



Εθνικό Μετσόβιο Πολυτεχνείο (Ε.Μ.Π)
Σχολή Μηχανολόγων Μηχανικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

“ΑΝΑΠΤΥΞΗ ΜΟΝΤΕΛΟΥ ΕΚΤΙΜΗΣΗΣ
ΥΠΟΛΕΙΠΟΜΕΝΗΣ ΩΦΕΛΙΜΗΣ
ΔΙΑΡΚΕΙΑΣ ΖΩΗΣ
ΕΔΡΑΝΩΝ ΚΥΛΙΣΗΣ”

Λάζαρος Χριστοφορίδης

Τομέας: Τεχνολογίας των Κατεργασιών

Επιβλέπων: Μπενάρδος Πανώριος, Επίκ. Καθηγητής ΕΜΠ

Αθήνα, 2023

Υπεύθυνη δήλωση για λογοκλοπή και για κλοπή πνευματικής ιδιοκτησίας:

Έχω διαβάσει και κατανοήσει τους κανόνες για τη λογοκλοπή και τον τρόπο σωστής αναφοράς των πηγών που περιέχονται στον οδηγό συγγραφής Διπλωματικών Εργασιών. Δηλώνω ότι, από όσα γνωρίζω, το περιεχόμενο της παρούσας Διπλωματικής Εργασίας είναι προϊόν δικής μου εργασίας και υπάρχουν αναφορές σε όλες τις πηγές που χρησιμοποίησα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή τη Διπλωματική εργασία είναι του συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις της Σχολής Μηχανολόγων Μηχανικών ή του Εθνικού Μετσόβιου Πολυτεχνείου.

Λάζαρος Χριστοφορίδης

Περιεχόμενα

Περίληψη	5
Abstract	6
1 Εισαγωγή	7
1.1 Σκοπός και δομή της διπλωματικής	7
1.2 Εισαγωγή στις βλάβες των εδράνων κύλισης	8
1.3 Εισαγωγή στην προγνωστική και το RUL των Εδράνων Κύλισης	9
1.4 Εισαγωγή στα Νευρωνικά Δίκτυα	10
2 Μέθοδος Έρευνας	12
3 Πειραματικές μετρήσεις	14
3.1 Περιγραφή πειράματος	14
3.2 Πειραματική Εγκατάσταση	14
3.3 Δομή δεδομένων	14
3.4 Ανασκόπηση Μετρήσεων	15
3.4.1 Ανασκόπηση Μετρήσεων στο πεδίο του χρόνου	15
3.4.2 Ανασκόπηση Μετρήσεων στο πεδίο συχνοτήτων	16
4 Εξαγωγή Χαρακτηριστικών	18
4.1 Αποθρομβοποίηση σήματος	18
4.2 Χαρακτηριστικά σημάτων	19
4.3 Κατηγοριοποίηση δεδομένων	26
4.4 Επιλογή Χαρακτηριστικών	28
4.4.1 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης RUL, με βλάβη στον εσωτερικό δακτύλιο	29
4.4.2 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης RUL, με βλάβη στον εξωτερικό δακτύλιο	30
4.4.3 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης RUL, με βλάβη στα στοιχεία κύλισης	32
4.4.4 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης κατάστασης του εδράνου κύλισης	34
5 Μοντέλο ανίχνευσης βλάβης του εδράνου κύλισης	36
5.1 Επιλογή Δεδομένων	36
5.2 Αρχιτεκτονική	37
5.3 Βελτιστοποίηση	38
6 Εκτίμηση Υπολειπόμενης Ωφέλιμης Διάρκειας Ζώνης του εδράνου κύλισης	42
6.1 Επιλογή Δεδομένων	42
6.2 Αρχιτεκτονική	43
6.3 Βελτιστοποίηση	46

6.3.1	Βελτιστοποίηση μοντέλου πρόβλεψης RUL μετρήσεων υπό βλάβη στον εσ. δακτύλιο (1)	49
6.3.2	Βελτιστοποίηση μοντέλου πρόβλεψης RUL μετρήσεων υπό βλάβη στον εξ. δακτύλιο (2)	51
6.3.3	Βελτιστοποίηση μοντέλου πρόβλεψης RUL μετρήσεων υπό βλάβη στα στοιχεία κύλισης (3)	53
7	Τελική αξιολόγηση	56
8	Συμπεράσματα και Μελλοντικές Εργασίες	60
	Κατάλογος Σχημάτων	62
	Κατάλογος Πινάκων	65
	Αναφορές	66
	Παράρτημα	68

Περίληψη

Η ασφαλής λειτουργία των περιστρεφόμενων μηχανών, συνδέεται άμεσα με την ασφαλή λειτουργία των εδράνων κύλισης, το οποίο είναι το κύριο μηχανολογικό εξάρτημα και πρωταρχικό αίτιο αστοχίας των περιστρεφόμενων μηχανών. Σκοπός της εργασίας αυτής, είναι η εκτίμηση της υπολειπόμενης ωφέλιμης διάρκειας ζωής (remaining useful life) των εδράνων κύλισης από μοντέλα τεχνητής νοημοσύνης, βασισμένα σε πειραματικά δεδομένα (Data Driven Model). Οι μετρήσεις που χρησιμοποιήθηκαν για την ανάλυση προέρχονται από ένα πείραμα που διεξήχθη στο Πανεπιστήμιο του Σινσινάτι, όπου επιταχυνσιόμετρα καταγράφουν κραδασμούς των εδράνων κύλισης, μέχρι την τελική τους αστοχία. Αρχικά, γίνεται αποθορυβοποίηση των μετρήσεων, με την μέθοδο αποσύνθεσης Wavelet, και από το πλέον επεξεργασμένο σήμα, εξάγονται χαρακτηριστικά του σήματος από το πεδίο του χρόνου και συχνότητας. Η τελική επιλογή των κατάλληλων χαρακτηριστικών, που έχουν άμεση συσχέτιση με το RUL, γίνεται χρησιμοποιώντας τα στατιστικά τεστ F-test και MI-test. Στην συνέχεια, αναπτύσσεται μοντέλο απλού νευρωνικού δικτύου, για την ανίχνευση της κατάστασης του έδρανου κύλισης όπου και γίνεται βελτιστοποίηση στην αρχιτεκτονική του. Έπειτα, αναπτύσσονται και βελτιστοποιούνται τρία μοντέλα νευρωνικών δικτύων με νευρώνες LSTM, για την εκτίμηση του RUL για τις τρεις κύριες βλάβες του έδρανου κύλισης. Το μοντέλο ανίχνευσης βλάβης παρουσιάζει ακρίβεια 99.1%, ενώ το μοντέλο εκτίμησης του υπολοίπου ωφέλιμου χρόνου ζωής παρουσιάζει ακρίβεια σφάλματος εντός του διανύσματος $[-6\%, +9\%]$ του συνολικού χρόνου αστοχίας. Η ανάπτυξη του μοντέλου εκτίμησης του RUL, είναι σημαντική για την ασφάλεια των συστημάτων, αλλά επιπλέον μπορεί μελλοντικά να βοηθήσει στην ανάπτυξη συστημάτων προληπτικής συντήρησης, με σκοπό τη μείωση των εξόδων συντήρησης.

Abstract

The safe operation of rotating machinery is directly related to the safe operation of bearings, which are the main mechanical component and primary cause of failure in rotating machinery. The purpose of this work is to estimate the remaining useful life (RUL) of bearings using artificial intelligence models based on experimental data (Data Driven Model). The measurements used for the analysis originate from an experiment conducted at the University of Cincinnati, where accelerometers record vibrations of the bearings until their final failure. Initially, the measurements are denoised using the Wavelet decomposition method, and from the processed signal, time and frequency domain features are extracted. The final selection of relevant features that have a direct correlation with the RUL is made using statistical tests such as the F-test and MI-test. Subsequently, a simple neural network model is developed to detect the condition of the bearings, and its architecture is optimized. Then, three Long Short-Term Memory (LSTM) neural network models are developed and optimized for RUL estimation for the three main bearing faults. The fault detection model achieves an accuracy of 99.1%, while the RUL estimation model has an error accuracy within the range of $[-6\%, +9\%]$ of the total failure time. The development of a model that estimates the RUL is significant for system safety and can also assist in the development of predictive maintenance systems, aiming to reduce maintenance costs in the future.

1 Εισαγωγή

1.1 Σκοπός και δομή της διπλωματικής

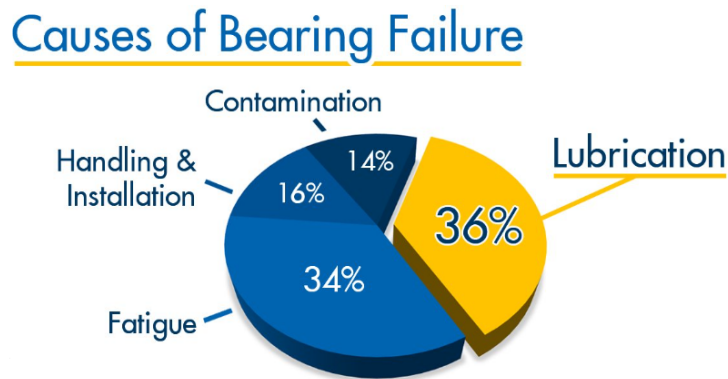
Ο στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός μοντέλου πρόβλεψης της υπολειπόμενης ωφέλιμης διάρκειας ζωής ενός συγκεκριμένου τύπου εδράνου κύλισης. Αυτό το μοντέλο θα επιτρέψει τη δημιουργία προγνωστικών συστημάτων συντήρησης και αντικατάστασης σε μελλοντικές εργασίες. Αξίζει να σημειωθεί ότι το μοντέλο που αναπτύσσεται στην παρούσα εργασία είναι αξιόπιστο μόνο για ένα συγκεκριμένο τύπο εδράνου κύλισης και για συγκεκριμένα φορτία λειτουργίας.

Η δομή της διπλωματικής διαμορφώνεται ως ακολούθως: Αρχικά, στην ενότητα **“Πειραματικές Μετρήσεις”**, περιγράφεται λεπτομερώς η πειραματική εγκατάσταση και αναφέρεται το πείραμα από το οποίο προέκυψαν τα σύνολα δεδομένων. Επιπλέον, γίνεται σύντομη ανασκόπηση των μετρήσεων στο πεδίο του χρόνου και των συχνοτήτων. Στη συνέχεια, στην ενότητα **“Εξαγωγή Χαρακτηριστικών”**, παρουσιάζεται λεπτομερώς η διαδικασία εξαγωγής χαρακτηριστικών από τα δεδομένα κραδασμών. Μετά την αποθορυβοποίηση των κραδασμών, εξετάζονται διάφορα χαρακτηριστικά του πεδίου χρόνου και των συχνοτήτων. Στη συνέχεια, μέσω της εφαρμογής στατιστικών τεστ, F-test και MI-test, επιλέγονται τα χαρακτηριστικά που συσχετίζονται σημαντικά με τον υπολειπόμενο ωφέλιμο χρόνο ζωής (RUL). Στη συνέχεια, στην ενότητα **“Μοντέλο Ανίχνευσης Βλάβης του εδράνου κύλισης”**, αναπτύσσεται το Τεχνητό Νευρωνικό Δίκτυο (ΤΝΔ) για την ανίχνευση βλάβης του εδράνου κύλισης. Στο πλαίσιο αυτό, πραγματοποιείται και βελτιστοποίηση του μοντέλου. Στην ενότητα **“Εκτίμηση Υπολειπόμενης Ωφέλιμης Διάρκειας Ζωής”**, αναπτύσσεται ένα Εντατικό Νευρωνικό Δίκτυο (ΕΝΔ) με χρήση κόμβων (LSTM) για κάθε τύπο βλάβης, με σκοπό την εκτίμηση του υπολειπόμενου ωφέλιμου χρόνου ζωής (RUL). Εδώ επίσης, πραγματοποιείται βελτιστοποίηση του μοντέλου. Στην ενότητα **“Τελική Αξιολόγηση”**, παρουσιάζονται και αναλύονται τα αποτελέσματα από όλες τις μετρήσεις που πραγματοποιήθηκαν. Τέλος, στην ενότητα **“Συμπεράσματα και Μελλοντικές Εργασίες”**, πραγματοποιείται σύγκριση των τελικών αποτελεσμάτων με άλλες σχετικές εργασίες, και κατατίθενται προτάσεις για μελλοντικές εργασίες.

Στην παρούσα εργασία, χρησιμοποιήθηκε το περιβάλλον MATLAB για την επεξεργασία των δεδομένων και την εξαγωγή των χαρακτηριστικών, ενώ η γλώσσα προγραμματισμού Python χρησιμοποιήθηκε για τη στατιστική ανάλυση και την ανάπτυξη των ΤΝΔ, ειδικότερα χρησιμοποιήθηκε η βιβλιοθήκη Keras. Οι κώδικες που χρησιμοποιήθηκαν μπορεί να βρεθούν στο **Παράρτημα**.

1.2 Εισαγωγή στις βλάβες των εδράνων κύλισης

Τα έδρανα κύλισης παίζουν κρίσιμο ρόλο στην ομαλή και αξιόπιστη λειτουργία των περιστρεφόμενων μηχανών. Αυτά τα συνήθως μικρά αλλά απαραίτητα μηχανολογικά εξαρτήματα επιτρέπουν την αποτελεσματική μεταφορά της περιστροφικής κίνησης, τη στήριξη αξόνων, τη μείωση της τριβής και τη διατήρηση της ακριβούς ευθυγράμμισης. Η κατανόηση της υπολειπόμενης ωφέλιμης ζωής (RUL) των εδράνων κύλισης είναι ζωτικής σημασίας για στρατηγικές προληπτικής συντήρησης, διασφαλίζοντας τη βέλτιστη απόδοση και αποτρέποντας απροσδόκητες βλάβες. Σύμφωνα με μια δημοσίευση που δημοσιεύτηκε από τον Irwin, 1996 [1], οι αστοχίες των εδράνων κύλισης θεωρούνται μία από τις κύριες αιτίες αστοχιών περιστρεφόμενων μηχανημάτων. Η δημοσίευση τονίζει ότι οι αστοχίες των εδράνων κύλισης αντιπροσωπεύουν περίπου το 34% όλων των μηχανικών αστοχιών στον περιστρεφόμενο εξοπλισμό. Μια μελέτη από την SKF το 2022 [2] αναφέρει ότι περίπου το 36% των αστοχιών στα έδρανα κύλισης οφείλονται σε ακατάλληλη λίπανση, το 34% σε κόπωση, το 16% σε λανθασμένο χειρισμό και εγκατάσταση, και το 14% σε μόλυνση (βλ. Σχήμα 1).



Σχήμα 1: Αιτίες βλάβης εδράνων κύλισης. Πηγή: SKF, 2022 [2].

Τα σημεία που εντοπίζονται οι τρεις κύριες βλάβες των εδράνων κύλισης, είναι, στον εξωτερικό δακτύλιο (Outer Race), στον εσωτερικό δακτύλιο (Inner Race) και στα στοιχεία κύλισης (Roller Elements). Στο Σχήμα 2, παρουσιάζονται οι τύποι των βλαβών.



Σχήμα 2: Οι τρεις κύριες βλάβες των εδράνων κύλισης. Βλάβη στον εξωτερικό δακτύλιο, βλάβη στον εσωτερικό δακτύλιο και βλάβη στα έδρανα κύλισης αντίστοιχα. Πηγή: An, 2014 [3].

Στην περίπτωση βλάβης, εμφανίζεται υψίσυχνη ιδιοσυχνότητα ταλάντωσης, όπου αποδιαμορφώνοντάς τη, εντοπίζεται η χαρακτηριστική συχνότητα βλάβης. Οι χαρακτηριστικές συχνότητες βλάβης, είναι γραμμικά εξαρτώμενες από την συχνότητα περιστροφής του άξονα, όπου ο γραμμικός συντελεστής υπολογίζεται αποκλειστικά από την γεωμετρία του εδράνου κύλισης. Οι σχέσεις των χαρακτηριστικών συχνωτήτων βλάβης λήφθηκαν από τον Randall, 2011 [4]:

$$BPFO = f_{outer} = \frac{N}{2} f_r \left(1 - \frac{BD}{D_m} \cos a\right) \quad (1)$$

$$BPFI = f_{inner} = \frac{N}{2} f_r \left(1 + \frac{BD}{D_m} \cos a\right) \quad (2)$$

$$BSF = f_{ball} = f_r \frac{D_m}{2BD} \left[1 - \left(\frac{BD}{D_m} \cos a\right)^2\right] \quad (3)$$

Όπου N ο αριθμός των στοιχείων κύλισης, f_r η συχνότητα περιστροφής του άξονα, BD η διάμετρος των στοιχείων κύλισης, D_m η διάμετρος θέσης των στοιχείων κύλισης και a η γωνία επαφής των στοιχείων κύλισης.

1.3 Εισαγωγή στην προγνωστική και το RUL των Εδράνων Κύλισης

Ως υπολειπόμενη ωφέλιμη διάρκεια ζωής (Remaining Useful Life) του εδράνου κύλισης, ορίζεται ο χρόνος λειτουργίας που απομένει στο εξάρτημα μέχρι αυτό να χρειαστεί επισκευή ή/και αντικατάσταση. Η εκτίμηση του υπολειπόμενου ωφέλιμου χρόνου ζωής ενός εξοπλισμού πριν από την αστοχία βοηθά τους ειδικούς στη βιομηχανία να αποφύγουν την ανάγκη για μη απαραίτητη δαπάνη χρόνου και χρημάτων στη συντήρηση. Έτσι, η προγνωστική έχει γίνει ένα σημαντικό βιομηχανικό ζήτημα και ένα ελκυστικό πεδίο έρευνας για πολλούς ερευνητές. Βασισμένοι στον Jaouher, 2014 [5], οι στρατηγικές συντήρησης χωρίζονται σε τρεις τύπους:

- Διορθωτική συντήρηση (corrective maintenance).
- Προληπτική συντήρηση (preventive maintenance).
- Συντήρηση με βάση την κατάσταση (condition-based maintenance - CBM).

Η διορθωτική συντήρηση και η προληπτική συντήρηση εξαφανίζονται από αρκετούς βιομηχανικούς τομείς, όπως αναφέρει ο Dong, 2017 [6], καθώς βασίζονται στην επισκευή ενός στοιχείου όταν αστοχεί και στην περιοδική επιθεώρηση και αντικατάσταση, αντίστοιχα. Το κύριο πλεονέκτημα της CBM είναι ο συνεχής έλεγχος μίας διαδικασίας για τον καθορισμό της βέλτιστης χρονικής παρέμβασης και τη μη διακοπή της κανονικής λειτουργίας. Επομένως, η CBM είναι ευρέως στοχευμένη στην έρευνα και τη βιομηχανία σήμερα. Στη βιβλιογραφία, υπάρχουν τρεις διαφορετικές προσεγγίσεις εκτίμησης του RUL:

- Φυσικά μοντέλα που αναπτύχθηκαν από ειδικούς.

- Συστήματα εμπειρογνομόνων βάσει κανόνων.
- Μοντέλα τεχνητής νοημοσύνης, βασισμένα σε δεδομένα (data-driven models).

Τα φυσικά μοντέλα αξιοποιούν την φυσική και μηχανική της διαδικασίας καταστροφής του εδράνου κύλισης για να εκτιμήσουν το RUL. Αυτά τα μοντέλα βασίζονται σε μαθηματικές εξισώσεις και θεωρητικές αρχές που περιγράφουν τους μηχανισμούς καταστροφής που εμπλέκονται. Λαμβάνοντας υπόψη παράγοντες όπως το φορτίο, τη θερμοκρασία, τη λίπανση και τις ιδιότητες των υλικών, τα φυσικά μοντέλα μπορούν να παρέχουν πολύτιμες πληροφορίες σχετικά με τη φθορά των εδράνων κύλισης. Μερικά αξιοσημείωτα φυσικά μοντέλα για την εκτίμηση RUL περιλαμβάνουν τα εμπειρικά μοντέλα που προτάθηκαν από τον Iooss, 2008 [7] και το μοντέλο στοχαστικής υποβάθμισης που αναπτύχθηκε από τον Wang, 2009 [8].

Οι μέθοδοι που βασίζονται σε κανόνες για την εκτίμηση του RUL στηρίζονται σε ειδικές γνώσεις και προκαθορισμένους κανόνες για τον προσδιορισμό της υπολειπόμενης ωφέλιμης διάρκειας ζωής των εδράνων κύλισης. Αυτές οι προσεγγίσεις συνήθως περιλαμβάνουν τη διαμόρφωση δεικτών κατάστασης που βασίζονται σε παρατηρήσιμα χαρακτηριστικά των εδράνων κύλισης, όπως κραδασμούς, θερμοκρασία και ακουστικές εκπομπές. Τα συστήματα που βασίζονται σε κανόνες εφαρμόζουν στη συνέχεια ένα σύνολο προκαθορισμένων κανόνων ή ορίων για να ταξινομήσουν την κατάσταση της υγείας των εδράνων κύλισης και να εκτιμήσουν την υπολειπόμενη ωφέλιμη διάρκεια ζωής τους. Ένα πολύ γνωστό παράδειγμα προσέγγισης που βασίζεται σε κανόνες είναι η μέθοδος Health Index που προτείνεται από τον Saxena, 2008 [9], το οποίο χρησιμοποιεί ένα σύνολο κανόνων και ορίων που βασίζονται σε στατιστική ανάλυση και κρίση εμπειρογνομόνων.

1.4 Εισαγωγή στα Νευρωνικά Δίκτυα

Τα μοντέλα τεχνητής νοημοσύνης, αξιοποιούν τη δύναμη των αλγορίθμων μηχανικής μάθησης για να μαθαίνουν αυτόματα μοτίβα και σχέσεις από ιστορικά δεδομένα. Αυτά τα μοντέλα μπορούν να συλλάβουν πολύπλοκες μη γραμμικές εξαρτήσεις εντός των δεδομένων, επιτρέποντας ακριβείς προβλέψεις RUL. Διάφοροι αλγόριθμοι μηχανικής μάθησης, όπως μηχανές διανυσμάτων υποστήριξης (SVM), τεχνητά νευρωνικά δίκτυα (ANN) και τυχαία δάση (RF), έχουν εφαρμοστεί με επιτυχία στην εκτίμηση RUL. Αξιοσημείωτα μοντέλα που βασίζονται σε δεδομένα για την εκτίμηση RUL περιλαμβάνουν το δίκτυο μακράς βραχυπρόθεσμης μνήμης (Long Short-Term Memory, Hochreiter, 1997 [10]) που προτείνεται από τον Li, 2017 [11], και το μοντέλο τυχαίας παλινδρόμησης δασών που εισήχθη από τον Saxena, 2008 [12].

Τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ), έχουν αναδειχθεί ως ισχυρά εργαλεία για την επίλυση σύνθετων προβλημάτων σε διάφορους τομείς. Τα ΤΝΔ, είναι εμπνευσμένα από την δομή και τη λειτουργία του ανθρώπινου εγκεφάλου, όπου οι διασυνδεδεμένοι νευρώνες, επεξεργάζονται και μεταδίδουν πληροφορίες. Ένα ΤΝΔ αποτελείται από διασυνδεδεμένους κόμβους, γνωστούς και ως νευρώνες (neurons), οι οποίοι είναι οργανωμένοι σε επίπεδα (layers). Αυτοί οι κόμβοι λαμβάνουν σήματα εισόδου, εκτελούν υπολογισμούς σε αυτούς και παράγουν σήματα εξόδου που μεταβιβάζονται σε άλλους κόμβους του δικτύου. Η ισχύς των συνδέσεων μεταξύ των κόμβων, γνωστών ως

βάρη, καθορίζει την επίδραση ενός κόμβου σε έναν άλλο. Προσαρμόζοντας αυτά τα βάρη, τα νευρωνικά δίκτυα μπορούν να μάθουν να αναγνωρίζουν μοτίβα και να κάνουν προβλέψεις.

Η απλούστερη μορφή ενός ΤΝΔ είναι ένα δίκτυο εμπρόσθιας τροφοδότησης (feed forward), όπου οι πληροφορίες ρέουν προς μία κατεύθυνση, από το επίπεδο εισόδου στο επίπεδο εξόδου. Κάθε κόμβος στο δίκτυο λαμβάνει σήματα εισόδου, εφαρμόζει μια μη γραμμική συνάρτηση ενεργοποίησης (activation function) στο σύνθετο άθροισμα των εισόδων του και παράγει μια έξοδο. Η συνάρτηση ενεργοποίησης εισάγει τη μη γραμμικότητα στο δίκτυο, επιτρέποντάς του να μοντελοποιεί σύνθετες σχέσεις μεταξύ εισόδων και εξόδων. Μαθηματικά, η έξοδος ενός κόμβου μπορεί να αναπαρασταθεί ως εξής:

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (4)$$

Όπου y η έξοδος, f η συνάρτηση ενεργοποίησης, w_i τα βάρη (weights) των εισόδων x_i και b ο συντελεστής (bias) του κάθε κόμβου. Ενώ η απλή αρχιτεκτονική των ΤΝΔ είναι αποτελεσματική σε πολλές εφαρμογές, δυσκολεύεται να καταγράψει μακροπρόθεσμες εξαρτήσεις σε διαδοχικά δεδομένα, όπως τα δεδομένα χρονοσειρών. Αυτό ο περιορισμός οδήγησε στην ανάπτυξη επαναλαμβανόμενων νευρωνικών δικτύων (ΕΝΔ-RNN). Ένας τύπος RNN, που είναι αρκετά διαδεδομένος, είναι το δίκτυο Long Short-Term Memory (LSTM), ο οποίος είναι ικανός να καταγράψει μακροπρόθεσμες εξαρτήσεις σε διαδοχικά δεδομένα. Τα RNN εισάγουν συνδέσεις ανάδρασης, επιτρέποντας στις πληροφορίες να ρέουν όχι μόνο προς τα εμπρός αλλά και προς τα πίσω στο δίκτυο, επιτρέποντας στο δίκτυο να διατηρεί πληροφορίες από προηγούμενα χρονικά βήματα. Τα δίκτυα LSTM επιτυγχάνουν με την εισαγωγή κυψελών μνήμης, που έχουν τη δυνατότητα να θυμούνται και να ξεχνούν πληροφορίες επιλεκτικά. Κάθε κύτταρο μνήμης αποτελείται από τρία κύρια στοιχεία: μια πύλη εισόδου, μια πύλη λήθης και μια πύλη εξόδου. Αυτές οι πύλες ελέγχουν τη ροή των πληροφοριών, επιτρέποντας στο δίκτυο να μάθει αποτελεσματικά τις εξαρτήσεις. Οι εξισώσεις που διέπουν τη συμπεριφορά μιας κυψέλης μνήμης LSTM είναι οι εξής:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (6)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (7)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (8)$$

$$h_t = o_t \odot \tanh(c_t) \quad (9)$$

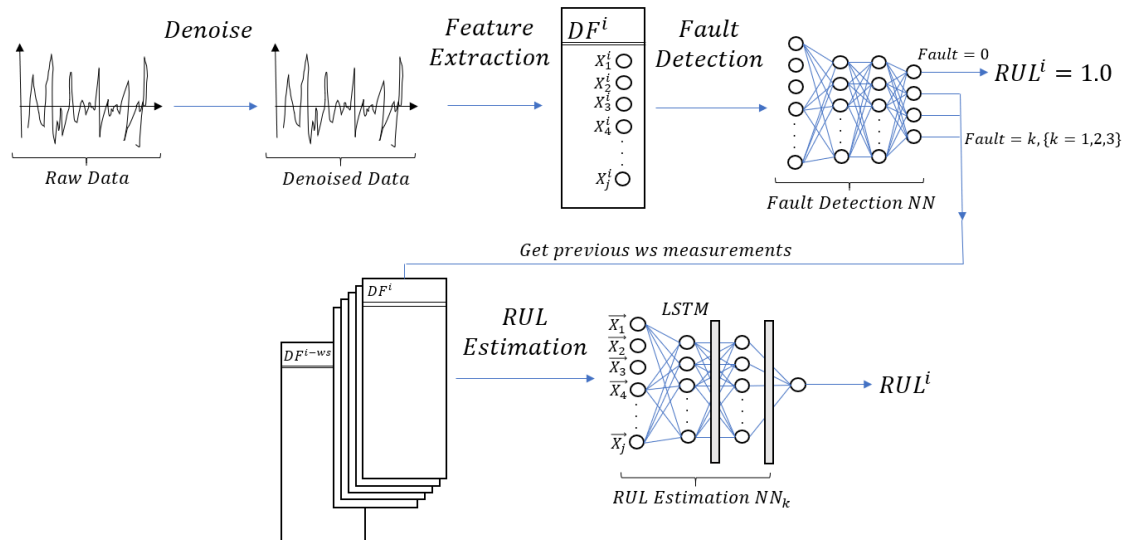
Όπου x_t η είσοδος την χρονική στιγμή t , h_t η κρυφή κατάσταση την χρονική στιγμή t , i_t , f_t και o_t είναι οι πύλες εισόδου, λήθης και εξόδου αντίστοιχα, σ η συνάρτηση ενεργοποίησης sigmoid και W , b ο πίνακας των βαρών και το διάνυσμα συντελεστών bias αντίστοιχα.

2 Μέθοδος Έρευνας

Η μεθοδολογία που χρησιμοποιείται για την ανάπτυξη του μοντέλου εκτίμησης του RUL του εδράνου κύλισης, είναι η εξής:

1. Συλλογή Δεδομένων: Χρησιμοποιήθηκαν δημόσια διαθέσιμα δεδομένα, από πειράματα που έγιναν στο πανεπιστήμιο της Σινσινάτι, τα οποία αποτελούν τους κραδασμούς του εδράνου κύλισης ως την τελική τους αστοχία.
2. Προ-επεξεργασία δεδομένων: Τα πειραματικά δεδομένα υπόκεινται σε αποθορυβοποίηση και επεξεργασία για τη βελτίωση της ποιότητάς τους.
3. Εξαγωγή Χαρακτηριστικών: Επιλέγονται διάφορα υποψήφια χαρακτηριστικά που αποτελούν τα χαρακτηριστικά που θα χρησιμοποιηθούν ως είσοδοι στα ΝΔ.
4. Στατιστική Ανάλυση: Πραγματοποιείται στατιστική ανάλυση μεταξύ των υποψήφιων χαρακτηριστικών και της υπολειπόμενης ωφέλιμης διάρκειας ζωής RUL προκειμένου να επιλεγθούν τα κατάλληλα χαρακτηριστικά που θα χρησιμοποιηθούν ως είσοδοι στα ΝΔ.
5. Ανάπτυξη ΤΝΔ για την ανίχνευση της κατάστασης του εδράνου κύλισης: Αναπτύσσεται ΤΝΔ για την αναγνώριση της κατάστασης του εδράνου κύλισης, βάσει των επιλεγμένων χαρακτηριστικών. Το συγκεκριμένο ΤΝΔ, προβλέπει αν υπάρχει και ποιά βλάβη έχει υποστεί το έδρανο κύλισης.
6. Ανάπτυξη ΕΝΔ για εκτίμηση της υπολειπόμενης ωφέλιμης διάρκειας ζωής (RUL): Παράλληλα, αναπτύσσεται ένα ΕΝΔ ανά τύπο βλάβης, για την εκτίμηση της υπολειπόμενης ωφέλιμης διάρκειας ζωής του εδράνου κύλισης.

Μετά την ανάπτυξη και βελτιστοποίηση των ΝΔ, το συνολικό μοντέλο εκτίμησης του RUL, είναι ικανό να εκτιμήσει την κατάσταση του εδράνου κύλισης και να προβλέψει τον υπολειπόμενο ωφέλιμο χρόνο ζωής του. Η διαδικασία των εκτιμήσεων για μια τυχαία μέτρηση βασίζεται στο **Σχήμα 3**.



