0	3	1	0	1	0	0	0	0	0
C_{11}	0	0	0	0	0	0	0	0	0	8	12	16	8	5	2	3	3	1	0	2	0	0	0	0	0	0
C_{12}	0	0	0	0	0	0	0	0	0	4	4	13	13	13	4	2	4	1	1	0	0	0	1	0	0	0
C_{13}	0	0	0	0	1	0	0	1	3	4	9	12	11	7	2	4	2	2	1	0	0	1	0	0	0	0
C_{14}	0	0	0	0	0	1	1	0	2	2	19	8	17	5	1	1	0	1	2	0	0	0	0	0	0	0
C_{15}	0	0	0	0	0	1	2	1	3	4	6	12	7	10	5	6	2	1	0	0	0	0	0	0	0	0
C_{16}	0	0	0	0	0	0	0	1	0	1	5	2	15	9	11	7	4	3	1	0	0	1	0	0	0	0
C_{17}	0	0	0	0	0	0	1	1	0	1	3	11	4	11	7	4	8	5	2	1	0	1	0	0	0	0
C_{18}	0	0	0	0	0	0	0	1	2	2	1	4	6	6	10	10	4	4	5	2	1	1	1	0	0	0
C_{19}	0	0	0	0	0	0	0	0	1	1	0	2	3	6	3	8	7	4	10	7	4	2	0	0	0	2
C_{20}	0	0	0	0	0	1	0	0	1	0	2	4	3	9	9	2	5	13	5	3	2	0	0	0	1	0
C_{21}	0	0	0	0	0	0	0	0	1	1	0	1	2	2	7	5	6	6	11	7	2	3	4	2	0	0
C_{22}	0	0	0	0	1	0	0	1	2	0	0	0	0	1	3	4	3	4	6	6	6	4	6	8	3	2
C_{23}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	4	2	3	1	8	9	7	9	8	3	5
C_{24}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	3	3	8	4	8	6	3	6	4	11
C_{25}	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	3	1	2	4	5	5	6	10	20
C_{26}	0	0	0	0	0	0	0	0	0	1	1	0	2	1	3	7	1	3	3	9	3	11	2	4	3	6

Πίνακας 18: Confusion Matrix του τελικού TSK μοντέλου

Cl→ ↓Met	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}
OA	**0.1257 (12.57 %)**																									
PA	0.2	0.1	0.3	0.1	0.1	0.1	0.1	0.3	0.1	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.0	0.1	0.2	0.1	0.2	0.1
UA	0.3	0.1	0.4	0.2	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.1	0.1	0.2	0.1	0.0	0.1	0.3	0.2	0.4	0.1
$\hat{k}$	0.0907 (9.07 %)																									

