Ασαφή Συστήματα 2018

Ομάδα 3

Regression – Series 06

ΕΙΣΗΓΗΤΗΣ: ΘΕΟΧΑΡΗΣ Ι.

Κοσμάς Τσιάκας
ΑΕΜ: 8255

10/10/2018

Περιεχόμενα

Σχήματα

Περιγραφή του προβλήματος

Στόχος αυτής της εργασίας είναι να μελετηθεί η ικανότητα των TSK μοντέλων για την επίλυση προβλημάτων regression. Θα χρησιμοποιηθούν δύο διαφορετικά σετ δεδομένων, για τα οποία γνωρίζοντας τα χαρακτηριστικά κάποιων δειγμάτων θα δημιουργηθεί μια συνάρτηση η οποία θα προσεγγίζει όσο τον δυνατόν καλύτερα την πραγματική απεικόνιση των στοιχείων της εισόδου στην έξοδο του συστήματος.

Εφαρμογή στο CCPP dataset

To Cobined Cycle Power Plant dataset περιέχει 9568 δείγματα, κάθε ένα από τα οποία χαρακτηρίζεται από 4 features, την μέση ωριαία θερμοκρασία (T), την μέση ωριαία πίεση (AP), την μέση ωριαία σχετική υγρασία (RH) και την μέση ωριαία exhaust vacuum (V). Χρησιμοποιώντας αυτά τα δεδομένα, προσπαθούμε να προβλέψουμε την ενεργειακή απόδοση του σταθμού.

Αρχικά, πραγματοποιούμε διαχωρισμό του σετ δεδομένων σε τρία μη επικαλυπτόμενα υποσύνολα:

- 60% : σύνολο εκπαίδευσης – training data
- 20% : σύνολο επικύρωσης – validation data
- 20% : σύνολο ελέγχου – check data

Επίσης, κανονικοποιούμε τις τιμές του dataset με τον εξής τρόπο: $x=2\left(\frac{(x-x_{min})}{(x_{max}-x_{min})}\right)-1$, όπου οι min και max τιμές αφορούν αποκλειστικά τα training data και με βάση αυτά κανονικοποιούμε και τα validation και check data.

Αυτό το κάνουμε, καθώς τα δεδομένα βρίσκονται σε αρκετά διαφορετικό εύρος και για το λόγο αυτό υπάρχει περίπτωση να μην υλοποιείται ορθά η εκπαίδευση του δικτύου. Μετά την κανονικοποίηση όλα τα χαρακτηριστικά βρίσκονται στο εύρος -1 έως 1.

Για την εκπαίδευση, θα χρησιμοποιησουμε τέσσερα διαφορετικά TSK μοντέλα, όπως φαίνονται στον παρακάτω πίνακα.

	Number of membership functions	**Output format**
TSK_model_1	2	Singleton
TSK_model_2	3	Singleton
TSK_model_3	2	Polynomial
TSK_model_4	3	Polynomial

Η εκπαίδευση γίνεται με την υβριδική μέθοδο, δηλαδή οι παράμετροι των συναρτήσεων συμμετοχής βελτιστοποιούνται με backpropagation και οι παράμετροι της πολυωνυμικής συνάρτησης εξόδου βελτιστοποιούνται με Least Squares Method.

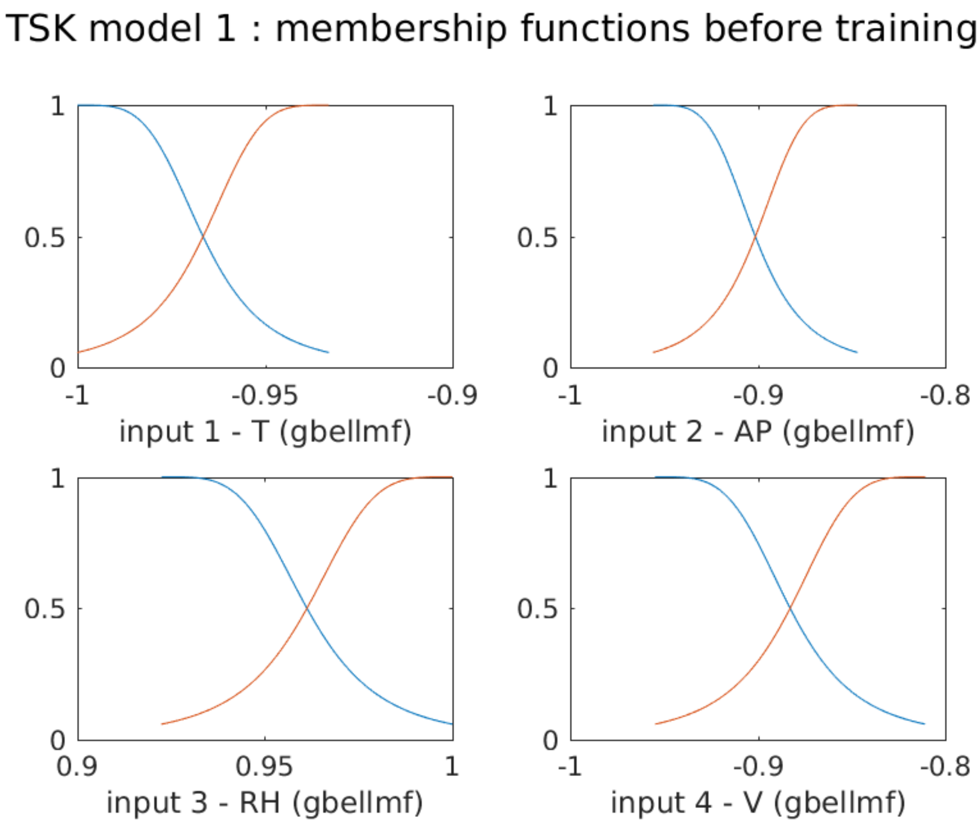
TSK model 1

Αρχικά, δημιουργούμε στο MATLAB το fuzzy inference system με τις κατάλληλες επιλογές, όσον αφορά την δημιουργία των συναρτήσεων συμμετοχής, μέσω της συνάρτησης *genfis1*.

Τα μοντέλα Singleton περιγράφονται με κανόνες της μορφής

$$R(i) = IF\ x1\ is\ A1\ AND \dots AND\ xm\ is\ Am\ THEN\ y\ is\ w$$

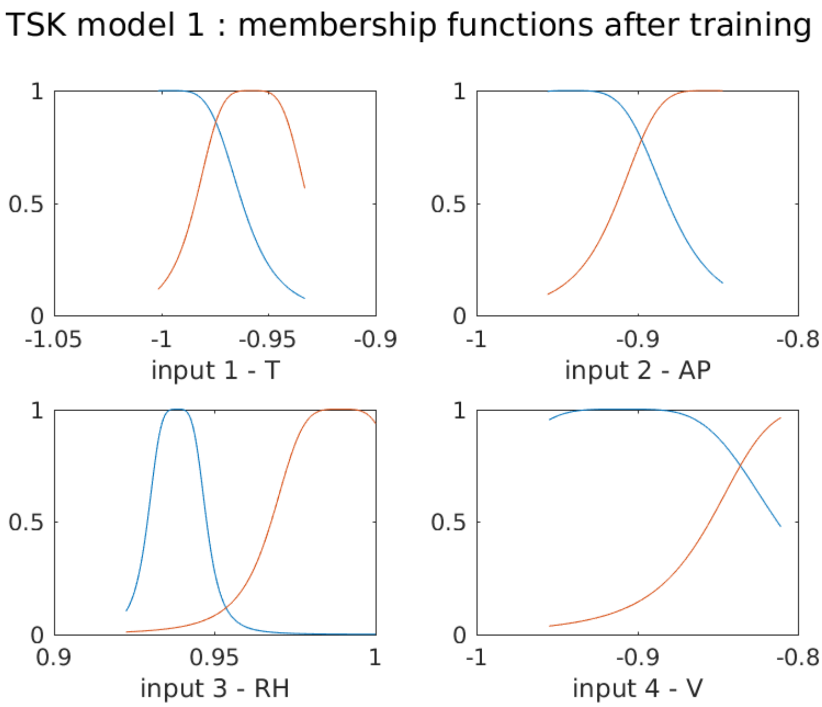
Στο πρώτο μοντέλο TSK οι συναρτήσεις συμμετοχής τύπου *gbellmf* με επικάλυψη 0.5 για κάθε μεταβλητή εισόδου φαίνονται στο παρακάτω σχήμα:



Σχήμα 1: TSK model 1 - Συναρτήσεις συμμετοχής

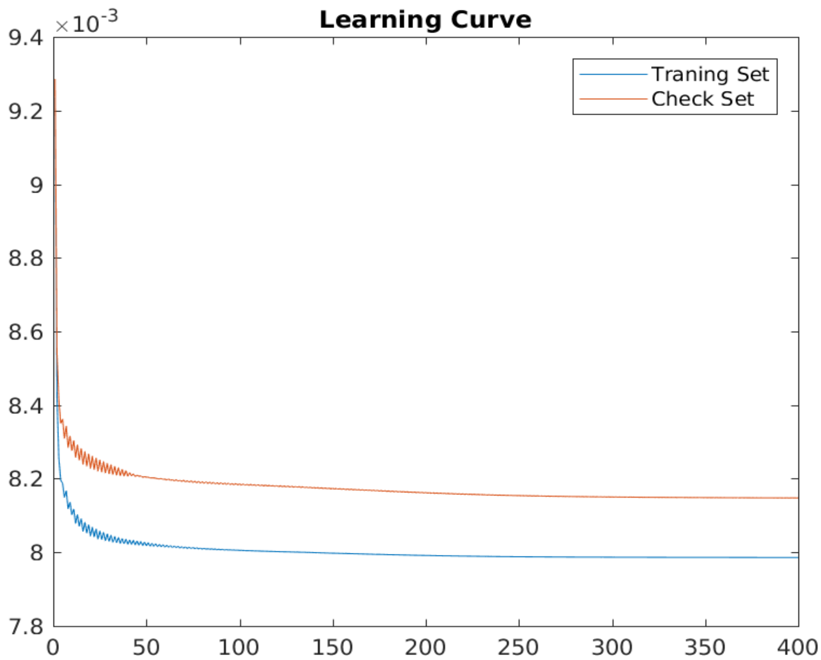
Στην συνέχεια, ρυθμίζουμε το σύστημα χρησιμοποιώντας τα δεδομένα εκπαίδευσης για την κατάλληλη επιλογή των τιμών στις παραμέτρους της συνάρτησης συμμετοχής και τα δεδομένα επικύρωσης ως παράμετρο στην *anfis*, ώστε η εκπαίδευση να συνεχίζεται χωρίς να φτάνουμε σε υπερεκπαίδευση. Η εκπαίδευση γίνεται για 400 εποχές και προκύπτουν τα παρακάτω αποτελέσματα, όπως φαίνονται στα διαγράμματα.