Σχήμα 3: Διαδικασία εκτίμησης της κατάστασης και υπολειπόμενης ωφέλιμης διάρκειας ζωής του εδράνου κύλισης.

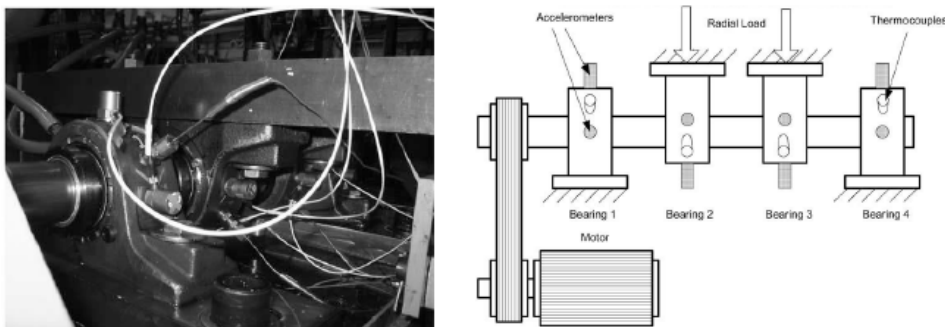
3 Πειραματικές μετρήσεις

3.1 Περιγραφή πειράματος

Οι μετρήσεις κραδασμών των εδράνων κύλισης που χρησιμοποιήθηκαν για την ανάλυση, προέρχονται από ένα διαθέσιμο δημόσιο σύνολο δεδομένων της NASA που παρασχέθηκε από το Κέντρο Intelligent Maintenance Systems (IMS) του Πανεπιστημίου της Σινσινάτι, με την υποστήριξη της εταιρείας Rexnord Corp. [13].

3.2 Πειραματική Εγκατάσταση

Κινητήρας AC σταθερής ταχύτητας περιστροφής $n = 2000rpm$, μεταδίδει κίνηση μέσω μάντα, σε άξονα που εδράζεται σε τέσσερα έδρανα κύλισης, τα οποία έχουν υποστεί λίπανση. Στα έδρανα κύλισης, ασκείται ακτινικά φορτίο $F = 6000lbs$, μέσω μηχανισμού ελατηρίου. Η πειραματική εγκατάσταση παρουσιάζεται στο **Σχήμα 4**.



Σχήμα 4: Πειραματική εγκατάσταση, και σακάριφμα εγκατάστασης. Πηγή από IMS, 2006 [13].

Τα έδρανα κύλισης είναι τύπου διπλής σειράς Rexnord ZA-2115, όπου στο κέλυφος τους είναι τοποθετημένα δύο επιταχυνσιόμετρα PCB 353B33 High Sensitivity Quartz ICP ένα στην οριζόντια (x) και ένα στην κατακόρυφη διεύθυνση (y). Σε αυτή την εργασία, χρησιμοποιούνται μόνο οι μετρήσεις στην κατακόρυφη διεύθυνση (y), καθώς στο δεύτερο και τρίτο σύνολο μετρήσεων, δεν έγιναν μετρήσεις στην οριζόντια διεύθυνση (x).

3.3 Δομή δεδομένων

Εκτελέστηκαν τρία (3) πειράματα, τα οποία ήταν αρκετά για να εντοπίσουν τις χαρακτηριστικές βλάβες των εδράνων κύλισης (εσ. δακτύλιος, εξ. δακτύλιος, στοιχεία κύλισης). Κάθε σύνολο δεδομένων (data set) περιέχει μετρήσεις μέχρι τελικής αστοχίας (test-to-failure). Όλες οι τελικές αστοχίες προέκυψαν μετά από υπέρβαση του σχεδιασμένου χρόνου ζωής του εδράνου κύλισης που αντιστοιχεί σε περισσότερες από 100 εκατομμύρια περιστροφές.

Κάθε σύνολο δεδομένων, περιέχει μετρήσεις κραδασμών από τα τέσσερα έδρανα κύλισης διάρκειας ενός δευτερολέπτου (1s), που γίνονται κάθε 10 λεπτά. Η συχνότητα

δειγματοληψίας, είναι $f_s = 20480\text{Hz}$, επομένως καθώς η διάρκεια μέτρησης των κραδασμών είναι 1s , κάθε μέτρηση αποτελείται από 20480 σημεία. Στον **Πίνακα 1**, παρουσιάζονται τα σύνολα μετρήσεων που έγιναν, οι συνολικές μετρήσεις μέχρι την τελική αστοχία, το έδρανο κύλισης αστοχίας και η χαρακτηριστική βλάβη.

Πίνακας 1: Σύνολα πειραματικών μετρήσεων.

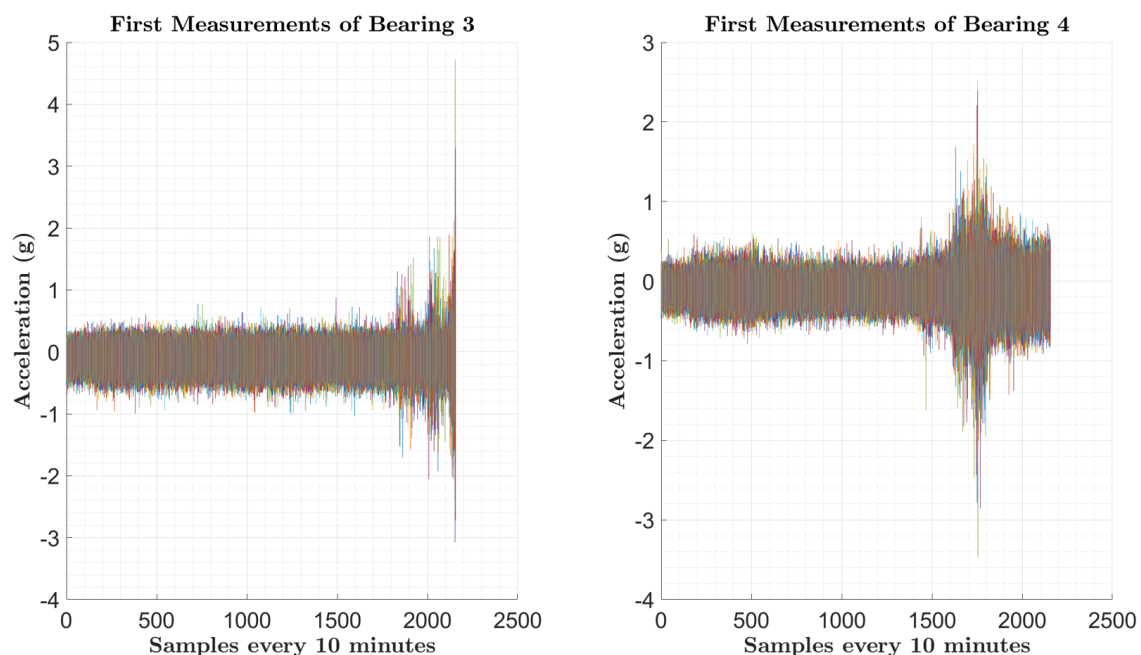
Σύνολα Μετρήσεων	Αρχεία Μετρήσεων *	Εδ. Κύλισης Αστοχίας	Βλάβη
No. 1	2156	Εδ. Κύλισης 3 Εδ. Κύλισης 4	Εσ. Δακτύλιος Στ. Κύλισης
No. 2	984	Εδ. Κύλισης 1	Εξ. Δακτύλιος
No. 3	6324	Εδ. Κύλισης 3	Εξ. Δακτύλιος

* Ένα κάθε 10 λεπτά, διάρκειας 1s .

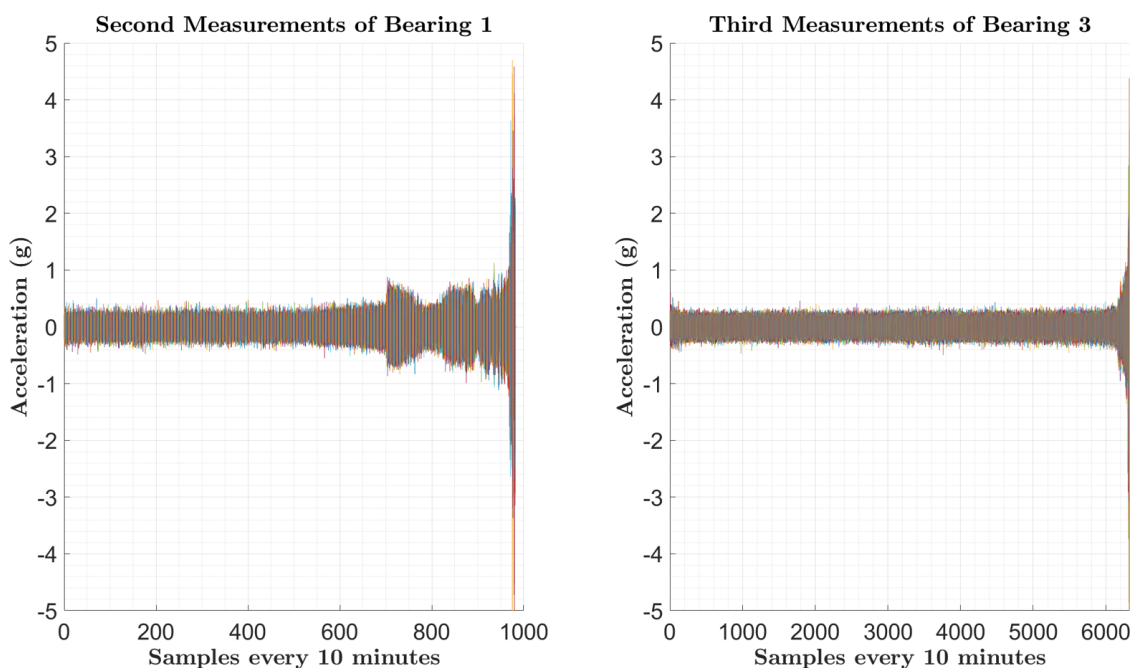
3.4 Ανασκόπηση Μετρήσεων

3.4.1 Ανασκόπηση Μετρήσεων στο πεδίο του χρόνου

Οι πρωτογενείς μετρήσεις (raw measurements) που λαμβάνονται από τα επιταχυνσιόμετρα, παρουσιάζονται στα **Σχήματα 5, 6**. Η βλάβη που εμφανίζεται κάθε φορά, βασίζεται στον **Πίνακα 1**. Είναι σημαντικό να σημειωθεί πως οι μετρήσεις διαρκούν 1s , και γίνονται κάθε 10min . Ωστόσο για να παρουσιαστούν σε γράφημα, παρουσιάζονται διαδοχικά.



Σχήμα 5: Ανασκόπηση μετρήσεων πρώτου συνόλου, στο πεδίο του χρόνου.



Σχήμα 6: Ανασκόπηση μετρήσεων δεύτερου και τρίτου συνόλου, στο πεδίο του χρόνου.

Από τα **Σχήματα 5, 6**, παρατηρείται πως όσο εξελίσσεται η βλάβη, τα επίπεδα των επιταχύνσεων αυξάνονται.

3.4.2 Ανασκόπηση Μετρήσεων στο πεδίο συχνότητας

Τα χαρακτηριστικά του εδράνου κύλισης Rexnord ZA-2115 παρουσιάζονται στον **Πίνακα 2**. Από τις **Σχέσεις (1),(2),(3)**, υπολογίζονται οι χαρακτηριστικές συχνότητες βλάβης των εδράνων κύλισης, όπου λαμβάνουν τις τιμές:

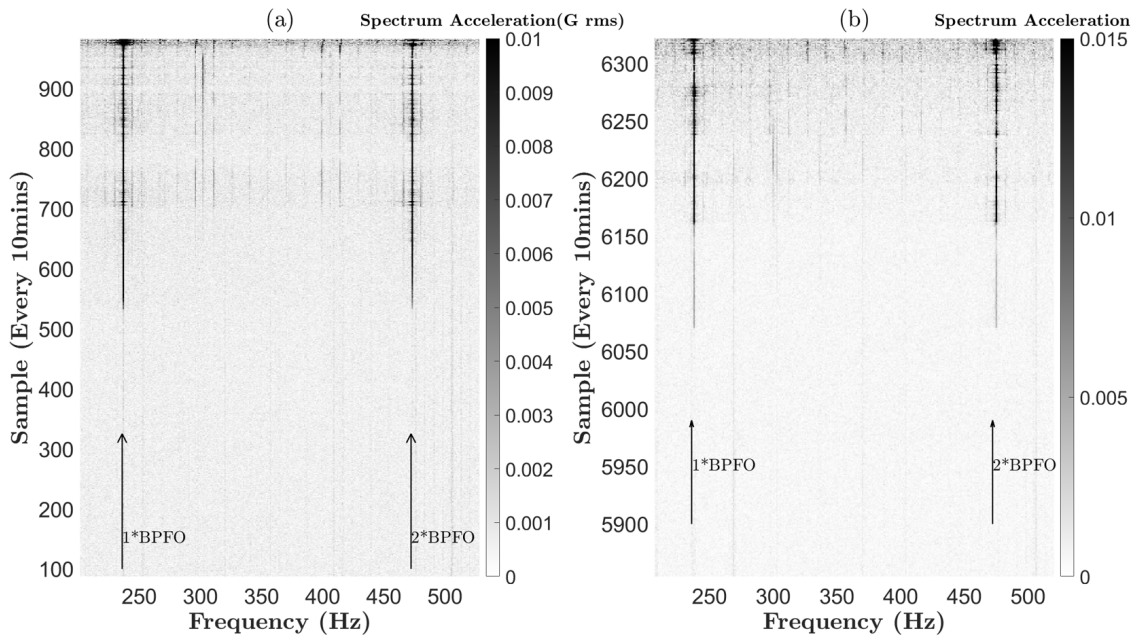
$$BPFO = 236Hz, \quad BPF1 = 297Hz, \quad BSF = 278Hz$$

Πίνακας 2: Χαρακτηριστικά των εδράνων κύλισης Rexnord ZA-2115

Rexnord ZA-2115		
Pitch diameter	2.815 inch	71.5mm
Rolling element diameter	0.331 inch	8.4mm
Number of rolling element per row	16	16
Contact angle	15.17deg	15.17deg

Η μετάβαση από το πεδίο του χρόνου στο πεδίο των συχνοτήτων γίνεται με τον διακριτό μετασχηματισμό Fourier (DFT). Στο **Σχήμα 7**, γίνεται αναπαράσταση της εξέλιξης του φάσματος των συχνοτήτων. Κάθε σημείο (δείγμα, συχνότητα) λαμβάνει μια τιμή επιτάχυνσης και για να απεικονιστεί σε διδιάστατο επίπεδο, χρωματίζεται

στην κλίμακα του γκρι. Τα σύνολα των μετρήσεων που τα φάσματα τους παρουσιάζονται, είναι αυτά που εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο του εδράνου κύλισης.



Σχήμα 7: Χρονικά φάσματα συχνοτήτων για (α) Σύνολο μετρήσεων 2 - έδρανο κύλισης 1, (β) Σύνολο μετρήσεων 3 - έδρανο κύλισης 3

Στο **Σχήμα 7**, γίνεται εμφανής η παρουσία της χαρακτηριστικής συχνότητα του εξωτερικού δακτύλιου (BPFO) και οι αρμονικές της (στο Σχήμα, φαίνεται μόνο η δεύτερη αρμονική). Για αρχή προσεγγιστικά αναφέρεται, πως η αστοχία των εδράνων κύλισης, βάσει των φασμάτων, ξεκινάει τις χρονικές στιγμές 500 και 6100 για τα σύνολα μετρήσεων 2 και 3 αντίστοιχα.

Στα χρονικά φάσματα του συνόλου μετρήσεων 1, όπου υπάρχει βλάβη στον εσωτερικό δακτύλιο του εδράνου κύλισης 3 και βλάβη στα στοιχεία κύλισης του εδράνου κύλισης 4, δεν παρατηρούνται οι χαρακτηριστικές συχνότητες βλάβης και για αυτό δεν παρουσιάζονται. Οι χαρακτηριστικές συχνότητες βλαβών, θα μπορούσαν να εμφανιστούν αποδιαμορφώνοντας το πρωτογενές σήμα σε μια ζώνη κοντά στην υψίσυχνη συχνότητα. Ωστόσο σε αυτή την ενότητα γίνεται απλά ανασκόπηση και δεν γίνεται περαιτέρω προ-επεξεργασία.

4 Εξαγωγή Χαρακτηριστικών

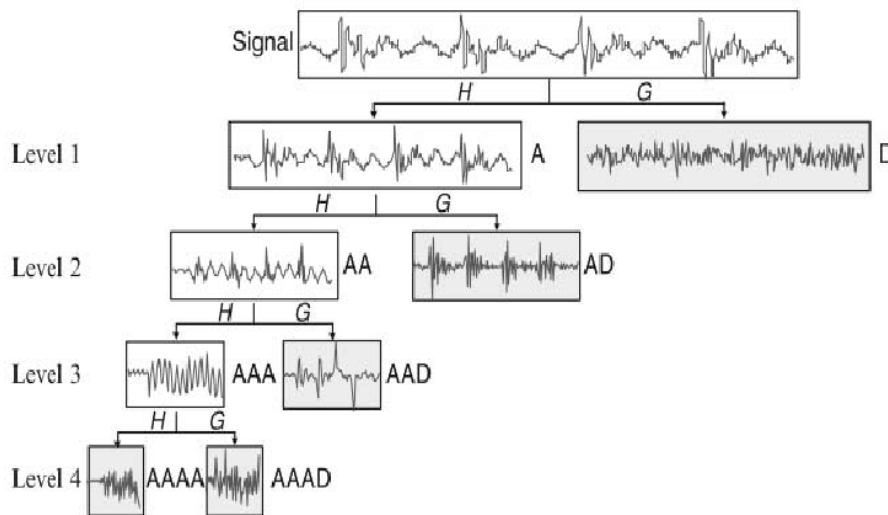
4.1 Αποθορυβοποίηση σήματος

Τα σήματα που λαμβάνονται από τα επιταχυνσιόμετρα (x_n) περιλαμβάνουν, εκτός από την επιθυμητή πληροφορία των μετρήσεων στο έδρανο κύλισης (s_n) που είναι εγκατεστημένα, θόρυβο από τα υπόλοιπα μηχανολογικά στοιχεία (σw_n). Αυτός ο θόρυβος πηγάζει από τα υπόλοιπα έδρανα κύλισης, τον κινητήρα, τον ιμάντα και τον άξονα της εγκατάστασης (βλ. **Σχήμα 4**). Η μορφή του σήματος x_n προκύπτει από τον συνδυασμό αυτών των δύο σημάτων.

$$x_n = s_n + \sigma w_n \quad (10)$$

Όπου σ τα επίπεδα θορύβου, και w_n στοιχεία από την γκαουσιανή κατανομή. Στόχος της αποθορυβοποίησης του σήματος, είναι η αφαίρεση του θορύβου σw_n από τις μετρήσεις του κάθε εδράνου κύλισης, ώστε να γίνει σωστά η διάγνωση του εδράνου κύλισης. Η αποθορυβοποίηση γίνεται με την βοήθεια της μεθόδου της κυματιδιακής αποσύνθεσης (Wavelet decomposition) κατά την οποία το αρχικό σήμα διαχωρίζεται σε δύο επιμέρους σήματα, την προσεγγιστική συνιστώσα και τη συνιστώσα πληροφορίας, οι οποίες περιέχουν πληροφορίες χαμηλών και υψηλών συχνοτήτων αντίστοιχα.

Η διαδικασία αποσύνθεσης γίνεται συνήθως σε παραπάνω από ένα επίπεδο, επαναλαμβάνεται δηλαδή N φορές. Εργασίες από τους Αντωνιάδη, 2016 [14] και Randall, 2011 [15], που επεξεργάζονται σήματα από μετρήσεις σε έδρανα κύλισης, προτείνουν τιμές $N=3-5$. Στο **Σχήμα 8**, παρουσιάζεται από τον Haider, 2015 [16] η πολυεπίπεδη αποσύνθεση.

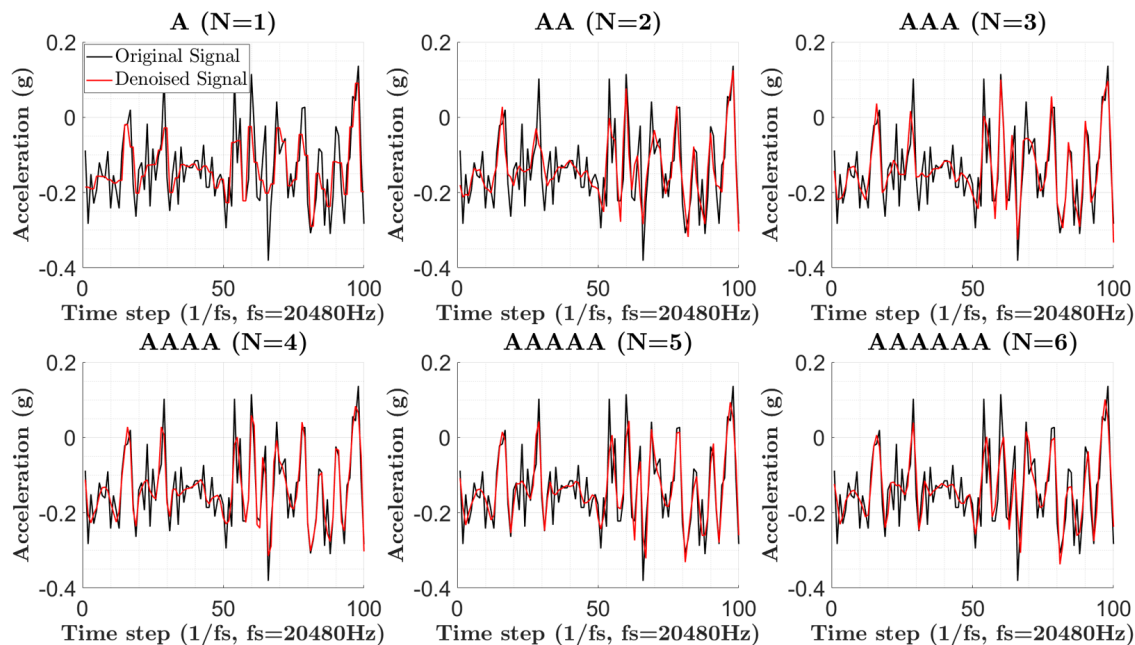


Note: H - Low pass filter; G - High pass filter; A - Approximate information; D - Detailed information

Σχήμα 8: Επίπεδα αποσύνθεσης Wavelet. Πηγή: Haider, 2015 [16]

Στο **Σχήμα 9**, παρουσιάζεται το θορυβώδες σήμα από τις πειραματικές μετρήσεις, σε συνδυασμό με κάθε φορά το N φορές αποσυνθεμένο σήμα. Στην εργασία αυτή,

κρίνεται πως η αποσύνθεση μετά την τέταρτη φορά δεν βελτιώνει την αποθρομβοποίηση του σήματος και μπορεί να προκαλέσει κίνδυνο για την εξαφάνιση σημαντικών πληροφοριών όπως είναι οι κρουστικοί παλμοί. Επομένως, για την εξαγωγή χαρακτηριστικών (Feature Extraction επιλέγεται το σήμα AAAA($N = 4$)).



Σχήμα 9: Αποτελέσματα της διαδικασίας αποθρομβοποίησης για διαφορετικό αριθμό επαναλήψεων.

4.2 Χαρακτηριστικά σημάτων

Το σήμα που λαμβάνεται από το επιταχυνσιόμετρο αποτελεί τη μοναδική μέτρηση που πραγματοποιείται σε κάθε έδρανο κύλισης. Ωστόσο, αυτές οι μετρήσεις δεν είναι κατάλληλες για την ανάπτυξη ενός νευρωνικού δικτύου. Μετά την αποθρομβοποίηση του σήματος, πραγματοποιείται εξαγωγή χαρακτηριστικών (features) από το πεδίο του χρόνου (time domain) και από το πεδίο των συχνοτήτων (frequency domain), με σκοπό τη συλλογή όσο το δυνατόν περισσότερων πληροφοριών σχετικά με την τρέχουσα κατάσταση του εδράνου κύλισης.

Στον **Πίνακα 3**, παρουσιάζονται τα χαρακτηριστικά στο πεδίο του χρόνου που επιλέχθηκαν να εξεταστούν, και οι εξισώσεις που τα περιγράφουν.

Τα πρώτα τέσσερα στοιχεία του **Πίνακα 3** περιγράφουν τη θέση και διασπορά μιας κανονικής κατανομής, τα σήματα που λαμβάνονται έχουν αυτήν την μορφή. Η μέση τιμή (1) συμβολίζει τη θέση της καμπάνας, η τυπική απόκλιση (2) το πλάτος της καμπάνας, η ασυμμετρία (3) την τάση της κατανομής προς την μία κατεύθυνση και η κύρτωση (4) τη μορφή της κατανομής. Η κορυφή προς κορυφή (5) είναι η μέγιστη διαφορά των άκρων του σήματος, συμβολικό χαρακτηριστικό για τον εντοπισμό κρουστικών παλμών. Η μέση τετραγωνική ρίζα (6) του σήματος, παρουσιάζει τα επίπεδα των κραδασμών, όπου παραδοσιακά, βάσει της τιμή αυτής, χρησιμοποιείται το πρότυπο ISO 10816-3 για την αξιολόγηση της ασφαλούς κατάστασης των εδράνων κύλισης και των υπόλοιπων μηχανολογικών στοιχείων. Οι συντελεστές (7-10), ανεξάρτητα

Πίνακας 3: Χαρακτηριστικά σήματος στο πεδίο του χρόνου.

Χαρακτηριστικά Σήματος	Εξισώσεις *
1. Mean	$\bar{v} = \frac{\sum v_i}{N}$
2. Standard Derivation	$std = \sqrt{\frac{\sum (v_i - \bar{v})^2}{N-1}}$
3. Skewness	$skew = \frac{1}{N} \frac{\sum (v_i - \bar{v})^3}{std^3}$
4. Kurtosis	$kurt = \frac{E((v - \bar{v})^4)}{std^4}$
5. Peak2Peak	$p2p = \max(v) - \min(v)$
6. RMS	$rms = \sqrt{\frac{1}{N} \sum v_i^2}$
7. CrestFactor	$CF = \max(v)/rms$
8. ShapeFactor	$SF = rms/E(v)$
9. ImpulseFactor	$IF = \max(v)/E(v)$
10. MarginFactor	$MF = rms/E(v)^2$
11. Energy	$ener = \sum v_i^2$
12. RMSEE	$rmsee = -rms \cdot \log(rms)$

* Με $E(x)$ συμβολίζεται η μέση τιμή του διανύσματος x .

από την ταχύτητα περιστροφής του εδράνου κύλισης, παραμένουν σταθεροί καθώς η τιμή RMS, η απόλυτη μέση τιμή και η μέγιστη τιμή αυξάνονται. Εάν η κατάσταση του εδράνου κύλισης επιδεινώνεται, η τιμή των συντελεστών μεταβάλλεται καθώς εμφανίζονται χρουστικά φαινόμενα. Η ενέργεια (11), συμβολίζει τη συνολική ενέργεια του επιταχυνσιόμετρου κατά το χρόνο της μέτρησης (1s). Τέλος το χαρακτηριστικό RMSEE, εισήχθη από τον Jaouher, 2015 [17] σε προσπάθεια να γίνουν πιο ομαλές οι περιοχές διακύμανσης.

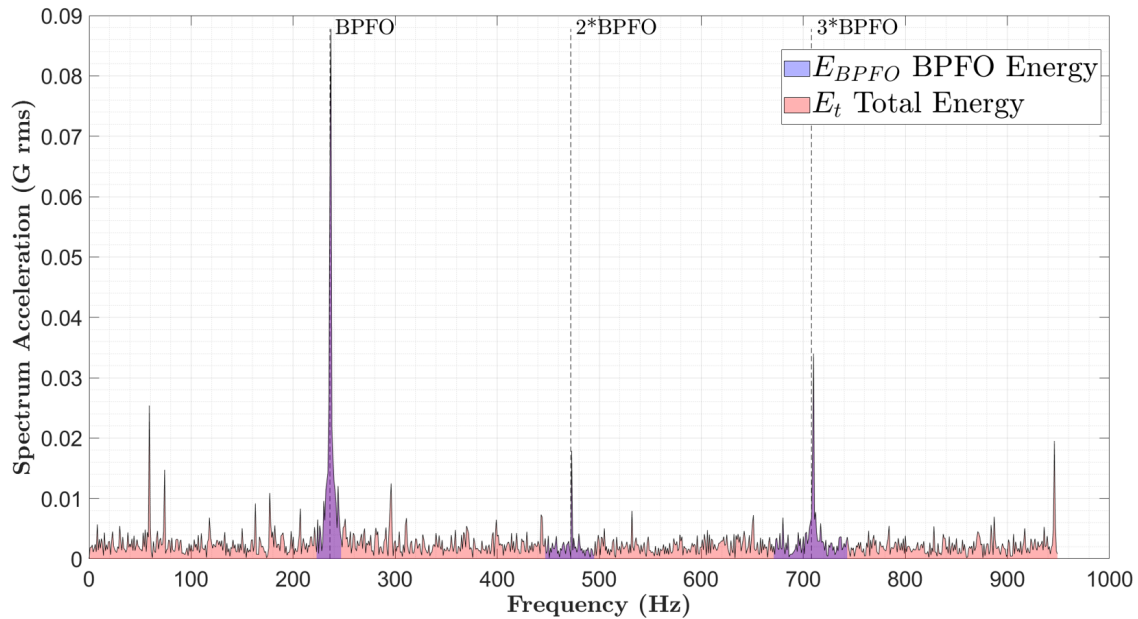
Η προσέγγιση για τα χαρακτηριστικά στο πεδίο των συχνοτήτων είναι παρόμοια, καθώς και εκεί η κατανομή των μέτρων (U_i) της κάθε συχνότητας (f_i), έχει τάση προς κανονική κατανομή. Στον **Πίνακα 4**, παρουσιάζονται τα χαρακτηριστικά στο πεδίο της συχνότητας που επιλέχθηκαν να εξεταστούν, και οι εξισώσεις που τα περιγράφουν.

Τα επιπλέον χαρακτηριστικά (17-19) περιγράφουν το ποσοστό της πιθανής ενέργειας βλάβης ως προς τη συνολική. Οι συγκεκριμένοι συντελεστές, όπως παρατηρήθηκε στην συγκεκριμένη διπλωματική, είναι αρκετά σημαντικοί στην ανίχνευση της βλάβης (Fault Detection) του εδράνου κύλισης. Στο **Σχήμα 10**, παρουσιάζεται το πεδίο συχνοτήτων σήματος όπου έχει εντοπιστεί βλάβη στον εξωτερικό δακτύλιο. Το εμβαδόν με κόκκινο χρώμα συμβολίζει την ενέργεια στο πεδίο των συχνοτήτων ($f = 0 : f_6$), ενώ με μπλε χρώμα συμβολίζει την ενέργεια στο πεδίο των πρώτων τριών αρμονικών της *BPFO*. Αντίστοιχος είναι και ο τρόπος υπολογισμού και για τους συντελεστές 18,19.

Πίνακας 4: Χαρακτηριστικά σήματος στο πεδίο των συχνοτήτων.