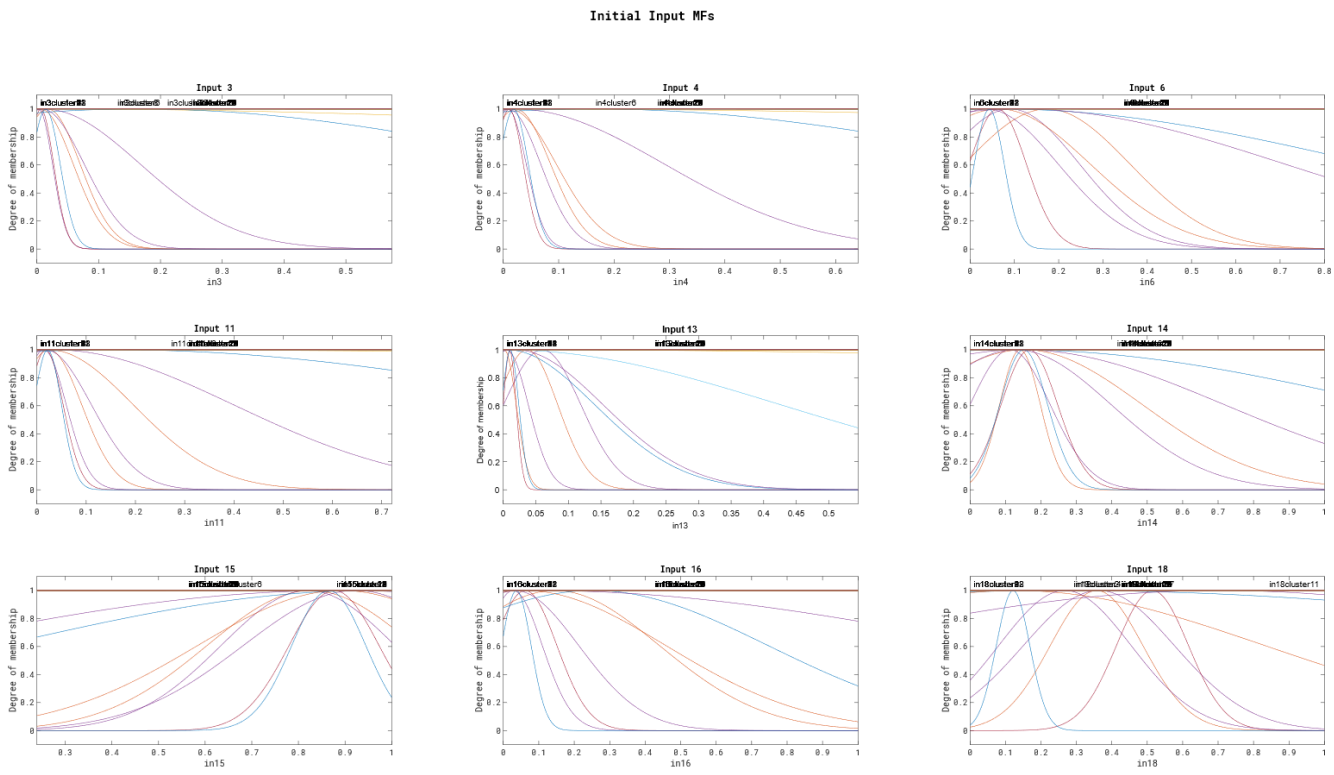
Πίνακας 19: Μετρικές Απόδοσης του τελικού TSK μοντέλου

Όπως φαίνεται από τα παραπάνω δεν μπορούμε να πούμε να πούμε ότι το τελικό εκπαιδευμένο TSK μοντέλο αποδίδει ικανοποιητικά στο classification task με

Overall Accuracy (OA) μόλις 12.57%. Ωστόσο πρέπει να λάβουμε υπόψη το γεγονός ότι για την εκπαίδευσή του χρησιμοποιείται μόλις το (περίπου) 3% των features του αρχικού dataset που αν και επιλέγονται τα πιο «σημαντικά» features και πάλι είναι σχετικά λίγα. Επίσης, πρέπει να ληφθεί υπόψη ότι ο αριθμός των χρησιμοποιούμενων ασαφών κανόνων που χρησιμοποιούνται με βάση το scatter partitioning είναι μόλις 25, πολύ λιγότεροι από αυτούς που θα απαιτούνταν για grid partitioning.