Αρχικά, οι συναρτήσεις συμμετοχής μετά την εκπαίδευση παίρνουν την εξής μορφή:

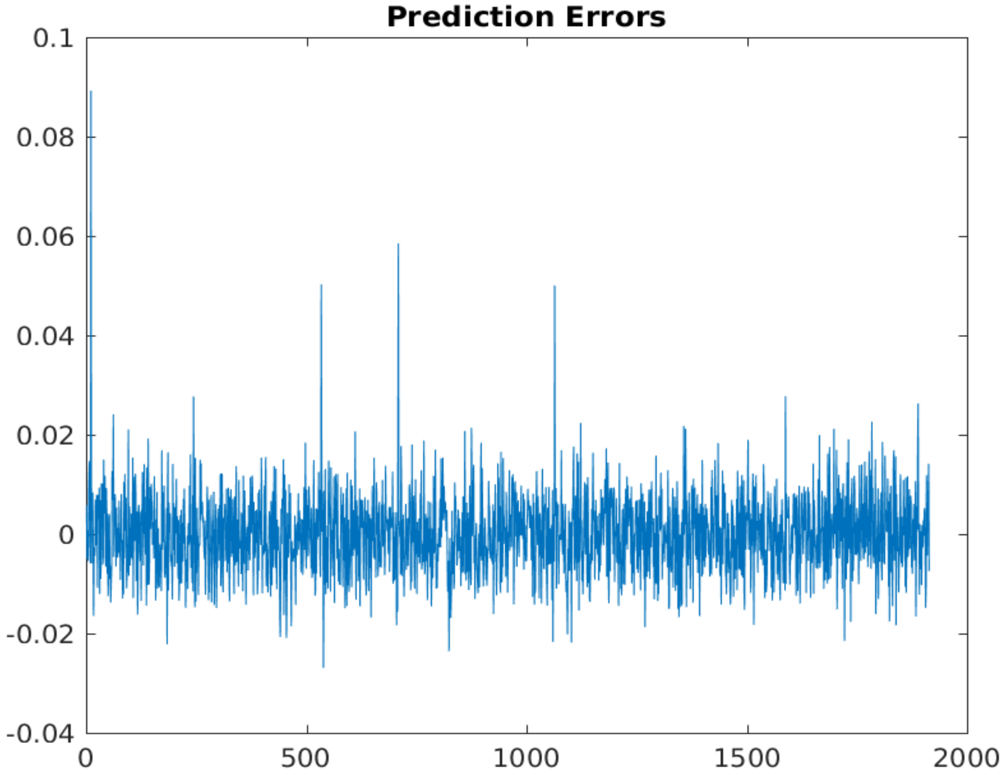


Σχήμα 2: TSK model 1 - Συναρτήσεις συμμετοχής μετά την εκπαίδευση

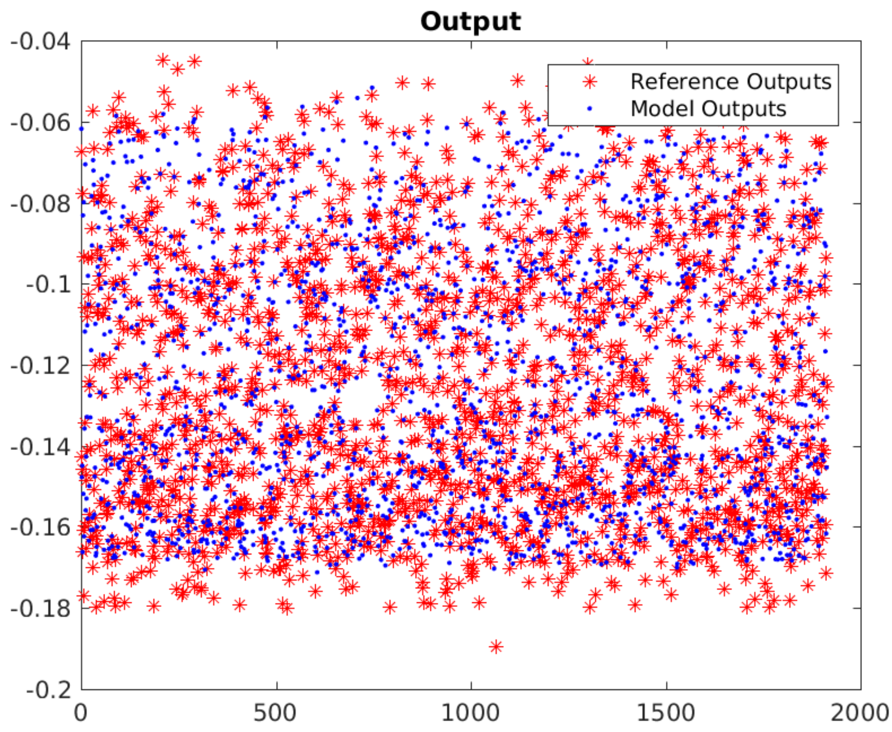
Οι καμπύλες εκμάθησης και σφάλματος φαίνονται παρακάτω:



Σχήμα 3: TSK model 1 - Καμπύλη εκμάθησης



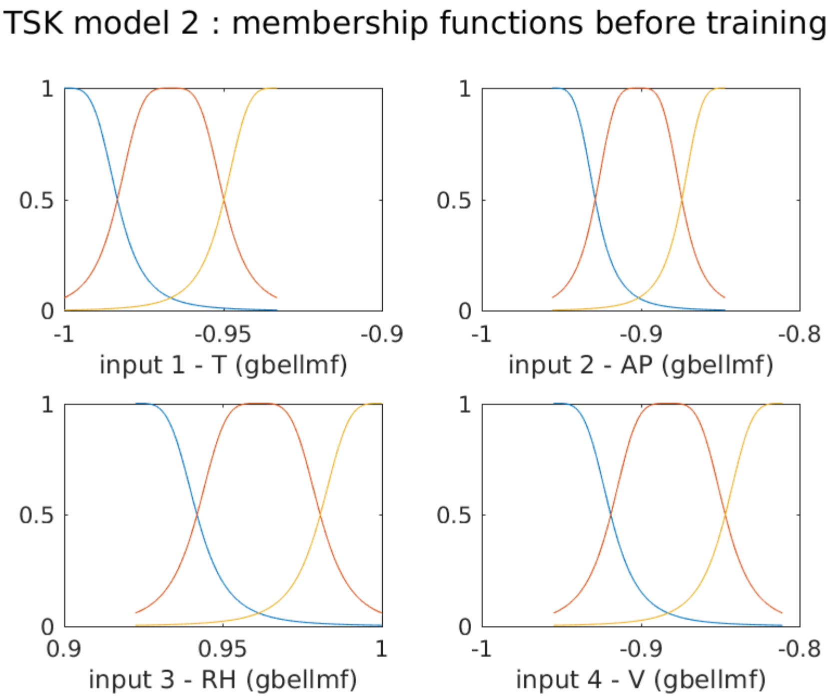
Σχήμα 4: TSK model 1 - Σφάλματα πρόβλεψης



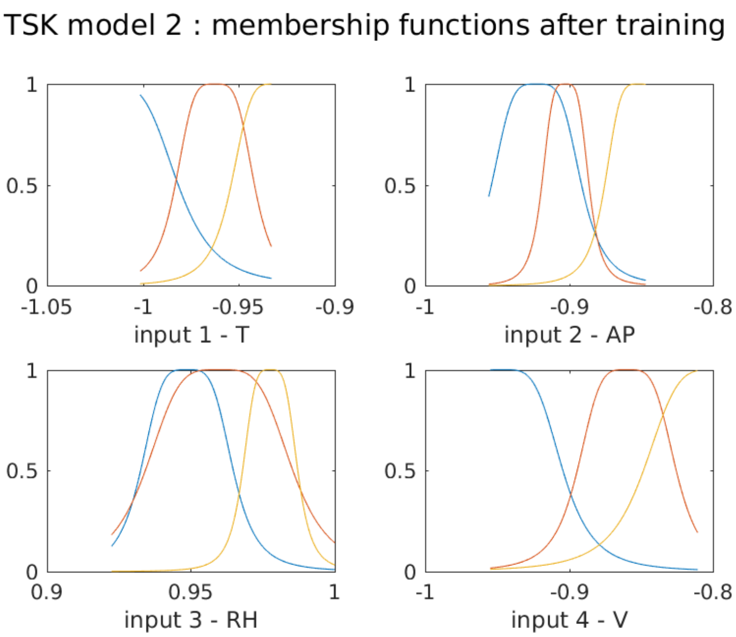
Σχήμα 5: TSK model 1 - Reference vs Model outputs

TSK model 2

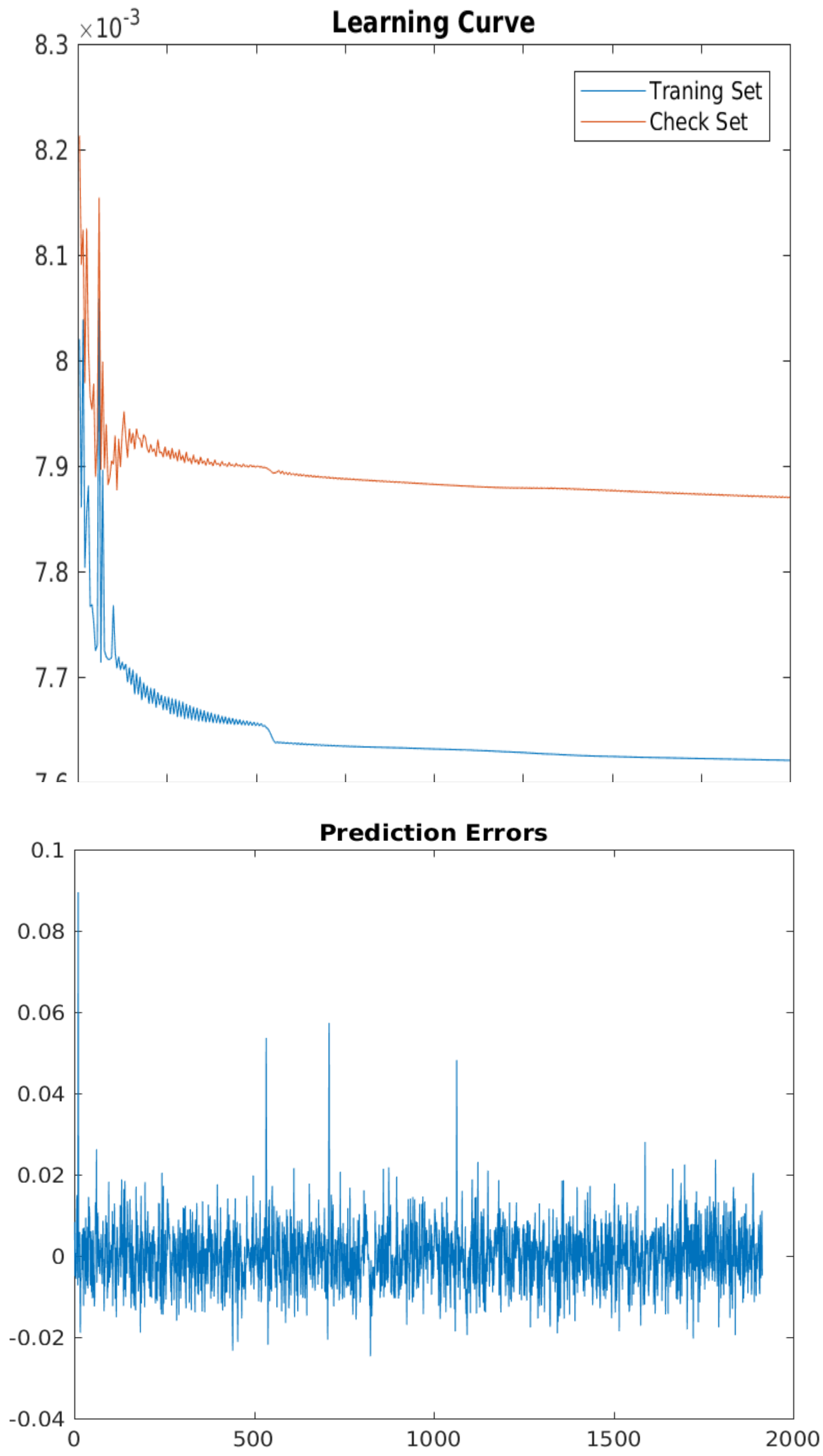
Εκτελούμε την ίδια διαδικασία για το 2° TSK μοντέλο με τρεις συναρτήσεις συμμετοχής για κάθε χαρακτηριστικό και singleton έξοδο και προκύπτουν τα παρακάτω αποτελέσματα.