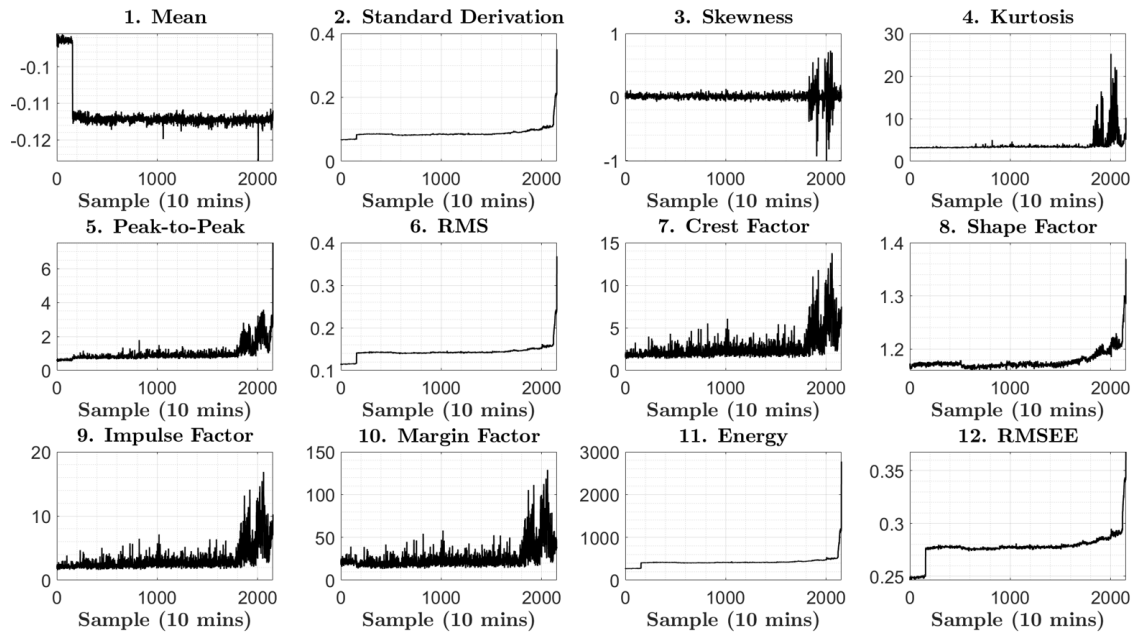
Χαρακτηριστικά Σήματος	Εξισώσεις *
13. SK Mean	$\bar{U} = \frac{\sum U_i}{N}$
14. SK Standard Derivation	$SKstd = \sqrt{\frac{\sum (U_i - \bar{U})^2}{N-1}}$
15. SK Skewness	$SKskew = \frac{1}{N} \frac{\sum (U_i - \bar{U})^3}{SKstd^3}$
16. SK Kurtosis	$SKkurt = \frac{E((U - \bar{U})^4)}{SKstd^4}$
17. BPFO Factor	$F_{BPFO} = \frac{1}{E_t} \sum^{j=1:3} \left(\sum^{f_i=f_1:f_2} U(j \cdot f_i) + 0.5(U(j \cdot f_1) + U(j \cdot f_2)) \right)$
18. BPF1 Factor	$F_{BPF1} = \frac{1}{E_t} \sum^{j=1:3} \left(\sum^{f_i=f_3:f_4} U(j \cdot f_i) + 0.5(U(j \cdot f_3) + U(j \cdot f_4)) \right)$
19. BSF Factor	$F_{BSF} = \frac{1}{E_t} \sum^{j=1:3} \left(\sum^{f_i=f_5:f_6} U(j \cdot f_i) + 0.5(U(j \cdot f_5) + U(j \cdot f_6)) \right)$

* Όπου $E_t = 0.5(U(1) + U(f_s)) + \sum U_i$, η συνολική ενέργεια του φάσματος των συχνοτήτων, ο υπολογισμός γίνεται με την μέθοδο των τραπεζιών. Οι συμβολικές συχνότητες $f_1, f_2, f_3, f_4, f_5, f_6$ συμβολίζουν τα άκρα της ολοκλήρωσης και συγκεκριμένα λαμβάνουν τις τιμές, $f_1 = 0.95f_{BPFO}, f_2 = 1.05f_{BPFO}, f_3 = 0.95f_{BPF1}, f_4 = 1.05f_{BPF1}, f_5 = 0.95f_{BSF}, f_6 = 1.05f_{BSF}$

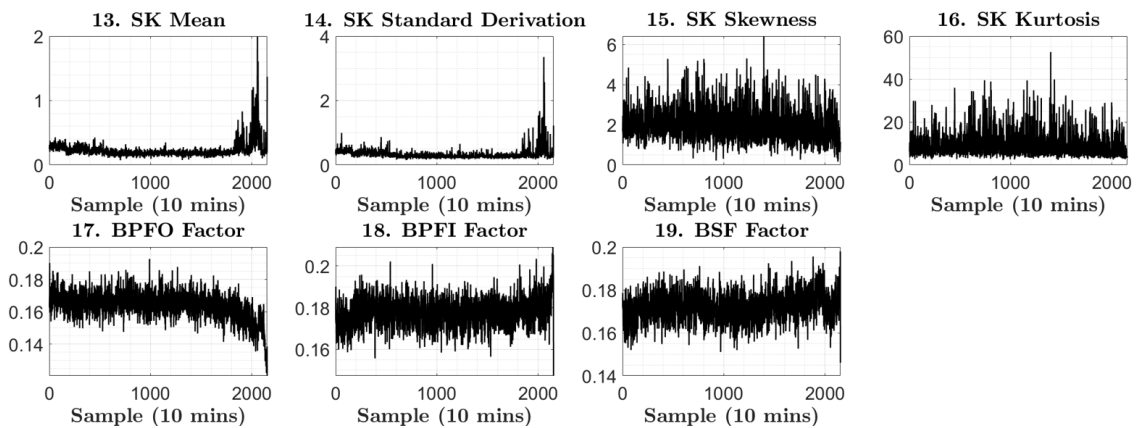


Σχήμα 10: Κατανόηση των χαρακτηριστικών συντελεστών ενέργειας .

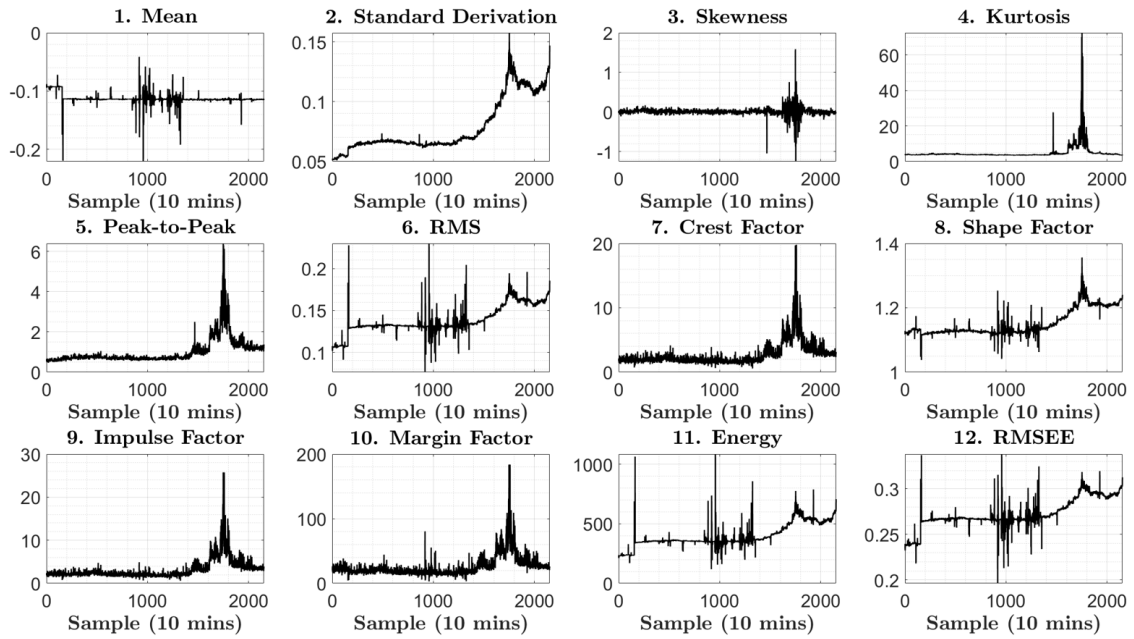
Τα δεκαεννιά (19) χαρακτηριστικά που επιλέχθηκαν υπολογίζονται για κάθε ένα από τα τέσσερα σύνολα μετρήσεων στα οποία εντοπίστηκαν βλάβες και παρουσιάζονται στα Σχήματα 11, 12, 13, 14, 15, 16, 17, 18.



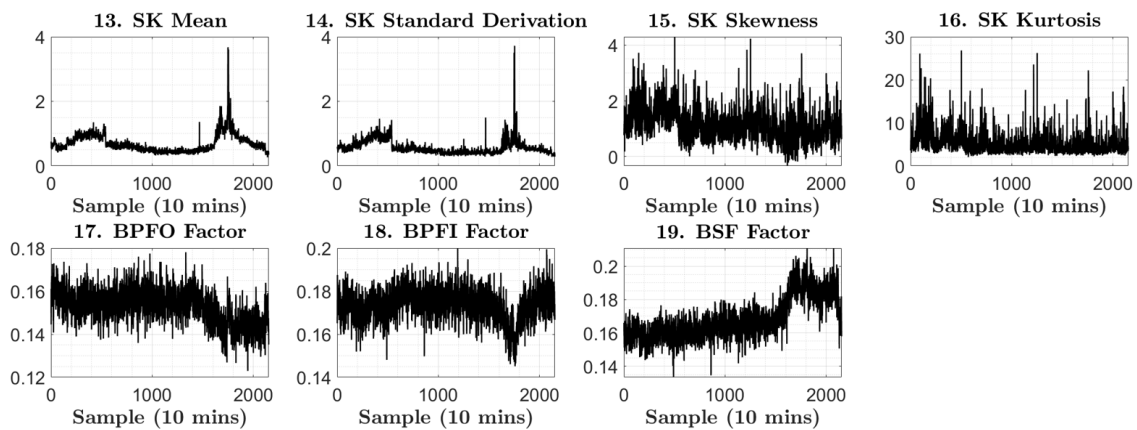
Σχήμα 11: Χαρακτηριστικά του πεδίου χρόνου, από το πρώτο σύνολο μετρήσεων στο 3^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εσωτερικό δακτύλιο.



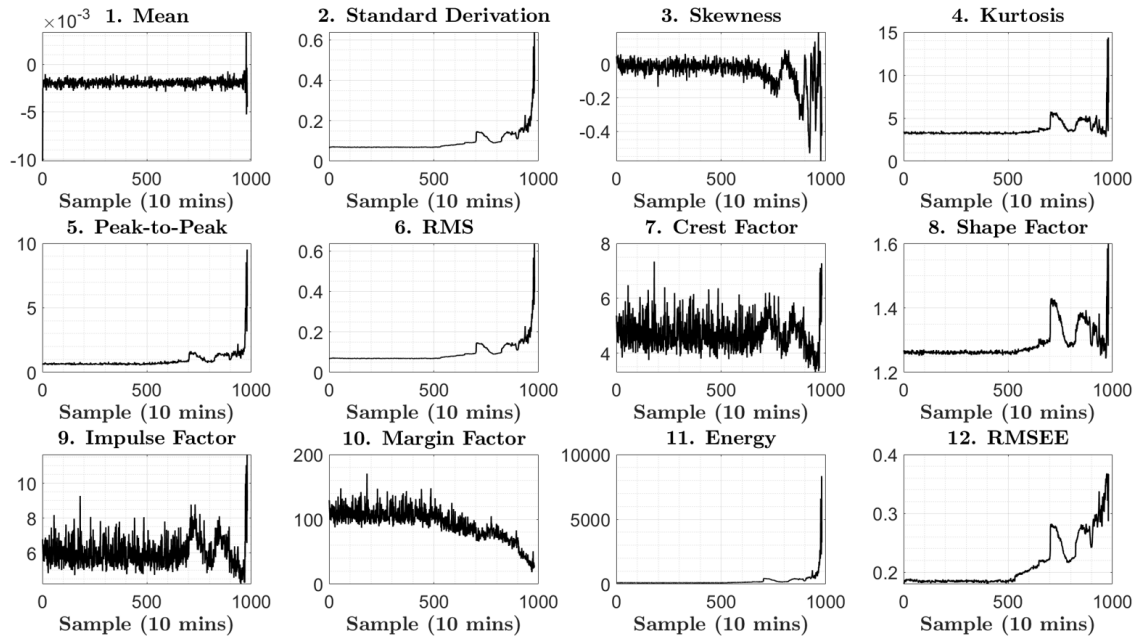
Σχήμα 12: Χαρακτηριστικά του πεδίου συχνοτήτων, από το πρώτο σύνολο μετρήσεων στο 3^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εσωτερικό δακτύλιο.



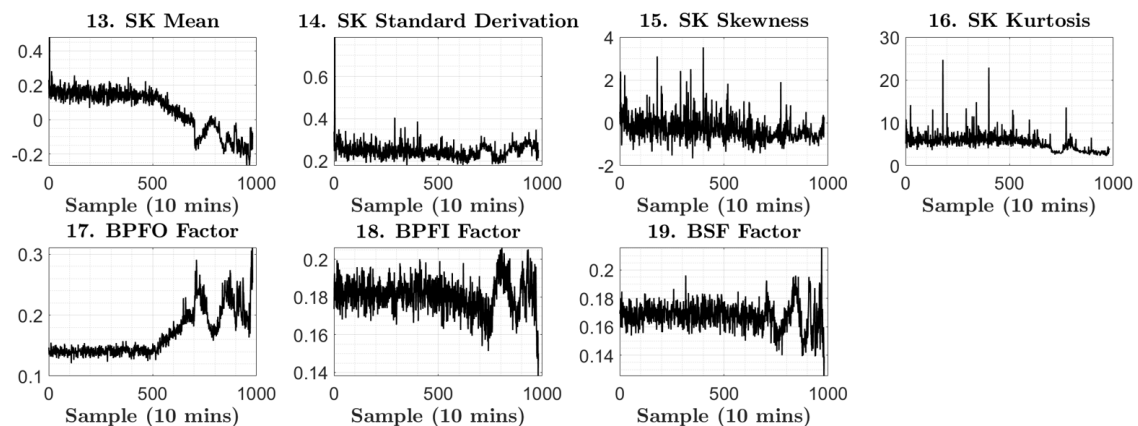
Σχήμα 13: Χαρακτηριστικά του πεδίου χρόνου, από το πρώτο σύνολο μετρήσεων στο 4^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στα στοιχεία κύλισης.



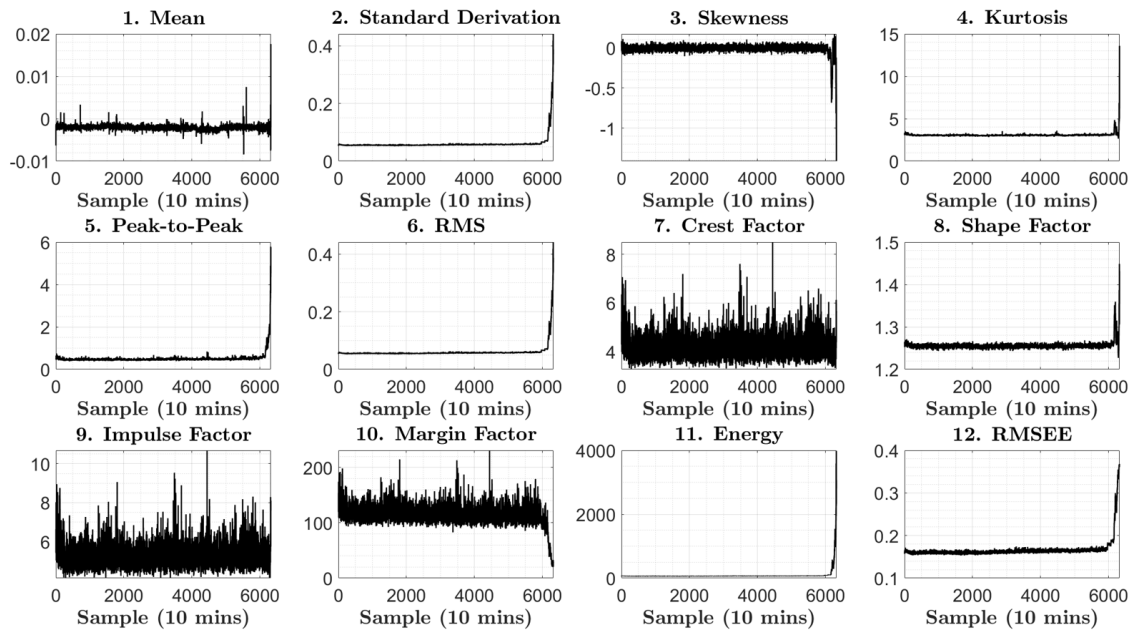
Σχήμα 14: Χαρακτηριστικά του πεδίου συχνοτήτων, από το πρώτο σύνολο μετρήσεων στο 4^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στα στοιχεία κύλισης.



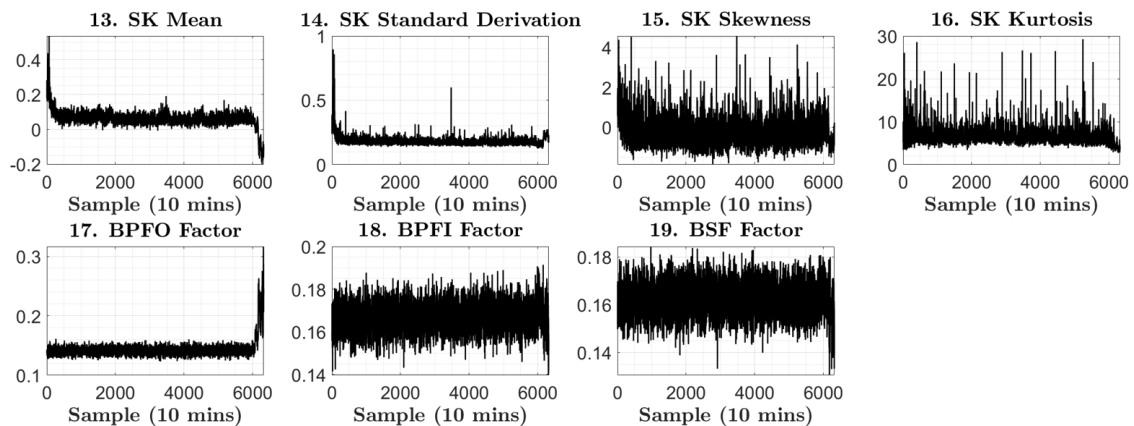
Σχήμα 15: Χαρακτηριστικά του πεδίου χρόνου, από το δεύτερο σύνολο μετρήσεων στο 1^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.



Σχήμα 16: Χαρακτηριστικά του πεδίου συχνοτήτων, από το δεύτερο σύνολο μετρήσεων στο 1^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.



Σχήμα 17: Χαρακτηριστικά του πεδίου χρόνου, από το τρίτο σύνολο μετρήσεων στο 3^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.



Σχήμα 18: Χαρακτηριστικά του πεδίου συχνότητας, από το τρίτο σύνολο μετρήσεων στο 3^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.

Γίνονται οι εξής παρατηρήσεις:

- Μερικά από τα χαρακτηριστικά που επιλέχθηκαν, έχουν παρόμοια συμπεριφορά S.D., RMS, Energy, το οποίο είναι αναμενόμενο καθώς οι σχέσεις του είναι παρόμοιες.
- Εμπειρικά, η μέση τετραγωνική ρίζα RMS του σήματος, ήταν κριτήριο για την πρόβλεψη της κατάστασης του εδράνου κύλισης. Εδώ επιβεβαιώνεται η επιλογή αυτού του μέγεθος, καθώς όταν αυξάνεται το RMS και τα αντίστοιχα μεγέθη αυξάνεται και η πρόοδος της βλάβης.
- Η χρονική συμπεριφορά πολλών χαρακτηριστικών δεν είναι ομαλή.

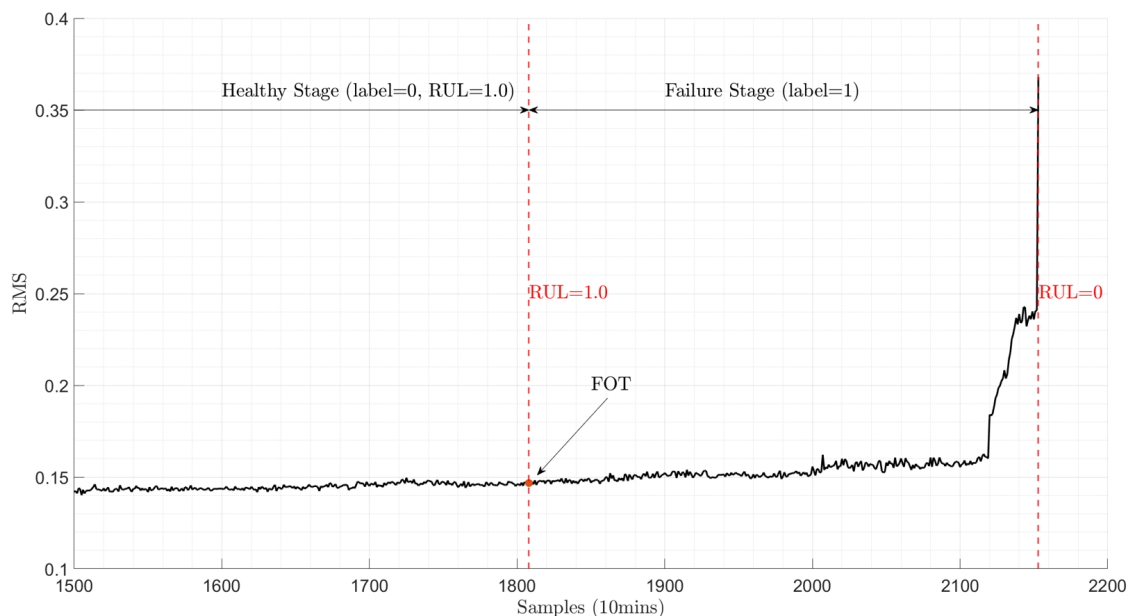
- Ανάλογα με την βλάβη που εμφανίζεται στο έδρανο κύλισης, μεταβάλλονται σημαντικά οι παράγοντες ενέργειες $FBPFO$, $FBPFI$, $FBSF$.

4.3 Κατηγοριοποίηση δεδομένων

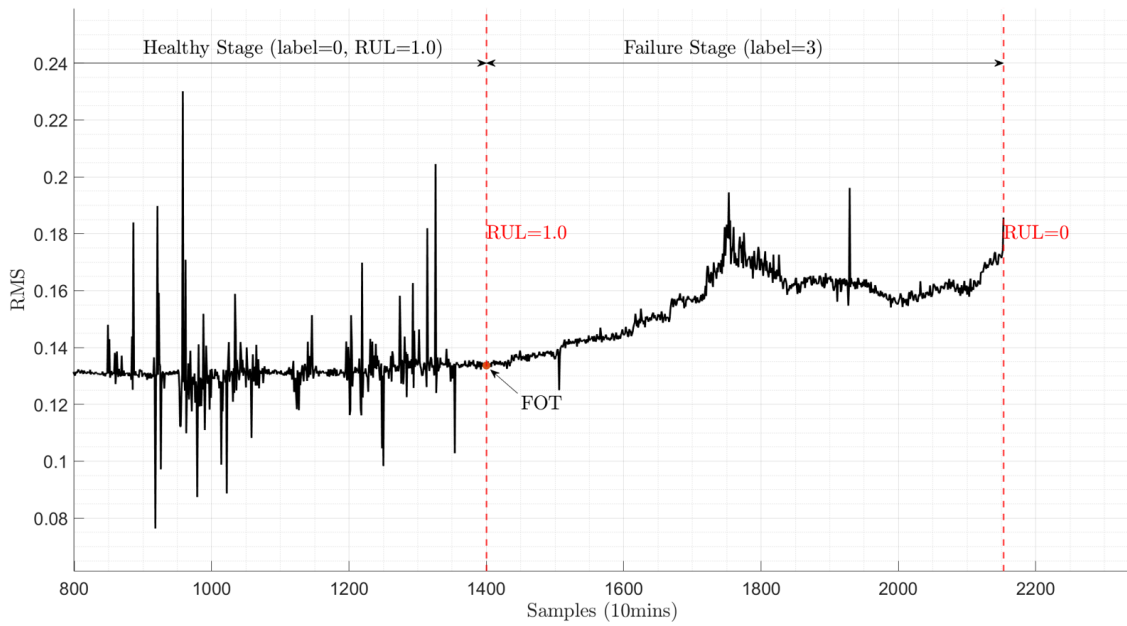
Η λειτουργία του εδράνου κύλισης κατηγοριοποιείται σε ασφαλή λειτουργία (Healthy Stage) και σε λειτουργία που έχει αναπτυχθεί βλάβη (Failure Stage). Το σημείο που εμφανίζεται η βλάβη, δηλαδή η μετάβαση από την ασφαλή λειτουργία, στην λειτουργία υπό βλάβη, ονομάζεται FOT (fault occurrence time). Για τον εντοπισμό του σημείου αυτού υπάρχουν διάφοροι μέθοδοι όπως η μέθοδος των κρυφών Μακροβιανών μοντέλων (Hidden Markov Models - HMM) που δοκιμάστηκε από τον Jun, 2020 [18]. Σε αυτήν την εργασία δεν γίνεται κάποια συγκεκριμένη ανάλυση για τον εντοπισμό των FOT, αλλά οι τιμές λαμβάνονται από τον Zhang, 2018 [19], ο οποίος επεξεργάζεται τις ίδιες μετρήσεις.

Η κατηγοριοποίηση (categorization) γίνεται ως εξής: Με $category = 0$, συμβολίζεται η ασφαλής λειτουργία, $category = 1$, η λειτουργία υπό βλάβη εσωτερικού δακτύλιου, $category = 2$, η λειτουργία υπό βλάβη εξωτερικού δακτύλιου, $category = 3$, η λειτουργία υπό βλάβη στοιχείων κύλισης. Στη συνέχεια προστίθενται στα δεδομένα, και οι πραγματικές τιμές του RUL. Οι χρονικές στιγμές όπου η λειτουργία του εδράνου κύλισης είναι ασφαλής, λαμβάνουν τιμή ($RUL = 1.0$ ή 100%), μετά τη στιγμή του FOT το RUL μεταβάλλεται γραμμικά (καθώς οι μετρήσεις έχουν σταθερό χρονικό βήμα) μέχρι την τελική αστοχία όπου ($RUL = 0$ ή 0%).

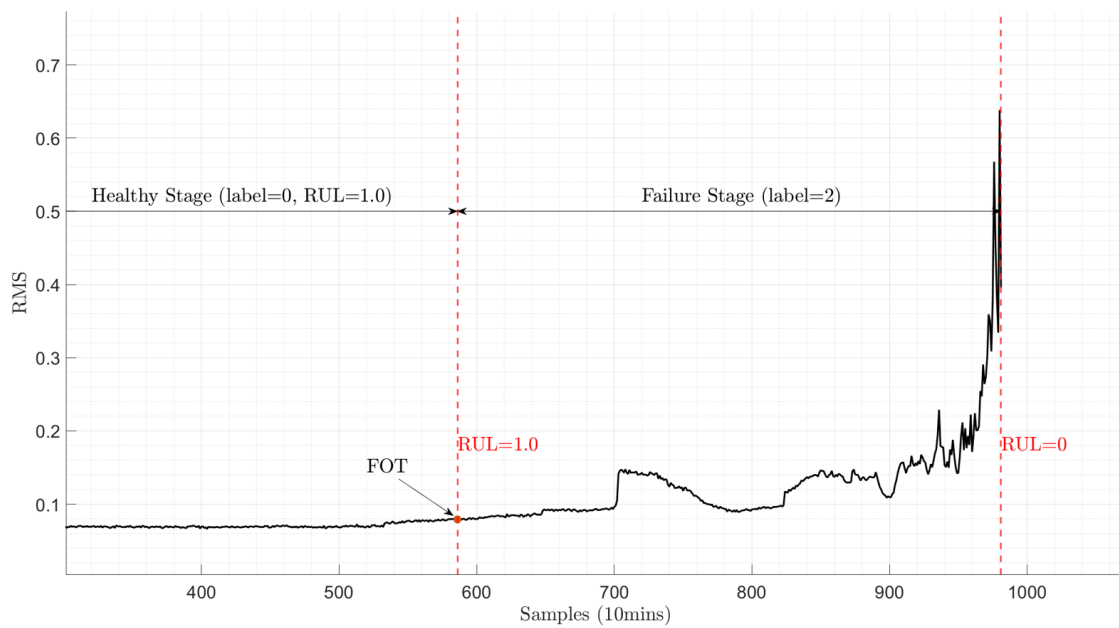
Στα **Σχήματα 19,20,21,22** παρουσιάζονται οι στιγμές FOT πάνω στο χρονοδιάγραμμα του RMS. Οι στιγμές FOT που λήφθηκαν από τον Zhang, 2018 [19] κρίνονται ικανοποιητικές. Αξίζει να σημειωθεί πως οι στιγμές FOT δεν μπορούν να εκτιμηθούν επακριβώς, καθώς η εμφάνιση βλάβης δεν γίνεται ακαριαία.



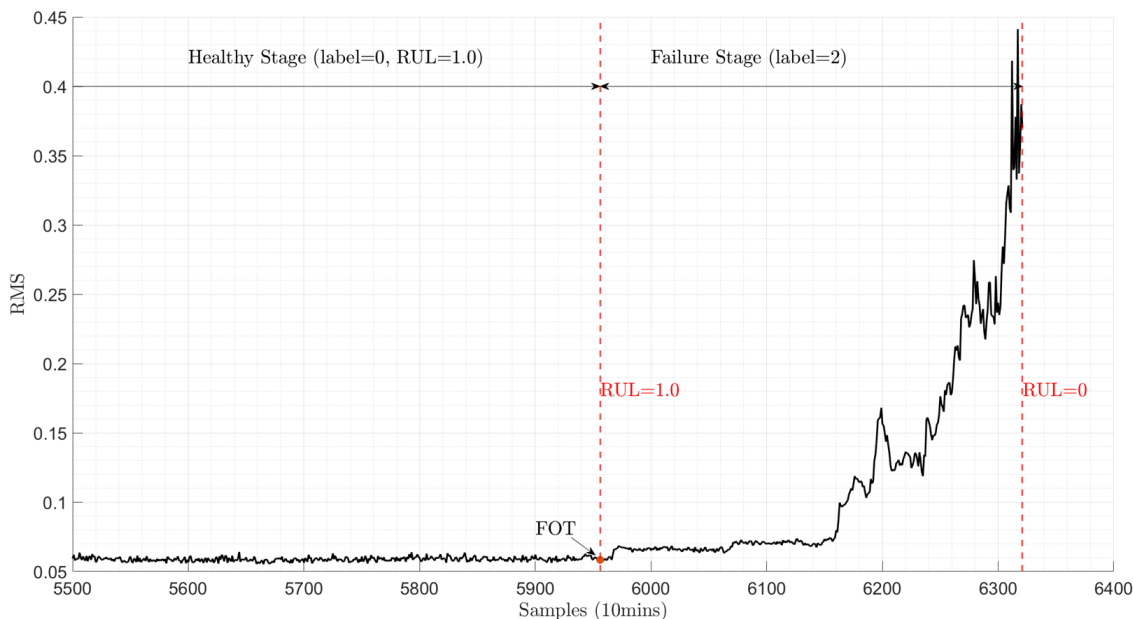
Σχήμα 19: Τιμές RMS από το πρώτο σύνολο μετρήσεων στο 3ο έδρανο κύλισης.



Σχήμα 20: Τιμές RMS από το πρώτο σύνολο μετρήσεων στο 4ο έδρανο κύλισης.



Σχήμα 21: Τιμές RMS από το δεύτερο σύνολο μετρήσεων στο 1ο έδρανο κύλισης.



