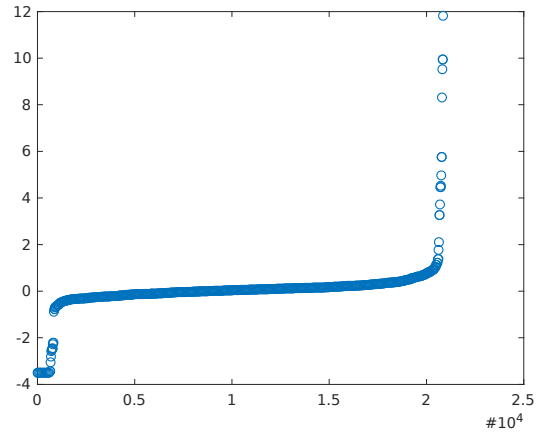
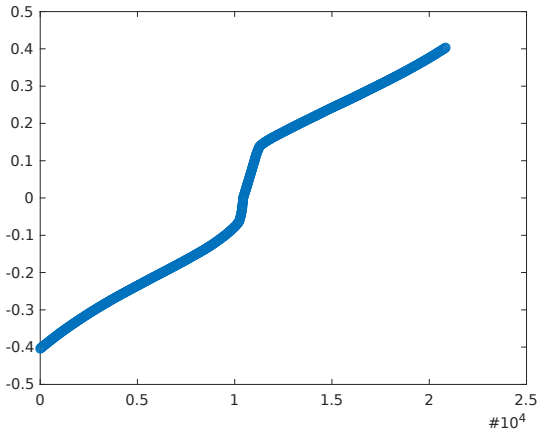
1. Εφαρμογή σε απλό dataset

1.1. Φόρτωση & Προ-επεξεργασία dataset

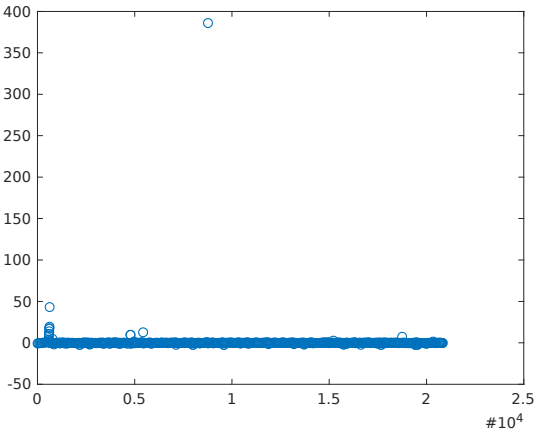
Το dataset του ερωτήματος, *avila*, αποδείχθηκε αρκετά “δύσκολο” dataset από την άποψη ότι έχει αρκετά σημεία τα οποία θα μπορούσαν να θεωρηθούν outliers. Για ισχυροποίηση του παραπάνω, παραθέτονται τα plots κάποιων features του dataset:



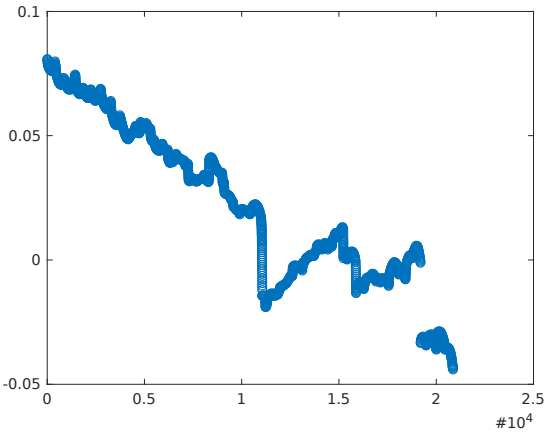
Εικ. 1: Feature 1 (αρχικό)