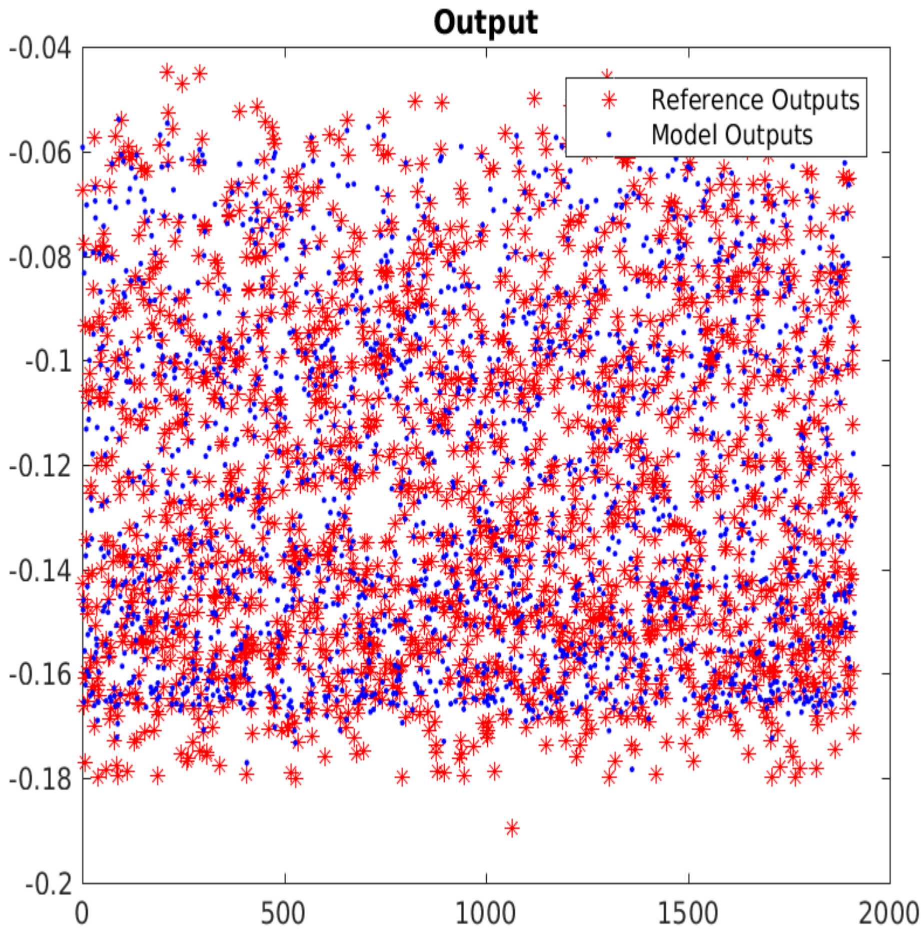
Σχήμα 6: TSK model 2 - Συναρτήσεις συμμετοχής πριν την εκπαίδευση



Σχήμα 7: TSK model 2 - Συναρτήσεις συμμετοχής μετά την εκπαίδευση



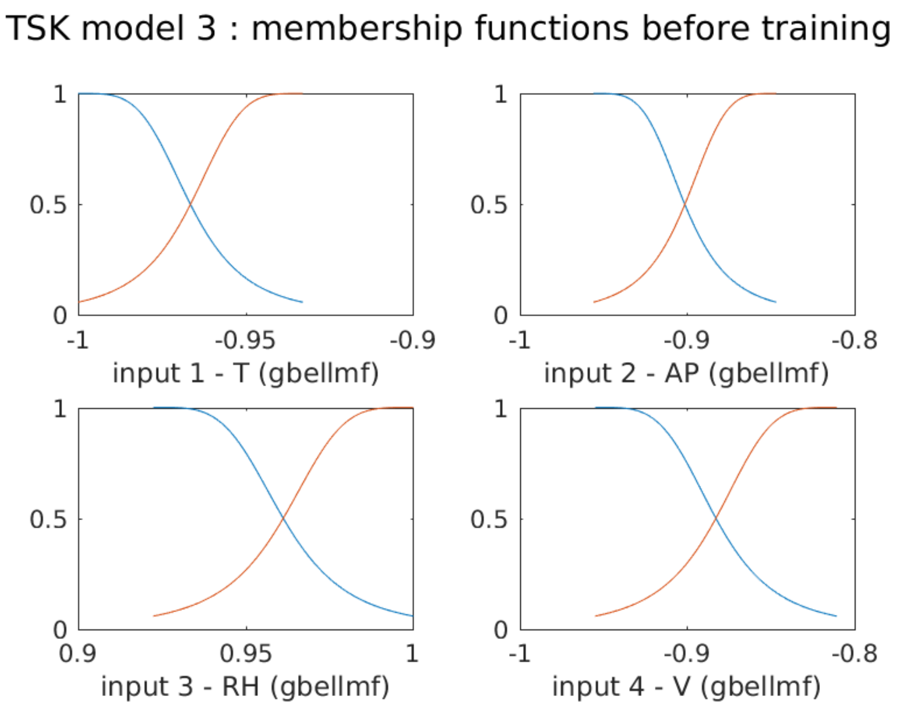
Σχήμα 9: TSK model 2 - Σφάλματα πρόβλεψης



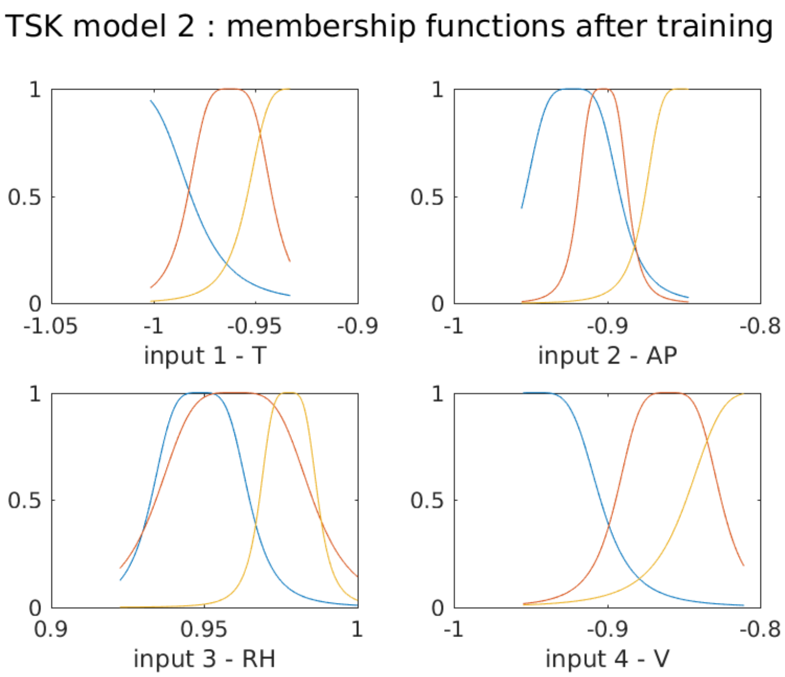
Σχήμα 10: TSK model 2 - Reference vs Model outputs

TSK model 3

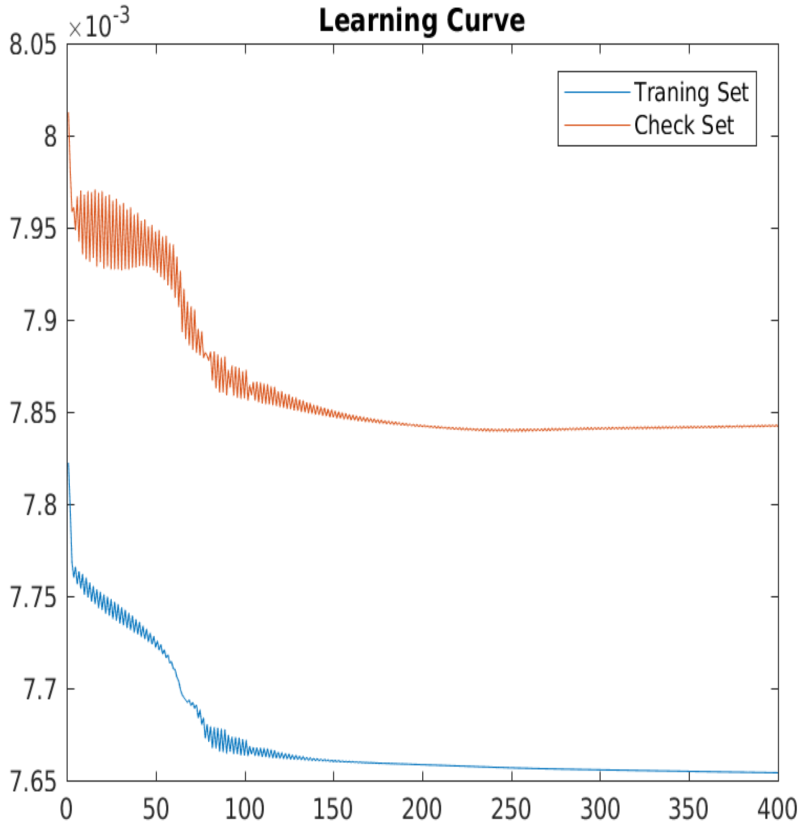
Εκτελούμε την ίδια διαδικασία για το 3ο TSK μοντέλο με δύο συναρτήσεις συμμετοχής για κάθε χαρακτηριστικό και γραμμική έξοδο, και παίρνουμε τα εξής αποτελέσματα.