Σχήμα 22: Τιμές RMS από το τρίτο σύνολο μετρήσεων στο 3ο έδρανο κύλισης.

4.4 Επιλογή Χαρακτηριστικών

Από τα χαρακτηριστικά του σήματος που έχουν επιλεγεί να εξεταστούν, επιλέγεται ένας κατάλληλος συνδυασμός ως είσοδοι στο νευρωνικό δίκτυο της κάθε βλάβης για την εκτίμηση της υπολειπόμενης ωφέλιμης διάρκειας ζωής (RUL). Υπάρχουν διάφορες προσεγγίσεις, για την επιλογή του κατάλληλου συνδυασμού. Παρακάτω αναφέρονται μερικές:

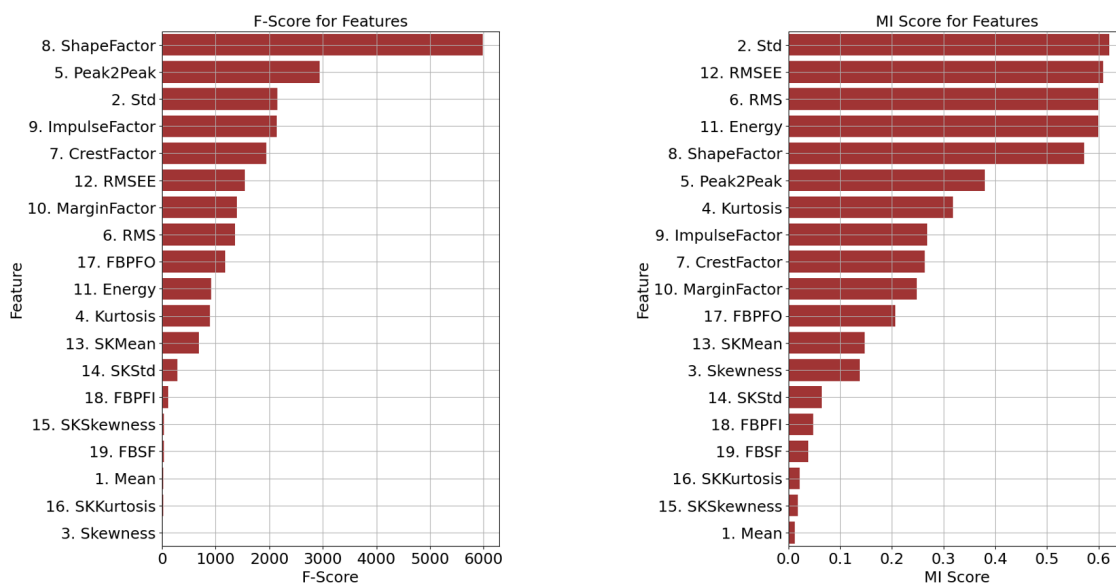
- **Εμπειρική προσέγγιση:** Εμπειρικά επιλέγονται τα χαρακτηριστικά που γνωρίζουμε ότι έχουν επίδραση στο προβλεπόμενο μέγεθος. Χαρακτηριστικό μέγεθος στην περίπτωση μας το RMS.
- **Επαναληπτική προσέγγιση:** Εξετάζονται όλοι οι συνδυασμοί των χαρακτηριστικών, και αξιολογούνται βάσει μιας αντικειμενικής συνάρτησης. Στην περίπτωση μας, οι συνδυασμοί είναι ($2^{19} = 524k$) πράγμα απαγορευτικό για την επιλογή αυτής της μεθόδου.
- **PCA προσέγγιση:** Η μέθοδος PCA (Principal Component Analysis), μετατρέπει το αρχικό σετ των χαρακτηριστικών, σε σετ με μικρότερες διαστάσεις (συνήθως 2-4). Πλέον οι μεταβλητές εισόδου είναι οι PCA_1, PCA_2, \dots . Αρνητικό της μεθόδου αυτής, είναι πως χάνεται η φυσική ερμηνεία των μεταβλητών.
- **Στατιστική προσέγγιση:** Γίνονται στατιστικά τεστ ως προς τη συσχέτιση μεταξύ χαρακτηριστικού/στόχου (feature/target), όπου στην περίπτωση μας ο στόχος είναι το RUL. Τα δύο στατιστικά τεστ που γίνονται είναι το F-test και το MI-test (Mutual Information). Το F-test εντοπίζει αν υπάρχει γραμμική συσχέτιση ενώ το MI-test οποιαδήποτε συσχέτιση μεταξύ δύο μεταβλητών. Επιλέγονται οι τιμές με τις υψηλότερες τιμές των τεστ, και στην συνέχεια εξετάζεται αν υπάρχει συσχέτιση μεταξύ των χαρακτηριστικών, για την αποφυγή

επιλογής παρόμοιων χαρακτηριστικών. Οι τιμές της συσχέτισης, παρουσιάζονται στον πίνακα συσχέτισης (correlation matrix), οι οποίες υπολογίζονται από τη σχέση του Pearson.

Στην παρούσα εργασία επιλέγεται η τελευταία μέθοδος. Στις παρακάτω υποενότητες, γίνεται η επιλογή του κατάλληλου συνδυασμού των χαρακτηριστικών για κάθε νευρωνικό δίκτυο του κάθε τύπου βλάβης. Για το συνδυασμό των χαρακτηριστικών του νευρωνικού δικτύου ανίχνευσης βλάβης, όπου στην περίπτωση αυτή είναι πρόβλημα πολυταξικής ταξινόμησης (Multiclass classification), επιλέγονται χαρακτηριστικά από τα προηγούμενα νευρωνικά δίκτυα.

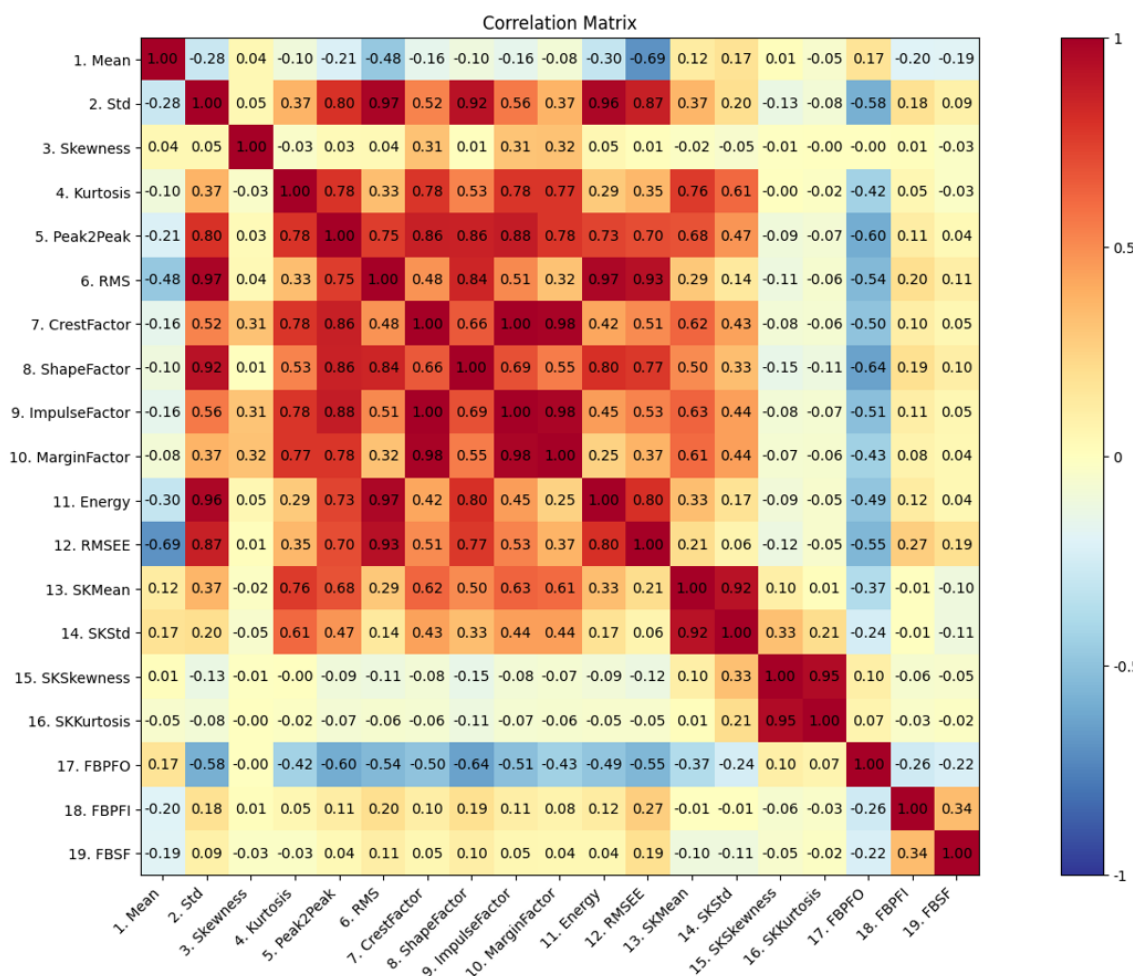
4.4.1 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης RUL, με βλάβη στον εσωτερικό δακτύλιο

Τα χαρακτηριστικά που επιλέγονται είναι αυτά των Σχημάτων 11, 12. Οι τιμές του στόχου RUL λαμβάνουν τιμές βάσει του Σχήματος 19. Τα στατιστικά τεστ F και MI στις μετρήσεις υπό βλάβη στον εσωτερικό δακτύλιο φαίνονται στο Σχήμα 23.



Σχήμα 23: Στατιστικά τεστ F, MI στις μετρήσεις υπό βλάβη στον εσωτερικό δακτύλιο.

Βάσει του F-test, αρχικά, επιλέγονται τα μεγέθη $F = [8, 5, 2, 9, 7, 12]$, και από το MI-test τα μεγέθη $MI = [2, 12, 11, 6, 8, 5, 4]$, καταλήγοντας στον συνδυασμό $array1 = [2, 4, 5, 6, 7, 8, 9, 11, 12]$. Μερικά από τα μεγέθη αυτά, ωστόσο, έχουν έντονη συσχέτιση (βλ. Σχήματος 11) μεταξύ τους. Στο Σχήμα 24 παρουσιάζονται οι συντελεστές συσχέτισης Pearson.



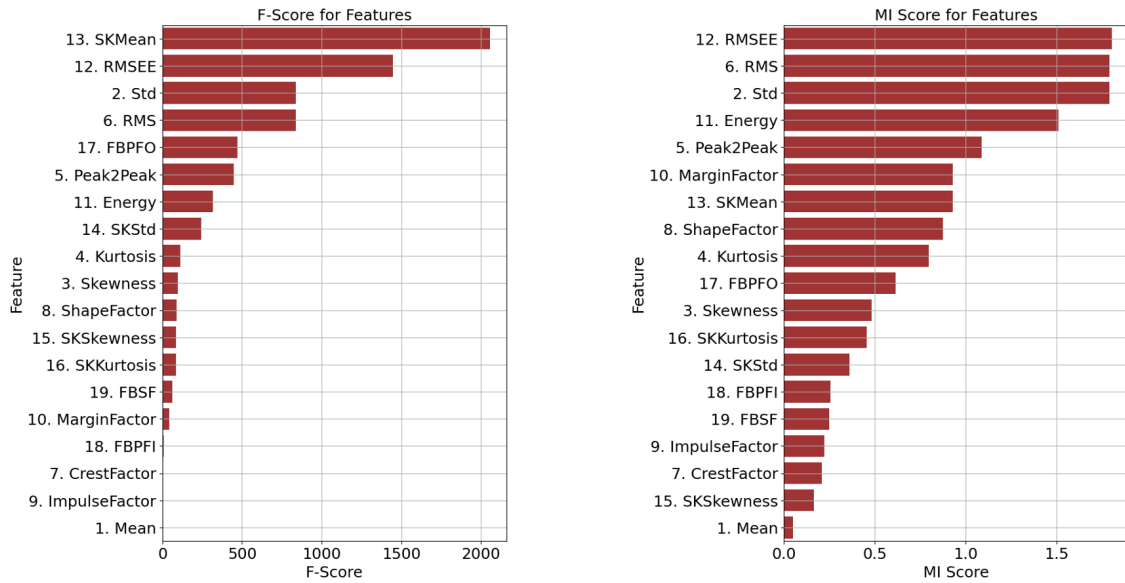
Σχήμα 24: Πίνακας συσχέτισης, στις μετρήσεις υπό βλάβη στον εσωτερικό δακτύλιο.

Τα χαρακτηριστικά [2, 5, 6, 8, 11, 12] έχουν έντονη συσχέτιση. Επιλέγονται δύο από αυτά, καθώς φαίνεται να έχουν τη σημαντικότερη επίδραση στο RUL. Τα χαρακτηριστικά που επιλέγονται είναι τα [8, 12] τα οποία έχουν σχετικά χαμηλότερη συσχέτιση $r(8, 12) = 0.77$ μεταξύ των άλλων. Υπάρχει επιπλέον συσχέτιση μεταξύ των χαρακτηριστικών [7, 9] ($r(7, 9) = 1$), επομένως αφαιρείται το χαρακτηριστικό [9]. Ο συνδυασμός χαρακτηριστικών, πλέον αποτελείται από:

$$\text{array1}=[4,7,8,12]=[\text{Kurtosis},\text{CrestFactor},\text{ShapeFactor},\text{RMSEE}]$$

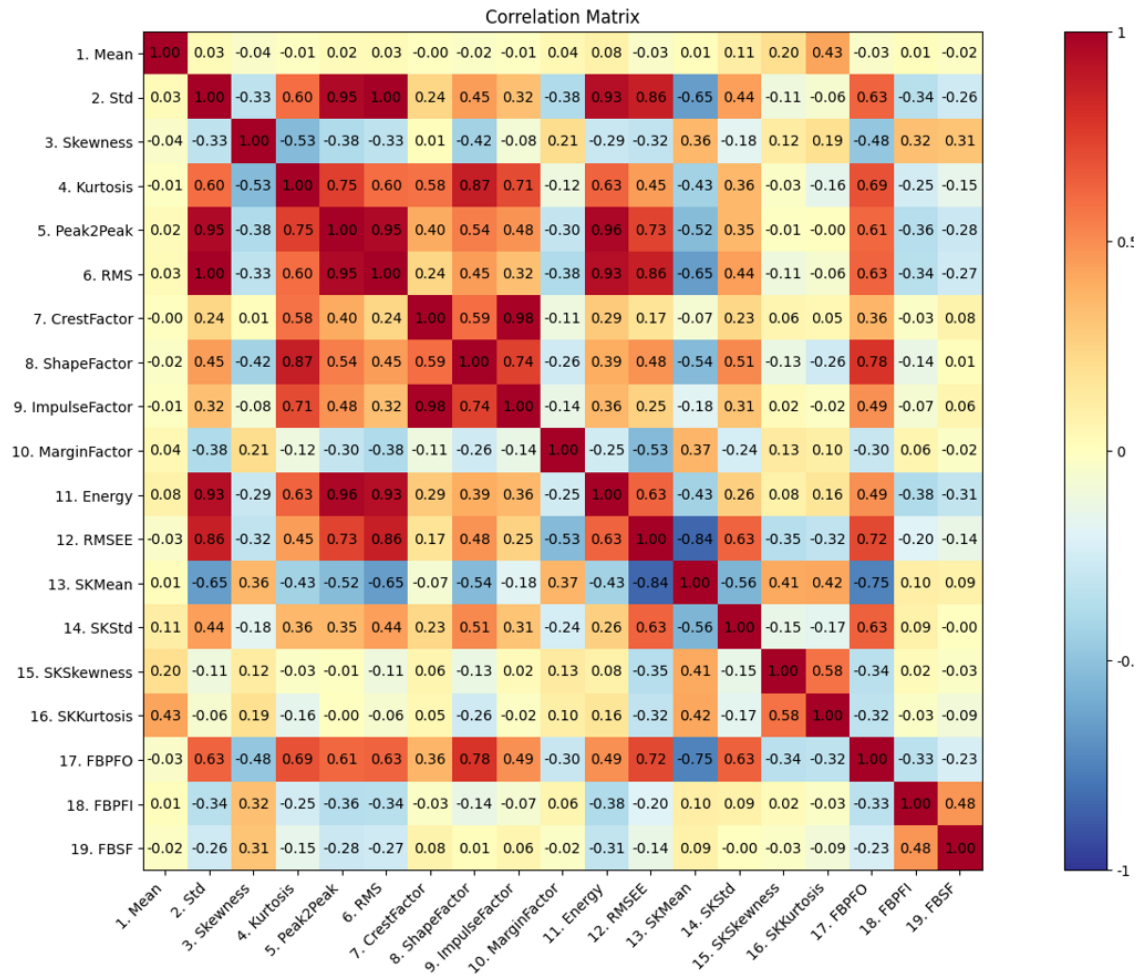
4.4.2 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης RUL, με βλάβη στον εξωτερικό δακτύλιο

Τα χαρακτηριστικά που επιλέγονται είναι συνδυασμός από τα Σχήματα 15,16 και 17,18, καθώς και τα δύο σύνολα μετρήσεων, αναφέρονται σε βλάβη στον εξωτερικό δακτύλιο. Οι τιμές του στόχου RUL λαμβάνουν τιμές βάσει των Σχημάτων 21, 22. Τα στατιστικά τεστ F και MI στις μετρήσεις υπό βλάβη στον εξωτερικό δακτύλιο φαίνονται στο Σχήμα 25.



Σχήμα 25: Στατιστικά τεστ F, MI στις μετρήσεις υπό βλάβη στον εξωτερικό δακτύλιο.

Βάσει του F-test, αρχικά, επιλέγονται τα μεγέθη $F = [13, 12, 2, 6, 17]$, και από το MI-test τα μεγέθη $MI = [12, 6, 2, 11, 5, 10, 13, 8]$, καταλήγοντας στον συνδυασμό $array2 = [2, 5, 6, 8, 10, 11, 12, 13, 17]$. Μερικά από τα μεγέθη αυτά, έχουν έντονη συσχέτιση μεταξύ τους. Στο **Σχήμα 26** παρουσιάζονται οι συντελεστές συσχέτισης Pearson.



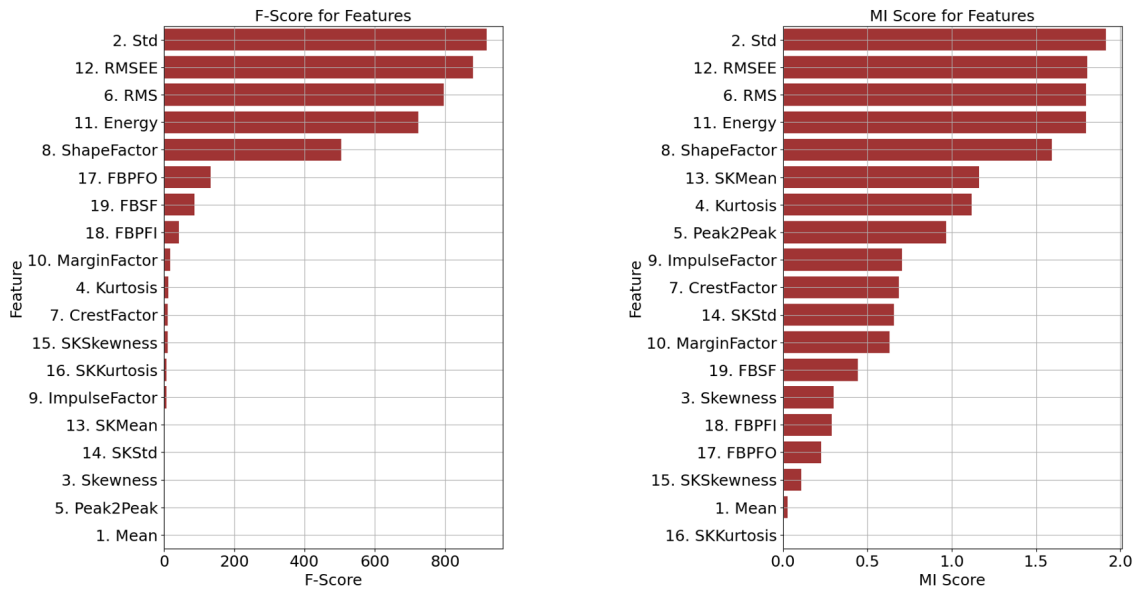
Σχήμα 26: Πίνακας συσχέτισης, στις μετρήσεις υπό βλάβη στον εξωτερικό δακτύλιο.

Τα χαρακτηριστικά [2, 5, 6, 11, 12] έχουν έντονη συσχέτιση. Επιλέγονται δύο από αυτά, καθώς φαίνεται να έχουν τη σημαντικότερη επίδραση στο RUL και τα υπόλοιπα αφαιρούνται. Τα χαρακτηριστικά που επιλέγονται είναι τα [5, 12] τα οποία έχουν σχετικά χαμηλότερη συσχέτιση $r(5, 12) = 0.73$ μεταξύ των άλλων. Ο συνδυασμός χαρακτηριστικών, πλέον αποτελείται από:

array2=[5,8,10,12,13,17]=[P2P,ShapeFactor,MarginFactor,RMSEE,SKMean,FBPF0]

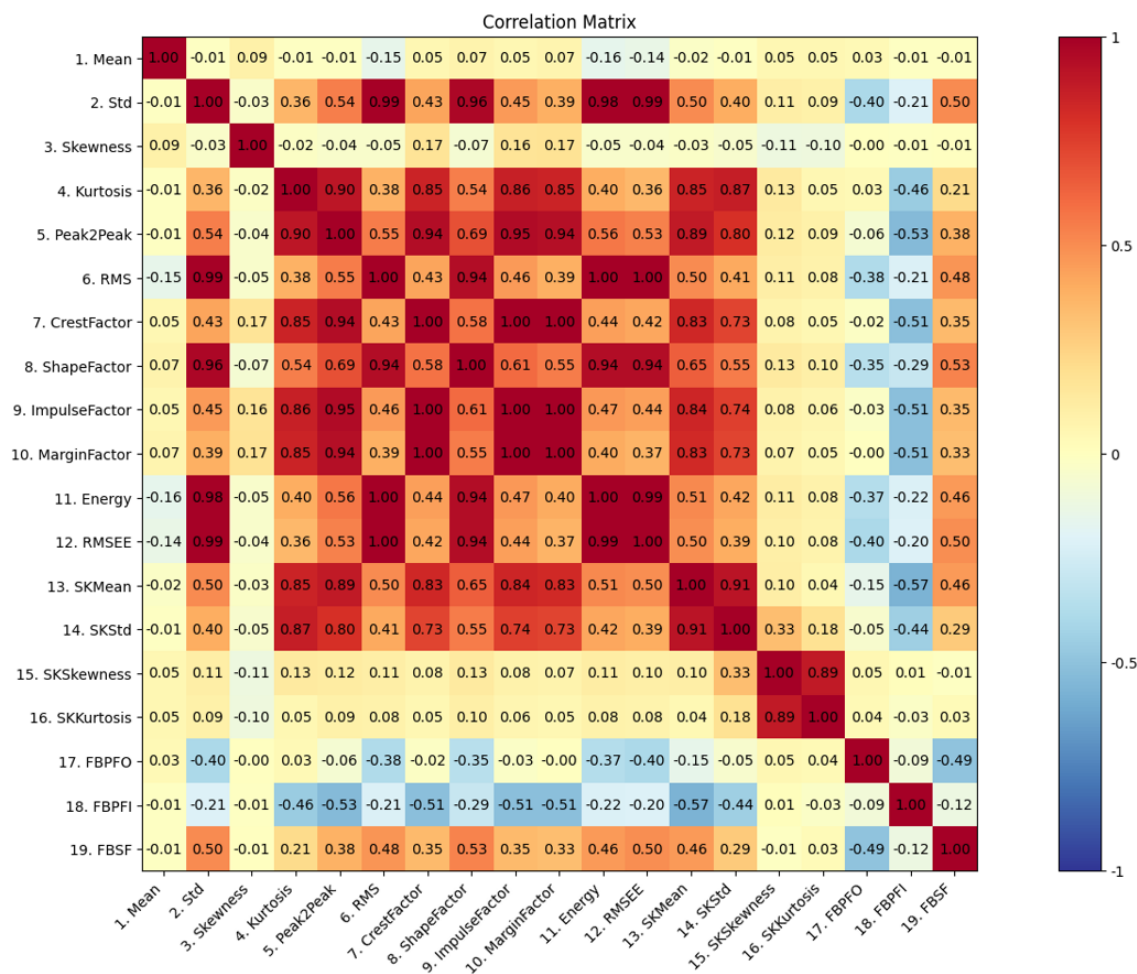
4.4.3 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης RUL, με βλάβη στα στοιχεία κύλισης

Τα χαρακτηριστικά που επιλέγονται είναι αυτά των Σχημάτων 13, 14. Οι τιμές του στόχου RUL λαμβάνουν τιμές βάσει του Σχήματος 20. Τα στατιστικά τεστ στις μετρήσεις υπό βλάβη στα στοιχεία κύλισης φαίνονται στο Σχήμα 27.



Σχήμα 27: Στατιστικά τεστ F, MI στις μετρήσεις υπό βλάβη στα στοιχεία κύλισης.

Βάσει του F-test, αρχικά, επιλέγονται τα μεγέθη $F = [2, 12, 6, 11, 8]$, και από το MI-test τα μεγέθη $MI = [2, 12, 6, 11, 8, 13, 14, 5, 9]$, καταλήγοντας στον συνδυασμό $array3 = [2, 5, 6, 8, 10, 11, 12, 13, 17]$. Μερικά από τα μεγέθη αυτά, έχουν έντονη συσχέτιση μεταξύ τους. Στο Σχήμα 28 παρουσιάζονται οι συντελεστές συσχέτισης Pearson.



Σχήμα 28: Πίνακας συσχέτισης, στις μετρήσεις υπό βλάβη στα στοιχεία κύλισης.

Τα χαρακτηριστικά [2, 6, 8, 11, 12] έχουν έντονη συσχέτιση. Το χαρακτηριστικό που επιλέγεται είναι το [12]. Επιπλέον έντονη συσχέτιση έχουν τα χαρακτηριστικά [4, 5, 9, 13], όπου από αυτά επιλέγονται τα [9, 13]. Ο συνδυασμός χαρακτηριστικών, πλέον αποτελείται από:

$$\text{array3}=[9,12,13]=[\text{ImpulseFactor},\text{RMSEE},\text{SKMean}]$$

4.4.4 Επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης κατάστασης του εδράνου κύλισης

Για την επιλογή χαρακτηριστικών στο μοντέλο πρόβλεψης κατάστασης του εδράνου κύλισης είναι αδύνατη η χρήση στατιστικών τεστ, καθώς οι τιμές του στόχου δηλώνουν κλάσεις (categories=[0,1,2,3]), οι οποίες είναι συμβολικές και δεν έχουν συσχέτιση μεταξύ τους. Επιλέχθηκε συνδυασμός χαρακτηριστικών, ο οποίος να δίνει έμφαση στον τύπο της βλάβης χωρίς να υπάρχει μεροληψία ως προς κάποια βλάβη. Ο συνδυασμός, αποτελείται από:

- τους συντελεστές των χαρακτηριστικών συχνότητων [17, 18, 19].

- από το κοινό χαρακτηριστικό μεταξύ των τριών μοντέλων [12].
- από τα κοινά χαρακτηριστικά μεταξύ των $array1 - array3$ και $array2 - array3$ που είναι τα [8] και [13] αντίστοιχα.
- από το κυρίαρχο χαρακτηριστικό του κάθε μοντέλου, όπου είναι τα [4, 9, 5].

Ο συνδυασμός χαρακτηριστικών, για το μοντέλο πρόβλεψης της κατάστασης του εδράνου κύλισης αποτελείται από:

$array=[4,5,8,9,12,13,17,18,19]=[Kurtosis,P2P,SF,IF,RMSEE,SKMean,fBPFO,fBPFI,fBSF]$

Στον **Πίνακα 5**, γίνεται ανασκόπηση από την τελική επιλογή των χαρακτηριστικών που θα δεχθεί το κάθε μοντέλο πρόβλεψης ως είσοδοι.

Πίνακας 5: Συνδυασμός χαρακτηριστικών για το κάθε μοντέλο πρόβλεψης.

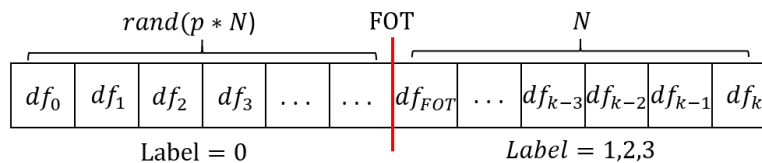
No.	Features	Prediction Models			
		Inner Fault	Outer Fault	Roller El. Fault	Fault Detection
1.	Mean	X	X	X	X
2.	Standard Deviation	X	X	X	X
3.	Skewness	X	X	X	X
4.	Kurtosis	✓	X	X	✓
5.	Peak2Peak	X	✓	X	✓
6.	RMS	X	X	X	X
7.	Crest Factor	✓	X	X	X
8.	Shape Factor	✓	✓	X	✓
9.	Impulse Factor	X	X	✓	✓
10.	Margin Factor	X	✓	X	X
11.	Energy	X	X	X	X
12.	RMSEE	✓	✓	✓	✓
13.	SKMean	X	✓	✓	✓
14.	SK Standard Deviation	X	X	X	X
15.	SK Skewness	X	X	X	X
16.	SK Kurtosis	X	X	X	X
17.	BPFO Factor	X	✓	X	✓
18.	BPFI Factor	X	X	X	✓
19.	BSF Factor	X	X	X	✓

5 Μοντέλο ανίχνευσης βλάβης του εδράνου κύλισης

Σε αυτήν την ενότητα αναπτύσσεται το μοντέλο ανίχνευσης βλάβης (Fault Detection), το οποίο αποτελεί το πρώτο κομμάτι για την εκτίμηση της υπολειπόμενης ωφέλιμης διάρκειας ζωής (RUL) του εδράνου κύλισης. Η πρόβλεψη της κατάστασης του εδράνου κύλισης γίνεται μέσω της ανάπτυξης ενός νευρωνικού δικτύου πρόσθιας τροφοδότησης (FFNN), με απαίτηση υψηλής ακρίβειας στις προβλέψεις, προκειμένου να επιλεγεί στη συνέχεια το κατάλληλο μοντέλο πρόβλεψης του (RUL).

5.1 Επιλογή Δεδομένων

Οι μετρήσεις για την εκπαίδευση (train) και την αξιολόγηση (test) του ΤΝΔ αποτελούν ένα συνδυασμό των τεσσάρων συνόλων μετρήσεων που περιλαμβάνουν την ανίχνευση βλάβης. Η κατηγοριοποίηση της κατάστασης του εδράνου κύλισης βασίζεται στα **Σχήματα 19, 20, 21, 22**. Κάθε σύνολο μετρήσεων περιλαμβάνει μετρήσεις για την ασφαλής κατάσταση (0) και μετρήσεις για τις καταστάσεις βλάβης (1,2,3), όπου οι μετρήσεις της ασφαλούς κατάστασης είναι περισσότερες. Από τις μετρήσεις της ασφαλούς κατάστασης, επιλέγεται τυχαίο ποσοστό του συνολικού αριθμού των μετρήσεων βλάβης, όπως φαίνεται στο **Σχήμα 29**, ώστε το μοντέλο να μην επικεντρώνεται υπερβολικά στις μετρήσεις της ασφαλούς κατάστασης.



Σχήμα 29: Επιλογή δεδομένων.

Στον **Πίνακα 6** παρουσιάζονται οι συνολικές μετρήσεις για το μοντέλο ανίχνευσης βλάβης. Οι μετρήσεις αυτές διαχωρίζονται σε μετρήσεις εκπαίδευσης (train), που αντιπροσωπεύουν το 80% του συνόλου, και μετρήσεις αξιολόγησης (test), που αντιπροσωπεύουν το 20% του συνόλου.