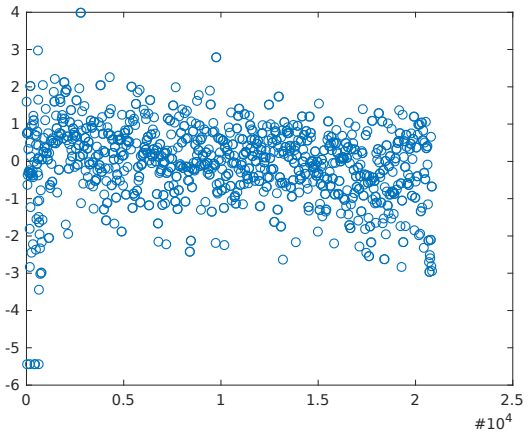
Εικ. 2: Feature 1 (smoothed, Smoothing Factor = 0.5)



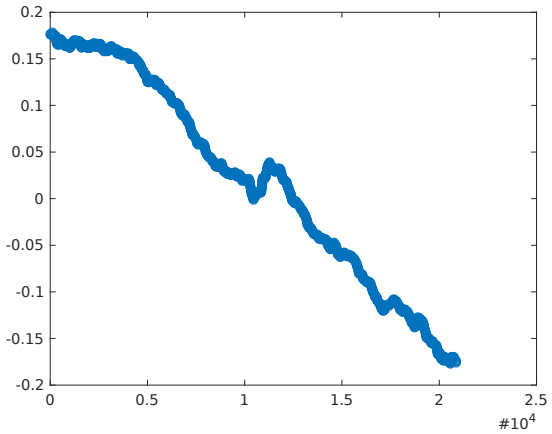
Εικ. 3: Feature 2 (αρχικό)



Εικ. 4: Feature 2 (smoothed, Smoothing Factor = 0.75)



Εικ. 5: Feature 4 (αρχικό)



Εικ. 6: Feature 4 (smoothed, Smoothing Factor = 0.5)

Στη δεξιά στήλη του παραπάνω πίνακα φαίνεται η βασική μέθοδος 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 χρήσιμοποίηκαν για κάθε feature του dataset. Αυτοί είναι αποτέλεσμα trial-and-error μεθοδολογίας και πιθανότατα δεν είναι βέλτιστοι. Αξίζει να σημειωθεί, ωστόσο, ότι χωρίς smoothing στο dataset δεν θα ήταν δυνατή η εκτέλεση του πρώτου μέρους της εργασίας καθώς κατά την ομαδοποίηση των training data points (clustering) προέκυπταν συνεχώς clusters με ένα στοιχείο, κάτι που απαγόρευε την εκπαίδευση του αντίστοιχου κανόνα του αρχικού FIS. Δοκιμάστηκε επίσης και **normalization** στο dataset ωστόσο δεν απέφερε καμία βελτίωση της απόδοσης του εκάστοτε μοντέλου (από τα πέντε που δοκιμάστηκαν).

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 η 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. Έτσι, οι συχνότητες εμφάνισης της κάθε κλάσης σε καθένα από τα τρία subsets, χρησιμοποιώντας την παραπάνω λογική, δίνεται παρακάτω:

Class	Dataset	Training	Validation	Testing
1	0.41079	0.41086	0.41074	0.41074
2	0.00047923	0.00047923	0.00047927	0.00047927
3	0.009872	0.0099042	0.010065	0.0095854
4	0.033785	0.033786	0.033549	0.034028
5	0.10495	0.10495	0.10496	0.10496
6	0.188	0.1881	0.18787	0.18787
7	0.042795	0.042652	0.043134	0.042895
8	0.049792	0.04984	0.049844	0.049605
9	0.079695	0.079633	0.079559	0.080038
10	0.0042651	0.0042332	0.0043134	0.0043134
11	0.050031	0.05016	0.049844	0.049844
12	0.025543	0.025399	0.025641	0.025641

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:

	Training	Validation	Testing
Training	0	0.0004827	0.0004058
Validation	0.0004827	0	0.0004793
Testing	0.0004058	0.0004793	0