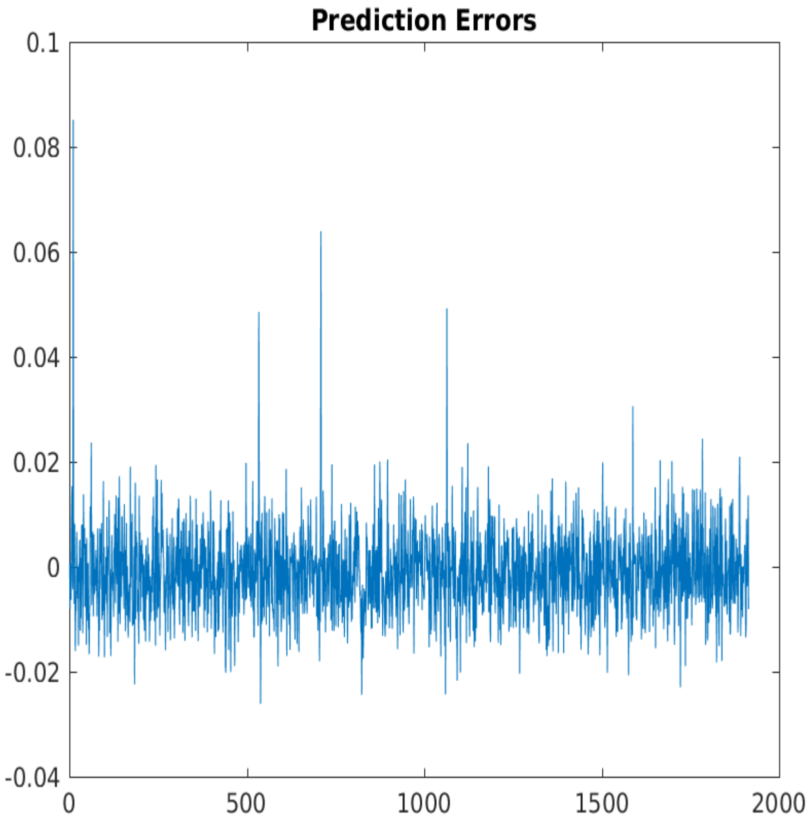
Σχήμα 11: TSK model 3 - Συναρτήσεις συμμετοχής πριν την εκπαίδευση



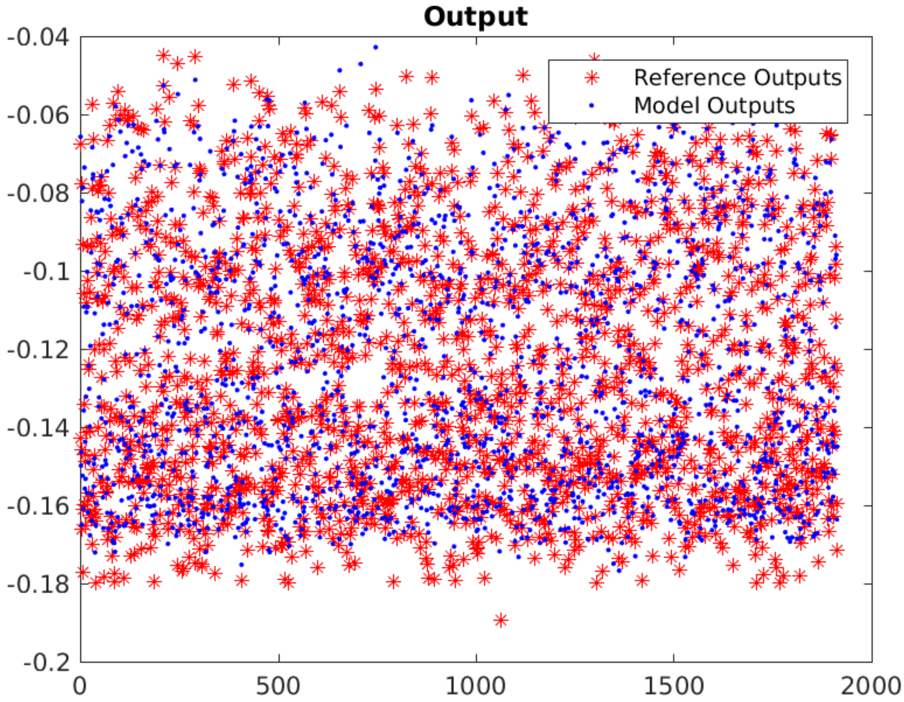
Σχήμα 12: TSK model 3 - Συναρτήσεις συμμετοχής μετά την εκπαίδευση



Σχήμα 13: TSK model 3 - Καμπύλη εκμάθησης



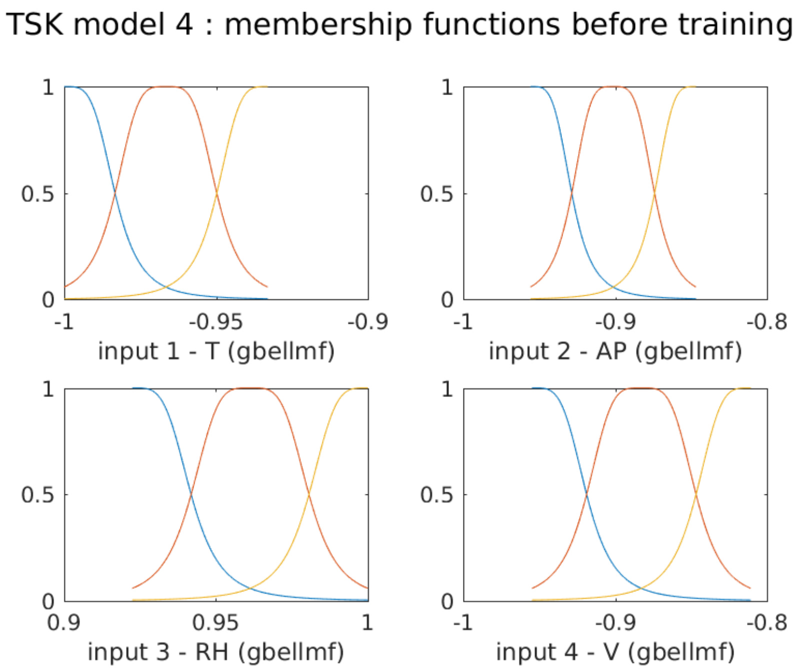
Σχήμα 14: TSK model 3 - Σφάλματα πρόβλεψης



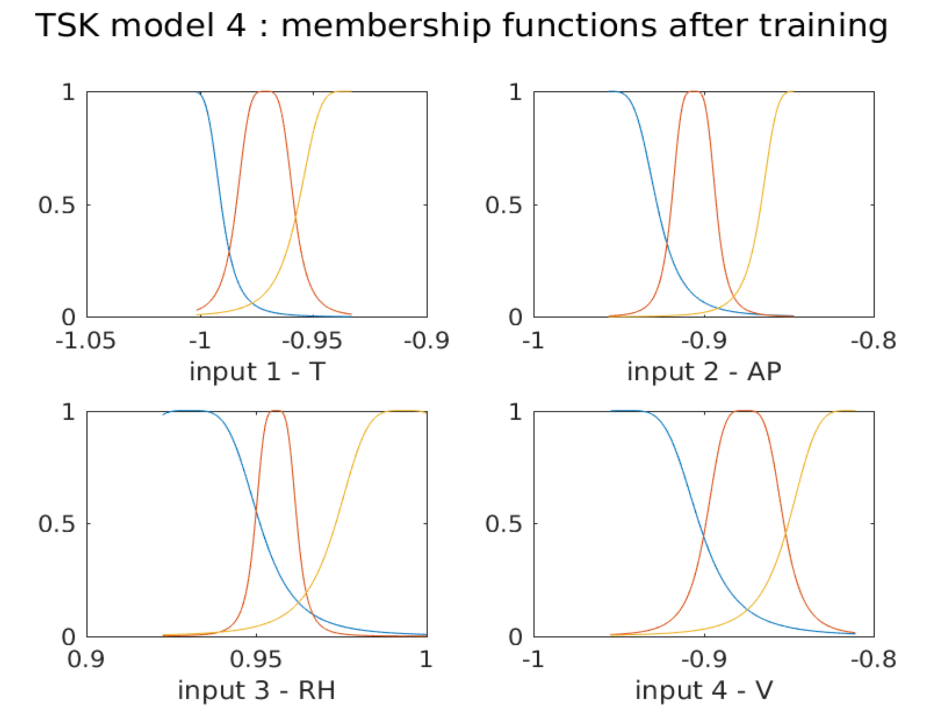
Σχήμα 15: TSK model 3 - Reference vs Model outputs

TSK model 4

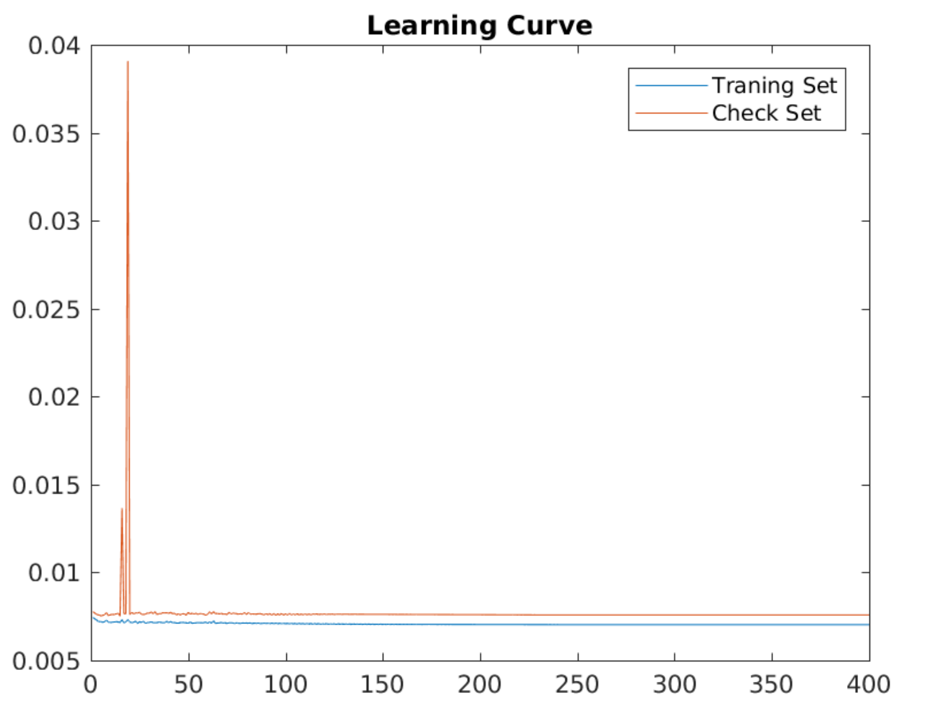
Εκτελούμε την ίδια διαδικασία για το 4ο TSK μοντέλο με τρεις συναρτήσεις συμμετοχής για κάθε χαρακτηριστικο και γραμμική έξοδο και παίρνουμε τα εξής αποτελέσματα.



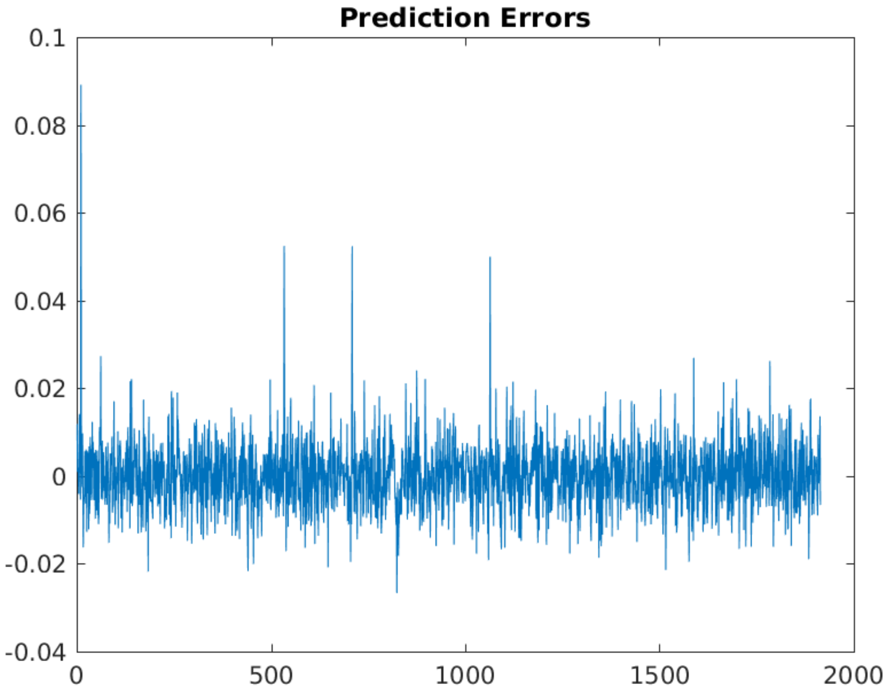
Σχήμα 16: TSK model 4 - Συναρτήσεις συμμετοχής πριν την εκπαίδευση