Πίνακας 6: Δεδομένα ανίχνευσης βλάβης.

Σύνολο Μετρήσεων*	Μετρήσεις	Βλάβη	#Ασφαλής Κατάστασης (0)	#Κατάσταση Βλάβης
No.13	2156	Εσ. Δακτύλιος (1)	347	347
No.14	2156	Στ. Κύλισης (3)	755	755
No.21	984	Εξ. Δακτύλιος (2)	397	397
No.33	6324	Εξ. Δακτύλιος (2)	367	367
Δεδομένα Αν. Βλάβης:			#(0)1866,	#(1)347
			#(2)764,	#(3)755

* Ο συμβολισμός No.13 συμβολίζει το πρώτο σύνολο μετρήσεων του εδράνου κύλισης 3. Επιλέχθηκε συντελεστής ($p = 1$).

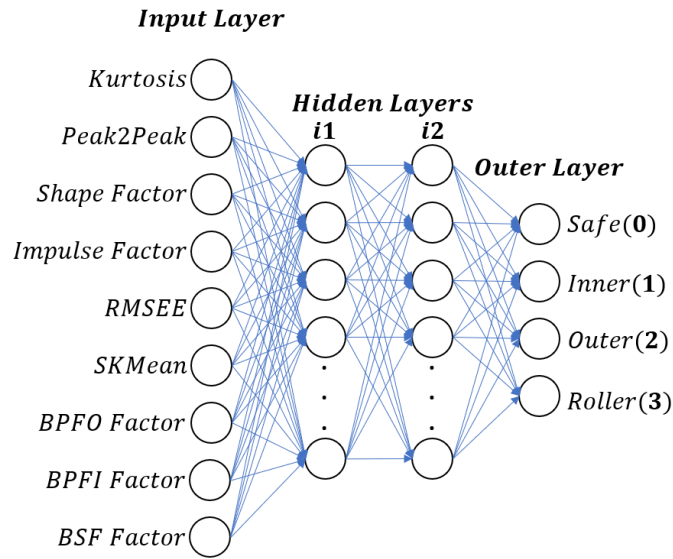
5.2 Αρχιτεκτονική

Ως είσοδοι στο ΤΝΔ, που αποτελούν το πρώτο επίπεδο (Input Layer) του ΤΝΔ, επιλέγονται τα χαρακτηριστικά (features) που διερευνήθηκαν στην **Ενότητα 4.4.4**, και παρουσιάζονται στον **Πίνακα 5**. Είναι σημαντικό να γίνει κλιμάκωση των επιλεγμένων εισόδων για να αποφευχθεί η μεροληψία προς ορισμένα χαρακτηριστικά. Η κλιμάκωση πραγματοποιείται ως εξής:

$$X_{i,scaled} = \frac{X_i - \bar{X}_i}{\sigma(X_i)} \quad (11)$$

Όπου X_i αναφέρεται στο σύνολο του χαρακτηριστικού i , \bar{X}_i αναπαριστά τη μέση τιμή και $\sigma(X_i)$ αναπαριστά την τυπική απόκλιση του χαρακτηριστικού i .

Μετά το πρώτο επίπεδο, ακολουθούν τα κρυφά επίπεδα (Hidden Layers) που λέγεται να είναι δύο -καθώς δεν βελτιώνεται η απόδοση του μοντέλου με παραπάνω επίπεδα και καθυστερείται η εκπαίδευση- όπου εκτελούνται οι μαθηματικές εκφράσεις των νευρώνων. Τέλος, στο τελευταίο επίπεδο (Output Layer) τοποθετούνται τέσσερις (4) νευρώνες, όπου προβλέπεται η κατάσταση του εδράνου κύλισης βάσει της μεγαλύτερης τιμής των νευρώνων. Η αρχιτεκτονική του ΤΝΔ, παρουσιάζεται στο **Σχήμα 30**.



Σχήμα 30: Αρχιτεκτονική ΤΝΔ ανίχνευσης βλάβης.

5.3 Βελτιστοποίηση

Η αξιολόγηση του κάθε ΤΝΔ γίνεται βάσει της αντικειμενικής συνάρτησης “κατηγορικής διασταυρούμενης εντροπίας” (categorical crossentropy) στα δεδομένα αξιολόγησης. Αυτή η συνάρτηση έχει σχεδιαστεί ειδικά για προβλήματα πολυκατηγορικής ταξινόμησης και υπολογίζεται από την παρακάτω σχέση:

$$Loss = - \sum \vec{y}_{true} \cdot (\log(\vec{y}_{pred}))^T \quad (12)$$

Όπου \vec{y}_{true} αναπαριστά το διάνυσμα των πραγματικών τιμών. Για παράδειγμα, στην περίπτωση της βλάβης εσωτερικού δακτυλίου, το διάνυσμα λαμβάνει τις τιμές $\vec{y}_{true} = [0, 1, 0, 0]$. Επίσης, \vec{y}_{pred} αναπαριστά το διάνυσμα των προβλεπόμενων τιμών. Για παράδειγμα, ένα πιθανό διάνυσμα μπορεί να είναι $\vec{y}_{pred} = [0.1, 0.8, 0.05, 0.05]$.

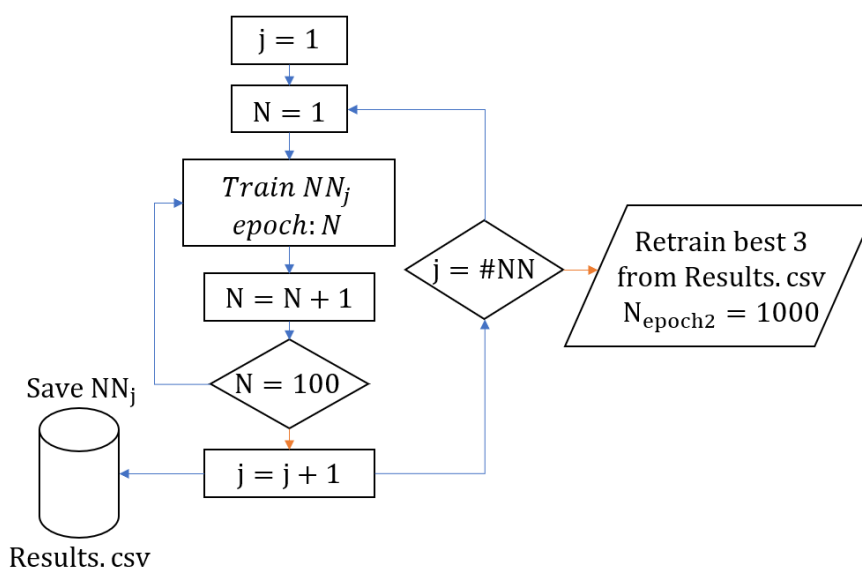
Οι παράμετροι βελτιστοποίησης του ΤΝΔ, που διερευνώνται είναι ο αριθμός των νευρώνων στα κρυφά επίπεδα $i1$, $i2$. Ο χώρος διερεύνησης καθώς και η αρχιτεκτονική των ΤΝΔ παρουσιάζονται στον Πίνακα 7.

Πίνακας 7: Αρχιτεκτονική των ΤΝΔ ανίχνευσης βλάβης.

Type	Layer	Nodes	Activation Function
Dense	Input	9	relu
Dense	Hidden $i1$	25:5:55	relu
Dense	Hidden $i2$	5:3:18	relu
Dense	Outer	4	softmax

Επιλέχθηκε η συνάρτηση ενεργοποίησης relu στα πρώτα τρία επίπεδα, καθώς είναι απλή και επιταχύνει τη σύγκλιση της εκπαίδευσης του ΤΝΔ. Στο τελευταίο επίπεδο επιλέχθηκε η συνάρτηση ενεργοποίησης softmax, ώστε να μετατρέπει το πρόβλημα κατηγοριοποίησης σε πρόβλημα πιθανότητας κλάσεων.

Κάθε ΤΝΔ επιλέγεται να εκπαιδευτεί για 100 εποχές ($N_{epoch} = 100$). Στη συνέχεια, γίνεται επανεκπαίδευση των τριών καλύτερων ΤΝΔ για επιπλέον 1000 εποχές ($N_{epoch2} = 1000$), και επιλέγεται το καλύτερο μοντέλο. Η επιλογή αυτή έγινε για να επιταχυνθεί η διαδικασία βελτιστοποίησης. Χρησιμοποιήθηκε ο αλγόριθμος Adams για την εύρεση των βαρών (weights) και των συντελεστών bias του ΤΝΔ. Η ταχύτητα επίλυσης αναφέρεται ότι είναι περίπου 400 εποχές ανά λεπτό ($v = 400epochs/min$). Το λογικό διάγραμμα της διαδικασίας βελτιστοποίησης του ΤΝΔ παρουσιάζεται στο **Σχήμα 31**.



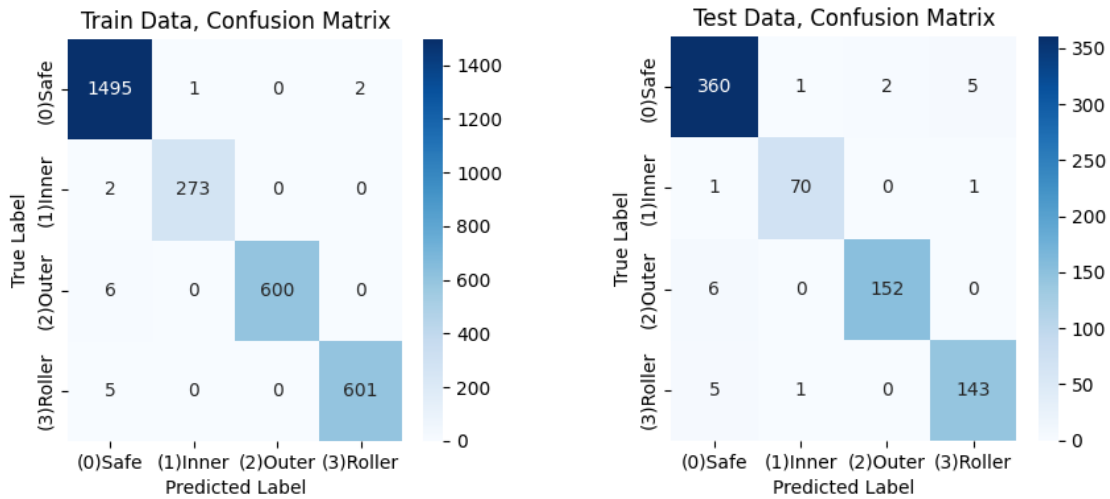
Σχήμα 31: Λογικό διάγραμμα βελτιστοποίησης αρχιτεκτονικής ΤΝΔ ανίχνευσης βλάβης.

Στον **Πίνακα 8** παρουσιάζονται οι τιμές της αντικειμενικής συνάρτησης για διάφορα ΤΝΔ που εκπαιδεύτηκαν, με την ταξινόμηση να ακολουθεί αύξουσα σειρά. Τα τρία (3) καλύτερα ΤΝΔ επανεκπαιδεύτηκαν, και επιλέγεται το ΤΝΔ με την μικρότερη τιμή της αντικειμενικής συνάρτησης, δηλαδή το ΤΝΔ με κρυφά επίπεδα $(i1, i2) = (25, 8)$.

Πίνακας 8: Αποτελέσματα βελτιστοποίησης για το ΤΝΔ ανίχνευσης βλάβης.

Rank	i1 Nodes	i2 Nodes	categorical crossentropy	
			at N_{epoch}	at N_{epoch2}
1.	50	8	0.1124	0.0096
2.	25	8	0.1281	0.0074
3.	30	8	0.1320	0.0085
4.	30	17	0.1401	
5.	35	5	0.1550	
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
35.	45	5	0.2749	
36.	35	8	0.2917	

Πέρα της αντικειμενικής συνάρτησης που ορίστηκε, ένα επιπλέον κριτήριο επιλογής του ΤΝΔ είναι ο πίνακας σύγχυσης (Confusion Matrix). Ο πίνακας σύγχυσης οπτικοποιεί τις πραγματικές και προβλεπόμενες τιμές κλάσης για όλες τις μετρήσεις. Ιδανικά, ο πίνακας πρέπει να είναι διαγώνιος. Στο **Σχήμα 32** παρουσιάζεται ο πίνακας σύγχυσης του ΤΝΔ με κρυφά επίπεδα $(i1, i2) = (25, 8)$, για τα δεδομένα εκπαίδευσης και αξιολόγησης.



Σχήμα 32: Πίνακας σύγχυσης για τα δεδομένα εκπαίδευσης και αξιολόγησης.

Οι κύριες μετρικές του πίνακα σύγχυσης υπολογίζονται από τις παρακάτω σχέσεις, και παρουσιάζονται στον **Πίνακα 9**.

$$\text{accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}} \quad (13)$$

$$\text{Precision}_i = \frac{TP_i}{FP_i}, \quad \text{Recall}_i = \frac{TP_i}{TP_i + FN_i}, \quad f1_i = 2 \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (14)$$

Πίνακας 9: Κύριες μετρικές που υπολογίζονται από τον πίνακα σύγκρισης των δεδομένων αξιολόγησης.

Class	Precision	Recall	f1-score
Safe (0)	0.978	0.967	0.973
Inner (1)	0.972	0.972	0.972
Outer (2)	0.962	0.987	0.974
Roller (3)	0.959	0.959	0.959
Accuracy	0.970		

Οι τιμές του πίνακα σύγκρισης και οι μετρικές του, είναι αρκετά ικανοποιητικές και η επιλογή της συγκεκριμένης αρχιτεκτονικής εξακολουθεί να ισχύει. Παρακάτω αναφέρονται οι κύριοι λόγοι:

- Οι τιμές των μετρικών του πίνακα σύγκρισης λαμβάνουν εξαιρετικές τιμές.
- Ο πίνακας σύγκρισης είναι κατά κύριο λόγο διαγώνιος.
- Στην περίπτωση που υπάρχει βλάβη $y_{true} = 1, 2, 3$ και η πρόβλεψη είναι ξανά βλάβη $y_{pred} = 1, 2, 3$ οι τιμές αυτές ταυτίζονται σχεδόν πάντα (βλ. υπομητρώο 3×3). Παρατηρούνται μόνο δύο (2) μετρήσεις στα δεδομένα αξιολόγησης (Test Data), όπου στην μία μέτρηση υπάρχει βλάβη στον εσωτερικό δακτύλιο και αναγνωρίστηκε ως βλάβη στα στοιχεία κύλισης και μία μέτρηση όπου υπάρχει βλάβη στα στοιχεία κύλισης και αναγνωρίστηκε ως βλάβη στον εσωτερικό δακτύλιο. Αυτό το γεγονός είναι αρκετά σημαντικό καθώς το ΤΝΔ σχεδόν πάντα θα ακολουθήσει το σωστό μοντέλο πρόβλεψης της υπολειπόμενης ωφέλιμης διάρκειας ζωής (RUL).
- Στην περίπτωση που η κατάσταση του εδράνου κύλισης είναι ασφαλές ($y_{true} = 0$) και η πρόβλεψη είναι βλάβη $y_{pred} = 1, 2, 3$, παρατηρούνται τρεις (3) μετρήσεις στα δεδομένα εκπαίδευσης (Train Data) και οκτώ (8) μετρήσεις στα δεδομένα αξιολόγησης (Test Data). Αυτό το γεγονός δεν είναι ανησυχητικό, καθώς οι μετρήσεις αυτές πιθανότερα βρίσκονται κοντά στο FOT, οπότε προκαλείται σύγκριση. Η ίδια παρατήρηση γίνεται και στις περιπτώσεις όπου η κατάσταση του εδράνου κύλισης είναι σε βλάβη $y_{ture} = 1, 2, 3$ και η προβλεπόμενη κατάσταση του εδράνου κύλισης είναι ($y_{pred} = 0$).

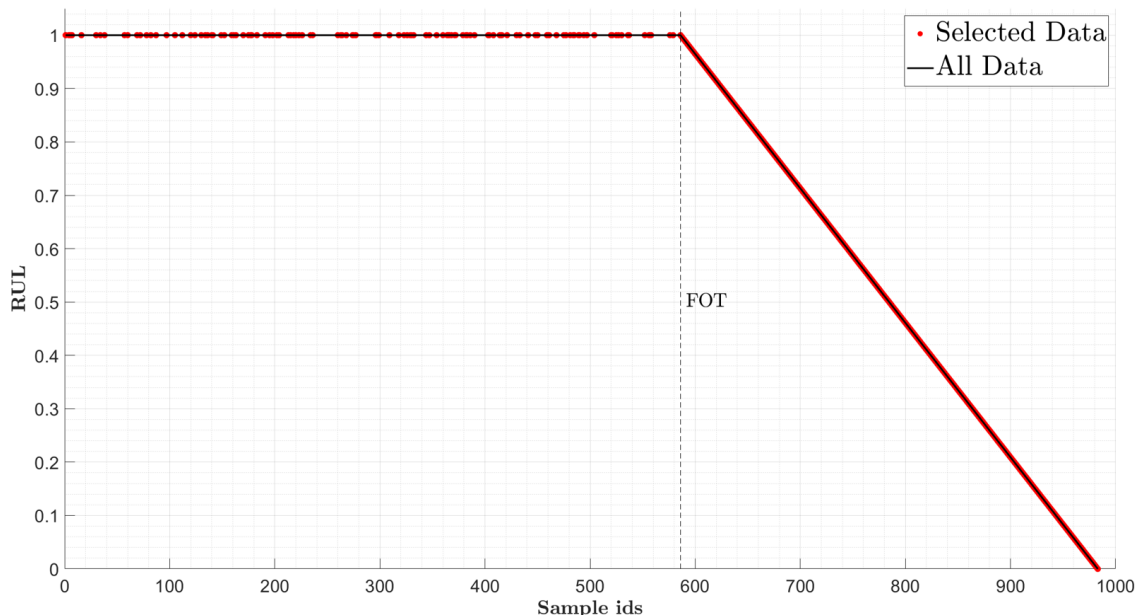
6 Εκτίμηση Υπολειπόμενης Ωφέλιμης Διάρκειας Ζώης του εδράνου κύλισης

Στην περίπτωση που έχει αναγνωριστεί βλάβη σε μία μέτρηση, από το πρωταρχικό μοντέλο πρόβλεψης, γίνεται εκτίμηση της υπολειπόμενης ωφέλιμης διάρκειας ζωής (RUL) από ένα από τα τρία (3) υπό-μοντέλα πρόβλεψης RUL. Η εκτίμηση του RUL, γίνεται με χρήση νευρωνικού δικτύου με κόμβους LSTM.

6.1 Επιλογή Δεδομένων

Η επιλογή των δεδομένων για την εκπαίδευση (train) και αξιολόγηση (test) των ΕΝΔ, έγινε αρχικά αποκλειστικά βασιζόμενη στις μετρήσεις μετά το FOT (όταν υπάρχει βλάβη). Το μοντέλο που δημιουργήθηκε με αυτές τις μετρήσεις, παρουσίαζε υψηλή ακρίβεια στις μετρήσεις που αντιστοιχούσαν στην εμφάνιση βλάβης, αλλά εμφάνιζε χαμηλή ακρίβεια στις μετρήσεις που βρίσκονταν σε ασφαλή κατάσταση και είχαν προβλεφθεί ως μετρήσεις βλάβης. Για αυτό το λόγο, τα τελικά επιλεγμένα δεδομένα περιλαμβάνουν μετρήσεις ασφαλούς κατάστασης ποσοστού p των μετρήσεων βλάβης, σύμφωνα με το **Σχήμα 29**.

Η τιμή του στόχου RUL λαμβάνει τιμές μεταξύ $[0.0, 1,0]$. Η τιμή 1.0 υποδηλώνει την ασφαλή κατάσταση του εδράνου κύλισης και μειώνεται γραμμικά μέχρι την τιμή 0.0, η οποία αντιστοιχεί στην τελική καταστροφή του εδράνου κύλισης (βλ. **Σχήματα 19, 20, 21, 22**). Το **Σχήμα 33** παρουσιάζει τις τιμές του RUL για το σύνολο μετρήσεων No.24, προκειμένου να επισημανθούν με μεγαλύτερη σαφήνεια οι προαναφερθέντες σχολιασμοί.



Σχήμα 33: Επιλογή δεδομένων για την εκτίμηση του RUL.

Τα δεδομένα χωρίζονται, όπως και στην περίπτωση της ανίχνευσης βλάβης, σε σύνολα εκπαίδευσης (train) και αξιολόγησης (test). Τα ποσοστά που χρησιμοποιού-

ύνται είναι 80% και 20% του συνολικού αριθμού δεδομένων, αντίστοιχα. Ο **Πίνακας 10** παρουσιάζει τα δεδομένα που χρησιμοποιούνται για κάθε μοντέλο πρόβλεψης του RUL.

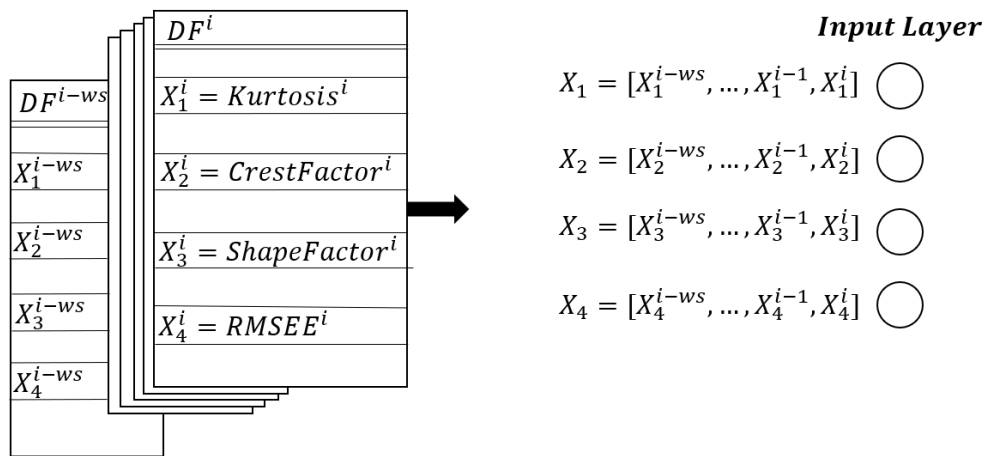
Πίνακας 10: Δεδομένα για την εκτίμηση του RUL.

Βλάβη	Σύνολο Μετρήσεων*	Μετρήσεις	#Ασφαλής Κατάστασης (0)	#Κατάσταση Βλάβης
Εσ. Δακτύλιος (1)	No.13	2156	139	347
Εξ. Δακτύλιος (2)	No.21	984	159	397
	No.33	6324	147	367
Στοιχεία Κύλισης (3)	No.14	6324	302	755

* Ο συμβολισμός No.13, συμβολίζει το πρώτο σύνολο μετρήσεων του εδράνου κύλισης 3. Επιλέχθηκε και στα τρία (3) υπό μοντέλα συντελεστής $p = 0.4$.

6.2 Αρχιτεκτονική

Η αρχιτεκτονική του νευρωνικού δικτύου για την εκτίμηση του RUL, που αρχικά διερευνήθηκε, ήταν παρόμοια με αυτή του νευρωνικού δικτύου για την ανίχνευση βλάβης. Ωστόσο, το σφάλμα στις εκτιμήσεις του RUL ήταν εμφανές, ειδικά στα αρχικά στάδια της βλάβης και δεν μπορούσε να αγνοηθεί. Η επόμενη και τελική επιλογή για το νευρωνικό δίκτυο εκτίμησης του RUL ήταν η αντικατάσταση των απλών νευρώνων του πρώτου επιπέδου με νευρώνες LSTM. Οι νευρώνες LSTM είναι κατάλληλοι για αυτό το πρόβλημα, καθώς οι μετρήσεις και τα χαρακτηριστικά τους (βλ. **Σχήματα 11-18**) εξαρτώνται χρονικά. Πριν από το επίπεδο με τους νευρώνες LSTM, υπάρχει το επίπεδο εισόδων (Input Layer) όπου κάθε νευρώνας αντιστοιχεί σε ένα διάνυσμα μήκους ws (Window Size) των διαδοχικών τιμών του k -χαρακτηριστικού (X_k) όπως φαίνεται στο **Σχήμα 34**. Τα χαρακτηριστικά που επιλέγονται σε κάθε μοντέλο για συγκεκριμένη βλάβη προέρχονται από τον **Πίνακα 5**.



Σχήμα 34: Διανύσματα εισόδων στα ΕΝΔ εκτίμησης του RUL.

Μετά το κρυφό επίπεδο με τους $i1$ νευρώνες LSTM, ακολουθεί το επίπεδο με τους απλούς $i2$ νευρώνες, το οποίο είναι συνδεδεμένο με το επίπεδο εξόδου (Outer Layer) για την πρόβλεψη του RUL. Προκειμένου να αποφευχθεί το φαινόμενο της υπερπροσαρμογής, δηλαδή η υπερβολική προσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης που οδηγεί σε χαμηλή ακρίβεια στα δεδομένα αξιολόγησης, προστίθενται δύο επιπλέον επίπεδα εγκατάλειψης (Dropout) ανάμεσα στο κρυφό επίπεδο με τους απλούς $i2$ νευρώνες. Κατά τη διάρκεια της εκπαίδευσης, το επίπεδο εγκατάλειψης επιλέγει τυχαία ένα ποσοστό των νευρώνων και τα απενεργοποιεί, αποκλείοντας τη συνεισφορά τους στην πρόβλεψη. Αυτό έχει ως αποτέλεσμα τη μείωση της υπερ-προσαρμογής του μοντέλου στα εκπαιδευτικά δεδομένα και τη βελτίωση της ικανότητάς του να γενικεύει σε νέα δεδομένα.

Στο **Σχήμα 35** συγκρίνονται δύο παρόμοια μοντέλα ως προς τη σύγκλισή τους, με τη διαφορά ότι το ένα μοντέλο περιλαμβάνει τα επίπεδα εγκατάλειψης.

6.3 Βελτιστοποίηση

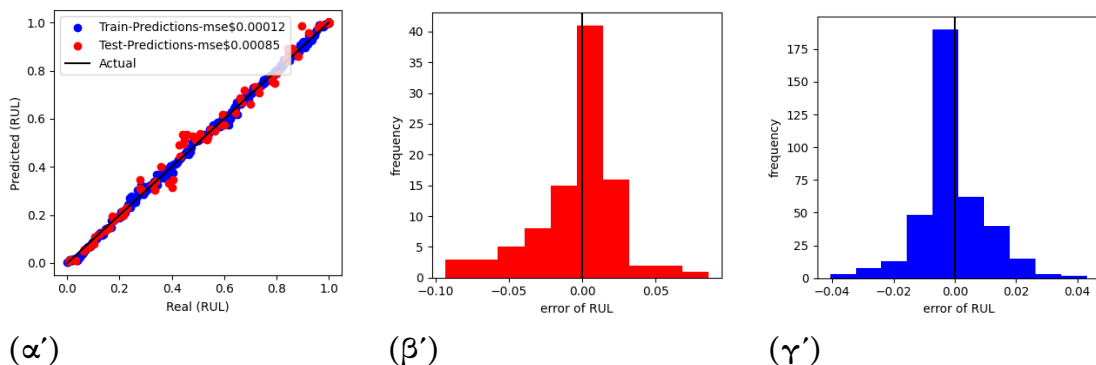
Τα κλασικά προβλήματα παλινδρόμησης (Regression), αξιολογούνται βάσει της αντικειμενικής συνάρτησης μέσου τετραγωνικού σφάλματος (Mean Squared Error), που δίδεται από τη σχέση:

$$MSE = \frac{1}{N} \sum (y_i - \hat{y}_i)^2 \quad (15)$$

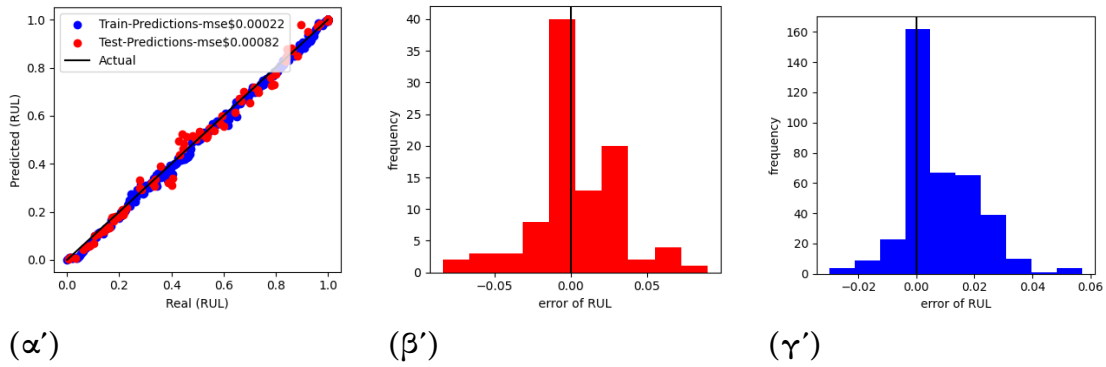
Στην εργασία αυτή πραγματοποιήθηκε τροποποίηση της αντικειμενικής συνάρτησης με σκοπό να δοθεί μεγαλύτερη βαρύτητα στις εκτιμήσεις του RUL, \hat{y}_i , όταν είναι μεγαλύτερες από τις πραγματικές τιμές y_i , δηλαδή $\hat{y}_i > y_i$. Η τροποποίηση πραγματοποιήθηκε για λόγους ασφάλειας, καθώς θεωρείται προτιμότερο να γίνεται συντηρητική αντικατάσταση του εδράνου κύλισης παρά να προβλεφθεί η αντικατάσταση μετά την αστοχία του. Η τροποποιημένη αντικειμενική συνάρτηση είναι:

$$\begin{aligned} &\text{for } i=1:\text{length}(y) \\ &\quad \text{if } \hat{y}_i > y_i : \\ &\quad \quad sd_i = \text{penalty} \cdot (y_i - \hat{y}_i)^2 \\ &\quad \text{else} \\ &\quad \quad sd_i = (y_i - \hat{y}_i)^2 \\ &\quad \text{end} \\ MSE^* &= \frac{1}{N} \sum sd_i \end{aligned} \quad (16)$$

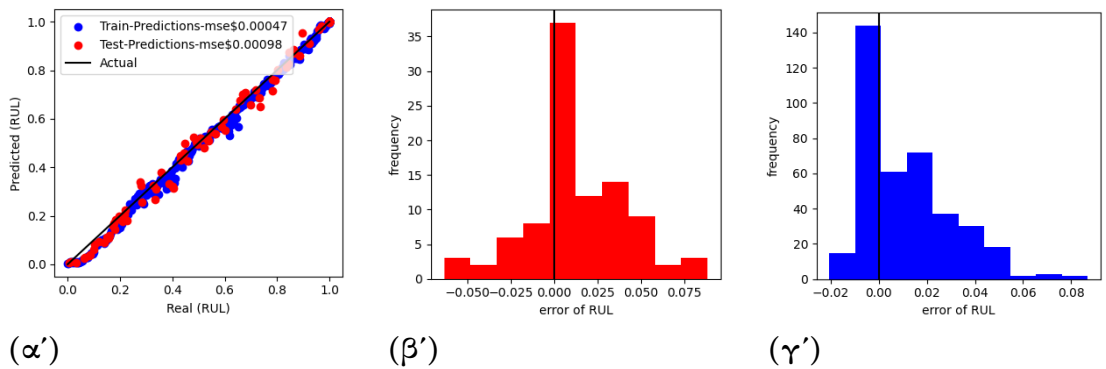
Όπου *penalty*, ο συντελεστής επιβάρυνσης των υπερεκτιμήσεων. Στα **Σχήματα 37, 38, 39, 40**, παρουσιάζονται οι εκτιμήσεις των RUL σε σχέση με τις πραγματικές τιμές των RUL, καθώς και τα ιστογράμματα των σφαλμάτων των μετρήσεων εκπαίδευσης και αξιολόγησης, για διάφορες τιμές *penalty*. Το μοντέλο αναπτύχθηκε βασιζόμενο σε μετρήσεις που έχει εντοπιστεί βλάβη στον εσωτερικό δακτύλιο, και στις τέσσερις περιπτώσεις είναι το ίδιο $(i_1, i_2) = (16, 17)$.