1.3. Εύρεση SC ακτίνων από αριθμό κανόνων

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

NR	*5**	8	12	16	20
RADII	**1.0**	**0.853125**	**0.7**	**0.5875**	**0.525**

*: Το ζητούμενο σετ κανόνων αρχικά ήταν {4,8,12,16,20}. Ωστόσο, για το συγκεκριμένο dataset και τη συγκεκριμένη μέθοδο ομαδοποίησης για το partitioning (SC) δεν είναι δυνατή η επίτευξη μοντέλου fis με λιγότερους από nr'=5 ασαφείς κανόνες (το οποίο αντιστοιχεί σε ακτίνα radii=1). Έτσι, το τελικό σετ κανόνων θα είναι {**5**,8,12,16,20}.

1.4. Εκπαίδευση TSK μοντέλων

1.4.1 Model 1: NR = 5

Confusion Matrix (στήλες = predicted, γραμμές = actual):

28	201	665	450	186	147	23	8	6	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0
0	0	5	12	7	16	0	0	0	0	0	0
0	0	4	26	9	90	13	0	0	0	0	0
0	6	21	57	103	153	90	8	0	0	0	0
0	26	265	234	133	102	18	0	3	3	0	0
0	5	56	62	20	20	14	2	0	0	0	0
0	0	46	57	56	30	10	8	0	0	0	0
0	0	18	22	45	68	67	49	48	17	0	0
0	0	0	0	18	0	0	0	0	0	0	0
0	0	1	21	30	76	68	12	0	0	0	0
0	0	10	13	11	55	12	6	0	0	0	0

Metrics:

OA = 0.08

PA = [0.016,0,0.125,0.183,0.235,0.13,0.078,0.038,0.143,0,0,0]

UA = [1,0,0.005,0.027,0.167,0.135,0.044,0.086,0.842,0,NaN,NaN]

1.4.2 Model 2: NR = 8

Confusion Matrix (στήλες = predicted, γραμμές = actual):

91	394	440	409	221	124	23	8	4	0	0	0
0	0	0	2	0	0	0	0	0	0	0	0
0	1	9	7	15	8	0	0	0	0	0	0
0	4	2	24	61	28	23	0	0	0	0	0
0	16	16	38	103	164	77	16	8	0	0	0
1	62	156	250	158	112	39	2	4	0	0	0

2	18	36	37	37	37	12	0	0	0	0	0
0	7	26	47	53	40	20	14	0	0	0	0
0	6	4	26	22	54	38	113	57	14	0	0
0	0	0	0	10	8	0	0	0	0	0	0
0	0	4	10	40	72	37	25	20	0	0	0
0	6	0	14	13	42	23	3	6	0	0	0

Metrics:

OA = 0.1011

PA = [0.053,0,0.225,0.169,0.235,0.143,0.067,0.068,0.171,0,0,0]

UA = [0.968,0,0.013,0.028,0.141,0.163,0.041,0.077,0.576,0,NaN,NaN]

1.4.3 Model 3: NR = 12

Confusion Matrix (στήλες = predicted, γραμμές = actual):

91	394	440	409	221	124	23	8	4	0	0	0
0	0	0	2	0	0	0	0	0	0	0	0
0	1	9	7	15	8	0	0	0	0	0	0
0	4	2	24	61	28	23	0	0	0	0	0
0	16	16	38	103	164	77	16	8	0	0	0
1	62	156	250	158	112	39	2	4	0	0	0
2	18	36	37	37	37	12	0	0	0	0	0
0	7	26	47	53	40	20	14	0	0	0	0
0	6	4	26	22	54	38	113	57	14	0	0
0	0	0	0	10	8	0	0	0	0	0	0
0	0	4	10	40	72	37	25	20	0	0	0
0	6	0	14	13	42	23	3	6	0	0	0

Metrics:

OA = 0.1011

PA = [0.053,0,0.225,0.169,0.235,0.143,0.067,0.068,0.171,0,0,0]

UA = [0.968,0,0.013,0.028,0.141,0.163,0.041,0.077,0.576,0,NaN,NaN]