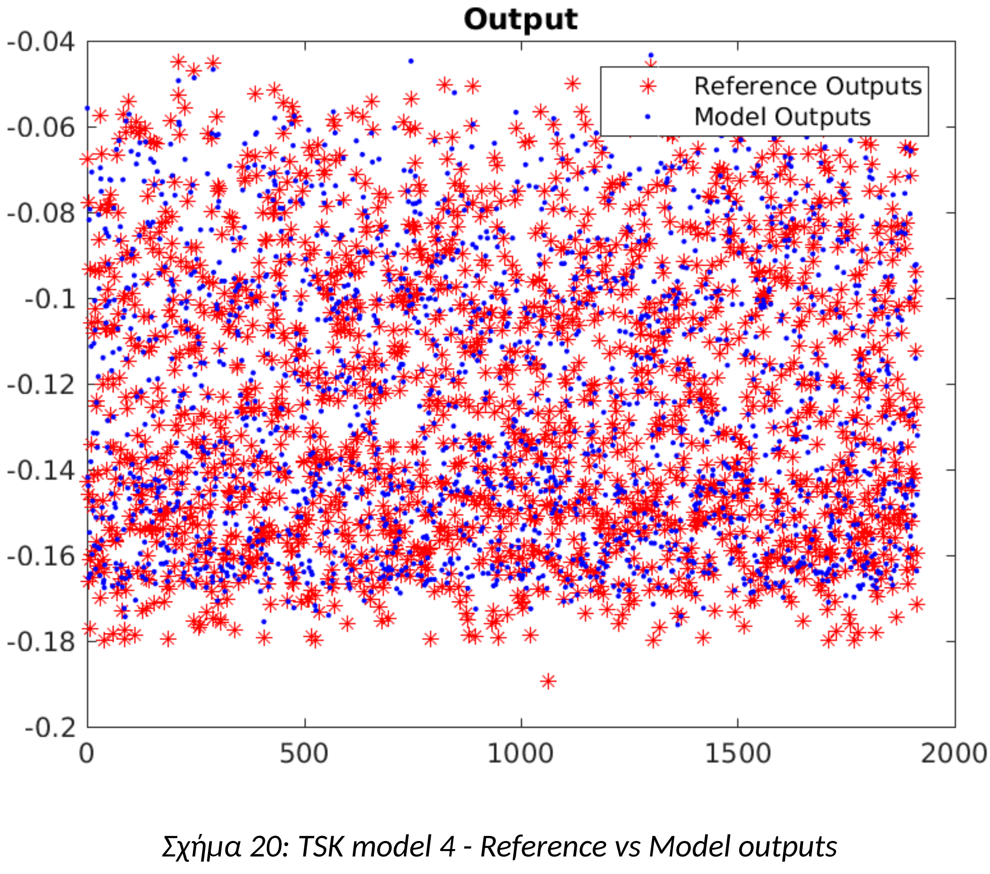
Σχήμα 17: TSK model 4 - Συναρτήσεις συμμετοχής μετά την εκπαίδευση



Σχήμα 18: TSK model 4 - Καμπύλη εκμάθησης



Σχήμα 19: TSK model 4 - Σφάλματα πρόβλεψης



Σχήμα 20: TSK model 4 - Reference vs Model outputs

Δείκτες απόδοσης & συμπεράσματα

Στον παρακάτω πίνακα παρουσιάζονται οι δείκτες απόδοσης $RMSE, NMSE, NDEI\ και\ R^2$.

TSK model	Number of membership functions	Output format	MSE	RMSE	NMSE	NDEI	R^2	Time (sec)
1	2	Singleton	0.000070	0.008346	0.064068	0.253117	0.935932	49.28
2	3	Singleton	0.000065	0.008038	0.059420	0.243762	0.940577	471.38
3	2	Polynomial	0.000066	0.008104	0.060402	0.245768	0.939598	133.91
4	3	Polynomial	0.000062	0.007877	0.057057	0.238867	0.942934	1798.6

Βλέπουμε ότι τα μοντέλα είναι πολύ κοντά μεταξύ τους στην εκτίμηση που κάνουν με βάση τις μετρικές που έχουν παραχθεί. Για το μοντέλο με τις περισσότερες συναρτήσεις συμμετοχής και πολυωνυμική έξοδο, το μέσο τετραγωνικό σφάλμα είναι μικρότερο, όπως και ο συντελεστής προσδιορισμού R^2 είναι πιο κοντά στη μονάδα, που σημαίνει ότι βρίσκεται πιο κοντά στην τέλεια εκτίμηση. Παρόλα αυτά γίνεται αρκετά πιο πολύπλοκο, καθώς ο χρόνος εκτέλεσης του είναι 35 φορές μεγαλύτερος από το πρώτο μοντέλο.

Γενικότερα, η χρήση γραμμικής πολυωνυμικής εξόδου βελτιώνει το αποτέλεσμα, ανεξάρτηση από τον αριθμό των συναρτήσεων συμμετοχής, το οποίο είναι λογικό, καθώς δίνει τη δυνατότητα να χρησιμοποιούνται πιο ακριβή αποτελέσματα στην έξοδο του μοντέλου μας.

Επίσης, στο τελευταίο TSK μοντέλο, βλέποντας την καμπύλη εκμάθησης, βλέπουμε ότι συγκλίνει σχεδόν αμέσως, εκτός από μια μικρή και απότομη απόκλιση που έχει στις πρώτες εποχές εκπαίδευσης. Επίσης, όλα τα μοντέλα συγκλίνουν στο τελικό σφάλμα σε λιγότερο από 150 εποχές, επομένως η εκπαίδευση μέχρι 400 μπορεί να θεωρηθεί και περιττή, καθώς δεν βελτιώνει άλλο το μοντέλο μας και η χρήση του validation set μας βοηθάει στη συνεχή εκπαίδευση χωρίς να φτάνουμε όμως σε overfitting του μοντέλου μηδενίζοντας εντελώς το τελικό σφάλμα.

Εφαρμογή στο Bank dataset

Το Bank dataset πρόκειται για ένα πολύ μεγαλύτερο dataset σε σχέση με το προηγούμενο, καθώς περιέχει 32 διαφορετικά χαρακτηριστικά τα οποία αποσκοπούν στην πρόβλεψης της πιθανότητας ένας πελάτης να αποχωρήσει από την ουρά αναμονής σε μια τράπεζας λόγω της μεγάλης αναμονής. Για το λόγο αυτό, θα επιλέξουμε έναν συγκεκριμένο αριθμό χαρακτηριστικών, των πιο αντιπροσωπευτικών του δείγματος με χρήση του αλγορίθμου ReliefF.

Αρχικά, εφαρμόζουμε ένα **shuffle** στο dataset για να υπάρχει τυχαίοτητα στην σειρά με την οποία υπάρχουν τα δεδομένα και αυτό να μην επηρεάσει το διαχωρισμό τους στη συνέχεια.

Μετά, διαχωρίζουμε το σετ δεδομένων ως εξής:

- 60% : σύνολο εκπαίδευσης – training data
- 20% : σύνολο επικύρωσης – validation data
- 20% : σύνολο ελέγχου – check data

Αρχικά όμως, είναι καλό να εφαρμόσουμε μια **προεπεξεργασία** στα δεδομένα μας, ώστε να αποφευχθούν οι διπλότυπες τιμές, να μην υπάρχουν κενές τιμές, καθώς και να είναι όλα τα δεδομένα στο ίδιο εύρος. Με τον τρόπο αυτό, θα είναι αποτελεσματικότερη, αλλά και ταχύτερη, η εκπαίδευση του δικτύου.

Τρέχοντας την εντολή του MATLAB $unique(Bank, 'rows')$ προκύπτει ένα σετ δεδομένων ίδιο με το αρχικό, επομένως διαπιστώνουμε ότι δεν υπάρχουν διπλότυπες τιμές. Επίσης, δεν υπάρχουν κενές ή NaN τιμές, κάτι που ήταν αναμενόμενο εφόσον το dataset προέκυψε από simulator. Στη συνέχεια, επειδή το εύρος είναι διαφορετικό για κάθε χαρακτηριστικό, θα εφαρμόσουμε **κανονικοποίηση** ώστε όλα να βρίσκονται στο διάστημα [-1, 1]. Στην συγκεκριμένη περίπτωση, κανονικοποιείται και η έξοδος, όμως αυτό δεν επηρεάζει τη συνάρτηση μας και απλά γίνεται το ανάλογο scale στο τελικό error. Έτσι για κάθε στοιχείο του dataset εφαρμόζουμε την εξής πράξη: $x=2\left(\frac{(x-x_{min})}{(x_{max}-x_{min})}\right)-1$, ενώ είναι σημαντικό να τονιστεί ότι αυτή η κανονικοποίηση εφαρμόζεται ξεχωριστά στα τρια διαφορετικά τμήματα του σετ δεδομένων, χρησιμοποιώντας όμως την μέγιστη και την ελάχιστη τιμή του training set, ώστε να βρίσκονται όλα στο ίδιο εύρος μεταξύ τους.

Εφαρμόζουμε τον αλγόριθμο **Relief** μέσω της συνάρτησης του MATLAB και επιλέγουμε να λειτουργεί με αριθμό γειτόνων 100. Η εντολή αυτή αρκεί να εκτελεστεί μία φορά και στη συνέχεια να χρησιμοποιούνται τα αντίστοιχα ορίσματα στο σετ δεδομένων με βάση το αποτέλεσμα της.

Στην συνέχεια, θα εκτελέσουμε 5-fold cross vaildation στα δεδομένα για διαφορετικό αριθμό χαρακτηριστικών και τιμής της μεταβλητής radii, ένα από τα ορίσματα που χρησιμοποιούνται για την δημιουργία των FIS και επηρεάζει το μέγεθος και τον αριθμό των cluster/κανόνων που θα δημιουργηθούν. Θα εξετάσουμε τις περιπτώσεις όπου έχουμε

- features: 3, radii = 0.1, 0.3, 0.5, 0.7, 0.9
- features: 9, radii = 0.5, 0.6, 0.7, 0.9, 1
- features: 15, radii = 0.5, 0.6, 0.7, 0.8, 1
- features: 21, radii = 0.6, 0.7, 0.8, 0.9, 1

Για κάθε διαφορετικό αριθμό από χαρακτηριστικά επιλέγουμε διαφορετικές τιμές για το radii και δεν διατηρούμε τον ίδιο σε κάθε περίπτωση. Αυτό συμβαίνει, γιατί καθώς μεταβάλλεται ο αριθμός των features επηρεάζεται πολύ ο αριθμός των κανόνων, με αποτέλεσμα να δημιούργούνται περιπτώσεις με πολύ μεγάλο αριθμό κανόνων, που είναι αδύνατο να εκτελεστεί σε λογικά χρονικά πλαίσια ή να δημιουργείται μόνο ένας κανόνας, που οδηγεί στην αδυναμία δημιουργίας ασαφούς συστήματος.

Πιο συγκεκριμένα, για λίγα χαρακτηριστικά, θέλουμε μικρές τιμές της ακτίνας, ώστε να δημιουργηθεί ένας λογικός αριθμός κανόνων, ενώ για πολλά χαρακτηριστικά, θέλουμε όσο το δυνατόν μεγαλύτερη ακτίνα. Να σημειωθει ότι το radii κυμαίνεται από 0 μέχρι 1.