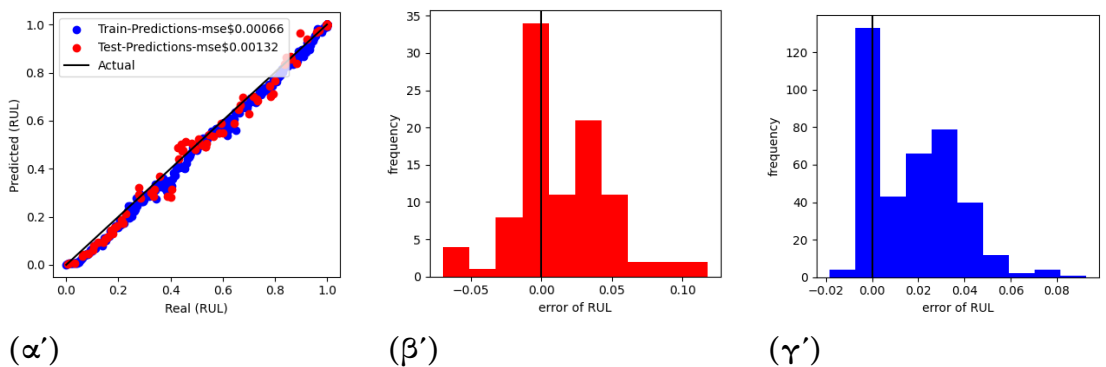
Σχήμα 37: Αποτελέσματα συντελεστή *penalty* = 1. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.



Σχήμα 38: Αποτελέσματα συντελεστή $penalty = 2$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.



Σχήμα 39: Αποτελέσματα συντελεστή $penalty = 4$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.



Σχήμα 40: Αποτελέσματα συντελεστή $penalty = 8$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.

Γίνονται οι εξής παρατηρήσεις πάνω στα Σχήματα 37, 38, 39, 40:

- Η κατανομή των σφαλμάτων, στο μοντέλο με την αντικειμενική συνάρτηση $Loss = MSE^*() = MSE()$, δηλαδή όταν ο συντελεστής *penalty* λαμβάνει την τιμή 1, είναι κανονική.
- Με την αύξηση του συντελεστή *penalty*, μειώνεται ο συντελεστής συμμετρίας της κατανομής ($Skewness < 0$) των σφαλμάτων ($error = y_i - \hat{y}$), δηλαδή η κατανομή τείνει προς τα θετικά σφάλματα.
- Τα σημεία (y_i, \hat{y}) , όσο αυξάνεται ο συντελεστής *penalty*, βρίσκονται κάτω από την ευθεία $y = \hat{y}$, υπάρχει δηλαδή υποεκτίμηση του RUL, επομένως η τροποποίηση της αντικειμενικής συνάρτησης ήταν αποτελεσματική.
- Το μέσο τετραγωνικό σφάλμα (MSE), αυξάνεται με την αύξηση του συντελεστή *penalty*.

Λαμβάνοντας υπόψη αυτές τις παρατηρήσεις, η επιλογή του συντελεστή *penalty* γίνεται με τρόπο ώστε να μεταβληθεί η κατανομή των σφαλμάτων και να διατηρηθεί το *MSE* σε χαμηλά επίπεδα. Η τιμή που επιλέγεται είναι *penalty* = 4, η οποία θεωρείται ικανοποιητική για όλα τα τρία (3) μοντέλα πρόβλεψης του RUL.

Οι παράμετροι βελτιστοποίησης των ΕΝΔ που διερευνώνται είναι ο αριθμός των νευρώνων στα κρυφά επίπεδα i_1, i_2 . Ο χώρος διερεύνησης καθώς και η αρχιτεκτονική των ΕΝΔ παρουσιάζονται στον **Πίνακα 11**.

Πίνακας 11: Αρχιτεκτονική των ΕΝΔ εκτίμησης του RUL.

Type	Layer	Nodes	Activation Function
	Input	(j x ws)	
LSTM	Hidden i_1	10:6:52	tanh
Dropout	Dropout1		propability=0.15
Dense	Hidden i_2	2:5:26	relu
Dropout	Dropout2		propability=0.15
Dense	Output	1	sigmoid

Η τιμή του μεγέθους του παραθύρου (window size) που επιλέγεται είναι $ws = 6$, προκειμένου να απαιτούνται μετρήσεις που λαμβάνονται σε μία ώρα (60 λεπτά). Είναι προφανές ότι η επιλογή ενός υψηλού μεγέθους του παραθύρου ευνοεί την ακρίβεια πρόβλεψης του μοντέλου. Ωστόσο, παρατηρήθηκε ότι τιμές μεγαλύτερες του $ws = 6$ δεν βελτιώνουν σημαντικά την ακρίβεια πρόβλεψης του μοντέλου και δεν επιλέγονται, καθώς αυξάνεται ο χρόνος καταγραφής των μετρήσεων. Επίσης, σημειώνεται ότι επιλέχθηκε η συνάρτηση ενεργοποίησης sigmoid (activation function) στο επίπεδο εξόδου, έτσι ώστε οι τιμές του RUL να βρίσκονται αυστηρά στο διάστημα [0,1].

Η διαδικασία εύρεσης του βέλτιστου συνδυασμού (i_1, i_2) για τα μοντέλα πρόβλεψης του RUL είναι αντίστοιχη με αυτήν της ανίχνευσης βλάβης (βλ. **Σχήμα 31**). Επιλέγεται να εκπαιδευτεί το κάθε μοντέλο $N_{epoch} = 150$ εποχές και επιπλέον για τα τρία (3) καλύτερα μοντέλα εκπαιδεύονται επιπλέον $N_{epoch2} = 2500$ εποχές.

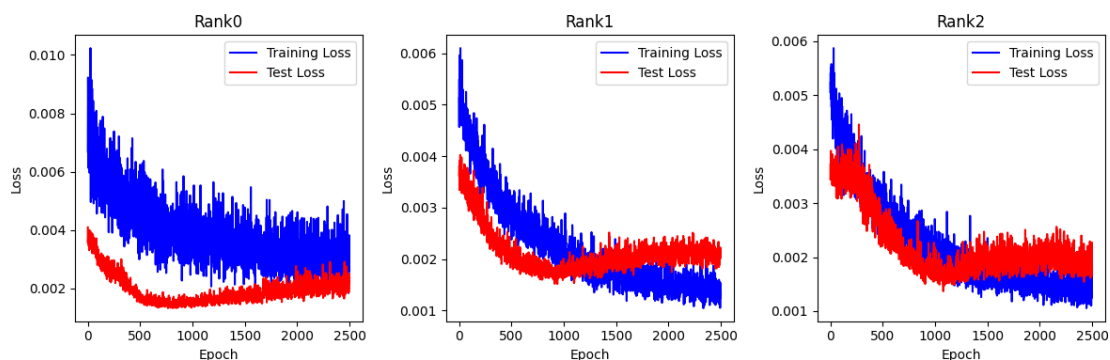
6.3.1 Βελτιστοποίηση μοντέλου πρόβλεψης RUL μετρήσεων υπό βλάβη στον εσ. δακτύλιο (1)

Στον Πίνακα 12 παρουσιάζονται οι τιμές της αντικειμενικής συνάρτησης για διάφορα ΕΝΔ που εκπαιδεύτηκαν, με την ταξινόμηση να ακολουθεί αύξουσα σειρά. Τα τρία (3) καλύτερα ΕΝΔ, όπως αναφέρεται, επανεκπαιδεύτηκαν.

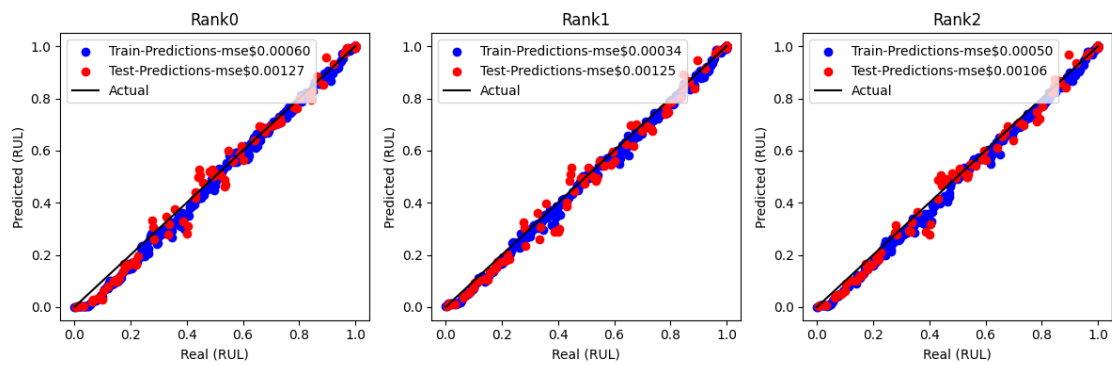
Πίνακας 12: Αποτελέσματα βελτιστοποίησης για το ΕΝΔ εκτίμησης RUL υπό βλάβη στον εσ. δακτύλιο(1).

Rank	i1 Nodes	i2 Nodes	custom MSE , (MSE^*)	
			at N_{epoch}	at N_{epoch2}
1.	28	22	0.0033	0.0012
2.	10	22	0.0035	0.0012
3.	16	17	0.0036	0.0010
4.	10	12	0.0037	
5.	40	22	0.0037	
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
39.	52	2	0.0354	
40.	10	2	0.0364	

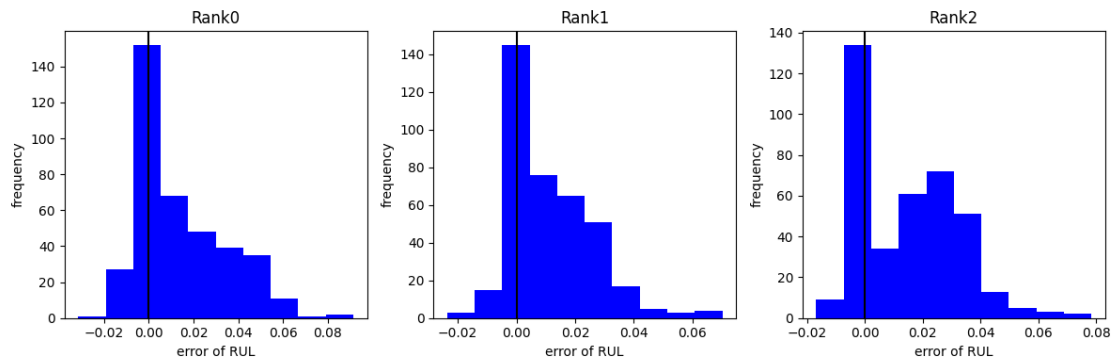
Οι τελικές τιμές της αντικειμενικής συνάρτησης για τα τρία (3) καλύτερα μοντέλα, είναι παρόμοιες. Η επιλογή του τελικού μοντέλου, γίνεται παρατηρώντας τα **Σχήματα 41, 42, 43, 44**.



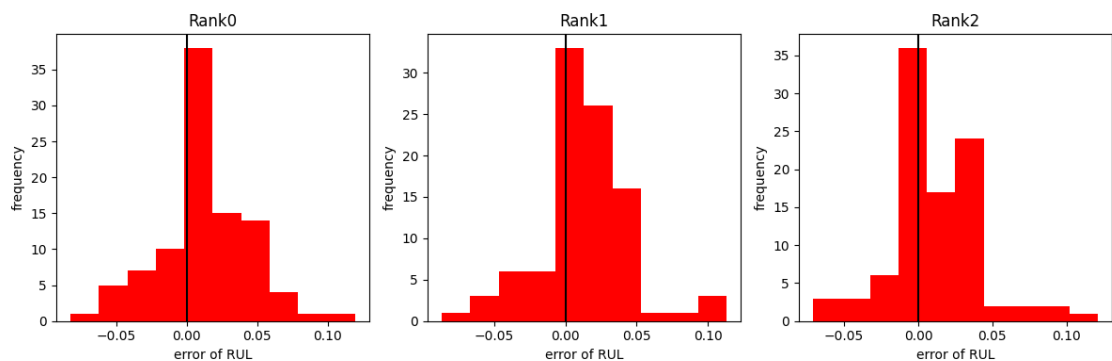
Σχήμα 41: Σύγκλιση εκπαίδευσης των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.



Σχήμα 42: Εκτιμήσεις του RUL των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.



Σχήμα 43: Ιστόγραμμα σφαλμάτων μετρήσεων εκπαίδευσης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.



Σχήμα 44: Ιστόγραμμα σφαλμάτων μετρήσεων αξιολόγησης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.

Η τελική επιλογή της αρχιτεκτονικής για το μοντέλο εκτίμησης του RUL για μετρήσεις υπό βλάβη στον εσωτερικό δακτύλιο είναι το τρίτο καλύτερο μοντέλο, δηλαδή $(i1, i2) = (16, 17)$ οι λόγοι είναι οι εξής:

- Υπάρχει σύγκλιση στις μετρήσεις εκπαίδευσης και αξιολόγησης (βλ. **Σχήμα 41**) και είναι σχεδόν κοινή.
- Ενώ οι τιμές MSE^* (βλ. **Σχήμα 42**) των τριών μοντέλων είναι κοντινές, στο τρίτο (3) μοντέλο γίνεται καλύτερη εκτίμηση στις τελευταίες τιμές του RUL, $RUL < 0.5$, όπου είναι σημαντικό για λόγους ασφάλειας.

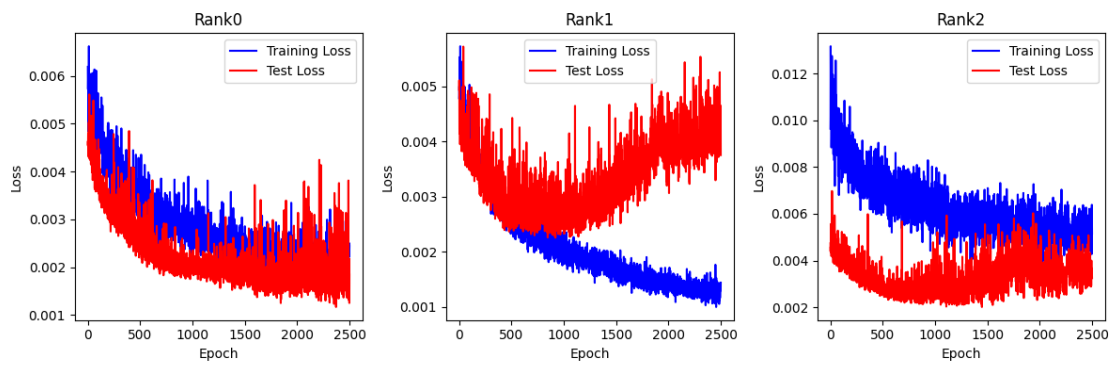
6.3.2 Βελτιστοποίηση μοντέλου πρόβλεψης RUL μετρήσεων υπό βλάβη στον εξ. δακτύλιο (2)

Στον **Πίνακα 13** παρουσιάζονται οι τιμές της αντικειμενικής συνάρτησης για διάφορα ΕΝΔ που εκπαιδεύτηκαν, με την ταξινόμηση να ακολουθεί αύξουσα σειρά. Τα τρία (3) καλύτερα ΕΝΔ, όπως αναφέρεται, επανεκπαιδεύτηκαν.

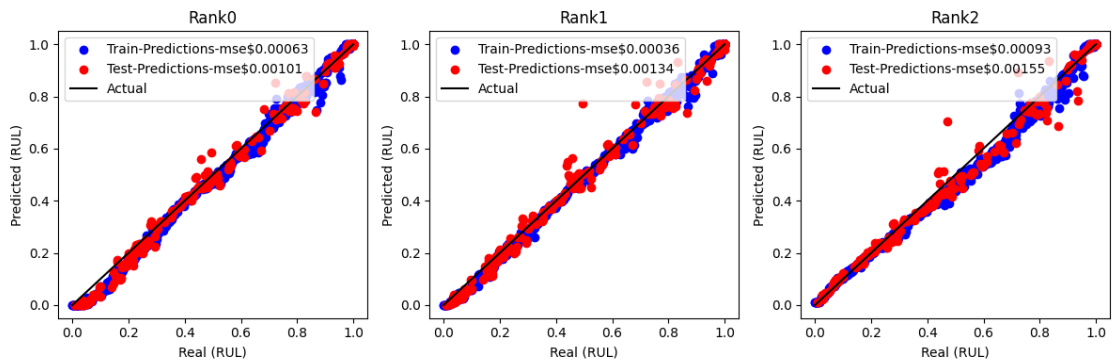
Πίνακας 13: Αποτελέσματα βελτιστοποίησης για το ΕΝΔ εκτίμησης RUL υπό βλάβη στον εξ. δακτύλιο(2).

Rank	i1 Nodes	i2 Nodes	custom MSE , (MSE^*)	
			at N_{epoch}	at N_{epoch2}
1.	34	12	0.0047	0.0010
2.	28	22	0.0049	0.0013
3.	40	7	0.0050	0.0015
4.	46	17	0.0050	
5.	40	12	0.0051	
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
39.	10	7	0.0077	
40.	46	2	0.022	

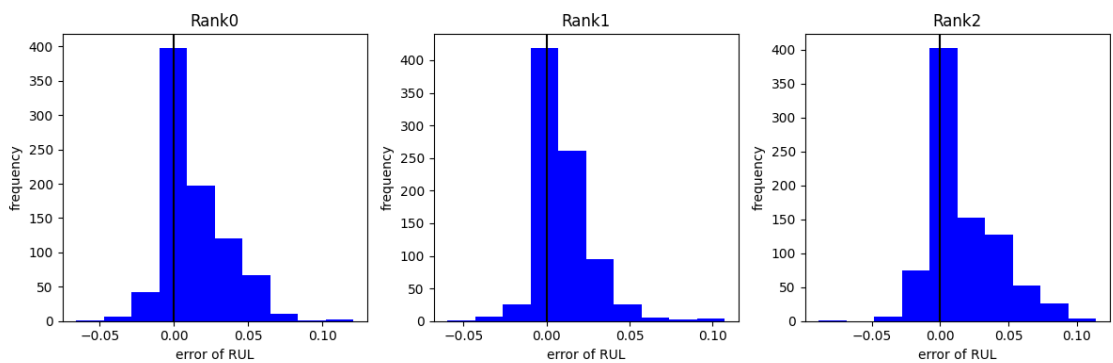
Οι τελικές τιμές της αντικειμενικής συνάρτησης για τα τρία (3) καλύτερα μοντέλα, είναι παρόμοιες. Η επιλογή του τελικού μοντέλου γίνεται παρατηρώντας τα **Σχήματα 45, 46, 47, 48**.



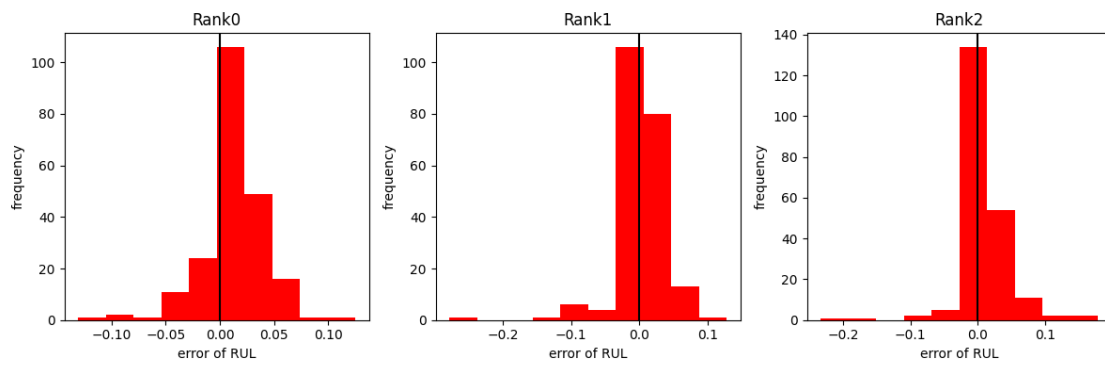
Σχήμα 45: Σύγκλιση εκπαίδευσης των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.



Σχήμα 46: Εκτιμήσεις του RUL των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.



Σχήμα 47: Ιστόγραμμα σφαλμάτων μετρήσεων εκπαίδευσης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.



Σχήμα 48: Ιστόγραμμα σφαλμάτων μετρήσεων αξιολόγησης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.

Η τελική επιλογή της αρχιτεκτονικής για το μοντέλο πρόβλεψης του RUL για μετρήσεις υπό βλάβη στον εξωτερικό δακτύλιο είναι το πρώτο καλύτερο μοντέλο, δηλαδή $(i1, i2) = (34, 12)$. Οι λόγοι για αυτήν την επιλογή είναι οι εξής:

- Υπάρχει σύγκλιση στις μετρήσεις εκπαίδευσης και αξιολόγησης (βλ. **Σχήμα 45**) και είναι σχεδόν κοινή (επιλέγεται το καλύτερο).
- Δεν υπάρχουν μετρήσεις με μεγάλα σφάλματα $error > 0.2$ (βλ. **Σχήματα 46, 48**).

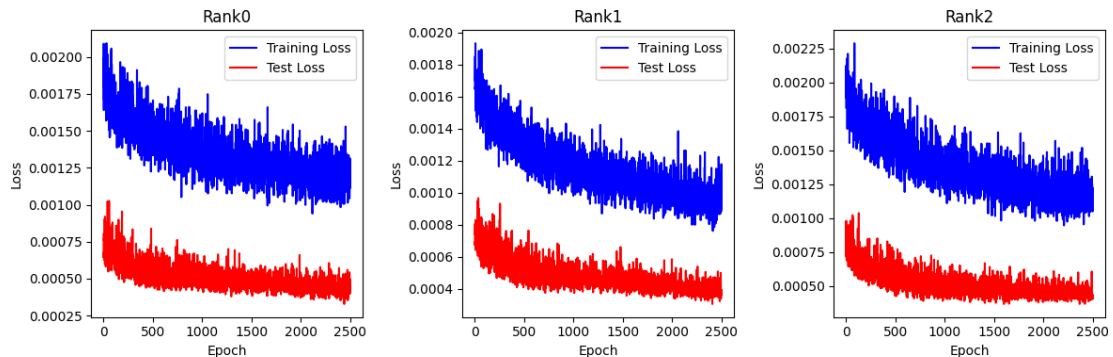
6.3.3 Βελτιστοποίηση μοντέλου πρόβλεψης RUL μετρήσεων υπό βλάβη στα στοιχεία κύλισης (3)

Στον **Πίνακα 14**, παρουσιάζονται οι τιμές της αντικειμενικής συνάρτησης για διάφορα ENΔ που εκπαιδεύτηκαν, με την ταξινόμηση να ακολουθεί αύξουσα σειρά. Τα τρία (3) καλύτερα ENΔ, όπως αναφέρεται, επανεκπαιδεύτηκαν.

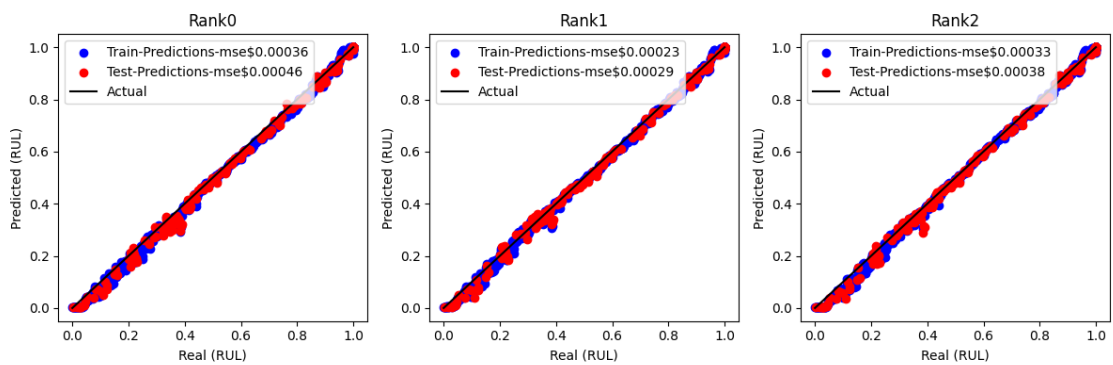
Πίνακας 14: Αποτελέσματα βελτιστοποίησης για το ENΔ εκτίμησης RUL υπό βλάβη στα στοιχεία κύλισης(3).

Rank	i1 Nodes	i2 Nodes	custom MSE , (MSE^*)	
			at N_{epoch}	at N_{epoch2}
1.	52	17	0.00061	0.00046
2.	46	22	0.00067	0.00029
3.	46	17	0.00068	0.00038
4.	40	22	0.00071	
5.	52	22	0.00072	
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
39.	34	2	0.01213	
40.	22	2	0.01229	

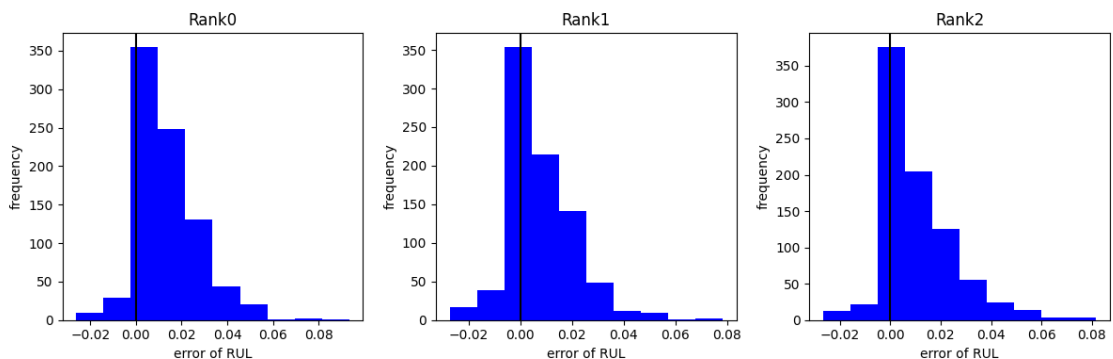
Οι τελικές τιμές της αντικειμενικής συνάρτησης για τα τρία (3) καλύτερα μοντέλα, είναι παρόμοιες. Η επιλογή του τελικού μοντέλου, γίνεται παρατηρώντας τα **Σχήματα 49, 50, 51, 52**.



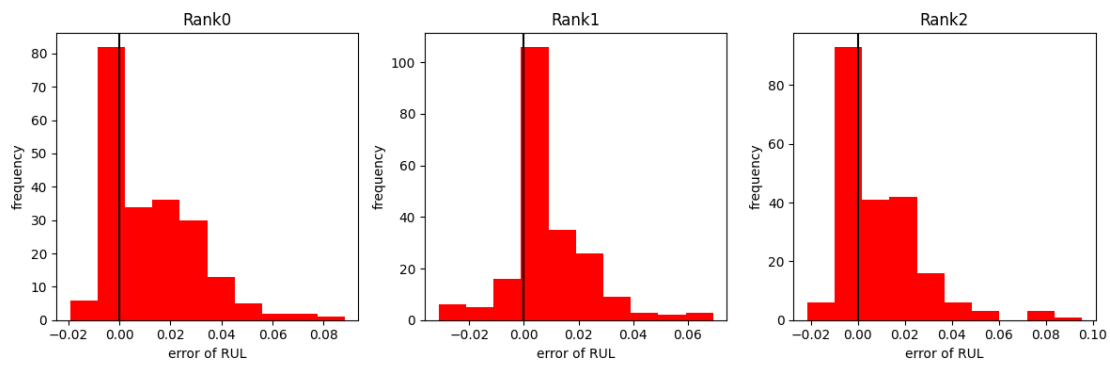
Σχήμα 49: Σύγκλιση εκπαίδευσης των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.



Σχήμα 50: Εκτιμήσεις του RUL των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.



Σχήμα 51: Ιστόγραμμα σφαλμάτων μετρήσεων εκπαίδευσης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.



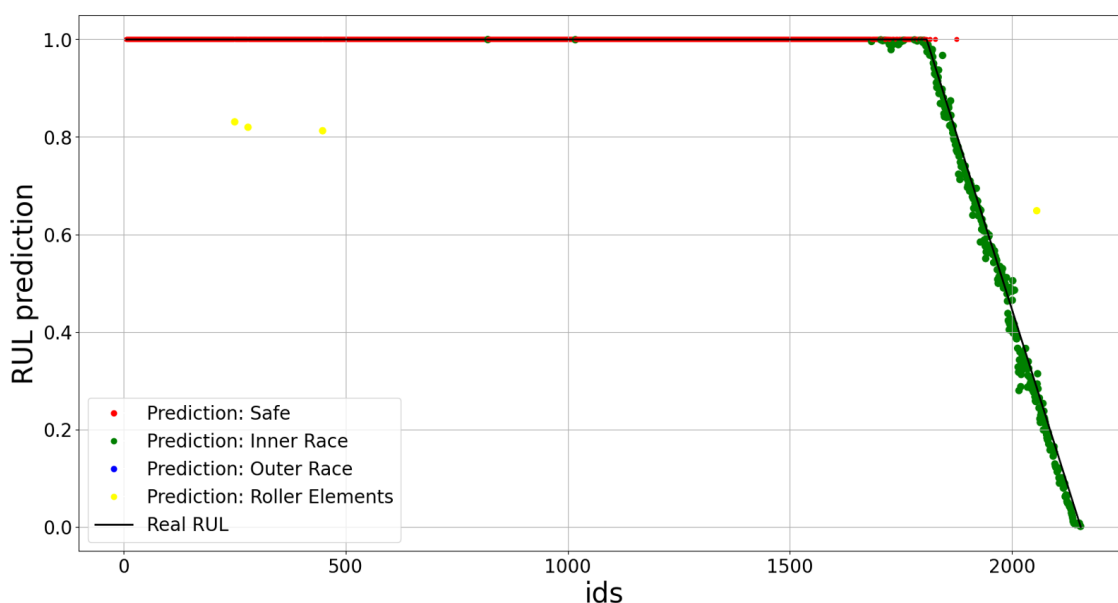
Σχήμα 52: Ιστόγραμμα σφαλμάτων μετρήσεων αξιολόγησης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.