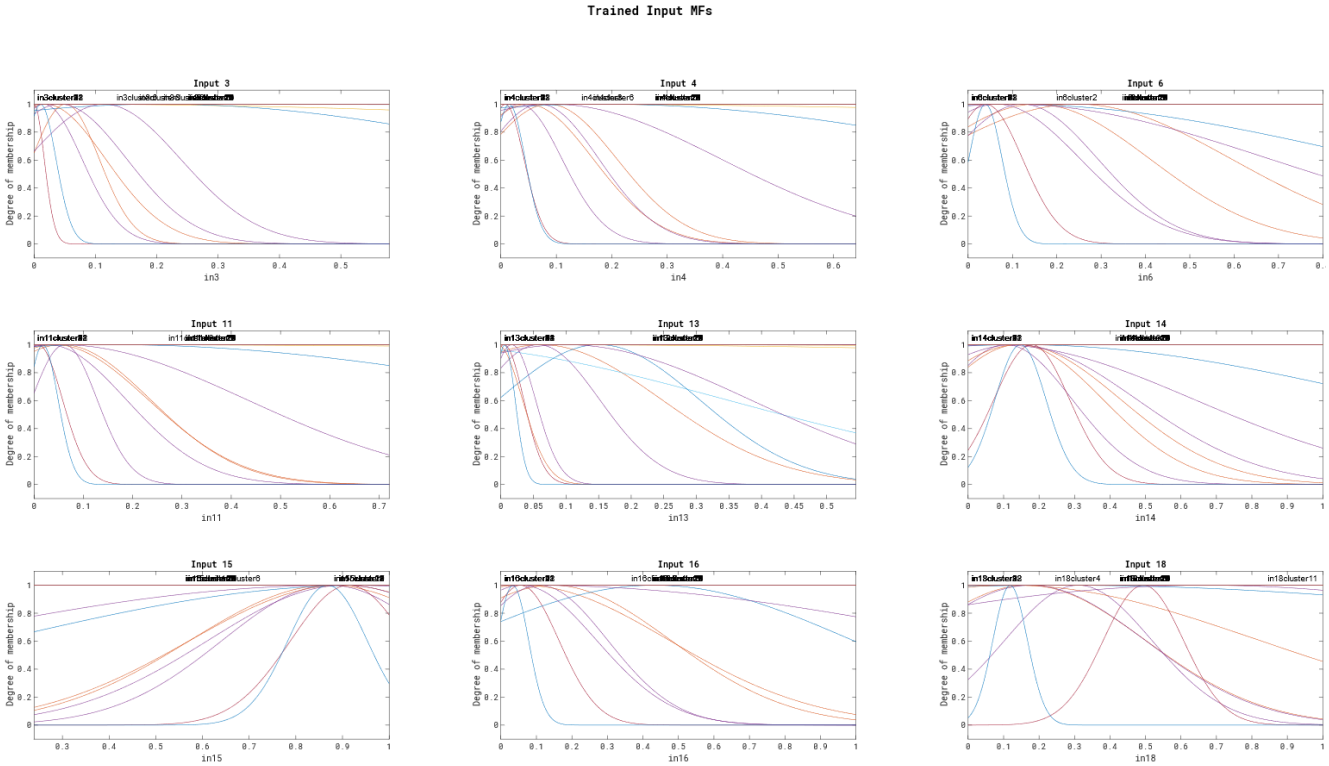
2.6.2 Συναρτήσεις Συμμετοχής (MFs) Τελικού Μοντέλου

Παρακάτω, δίνονται οι αρχικές συναρτήσεις συμμετοχής (MFs) των 25 λεκτικών τιμών - clusters για τις 20 εισόδους - features του τελικού TSK μοντέλου (έχουν επιλεγεί τυχαία 9 είσοδοι για απεικόνιση):



Εικόνα 30: Αρχική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου τελικού μοντέλου

ενώ ακολούθως δίνονται οι τελικές μορφές των παραπάνω MFs:



Εικόνα 31: Τελική μορφή συναρτήσεων συμμετοχής μεταβλητών εισόδου τελικού μοντέλου

2.7. Συμπερασματικές Παρατηρήσεις – Σχόλια

Παρατηρούμε ότι για το δοσμένο dataset η ανάπτυξη ενός FNN κρατώντας μόνο τα 20 (σημαντικότερα) από τα συνολικά 617 χαρακτηριστικά οδηγεί σε όχι αρκετά ικανοποιητικά αποτελέσματα ταξινόμησης (classification). Αυτό φαίνεται στους δείκτες απόδοσης που δίνονται στον Πίνακα 18 παραπάνω και ειδικά στους δείκτες OA και $\hat{k}$. Προφανώς, οι δείκτες απόδοσης αυτοί είναι χειρότεροι σε σύγκριση με τους αντίστοιχους του μη-υψηλής διαστασιμότητας dataset του 1ου μέρους, κάτι απολύτως αναμενόμενο αν λάβουμε υπόψη το feature subset selection που έλαβε μέρος στο dataset αυτού του μέρους της εργασίας. Επίσης, η χρήση μόνο 25 κανόνων σε σύγκριση με τους 2^{20} (για 2 λεκτικές τιμές) ή 3^{20} (για 3) που θα απαιτούνταν για grid partitioning, φανερώνει το πλεονέκτημα της χρήσης ομαδοποίησης στο χώρο των εισόδων (scatter partitioning), χειροτερεύει όμως την απόδοση του μοντέλου.

Τέλος, για λόγους πληρότητας, παραθέτουμε τα 20 πιο σημαντικά features που επιλέχθηκαν με τον ReliefF για την εκπαίδευση του τελικού μοντέλου:

Indices των 20 πιο σημαντικών features																			
419	418	417	416	480	581	391	426	392	582	415	390	389	433	430	585	395	425	414	422

Πίνακας 20: Indices των 20 πιο σημαντικών features από ReliefF (K=100)

Θεσσαλονίκη, 28/2/2020