Με τις τιμές που επιλέχθηκαν, θα αντιμετωπιστούν όσο τον δυνατόν περισσότερες περιπτώσεις γίνεται, ενώ στη συνέχεια φαίνεται και ο τελικός αριθμός κανόνων που δηιουργείται.

Για κάθε ένα συνδυασμό των παραπάνω τιμών εκτελούμε την παρακάτω διαδικασία:

- Διαχωρίζουμε το training set με χρήση της cvpartition() σε 5 folds και δημιουργούμε έναν πίνακα για την αποθήκευση του σφάλματος σε κάθε επανάληψη. Έτσι, το 80% των δεδομένων αυτών χρησιμοποιείται για εκπαίδευση και το υπόλοιπο 20% αυτών για επικύρωση.
- Δημιουργούμε ένα Fuzzy Inference System με τα δεδομένα αυτά, να λειτουργεί με Subtractive Clustering και την τιμή του radii που επιλέγουμε. Το FIS αυτό θα εκπαιδεύεται σε κάθε fold του αλγορίθμου. Στην περίπτωση που δημιουργείται μόνο ένας κανόνας ή περισσότεροι από 100, συνεχίζουμε στην επόμενη επανάληψη, για να αποφευχθεί σφάλμα του κώδικα ή πολύ μεγάλη διάρκεια εκτέλεσης.
- Για κάθε ένα από τα 5 folds εκτελείται η εκπαίδευση για 50 εποχές και ελέγχουμε την λειτουργία του εκπαιδευμένου συστήματος μας στα validation data του συνολικού dataset που δεν έχουν εμπλακεί στην διαδικασία της εκπαίδευσης.
- Ως σφάλμα κρατάμε το τετραγωνικό άθροισμα των διαφορών σφάλματος. Για να βρούμε το μέσο σφάλμα, προσθέτουμε τα σφάλματα και τα διαιρούμε με το 5, ενώ στη συνέχεια

το διαιρούμε με τον αριθμό των στοιχείων που χρησιμοποιήθηκαν στην εκπαίδευση για να προκύπτει στο Mean Square Error.

- Στο τέλος, αποθηκεύουμε για κάθε μία περίπτωση αριθμού χαρακτηριστικών και ακτίνας το σφάλμα στον πίνακα error_grid και το αποτέλεσμα είναι αυτό που φαίνεται παρακάτω.

Μέσο Τετραγωνικό Σφάλμα

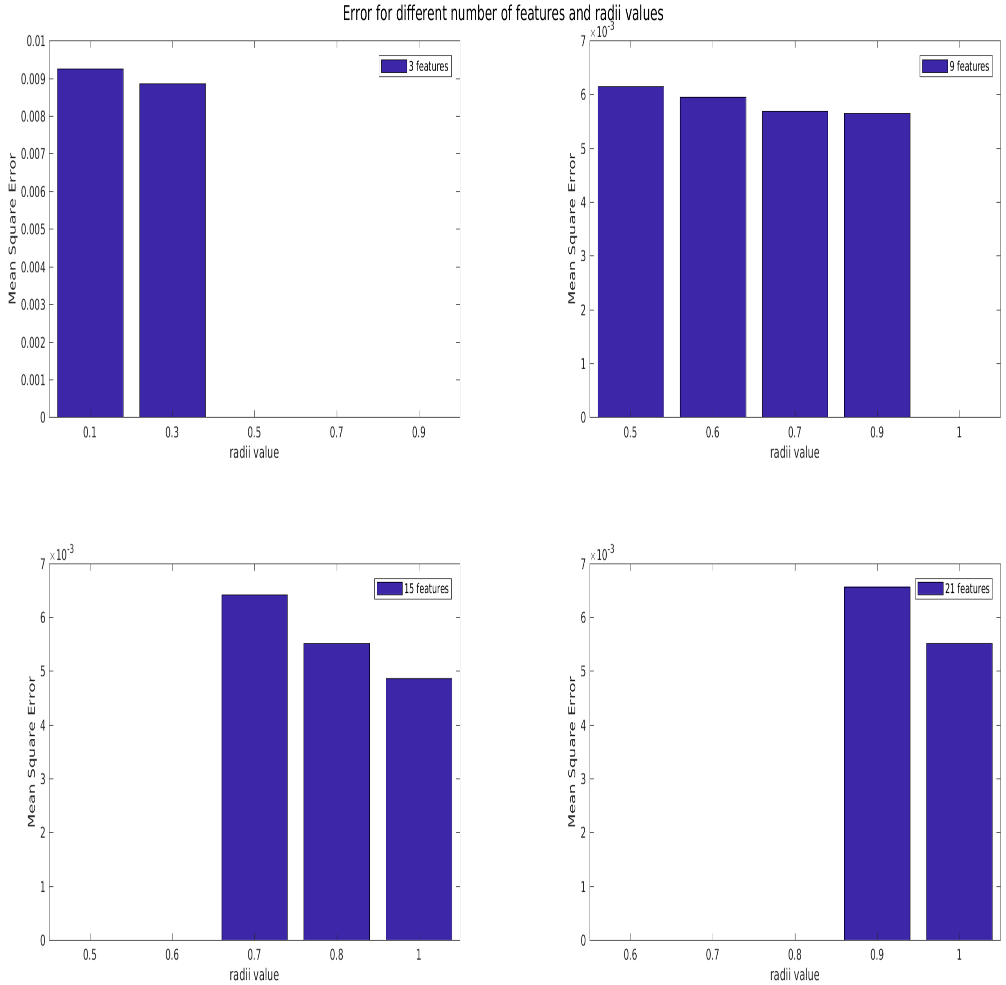
Radii / Features	0.1	0.3	0.5	0.6	0.7	0.8	0.9	1
3	0.0093	0.0089	X		X		X	
9			0.0061	0.0059	0.0057		0.0056	X
15			X	X	0.0064	0.0055		0.0049
21				X	X	X	0.0066	0.0055

Αριθμός κανόνων που δημιουργήθηκαν

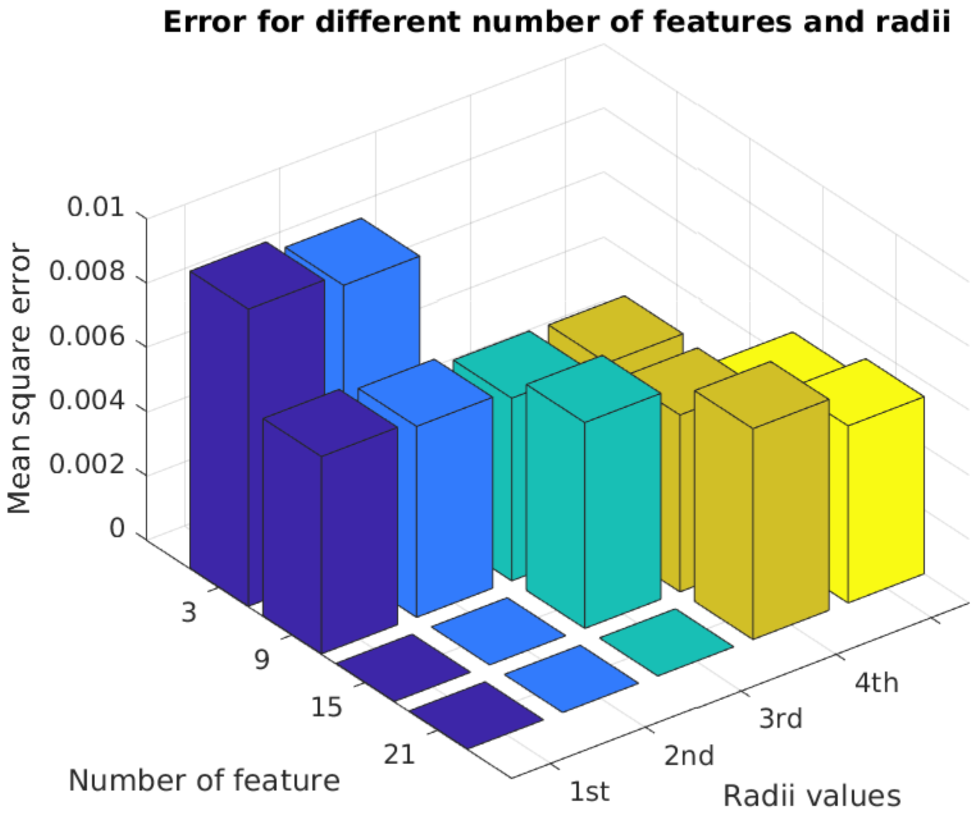
Radii / Features	0.1	0.3	0.5	0.6	0.7	0.8	0.9	1
3	5	2	1		1		1	
9			23	13	6		2	1
15			3208	227	44	18		5
21				4880	3679	632	30	14

Τα πεδία με γκρι είναι αυτά τα οποία δεν δοκιμάστηκε η εκτέλεση του αλγορίθμου, ενώ στα πεδία του σφάλματος που υπάρχει το σύμβολο Χ, σημαίνει ότι δεν εκτελέστηκε ο αλγόριθμος είτε γιατί υπήρχε μόνο ένας κανόνας είτε γιατί υπήρχαν πάρα πολλοί, που δεν επέτρεπαν την εκτέλεση του. Επίσης, δεδομένου ότι η μέθοδος Subtractive Clustering χρησιμοποιείται για την αντιμετώπιση του προβλήματος της διαστασιμότητας, δεν έχει νόημα να τον χρησιμοποιήσουμε για πολύ μεγάλο αριθμό cluster/κανόνων.

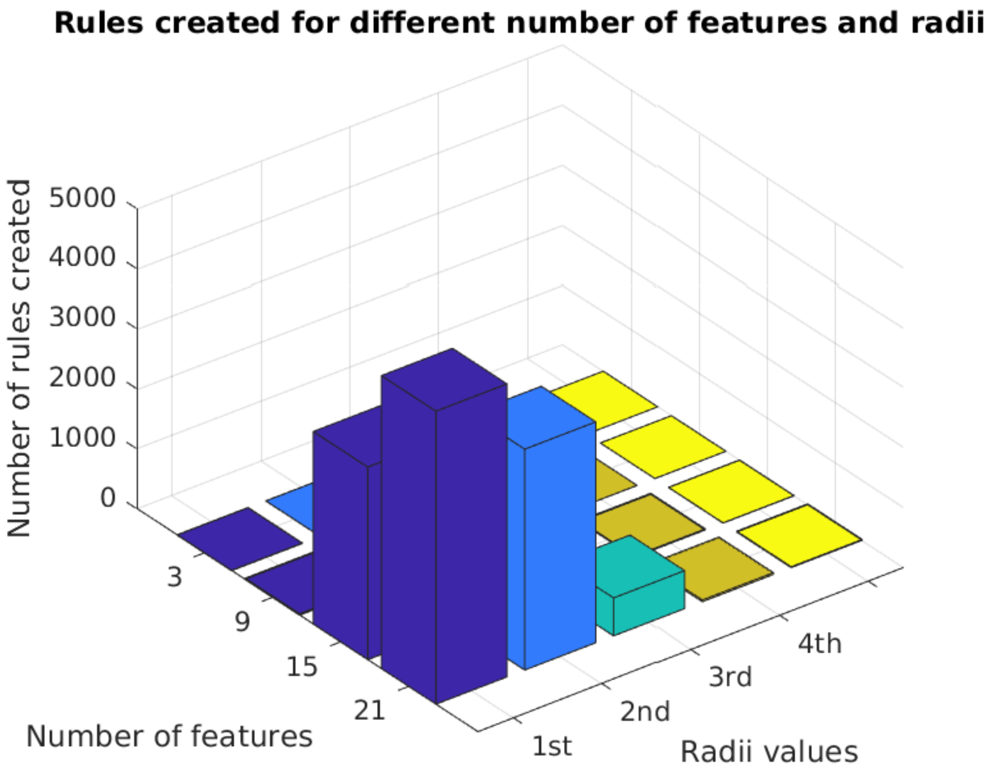
Στο παρακάτω διάγραμμα φαίνεται και καλύτερα η πορεία του σφάλματος για τις διάφορες τιμές χαρακτηριστικών και ακτίνας.