Η τελική επιλογή της αρχιτεκτονικής για το μοντέλο εκτίμησης του RUL για μετρήσεις υπό βλάβη στα στοιχεία κύλισης είναι το δεύτερο καλύτερο μοντέλο, δηλαδή $(i1, i2) = (46, 22)$. Οι λόγοι για αυτή την επιλογή είναι οι εξής:

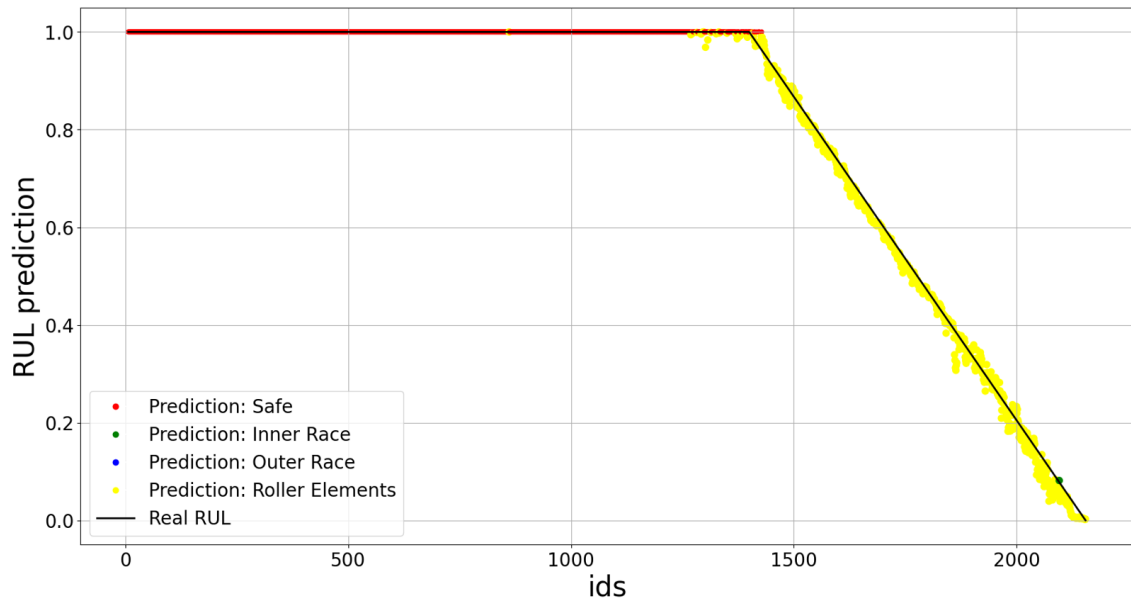
- Οι τελικές τιμές των MSE για τις μετρήσεις της εκπαίδευσης και αξιολόγησης, είναι χαμηλότερες από τα άλλα δύο μοντέλα (βλ. **Σχήμα 50**).
- Οι τιμές των σφαλμάτων λαμβάνουν τιμές $error < 0.06$ (βλ. **Σχήματα 51, 52**).

7 Τελική αξιολόγηση

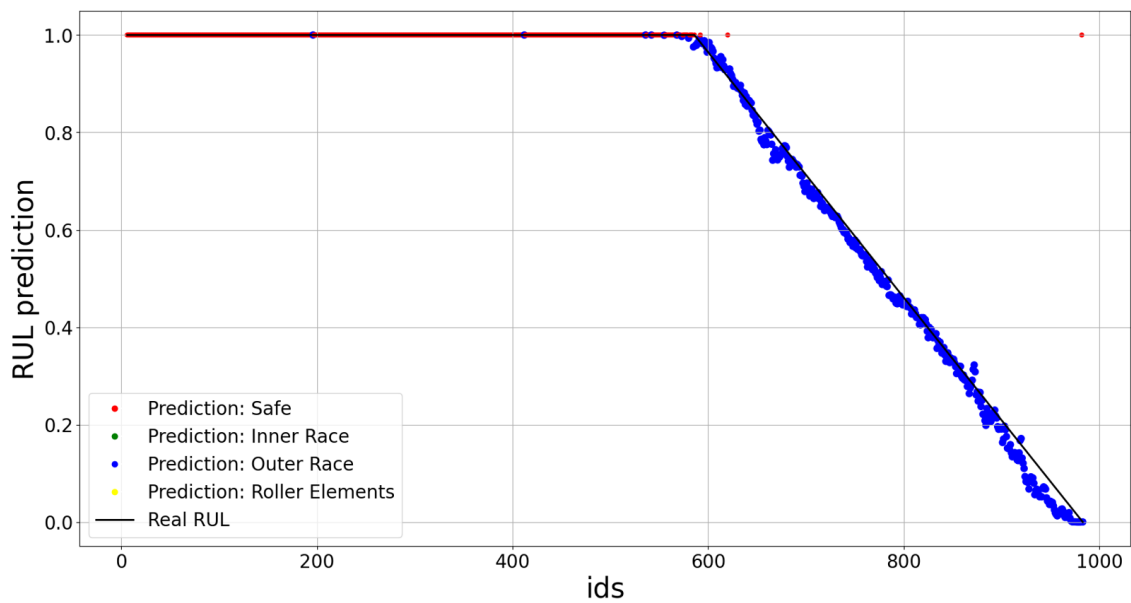
Οι συνολικές μετρήσεις που χρησιμοποιήθηκαν για την εκπαίδευση και αξιολόγηση των βελτιστοποιημένων μοντέλων είναι 11,620. Κάθε μέτρηση περιγράφει κραδασμούς επιτάχυνσης διάρκειας 1s. Αφού γίνει αποθρομβοποίηση της μέτρησης, εξάγονται χαρακτηριστικά του σήματος, με τα οποία γίνεται η εκτίμηση της κατάστασης του εδράνου κύλισης στο πρωταρχικό ΤΝΔ. Στην περίπτωση που εκτιμηθεί βλάβη, χρησιμοποιούνται έξι (6) διαδοχικές μετρήσεις, επιλέγοντας κάθε φορά τα κατάλληλα χαρακτηριστικά βλάβης, και γίνεται η εκτίμηση του RUL χρησιμοποιώντας το κατάλληλο ΕΝΔ. Η διαδικασία εκτίμησης του RUL κάθε μέτρησης βασίζεται στο **Σχήμα 3**. Στα **Σχήματα 53,54,55,56,57** γίνεται η τελική εκτίμηση της κατάστασης του εδράνου κύλισης, καθώς και του RUL κάθε μέτρησης.



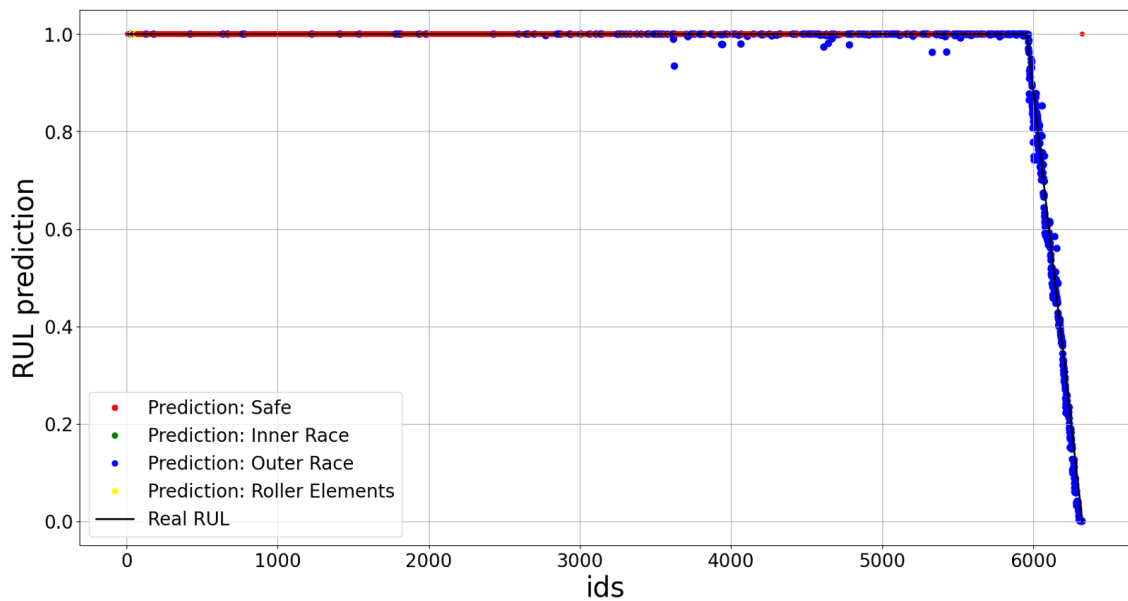
Σχήμα 53: Εκτιμήσεις του RUL για το σύνολο μετρήσεων 1 του εδράνου κύλισης 3.



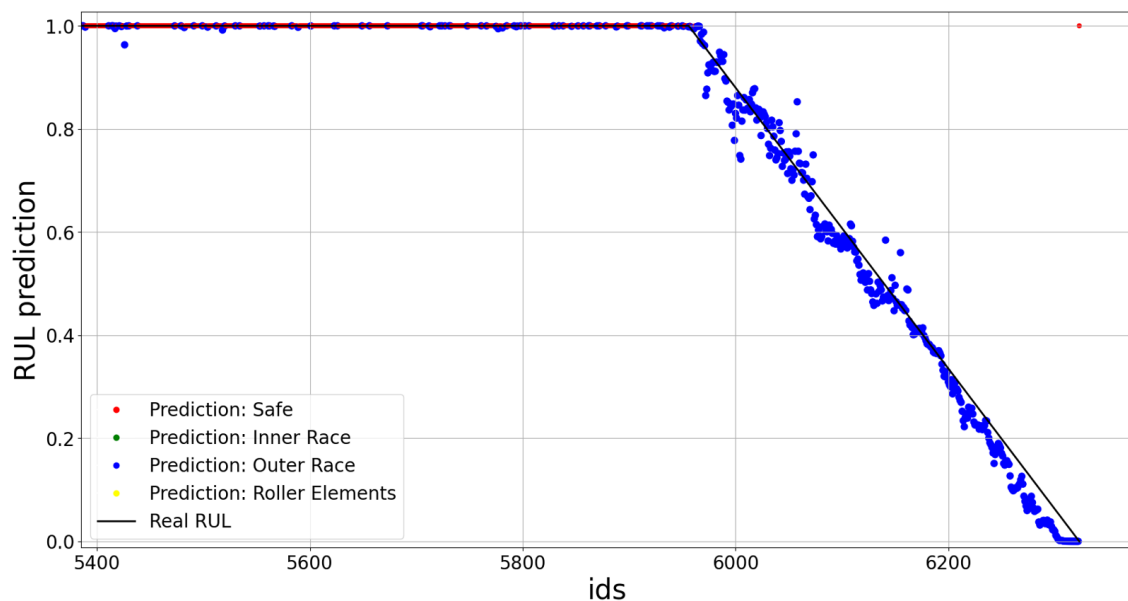
Σχήμα 54: Εκτιμήσεις του RUL για το σύνολο μετρήσεων 1 του εδράνου κύλισης 4.



Σχήμα 55: Εκτιμήσεις του RUL για το σύνολο μετρήσεων 2 του εδράνου κύλισης 1.



Σχήμα 56: Εκτιμήσεις του RUL για το σύνολο μετρήσεων 3 του εδράνου κύλισης 3.



Σχήμα 57: Εκτιμήσεις του RUL για το σύνολο μετρήσεων 3 του εδράνου κύλισης 3, στην περιοχή κοντά στην αστοχία

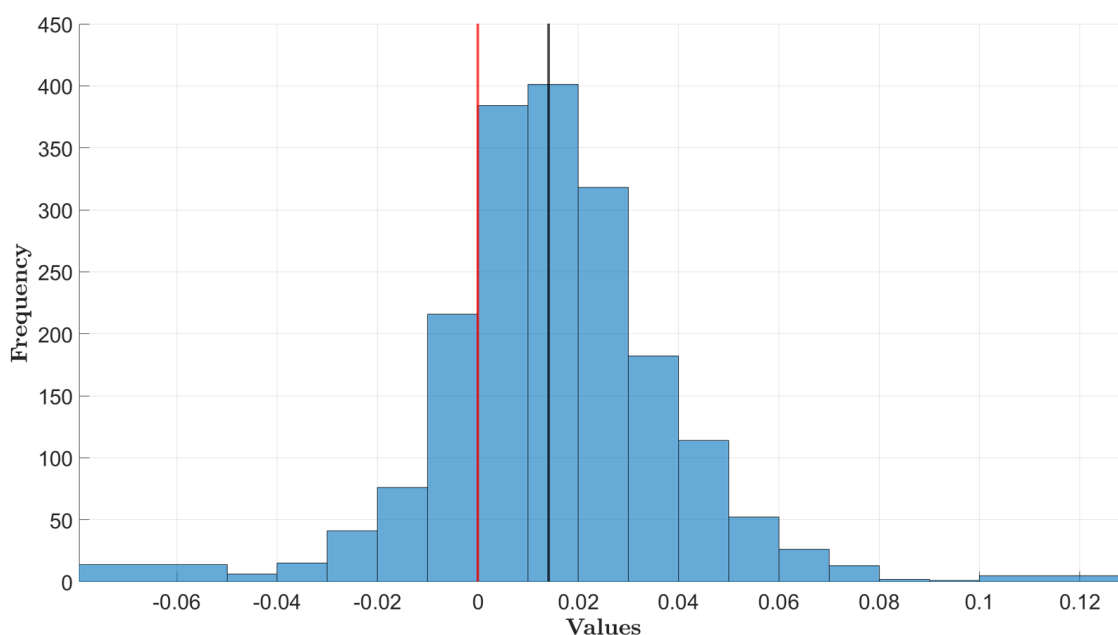
Βάσει των Σχημάτων 53,54,55,56,57, γίνονται οι εξής παρατηρήσεις:

- Η πρόβλεψη της κατάστασης του εδράνου κύλισης είναι σχεδόν πάντα επιτυχής. Υπάρχει μόνο μια μικρή σύγχυση μεταξύ των μετρήσεων με βλάβη στον εσωτερικό δακτύλιο και των μετρήσεων με βλάβη στα στοιχεία κύλισης.
- Δεν υπάρχουν σημαντικά έκτοπα σημεία (outliers). Οι μόνες μετρήσεις που

παρατηρούνται είναι στα **Σχήματα 55** και **57**, που είναι οι τελικές μετρήσεις, και φαίνεται ότι είναι κακής ποιότητας. Το γεγονός αυτό δεν προκαλεί ανησυχία.

- Παρατηρείται υποεκτίμηση σε όλα τα σύνολα μετρήσεων.
- Η προσθήκη μετρήσεων ασφαλούς κατάστασης στα δεδομένα του ΕΝΔ για την εκτίμηση του RUL ήταν κατάλληλη επιλογή, ειδικά στο **Σχήμα 56**, όπου φαίνεται ότι πολλές μετρήσεις ασφαλούς κατάστασης προβλέπονται ως βλαβερές καταστάσεις.

Στο **Σχήμα 58**, παρουσιάζεται το ιστόγραμμα των σφαλμάτων για όλες τις μετρήσεις που τα έδρανα κύλισης είναι φθαρμένα (μετά τα σημεία FOT).



Σχήμα 58: Ιστόγραμμα σφαλμάτων για όλες τις μετρήσεις που τα έδρανα κύλισης λειτουργούν υπό βλάβη.

Η κατανομή των σφαλμάτων ακολουθεί κανονική κατανομή με μέση τιμή και τυπική απόκλιση ως εξής:

$$\mu = 1.4\%, \quad \sigma = 3.7\%$$

Επομένως, το μοντέλο εκτίμησης του RUL προβλέπει με βεβαιότητα 95.4% την υπολειπόμενη ωφέλιμη διάρκεια ζωής του εδράνου κύλισης, με σφάλμα που κυμαίνεται στο διάστημα:

$$[\mu - 2\sigma, \mu + 2\sigma] = [-6\%, +9\%] \quad (17)$$

Σημειώνεται ότι οι τιμές δίνονται σε ποσοστά του συνολικού χρόνου αστοχίας. Συγκεκριμένα, στην περίπτωση βλάβης εσωτερικού δακτυλίου, η τιμή είναι $t_I = 57.8h$, στην περίπτωση βλάβης εξωτερικού δακτυλίου είναι $t_O = 66.1h$, και στην περίπτωση βλάβης των στοιχείων κύλισης είναι $t_R = 125.8h$.

8 Συμπεράσματα και Μελλοντικές Εργασίες

Η ακριβής πρόβλεψη του υπολειπόμενου ωφέλιμου χρόνου ζωής των εδράνων κύλισης αποτελεί κρίσιμο παράγοντα για την αποτελεσματική συντήρηση, με σκοπό τη βελτίωση της αξιοπιστίας και τη μείωση του συνολικού κόστους συντήρησης. Στο πλαίσιο αυτής της εργασίας, πραγματοποιείται μια προσπάθεια χαρακτηρισμού της κατάστασης του εδράνου κύλισης και έπειτα πραγματοποιείται εκτίμηση του υπολειπόμενου ωφέλιμου χρόνου ζωής των εδράνων κύλισης. Γίνεται αρχικά εστίαση στη βελτίωση της ποιότητας των μετρήσεων μέσω της αποθρομβοποίηση των δεδομένων. Επιπλέον, πραγματοποιήθηκαν στατιστικά τεστ για την επιλογή των κατάλληλων χαρακτηριστικών. Τα χαρακτηριστικά που εξετάστηκαν προέρχονται τόσο από τη βιβλιογραφία όσο και από ειδικά χαρακτηριστικά που προτάθηκαν για την ανίχνευση βλάβης. Έπειτα, αναπτύχθηκε και βελτιστοποιήθηκε ένα απλό νευρωνικό δίκτυο με στόχο την υψηλή ακρίβεια στην εκτίμηση της κατάστασης του εδράνου κύλισης. Τέλος, αναπτύχθηκε ένα νευρωνικό δίκτυο με κόμβους LSTM για την πρόβλεψη του υπολοιπούμενου ωφέλιμου χρόνου ζωής, επιλέγοντας με έξυπνο τρόπο την αντικειμενική συνάρτηση ώστε να υποεκτιμάται ο υπολειπόμενος ωφέλιμος χρόνος ζωής.

Το μοντέλο ανίχνευσης βλάβης που αναπτύχθηκε έχει ακρίβεια 99.1% στις μετρήσεις εκπαίδευσης και 96.3% στις μετρήσεις αξιολόγησης, στην περίπτωση που υπάρχει βλάβη στα έδρανα κύλισης. Σε αντίστοιχη εργασία ανίχνευσης βλάβης από τον Κοτσανίδη, 2021 [20], αναπτύχθηκε ένα απλό μοντέλο νευρωνικού δικτύου μοντέλο με εισόδους τα χαρακτηριστικά Amplitude, Crest Factor, RMS, Kurtosis από το αρχικό σήμα αποσυνθεμένο σε διάφορα φάσματα, το οποίο πρόβλεπε ακρίβεια 100% στις μετρήσεις εκπαίδευσης και 90.2% στις μετρήσεις αξιολόγησης. Το μοντέλο που αναπτύχθηκε σε αυτή την εργασία, έχει παρόμοια απόδοση στις μετρήσεις εκπαίδευσης αλλά σημαντικά καλύτερη απόδοση στις μετρήσεις αξιολόγησης.

Το μοντέλο εκτίμησης του υπολειπόμενου ωφέλιμου χρόνου ζωής που αναπτύχθηκε σε αυτή τη μελέτη εξαιρετικά αποτελέσματα συγκριτικά με τη βιβλιογραφία. Ο Zhang, 2018 [19], ανέπτυξε ένα απλό μοντέλο νευρωνικού δικτύου με εισόδους που είναι στατιστικά ισοδύναμες με το RMS. Οι τιμές αυτών των χαρακτηριστικών υποβλήθηκαν σε παρεμβολή Weibull και χρησιμοποιήθηκαν για την εκπαίδευση του ΤΝΔ. Από την άλλη πλευρά, ο Qian, 2017 [21] παρουσιάζει μια μεθοδολογία που συνδυάζει ανάλυση στοχαστικής διαδικασίας, ανάλυση κυματομορφής και ανάλυση μη γραμμικής παραμέτρου για να αναλύσει τα δεδομένα που σχετίζονται με τη λειτουργία και την κατάσταση του εδράνου κύλισης.

Το μέτρο σύγκρισης που χρησιμοποιήθηκε είναι το μέσο απόλυτο σφάλμα (MAE) της εκτίμησης του RUL, όπως ορίζεται από την παρακάτω σχέση:

$$e = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (18)$$

Οι τιμές του μέτρου σύγκρισης e παρουσιάζονται στον **Πίνακα 15**.

Πίνακας 15: Αξιολόγηση μοντέλου εκτίμησης RUL συγκριτικά με βιβλιογραφία.

Σύνολο Μετρήσεων	Βλάβη	Αλγόριθμος	e
No.13	Εσ. Δακτύλιος	LSTM	0.0260
		Αναφορά [19]	0.1173
		Αναφορά [21]	4.61
No.14	Στ. Κύλισης	LSTM	0.0145
		Αναφορά [19]	0.6345
No.21	Εξ. Δακτύλιος	LSTM	0.0194
		Αναφορά [19]	0.0402
		Αναφορά [21]	0.980
No.33	Εξ. Δακτύλιος	LSTM	0.0350

Με βεβαιότητα μπορούμε να πούμε ότι η εργασία αυτή έχει καλή ακρίβεια στην εκτίμηση του RUL σε σύγκριση με τη βιβλιογραφία. Για μελλοντικές εργασίες, προτείνεται η μελέτη της ευρωστίας (robustness) του μοντέλου εκτίμησης του RUL, δηλαδή η αξιολόγηση της απόδοσής του σε πιθανά σενάρια αβεβαιότητας ή αντικατάστασης δεδομένων. Επίσης, προτείνεται η ανάπτυξη ενός μοντέλου νευρωνικού δικτύου με LSTM νευρώνες, αλλά ως είσοδοι να λαμβάνονται μετρήσεις από το χρονικό σήμα, προκειμένου να εξεταστεί η επίδραση της χρονικής σειράς των δεδομένων στην ακρίβεια της εκτίμησης του RUL.

Κατάλογος Σχημάτων

1	Αιτίες βλάβης εδράνων κύλισης. Πηγή: SKF, 2022 [2].	8
2	Οι τρεις κύριες βλάβες των εδράνων κύλισης. Βλάβη στον εξωτερικό δακτύλιο, βλάβη στον εσωτερικό δακτύλιο και βλάβη στα έδρανα κύλισης αντίστοιχα. Πηγή: An, 2014 [3].	8
3	Διαδικασία εκτίμησης της κατάστασης και υπολειπόμενης ωφέλιμης διάρκειας ζωής του εδράνου κύλισης.	13
4	Πειραματική εγκατάσταση, και σκαρίφημα εγκατάστασης. Πηγή από IMS, 2006 [13].	14
5	Ανασκόπηση μετρήσεων πρώτου συνόλου, στο πεδίο του χρόνου. . . .	15
6	Ανασκόπηση μετρήσεων δεύτερου και τρίτου συνόλου, στο πεδίο του χρόνου.	16
7	Χρονικά φάσματα συχνοτήτων για (α) Σύνολο μετρήσεων 2 - έδρανο κύλισης 1, (β) Σύνολο μετρήσεων 3 - έδρανο κύλισης 3	17
8	Επίπεδα αποσύνθεσης Wavelet. Πηγή: Haider, 2015 [16]	18
9	Αποτελέσματα της διαδικασίας αποθορυβοποίησης για διαφορετικό αριθμό επαναλήψεων.	19
10	Κατανόηση των χαρακτηριστικών συντελεστών ενέργειας	21
11	Χαρακτηριστικά του πεδίου χρόνου, από το πρώτο σύνολο μετρήσεων στο 3 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εσωτερικό δακτύλιο.	22
12	Χαρακτηριστικά του πεδίου συχνοτήτων, από το πρώτο σύνολο μετρήσεων στο 3 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εσωτερικό δακτύλιο.	22
13	Χαρακτηριστικά του πεδίου χρόνου, από το πρώτο σύνολο μετρήσεων στο 4 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στα στοιχεία κύλισης.	23
14	Χαρακτηριστικά του πεδίου συχνοτήτων, από το πρώτο σύνολο μετρήσεων στο 4 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στα στοιχεία κύλισης.	23
15	Χαρακτηριστικά του πεδίου χρόνου, από το δεύτερο σύνολο μετρήσεων στο 1 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.	24
16	Χαρακτηριστικά του πεδίου συχνοτήτων, από το δεύτερο σύνολο μετρήσεων στο 1 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.	24
17	Χαρακτηριστικά του πεδίου χρόνου, από το τρίτο σύνολο μετρήσεων στο 3 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.	25
18	Χαρακτηριστικά του πεδίου συχνοτήτων, από το τρίτο σύνολο μετρήσεων στο 3 ^ο έδρανο κύλισης. Εντοπίστηκε βλάβη στον εξωτερικό δακτύλιο.	25
19	Τιμές RMS από το πρώτο σύνολο μετρήσεων στο 3 ^ο έδρανο κύλισης.	26
20	Τιμές RMS από το πρώτο σύνολο μετρήσεων στο 4 ^ο έδρανο κύλισης.	27
21	Τιμές RMS από το δεύτερο σύνολο μετρήσεων στο 1 ^ο έδρανο κύλισης.	27
22	Τιμές RMS από το τρίτο σύνολο μετρήσεων στο 3 ^ο έδρανο κύλισης.	28
23	Στατιστικά τεστ F, MI στις μετρήσεις υπό βλάβη στον εσωτερικό δακτύλιο.	29
24	Πίνακας συσχέτισης, στις μετρήσεις υπό βλάβη στον εσωτερικό δακτύλιο.	30

25	Στατιστικά τεστ F, MI στις μετρήσεις υπό βλάβη στον εξωτερικό δακτύλιο.	31
26	Πίνακας συσχέτισης, στις μετρήσεις υπό βλάβη στον εξωτερικό δακτύλιο.	32
27	Στατιστικά τεστ F, MI στις μετρήσεις υπό βλάβη στα στοιχεία κύλισης.	33
28	Πίνακας συσχέτισης, στις μετρήσεις υπό βλάβη στα στοιχεία κύλισης.	34
29	Επιλογή δεδομένων.	36
30	Αρχιτεκτονική ΤΝΔ ανίχνευσης βλάβης.	38
31	Λογικό διάγραμμα βελτιστοποίησης αρχιτεκτονικής ΤΝΔ ανίχνευσης βλάβης.	39
32	Πίνακας σύγκρισης για τα δεδομένα εκπαίδευσης και αξιολόγησης. . .	40
33	Επιλογή δεδομένων για την εκτίμηση του RUL.	42
34	Διανύσματα εισόδων στα ΕΝΔ εκτίμησης του RUL.	44
35	Πορεία σύγκλισης μοντέλων (α) χωρίς επίπεδο εγκατάλειψης, (β) με επίπεδο εγκατάλειψης.	45
36	Αρχιτεκτονική των ΕΝΔ εκτίμησης του RUL.	45
37	Αποτελέσματα συντελεστή $penalty = 1$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.	46
38	Αποτελέσματα συντελεστή $penalty = 2$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.	47
39	Αποτελέσματα συντελεστή $penalty = 4$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.	47
40	Αποτελέσματα συντελεστή $penalty = 8$. (α) Εκτιμήσεις των RUL σε σύγκριση με τις πραγματικές τιμές και ιστογράμματα σφαλμάτων για τις τιμές (β) εκπαίδευσης και (γ) αξιολόγησης.	47
41	Σύγκλιση εκπαίδευσης των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.	49
42	Εκτιμήσεις του RUL των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.	50
43	Ιστόγραμμα σφαλμάτων μετρήσεων εκπαίδευσης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.	50
44	Ιστόγραμμα σφαλμάτων μετρήσεων αξιολόγησης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εσ. δακτύλιο.	50
45	Σύγκλιση εκπαίδευσης των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.	52
46	Εκτιμήσεις του RUL των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.	52
47	Ιστόγραμμα σφαλμάτων μετρήσεων εκπαίδευσης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.	52
48	Ιστόγραμμα σφαλμάτων μετρήσεων αξιολόγησης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στον εξ. δακτύλιο.	53
49	Σύγκλιση εκπαίδευσης των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.	54

50	Εκτιμήσεις του RUL των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.	54
51	Ιστόγραμμα σφαλμάτων μετρήσεων εκπαίδευσης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.	54
52	Ιστόγραμμα σφαλμάτων μετρήσεων αξιολόγησης, των τριών (3) καλύτερων μοντέλων για μετρήσεις υπό βλάβη στα στοιχεία κύλισης.	55
53	Εκτιμήσεις του RUL για το σύνολο μετρήσεων 1 του εδράνου κύλισης 3.	56
54	Εκτιμήσεις του RUL για το σύνολο μετρήσεων 1 του εδράνου κύλισης 4.	57
55	Εκτιμήσεις του RUL για το σύνολο μετρήσεων 2 του εδράνου κύλισης 1.	57
56	Εκτιμήσεις του RUL για το σύνολο μετρήσεων 3 του εδράνου κύλισης 3.	58
57	Εκτιμήσεις του RUL για το σύνολο μετρήσεων 3 του εδράνου κύλισης 3, στην περιοχή κοντά στην αστοχία	58
58	Ιστόγραμμα σφαλμάτων για όλες τις μετρήσεις που τα έδρανα κύλισης λειτουργούν υπό βλάβη.	59

Κατάλογος Πινάκων

1	Σύνολα πειραματικών μετρήσεων.	15
2	Χαρακτηριστικά των εδράνων κύλισης Rexpord ZA-2115	16
3	Χαρακτηριστικά σήματος στο πεδίο του χρόνου.	20
4	Χαρακτηριστικά σήματος στο πεδίο των συχνοτήτων.	21
5	Συνδυασμός χαρακτηριστικών για το κάθε μοντέλο πρόβλεψης.	35
6	Δεδομένα ανίχνευσης βλάβης.	37
7	Αρχιτεκτονική των ΤΝΔ ανίχνευσης βλάβης.	38
8	Αποτελέσματα βελτιστοποίησης για το ΤΝΔ ανίχνευσης βλάβης.	40
9	Κύριες μετρικές που υπολογίζονται από τον πίνακα σύγκυσης των δεδομένων αξιολόγησης.	41
10	Δεδομένα για την εκτίμηση του RUL.	43
11	Αρχιτεκτονική των ΕΝΔ εκτίμησης του RUL.	48
12	Αποτελέσματα βελτιστοποίησης για το ΕΝΔ εκτίμησης RUL υπό βλάβη στον εσ. δακτύλιο(1).	49
13	Αποτελέσματα βελτιστοποίησης για το ΕΝΔ εκτίμησης RUL υπό βλάβη στον εξ. δακτύλιο(2).	51
14	Αποτελέσματα βελτιστοποίησης για το ΕΝΔ εκτίμησης RUL υπό βλάβη στα στοιχεία κύλισης(3).	53
15	Αξιολόγηση μοντέλου εκτίμησης RUL συγκριτικά με βιβλιογραφία.	61

Αναφορές

- [1] Irwin, J. Summary of the NLC accelerator working group. United States: N. p., 1996. Web.
- [2] SKF 2022. SKF Maintenance and Lubrication Products
- [3] An, Xueli & Jiang, Dongxiang. (2014). Bearing fault diagnosis of wind turbine based on intrinsic time-scale decomposition frequency spectrum. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability. 228. 558-566. 10.1177/1748006X14539678.
- [4] R. B. Randall and J. Antoni. Rolling element bearing diagnostics-A tutorial. Mechanical Systems and Signal Processing 2011; 25:485-520.
- [5] Jaouher Ben Ali, Nader Fnaiech, Lotfi Saidi, Brigitte Chebel-Morello, Farhat Fnaiech, Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals, Applied Acoustics, Volume 89, 2015
- [6] Dong, M., & Zhang, J. (2017). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications.
- [7] Iooss, B., & Marais, A. (2008). A stochastic model for predicting the remaining useful life of rolling element bearings. Journal of Statistical Planning and Inference, 138(8), 2625-2638.
- [8] Wang, W., Hong, Y., Zhou, D. H., & Chen, J. (2009). A stochastic model for predicting remaining useful life of individual rolling element bearings. Mechanical Systems and Signal Processing, 23(3), 700-714.
- [9] Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In International Conference on Prognostics and Health Management (PHM08).
- [10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. This seminal paper introduced the LSTM architecture and has been highly influential in the field of deep learning.
- [11] Li, S., Xu, J., Wang, Q., Zhou, Z., & Cheng, Y. (2017). RUL prediction of bearings using LSTM networks. Journal of Intelligent Manufacturing, 28(4), 975-986.
- [12] Saxena, A., Celaya, J., & Saha, B. (2008). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. Mechanical Systems and Signal Processing, 22(6), 1393-1417.
- [13] Bearing Data of Center for Intelligent Maintenance Systems (IMS). University of Cincinnati. Available online: <http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/bearing>

- [14] Antoniadis, I., & Freyermuth, J. (2016). Wavelet methods in statistics: Some recent developments and their applications. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(3), 617-643.
- [15] Randall, R. B., & Antoni, J. (2011). Rolling element bearing diagnostics—a tutorial. *Mechanical Systems and Signal Processing*, 25(2), 485-520.
- [16] Haider, Ijlal & Shahbaz, Muhammad & Abdullah, Muhammad & Nazim, Muhammad. (2015). Feature Extraction for Identification of Extension and Flexion Movement of Wrist using EMG Signals. 10.1109/C-CECE.2015.7129375.
- [17] Jaouher Ben Ali, Brigitte Chebel-Morello, Lotfi Saidi, Simon Malinowski, Farhat Fnaiech, Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network, *Mechanical Systems and Signal Processing*, Volumes 56–57, 2015
- [18] Jun Zhu, Nan Chen, Changqing Shen, A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions, *Mechanical Systems and Signal Processing*, Volume 139, 2020
- [19] Zhang, N.; Wu, L.; Wang, Z.; Guan, Y. Bearing Remaining Useful Life Prediction Based on Naive Bayes and Weibull Distributions. *Entropy* 2018, 20, 944
- [20] Kotsanidis, K & Benardos, Panorios. (2021). Rolling element bearings fault classification based on feature extraction from acceleration data and artificial neural networks. *IOP Conference Series: Materials Science and Engineering*. 1037. 012008. 10.1088/1757-899X/1037/1/012008.
- [21] Qian, Y.; Yan, R.; Gao, R.X. A multi-time scale approach to remaining useful life prediction in rolling bearing. *Mech. Syst. Signal Process.* 2017, 83, 549–567.