Σχήμα 21: Σφάλμα μοντέλου για διαφορετικές τιμές χαρακτηριστικών και κανόνων



Σχήμα 22: Μέσο τετραγωνικό σφάλμα για διάφορες τιμές χαρακτηριστικών και ακτινών



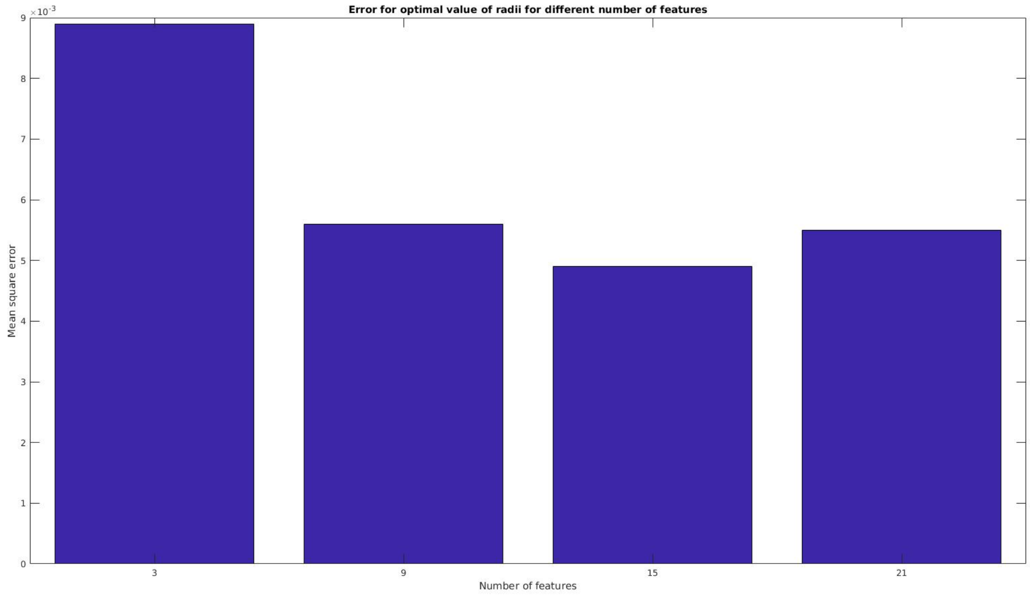
Σχήμα 23: Αριθμός κανόνων συναρτήσει του αριθμού χαρακτηριστικών και του radii

Συμπεράσματα από το Grid Search

Όπως προκύπτει από τα παραπάνω αποτελέσματα, υπάρχουν πολλές περιπτώσεις για την οποίες δεν είναι δυνατή η εκτέλεση του αλγορίθμου. Όταν έχουμε **λίγα χαρακτηριστικά** και **μεγάλη ακτίνα** δημιουργείται μόνο ένας κανόνας, πράγμα που σημαίνει ότι δεν μπορεί να εκτελεστεί η μέθοδος Subtractive Clustering. Επίσης, για **πολλά χαρακτηριστικά** και ακτίνα **σχετικά μικρή**, δημιουργείται πολύ μεγάλος αριθμός κανόνων, γεγονός που δεν μας επιτρέπει να εκτελέσουμε τον αλγόριθμο, καθώς θα χρειαστεί πάρα πολύς χρόνος εκτέλεσης. Για **σταθερό αριθμό χαρακτηριστικών**, όποιος κι αν είναι αυτός, όσο **αυξάνεται η ακτίνα** επιρροής των cluster **μειώνεται και το μέσο σφάλμα**.

Η βέλτιστη περίπτωση προκύπτει για 15 features και την μέγιστη ακτίνα, όπου δημιουργούνται 5 κανόνες. Βλέπουμε δηλαδή ότι η **ύπαρξη πολλών κανόνων**, **δεν οδηγεί στην μείωση του σφάλματος**, αλλά το αντίθετο. Αυξάνεται η πολυπλοκότητα, άρα και ο χρόνος εκτέλεσης του αλγορίθμου, όμως δεν βελτιώνεται η εκτίμηση του μοντέλου. Η πιο απλή περίπτωση με τη μικρότερη ακτίνα και τα λιγότερα χαρακτηριστικά, επιφέρει το διπλάσιο περίπου σφάλμα από την βέλτιστη περίπτωση.

Στο παρακάτω διάγραμμα φαίνεται η σχέση ανάμεσα στη βέλτιστη περίπτωση ακτίνας για κάθε αριθμό χαρακτηριστικών.

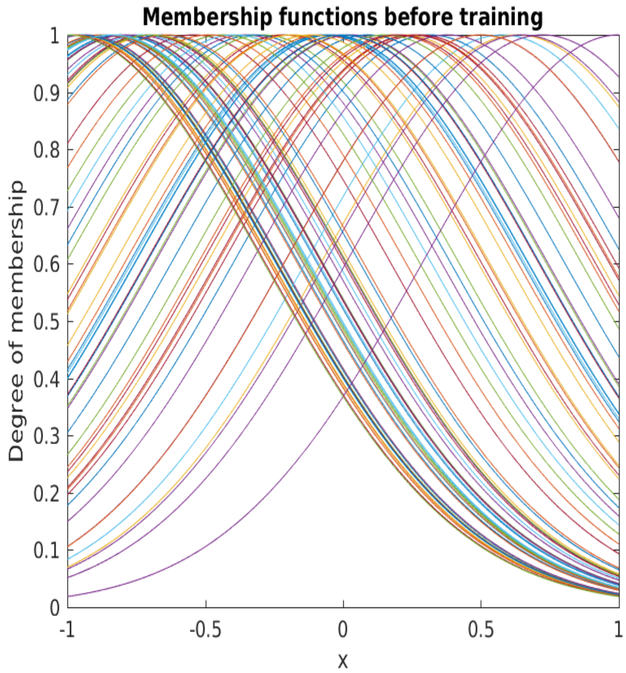


Σχήμα 24: Μέσο σφάλμα για την βέλτιστη τιμή ακτίνας για διάφορες τιμές χαρακτηριστικών

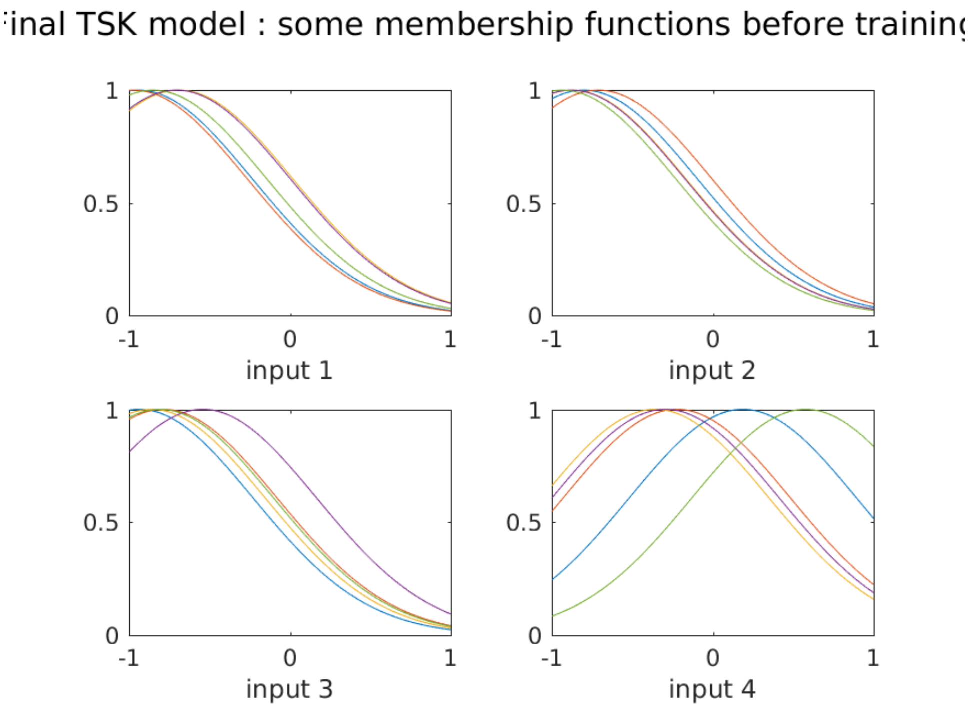
Άρα, όπως είπαμε, ο βέλτιστος συνδυασμός είναι **15 χαρακτηριστικά** και **ακτίνα εμβέλειας των cluster 1,** με την οποία δημιουργούνται 5 κανόνες και με αυτές τις τιμές μπορούμε να προχωρήσουμε στην εκπαίδευση του τελικού TSK μοντέλου.

Εκπαίδευση τελικού TSK μοντέλου

Όσον αφορά το τελικό μοντέλο, αρχικά βλέπουμε ότι οι συναρτήσεις συμμετοχής πριν την εκπαίδευση είναι οι εξής:

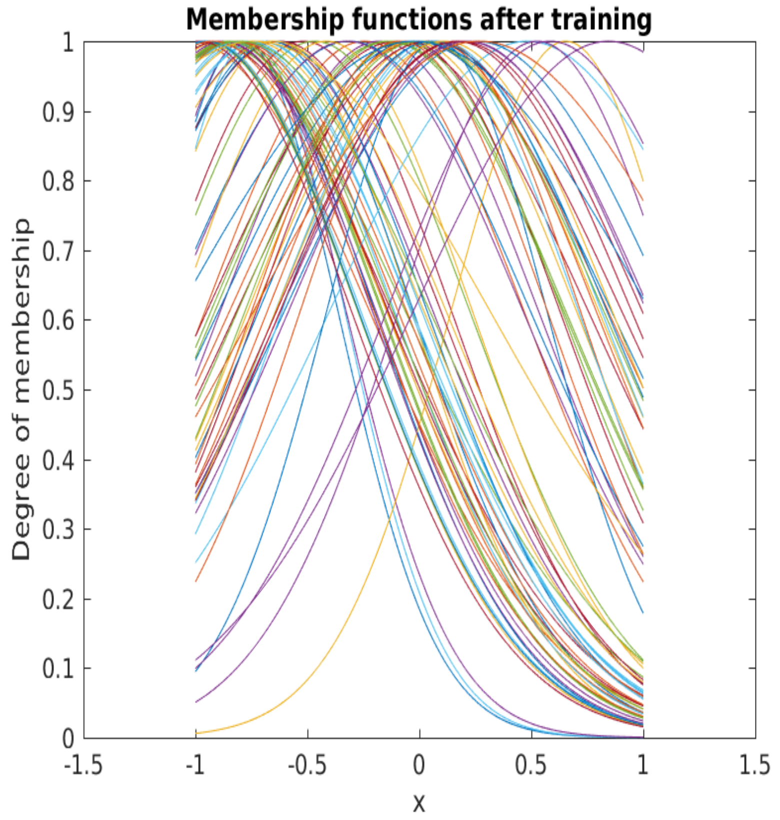


Σχήμα 25: Τελικό TSK model - Συναρτήσεις συμμετοχής πριν την εκπαίδευση

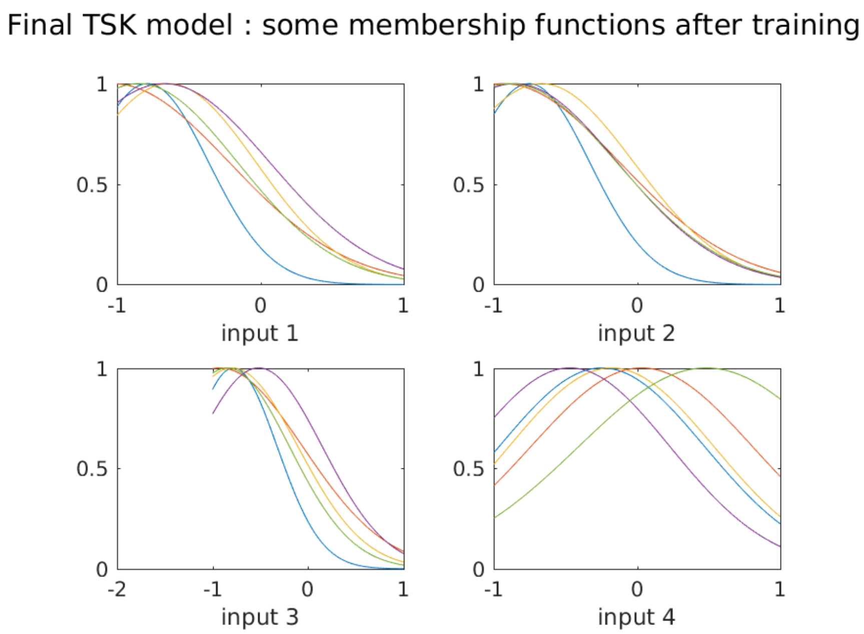


Σχήμα 26: Τελικό TSK model - Ορισμένες συναρτήσεις συμμετοχής πριν την εκπαίδευση

Μετά από εκπαίδευση για 150 εποχές, το τελικό TSK μοντέλο έχει τις παρακάτω συναρτήσεις συμμετοχής.

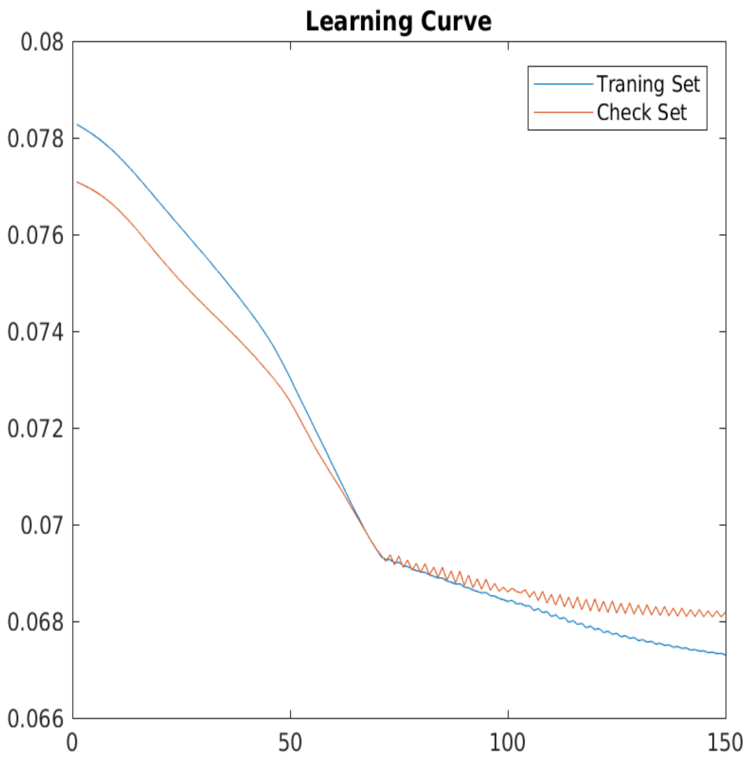


Σχήμα 27: Τελικό TSK model - Συναρτήσεις συμμετοχής μετά την εκπαίδευση

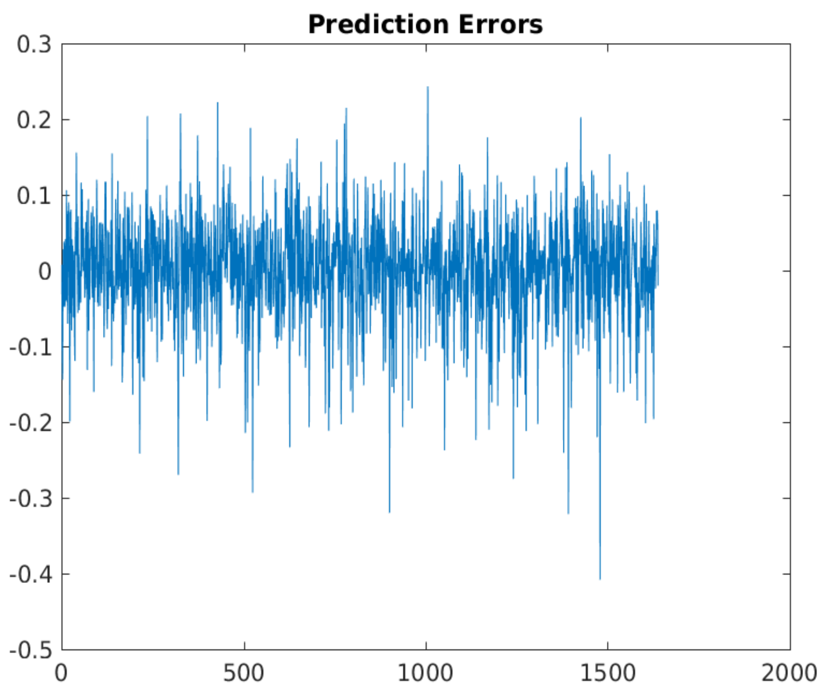


Σχήμα 28: Τελικό TSK model - Ορισμένες συναρτήσεις συμμετοχής μετά την εκπαίδευση

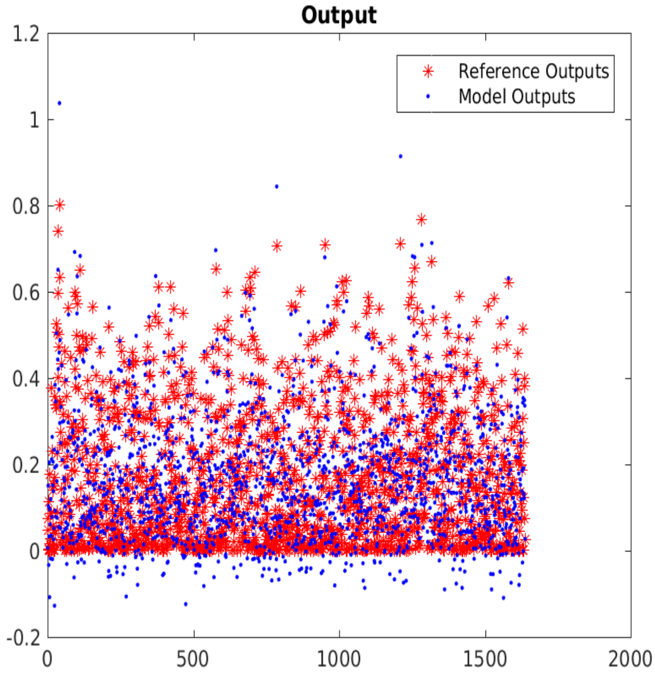
Η καμπύλη εκμάθησης και τα σφάλματα πρόβλεψης φαίνονται στα παρακάτω διαγράμματα.



Σχήμα 29: Τελικό TSK model - Καμπύλη εκμάθησης



Σχήμα 30: Τελικό TSK model - Σφάλματα πρόβλεψης



Σχήμα 31: Τελικό TSK model - Τελική vs Επιθυμητή έξοδος

Στον παρακάτω πίνακα παρουσιάζονται οι δείκτες απόδοσης $RMSE, NMSE, NDEI$ και R^2.

TSK model	Number of features	Number of rules	MSE	RMSE	NMSE	NDEI	R^2	Time (sec)
Final	15	5	0.004832	0.069511	0.162823	0.403513	0.837169	102.8

Βλέπουμε ότι το μοντέλο παρουσιάζει αρκετά μικρό σφάλμα, όσο είχε προκύψει και κατά το Grid Search για την αναζήτηση των βέλτιστων αριθμών για να πραγματοποιηθεί η εκπαίδευση. Ο συντελεστής R^2 βρίσκεται σχετικά κοντά στο 1, που σημαίνει ότι το μοντέλο δεν μας δίνει ως εκτίμηση τη μέση τιμή, αλλα πλησιάζει προς την τέλεια εκτίμηση. Το μοντέλο μετά από 70 περίπου εποχές συγκλίνει προς το τελικό σφάλμα, ενώ οι συναρτήσεις συμμετοχής δεν παρουσιάζουν πολύ μεγάλη διαφορά μετά την εκπαίδευση του δικτύου, προσαρμόζονται όμως ώστε να δίνουν την καλύτερη πρόβλεψη.

Η σημαντική διαφορά όμως βρίσκεται στον αριθμό των κανόνων που υπάρχει σε σύγκριση με τον χρόνο εκτέλεσης και την μνήμη που απαιτείται, λόγω του Subtractive Clustering. Στην περίπτωση που θα χρησιμοποιούσαμε Grid Partitioning, για να έχουμε 15 χαρακτηριστικά με 2 συναρτήσεις συμμετοχής για το καθένα, θα καταλήγαμε με 2^{15} κανόνες, ενώ με 3 συναρτήσεις θα καταλήγαμε με 3^{15} κανόνες. Αντίθετα τώρα, μόνο με 5 IF-THEN κανόνες μπορούμε να εξάγουμε πολύ καλή πρόβλεψη και πολύ σύντομο χρονικό διάστημα εκτέλεσης. Επομένως, για την περίπτωση που έχουμε μεγάλο αριθμό χαρακτηριστικών, είναι απαγορευτική η χρήση του Grid Partitioning.

Επεξήγηση των παραδοτέων αρχείων MATLAB

- ccpp_tsk_model_1.m: Εκπαίδευση και αξιολόγηση του 1ου TSK μοντέλου
- ccpp_tsk_model_2.m: Εκπαίδευση και αξιολόγηση του 2ου TSK μοντέλου
- ccpp_tsk_model_3.m: Εκπαίδευση και αξιολόγηση του 3ου TSK μοντέλου
- ccpp_tsk_model_4.m: Εκπαίδευση και αξιολόγηση του 4ου TSK μοντέλου
- grid_search.m: Επιλογή των βέλτιστων παραμέτρων
- final_tsk_model.m: Εκπαίδευση και αξιολόγηση του τελικού μοντέλου

Η ύπαρξη φακέλων για κάθε διαφορετικό μοντέλο, οφείλεται στην απευθείας αποθήκευση των διαγραμμάτων κατά την εκτέλεση των αλγορίθμων.