Παράρτημα

```

1 clear, clc
2 BPF0=236;
3 BPF1=297;
4 BSF=278;
5
6 timeUnit = 'day';
7 hsbearing = fileEnsembleDatastore(...
8     fullfile('.', 'Bearing33'), ...
9     '.mat');
10 hsbearing.DataVariables = "vibration";
11 hsbearing.IndependentVariables = "Date";
12 hsbearing.SelectedVariables = ["Date", "vibration"];
13 hsbearing.ReadFcn = @helperReadData;
14 hsbearing.WriteToMemberFcn = @helperWriteToHSBearing;
15 tall(hsbearing)
16
17 fs = 20480; % Hz
18 reset(hsbearing)
19 hsbearing.DataVariables = ["vibration", "SpectralKurtosis"];
20
21
22 hsbearing.DataVariables = [hsbearing.DataVariables; ...
23     "Mean"; "Std"; "Skewness"; "Kurtosis"; "Peak2Peak"; ...
24     "RMS"; "CrestFactor"; "ShapeFactor"; "ImpulseFactor"; "MarginFactor"; "Energy";
25     ...
26     "SKMean"; "SKStd"; "SKSkewness"; "SKKurtosis"; "EnBPF0"; "EnBPF1"; "EnBSF";
27     "RMSE"; "aBPF0"; "aBPF1"; "aBSF"];
28
29 hsbearing.SelectedVariables = ["vibration", "SpectralKurtosis"];
30 reset(hsbearing)
31
32 i=1;
33 while hasdata(hsbearing)
34     data = read(hsbearing);
35     v = data.vibration{1};
36
37     % order = 15;
38     % framelen = 29;
39     %
40     % v = sgolayfilt(v,order,framelen);
41     [cA,cD] = dwt(v,'db4');
42     v = idwt(cA,zeros(size(cA)),'db4');
43
44     SK = data.SpectralKurtosis{1}.SK;
45     features = table;
46     h1=abs(hilbert(v));
47     U=abs(fft(h1,fs)/fs);
48     U(1)=0; %Set DC to zero
49
50     Eo=0; Ei=0; Eb=0;
51     for k=1:3
52         f1o=floor(0.95*k*BPF0);
53         f2o=ceil(1.05*k*BPF0);
54
55         f1i=floor(0.95*k*BPF1);
56         f2i=ceil(1.05*k*BPF1);
57
58         f1b=floor(0.95*k*BSF);
59         f2b=ceil(1.05*k*BSF);
60
61         Eo=Eo+(.5*U(f1o)+.5*U(f2o)+sum(U(f1o+1:f2o-1)));
62         Ei=Ei+(.5*U(f1i)+.5*U(f2i)+sum(U(f1i+1:f2i-1)));
63         Eb=Eb+(.5*U(f1b)+.5*U(f2b)+sum(U(f1b+1:f2b-1)));
64     end
65     fstop=950;
66     En=(.5*U(1)+.5*U(fstop)+sum(U(1:fstop)));

```

Listing 1: Κώδικας matlab εξαγωγής χαρακτηριστικών.

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from LabelCreation import *
5 import seaborn as sns
6 from sklearn.feature_selection import f_regression, mutual_info_regression
7 #import matplotlib as mpl
8
9 percentage=0.4
10
11 def TimeCreation(df,size,in1):
12     timecolumn=np.zeros(size)
13     for i in range(size):
14         if i <= in1:
15             timecolumn[i]=1.0
16         else:
17             timecolumn[i]=1-(i-in1)/(size-in1-1)
18     df["Time"] = timecolumn.tolist()
19     print(df)
20
21 def remove_random_rows(dataframe):
22
23     N = dataframe[dataframe['Time'] != 1.0].shape[0] # Count the number of rows
24         with non-zero Fault values
25     zero_rows = dataframe[dataframe['Time'] == 1.0] # Get rows where Fault is 0
26     random_rows = zero_rows.sample(round(dataframe.shape[0]-N*(percentage+1))) #
27         Randomly select N rows from the zero_rows dataframe
28     dataframe = dataframe.drop(random_rows.index) # Drop the randomly selected
29         rows from the original dataframe
30     dataframe.reset_index(inplace=True, drop=True)
31     return dataframe
32
33 # Set the font to use LaTeX
34 #mpl.rcParams['text.usetex'] = True
35 #mpl.rcParams['font.family'] = 'serif'
36
37 ki = 10
38 Fault=3
39 ystring="Time"
40 type="WSn"
41 N=17+2
42 miTotal = np.zeros([ki,N])
43 FiTotal = np.zeros([ki,N])
44
45 miT = np.zeros(N)
46 FiT = np.zeros(N)
47
48 if Fault==1:
49     df = pd.read_csv("featureTableEnergies13"+type+".csv")
50     size=len(df["RMS"].to_numpy())
51
52     in1=1808
53     TimeCreation(df,size,in1)
54     remove_random_rows(df)
55     #df.drop(index=df.index[:in1], axis=0, inplace=True)
56
57 elif Fault==2:
58     df1 = pd.read_csv("featureTableEnergies24"+type+".csv")
59     size1=len(df1["RMS"].to_numpy())
60
61     df2 = pd.read_csv("featureTableEnergies33"+type+".csv")
62     size2=len(df2["RMS"].to_numpy())
63
64     id11=530
65     id21=5965
66     TimeCreation(df1,size1,id11)

```

```

64     remove_random_rows(df1)
65     TimeCreation(df2,size2,id21)
66     remove_random_rows(df2)
67     df1.drop(index=df1.index[:id11], axis=0, inplace=True)
68     df2.drop(index=df2.index[:id21], axis=0, inplace=True)
69
70     df = df1.append(df2, ignore_index=True)
71
72
73 elif Fault==3:
74     df = pd.read_csv("featureTableEnergies14"+type+".csv")
75     size=len(df["RMS"].to_numpy())
76
77     id1=1430
78
79     TimeCreation(df,size,id1)
80     remove_random_rows(df)
81     df.drop(index=df.index[:id1], axis=0, inplace=True)
82
83
84 #-----Seperate Matrices
85 #Target
86 Y_df = df[ystring]
87 #print(Y_df)
88
89 #drop other outputs
90 X_df = df.drop(['Time', 'Date'],axis=1)
91 X_df=X_df.iloc[:, :-3]
92
93 columns = X_df.columns.tolist()
94
95 # Move the last column to the 12th position
96 columns.insert(11, columns[-1])
97 columns.pop()
98
99 new_column_names=['1. Mean', '2. Std', '3. Skewness', '4. Kurtosis', '5. Peak2Peak',
100                  '6. RMS', '7. CrestFactor', '8. ShapeFactor', '9. ImpulseFactor', '10. MarginFactor',
101                  '11. Energy', '12. RMSEE', '13. SKMean', '14. SKStd', '15. SKSkewness', '16. SKKurtosis',
102                  '17. FBPF0', '18. FBPFI', '19. FBSF']
103
104
105
106 #-----Make nd arrays
107 X = X_df.to_numpy()
108 Y = Y_df.to_numpy()
109
110 # F-test for regression
111 f_scores, f_p_values = f_regression(X_df, Y)
112
113 # MI test for regression
114 mi_scores = mutual_info_regression(X_df, Y)
115
116 # Create a DataFrame to store the scores
117 scores_df = pd.DataFrame({'Feature': X_df.columns, 'F-Score': f_scores, 'MI-Score': mi_scores})
118
119 # Sort the DataFrame by F-Score in descending order
120 scores_df = scores_df.sort_values('F-Score', ascending=False)
121 font_size = 18
122 # Set up the figure with subplots
123 fig, axes = plt.subplots(1, 2, figsize=(12, 6))
124
125 # Define the colors
126 color_f_score = '#B22222' # Reddish color for F-Score
127 color_mi_score = '#B22222' # Darker blue color for MI Score
128
129 # Set the font size

```

```

130 font_size = 18
131
132 # Plot the F-Scores
133 sns.barplot(x='F-Score', y='Feature', data=scores_df, ax=axes[0], color=
    color_f_score)
134 axes[0].set_title('F-Score for Features', fontsize=font_size)
135 axes[0].set_xlabel('F-Score', fontsize=font_size)
136 axes[0].set_ylabel('Feature', fontsize=font_size)
137 axes[0].grid(True)
138
139 # Sort the DataFrame by MI-Score in descending order
140 scores_df = scores_df.sort_values('MI-Score', ascending=False)
141
142 # Plot the MI scores
143 sns.barplot(x='MI-Score', y='Feature', data=scores_df, ax=axes[1], color=
    color_mi_score)
144 axes[1].set_title('MI-Score for Features', fontsize=font_size)
145 axes[1].set_xlabel('MI-Score', fontsize=font_size)
146 axes[1].set_ylabel('Feature', fontsize=font_size)
147 axes[1].grid(True)
148
149 # Set the tick label font size
150 for ax in axes:
151     ax.tick_params(axis='both', labelsize=font_size)
152
153 # Adjust the spacing between subplots
154 plt.tight_layout()
155
156 # Show the figure
157 plt.show(block=False)
158 from matplotlib.colors import LinearSegmentedColormap
159 # Calculate the correlation matrix
160
161 # Calculate the correlation matrix
162 # Calculate the correlation matrix
163 correlation_matrix = X_df.corr()
164
165 # Set up the plot figure
166 fig, ax = plt.subplots(figsize=(10, 8))
167
168 # Create a colormap ranging from black to red
169 cmap = plt.cm.RdYlBu_r
170
171 # Plot the correlation matrix
172 heatmap = ax.imshow(correlation_matrix, cmap=cmap, vmin=-1, vmax=1)
173
174 # Add colorbar
175 cbar = plt.colorbar(heatmap)
176
177 # Set the colorbar ticks and labels
178 cbar.set_ticks([-1, -0.5, 0, 0.5, 1])
179 cbar.set_ticklabels([-1, -0.5, 0, 0.5, 1])
180
181 # Set the tick labels and rotate them
182 ax.set_xticks(np.arange(correlation_matrix.shape[1]))
183 ax.set_yticks(np.arange(correlation_matrix.shape[0]))
184 ax.set_xticklabels(correlation_matrix.columns, rotation=45, ha='right')
185 ax.set_yticklabels(correlation_matrix.columns)
186
187 # Print the correlation values on each cell
188 for i in range(correlation_matrix.shape[0]):
189     for j in range(correlation_matrix.shape[1]):
190         ax.text(j, i, f'{correlation_matrix.iloc[i,j]:.2f}', ha='center', va='
    center', color='black')
191
192 # Set the title
193 ax.set_title('Correlation Matrix')
194
195 # Display the plot
196 plt.show(block=False)

```


Listing 2: Κώδικας python για την στατιστική ανάλυση.

```

1 import numpy as np
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler, LabelEncoder
5 import matplotlib.pyplot as plt
6 from LabelCreation import *
7 from keras.models import Sequential
8 from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense
9 from keras.utils import np_utils
10 from sklearn.metrics import confusion_matrix
11 import seaborn as sns
12 import tensorflow as tf
13 from keras.callbacks import ModelCheckpoint
14 from keras.layers import Dropout
15
16 import os
17 import joblib
18
19 def FaultLabel(df, size, in1, type):
20     classcolumn=np.zeros(size)
21
22
23
24     for i in range(size):
25         if i <= in1:
26             classcolumn[i]=0
27         else:
28             classcolumn[i]=type
29
30     df["Fault"] = classcolumn.tolist()
31 def fldcreation(fld):
32     # Check if the directory exists
33     if not os.path.exists(fld):
34         # If it doesn't exist, create it
35         os.makedirs(fld)
36 def epochplot(train_loss, val_loss, path):
37     plt.clf()
38     plt.plot(range(1, len(train_loss) + 1), train_loss, color="blue", label='
39         Training Loss')
40     plt.plot(range(1, len(val_loss) + 1), val_loss, color="red", label='Test Loss')
41     plt.xlabel('Epoch')
42     plt.ylabel('Loss')
43     plt.title('Training and Test Loss')
44     plt.legend()
45     plt.savefig(path+'epoch.png')
46     plt.close('all')
47 def remove_random_rows(dataframe):
48     N = dataframe[dataframe['Fault'] != 0].shape[0] # Count the number of rows
49     # with non-zero Fault values
50     zero_rows = dataframe[dataframe['Fault'] == 0] # Get rows where Fault is 0
51     random_rows = zero_rows.sample(dataframe.shape[0]-2*N) # Randomly select N
52     # rows from the zero_rows dataframe
53     dataframe = dataframe.drop(random_rows.index) # Drop the randomly selected
54     # rows from the original dataframe
55     dataframe.reset_index(inplace=True, drop=True)
56     return dataframe
57
58 scalerX = StandardScaler()
59 ft1 = pd.read_csv("featureTableEnergies13WSn.csv")
60 size1=len(ft1["RMS"].to_numpy())
61
62 ft2 = pd.read_csv("featureTableEnergies14WSn.csv")
63 size2=len(ft2["RMS"].to_numpy())
64
65 ft3 = pd.read_csv("featureTableEnergies24WSn.csv")
66 size3=len(ft3["RMS"].to_numpy())

```

```

63
64 ft4 = pd.read_csv("featureTableEnergies33WSn.csv")
65 size4=len(ft4["RMS"].to_numpy())
66
67
68 id1=1808
69 id2=1400
70 id3=586
71 id4=5956
72 Nepoch=100
73 smallerDF=1
74
75 FaultLabel(ft1,size1,id1,1)
76 FaultLabel(ft2,size2,id2,3)
77 FaultLabel(ft3,size3,id3,2)
78 FaultLabel(ft4,size4,id4,2)
79
80
81 if smallerDF==1:
82     df1=remove_random_rows(ft1)
83     df2=remove_random_rows(ft2)
84     df3=remove_random_rows(ft3)
85     df4=remove_random_rows(ft4)
86     df = pd.concat([df1, df2, df3, df4])
87 else:
88     df = pd.concat([ft1, ft2, ft3, ft4])
89
90
91 df.reset_index(inplace=True, drop=True)
92
93 fault_values = df['Fault'].tolist()
94 indexes = df.index.tolist()
95
96
97
98
99 MATRIX=[]
100 cn=0
101 #maximum 2 layers
102 for i in range(25,55,5):
103     MATRIX.append([i, 0])
104     cn=cn+1
105 for i in range(25,55,5):
106     for j in range(5,18,3):
107         MATRIX.append([i, j])
108         cn=cn+1
109
110
111 DF=pd.DataFrame()
112 DF = pd.DataFrame([{'Layer1': 0, 'Layer2': 0, 'Test_loss': 0, 'Test_accuracy': 0, 'Train_loss': 0}])
113 DF = DF.drop(labels=0, axis=0)
114 #FEATURES LABELS
115 #0. Date, 1. Mean, 2. Std, 3. Skewness, 4. Kurtosis, 5. Peak2Peak, 6. RMS, 7.
    CrestFactor
116 #8. ShapeFactor, 9. ImpulseFactor, 10. MarginFactor, 11. Energy, 12. SKMean, 13.
    SKStd, 14. SKSkewness
117 #15. SKKurtosis, 16. EnBPF0,17. EnBPF1, 18. EnBSF 19. RMSEE
118
119 #Array Selection
120 #array=[1,6,12,16,19,8,5]
121 array=[4, 8, 11, 12, 5, 16, 17, 18, 19]
122 LOG_DIR = './Apotelsmata15/'+"FaultDitECTION_"+str(array)
123 fldcreation(LOG_DIR)
124
125 X=df.iloc[:,array]
126 y=df.iloc[:, -1]
127
128 X = X.values
129 y = y.values

```

```

130
131 # Ensure X and y have the same number of samples
132 assert X.shape[0] == y.shape[0], "X and y must have the same number of samples"
133
134 # Split the data into training and testing sets
135 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
136         random_state=42)
137
138 # Scale the data using StandardScaler
139 scaler = StandardScaler()
140 X_train_scaled = scaler.fit_transform(X_train)
141 X_test_scaled = scaler.transform(X_test)
142
143 scaler_path = os.path.join(LOG_DIR, 'scaler.pkl')
144 joblib.dump(scaler, scaler_path)
145 # Encode the class labels
146 label_encoder = LabelEncoder()
147 y_train_encoded = label_encoder.fit_transform(y_train)
148 y_test_encoded = label_encoder.transform(y_test)
149
150 # Convert the encoded labels to one-hot encoded format
151 num_classes = len(label_encoder.classes_)
152 y_train_onehot = np_utils.to_categorical(y_train_encoded, num_classes)
153 y_test_onehot = np_utils.to_categorical(y_test_encoded, num_classes)
154
155 # Define the model
156 for i in range(cn):
157     print('-----')
158     print(str(i)+"/"+str(cn))
159     print('-----')
160
161     model = Sequential()
162     model.add(Dense(MATRIX[i][0], activation='relu', input_shape=(X_train_scaled.
163         shape[1],)))
164     if MATRIX[i][1]!=0:
165         model.add(Dropout(0.15))
166         model.add(Dense(MATRIX[i][1], activation='relu'))
167         model.add(Dropout(0.15))
168     model.add(Dense(num_classes, activation='softmax'))
169
170     FLD2=LOG_DIR+"/Models/"+str(MATRIX[i][0])+" "+str(MATRIX[i][1])+"/model"
171     checkpoint_path = os.path.join(FLD2, 'model_checkpoint.h5')
172     checkpoint = ModelCheckpoint(checkpoint_path,
173         save_weights_only=True,
174         save_best_only=False,
175         save_freq=1)
176
177     # Compile the model
178     model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['
179         accuracy'])
180
181     # Train the model
182     history=model.fit(X_train_scaled, y_train_onehot, validation_data=(
183         X_test_scaled, y_test_onehot), batch_size=32, epochs=Nepoch, verbose=0)
184
185     # Evaluate the model on the test set
186     score = model.evaluate(X_test_scaled, y_test_onehot, verbose=0)
187     print("Test loss:", score[0])
188     print("Test accuracy:", score[1])
189     score1 = model.evaluate(X_train_scaled, y_train_onehot, verbose=0)
190
191     DF.loc[len(DF.index)] = [MATRIX[i][0], MATRIX[i][1], score[0], score[1], score1
192         [0]]
193
194     FLD=LOG_DIR+"/Models/"+str(MATRIX[i][0])+" "+str(MATRIX[i][1])
195     fldcreation(FLD)
196     X1=scaler.fit_transform(ft1.iloc[:,array])
197     X2=scaler.fit_transform(ft2.iloc[:,array])
198     X3=scaler.fit_transform(ft3.iloc[:,array])
199     X4=scaler.fit_transform(ft4.iloc[:,array])
200     Y1=np.argmax(model.predict(X1), axis=1)

```

```

195 Y2=np.argmax(model.predict(X2), axis=1)
196 Y3=np.argmax(model.predict(X3), axis=1)
197 Y4=np.argmax(model.predict(X4), axis=1)
198
199
200
201 plt.scatter(range(0,len(Y1)),Y1)
202 plt.xlabel("Predicted_Label")
203 plt.ylabel("True_Label")
204 plt.savefig(FLD+'Sxima.png')
205 plt.close('all')
206
207
208
209 y_train_pred = np.argmax(model.predict(X_train_scaled), axis=1)
210 y_test_pred = np.argmax(model.predict(X_test_scaled), axis=1)
211
212 # Get the true labels for the training and test sets
213 y_train_true = np.argmax(y_train_onehot, axis=1)
214 y_test_true = np.argmax(y_test_onehot, axis=1)
215
216 # Compute the confusion matrices for training and test sets
217 cm_train = confusion_matrix(y_train_true, y_train_pred)
218 cm_test = confusion_matrix(y_test_true, y_test_pred)
219
220 # Get the class labels
221 class_labels = np.unique(np.concatenate((y_train_true, y_test_true)))
222
223 train_loss = history.history['loss']
224 val_loss = history.history['val_loss']
225 epochplot(train_loss, val_loss, FLD)
226 #epochplot(train_loss, val_loss, FLD)
227
228 # Plot the confusion matrices in subplots
229 fig, axes = plt.subplots(1, 2, figsize=(12, 6))
230
231 # Training set confusion matrix
232 plt.sca(axes[0])
233 sns.heatmap(cm_train, annot=True, fmt="d", cmap="Blues", xticklabels=
class_labels, yticklabels=class_labels)
234 plt.title("Training_Set_Confusion_Matrix")
235 plt.xlabel("Predicted_Label")
236 plt.ylabel("True_Label")
237
238 # Test set confusion matrix
239 plt.sca(axes[1])
240 sns.heatmap(cm_test, annot=True, fmt="d", cmap="Blues", xticklabels=
class_labels, yticklabels=class_labels)
241 plt.title("Test_Set_Confusion_Matrix")
242 plt.xlabel("Predicted_Label")
243 plt.ylabel("True_Label")
244
245 plt.tight_layout()
246 plt.savefig(FLD+'ConfusiomMatrix.png')
247 plt.close('all')
248
249 DF = DF.sort_values(DF.columns[2])
250 filenm="results.xlsx"
251 DF.to_excel(f"{LOG_DIR}/{filenm}", index=False)
252
253 model_path = os.path.join(FLD2, 'model.h5')
254 model.save(model_path)
255
256
257
258
259
260
261 print(DF)
262

```

```

263 np.save(os.path.join(LOG_DIR, 'X_train_scaled.npy'), X_train_scaled)
264 np.save(os.path.join(LOG_DIR, 'y_train_onehot.npy'), y_train_onehot)
265 np.save(os.path.join(LOG_DIR, 'X_test_scaled.npy'), X_test_scaled)
266 np.save(os.path.join(LOG_DIR, 'y_test_onehot.npy'), y_test_onehot)

```

Listing 3: Κώδικας python βελτιστοποίησης του ΝΔ ανύχνευσης βλάβης.

```

1  import tensorflow as tf
2  import os
3  import pandas as pd
4  import numpy as np
5  import matplotlib.pyplot as plt
6  from sklearn.model_selection import train_test_split
7  from sklearn.preprocessing import StandardScaler
8  from tensorflow import keras
9  from keras.models import Model
10 from keras.layers import Dense
11 from keras.models import Sequential
12 from keras.layers import *
13 from keras.callbacks import ModelCheckpoint
14 from keras.losses import MeanSquaredError
15 from keras.metrics import RootMeanSquaredError
16 from keras.optimizers import Adam
17 from sklearn.metrics import mean_squared_error as mse
18 import os
19 from scipy.stats import skew
20 from scipy.stats import kurtosis
21 import joblib
22 from keras.layers import LSTM, Conv1D, MaxPooling1D, Flatten, Dense, Reshape
23 from keras.callbacks import EarlyStopping
24 from keras import regularizers
25
26 # Set random seeds for reproducibility
27 seed_value = 300
28 os.environ['PYTHONHASHSEED'] = str(seed_value)
29 np.random.seed(seed_value)
30 tf.random.set_seed(seed_value)
31 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
32
33 #1. Inner, 2. Outer, 3. Rollers
34 Fault=1
35 safe=1
36 window_size=6
37 type="WS"
38 layertype="LSTM"
39 Nepoch=150
40 penalty=4
41 percentage=0.4
42 typemodel=1
43 def custom_loss(y_true, y_pred):
44     squared_diff_sum = tf.TensorArray(tf.float32, size=0, dynamic_size=True)
45
46     for i in range(len(y_true)):
47         if y_true[i]>y_pred[i]:
48             squared_diff = (y_true[i] - y_pred[i]) ** 2
49         else:
50             squared_diff = penalty*(y_true[i] - y_pred[i]) ** 2
51         squared_diff_sum = squared_diff_sum.write(i, squared_diff)
52
53     squared_diff_sum = squared_diff_sum.stack()
54     mse = tf.reduce_mean(squared_diff_sum)
55     return mse
56 def TimeCreation(df,size,in1):
57     timecolumn=np.zeros(size)
58     for i in range(size):
59         if i <= in1:
60             timecolumn[i]=1.0
61         else:
62             timecolumn[i]=1-(i-in1)/(size-in1-1)
63     df["Time"] = timecolumn.tolist()

```

```

64 def df_to_X_y2(df, window_size):
65     df_as_np = df
66     X = []
67     for i in range(len(df_as_np)-window_size):
68         row = [r for r in df_as_np[i:i+window_size]]
69         X.append(row)
70     return np.array(X)
71 def plot_predictions1(model, X1, y1, X2, y2,path):
72     predictions1 = model.predict(X1).flatten()
73
74     df1 = pd.DataFrame(data={'Predictions':predictions1, 'Actuals':y1})
75
76     predictions2 = model.predict(X2).flatten()
77     df2 = pd.DataFrame(data={'Predictions':predictions2, 'Actuals':y2})
78     plt.clf()
79     plt.scatter(y2,df2['Predictions'], color="blue", label = "Train-Predictions-mse$
80         {:.5f}".format(mse(y2, predictions2)))
81     plt.scatter(y1,df1['Predictions'], color="red", label = "Test-Predictions-mse$
82         {:.5f}".format(mse(y1, predictions1)))
83     plt.plot([0,1],[0,1], color="black", label = "Actual")
84     plt.legend()
85     plt.xlabel('Real (RUL)')
86     plt.ylabel('Predicted (RUL)')
87     plt.savefig(path+'/predictions.png')
88     plt.close('all')
89 def histogram(X1,X2,path):
90     plt.clf()
91     fig, (ax1, ax2) = plt.subplots(1, 2)
92     ax1.hist(X1, color='blue')
93     ax2.hist(X2, color='red')
94     ax1.axvline(0, color='black')
95     ax2.axvline(0, color='black')
96
97     ax1.set_xlabel('error of RUL')
98     ax1.set_ylabel('frequency')
99     ax1.set_title('Test error histogram')
100
101     ax2.set_xlabel('error of RUL')
102     ax2.set_ylabel('frequency')
103     ax2.set_title('Train error histogram')
104     plt.savefig(path+'/error.png')
105     plt.close('all')
106 def fldcreation(fld):
107     # Check if the directory exists
108     if not os.path.exists(fld):
109         # If it doesn't exist, create it
110         os.makedirs(fld)
111 def epochplot(train_loss, val_loss,path):
112     plt.clf()
113     plt.plot(range(1, len(train_loss) + 1), train_loss, color="blue", label='
114         Training Loss')
115     plt.plot(range(1, len(val_loss) + 1), val_loss, color="red", label='Test Loss')
116     plt.xlabel('Epoch')
117     plt.ylabel('Loss')
118     plt.title('Training and Test Loss')
119     plt.legend()
120     plt.savefig(path+'/epoch.png')
121     plt.close('all')
122 def remove_random_rows(dataframe):
123
124     N = dataframe[dataframe['Time'] != 1.0].shape[0] # Count the number of rows
125     with non-zero Fault values
126     zero_rows = dataframe[dataframe['Time'] == 1.0] # Get rows where Fault is 0
127     random_rows = zero_rows.sample(round(dataframe.shape[0]-N*(percentage+1))) #
128     Randomly select N rows from the zero_rows dataframe
129     dataframe = dataframe.drop(random_rows.index) # Drop the randomly selected
130     rows from the original dataframe
131     dataframe.reset_index(inplace=True, drop=True)
132     return dataframe

```

```

128 MATRIX=[]
129 cn=0
130 #maximum 2 layers
131 # for i in range(20,55,5):
132 #     MATRIX.append([i, 0])
133 #     cn=cn+1
134 for i in range(10,53,6):
135     for j in range(2,26,5):
136         MATRIX.append([i, j])
137         cn=cn+1
138
139 #FEATURES LABELS
140 #0. Date, 1. Mean, 2. Std, 3. Skewness, 4. Kurtosis, 5. Peak2Peak, 6. RMS, 7.
    CrestFactor
141 #8. ShapeFactor, 9. ImpulseFactor, 10. MarginFactor, 11. Energy, 12. SKMean, 13.
    SKStd, 14. SKSkewness
142 #15. SKKurtosis, 16. EnBPF0,17. EnBPFI, 18. EnBSF 19. RMSEE
143
144 if Fault==1:
145     df = pd.read_csv("featureTableEnergies13"+type+".csv")
146     size=len(df["RMS"].to_numpy())
147
148     in1=1808
149     TimeCreation(df,size,in1)
150     #array=[5,7,8,17,19]
151     #array=[5,7,19,11,10]
152     array=[8,19,7,4]
153 elif Fault==2:
154     df1 = pd.read_csv("featureTableEnergies24"+type+".csv")
155     size1=len(df1["RMS"].to_numpy())
156
157     df2 = pd.read_csv("featureTableEnergies33"+type+".csv")
158     size2=len(df2["RMS"].to_numpy())
159
160     id11=586
161     id21=5956
162     TimeCreation(df1,size1,id11)
163     TimeCreation(df2,size2,id21)
164
165
166 if safe==1:
167     df1=remove_random_rows(df1)
168     df2=remove_random_rows(df2)
169 else:
170     df1 = df1[df1.Time != 0]
171     df2 = df2[df2.Time != 0]
172
173     size2=len(df2["RMS"].to_numpy())
174     size1=len(df1["RMS"].to_numpy())
175     #array=[12,19,5,10]
176     #array=[12,19,6,8,10,3]
177     array=[5,8,10,19,12,16]
178     Y1 = df1['Time']
179     Y2 = df2['Time']
180
181
182     df1=df1.iloc[:,array]
183     df2=df2.iloc[:,array]
184
185     print(df1)
186     X1= df1
187     X2= df2
188     if window_size!=0:
189         Y1=Y1.iloc[window_size:]
190         Y2=Y2.iloc[window_size:]
191
192     scaler = StandardScaler()
193     scaler.fit(pd.concat([X1, X2]))
194     X1=scaler.transform(X1)
195     X2=scaler.transform(X2)

```

```

196
197     if window_size!=0:
198         X1= df_to_X_y2(X1,window_size)
199     X1_train, X1_test, y1_train, y1_test = train_test_split(X1, Y1, test_size=0.2,
200         random_state=1)
201
202     if window_size!=0:
203         X2= df_to_X_y2(X2,window_size)
204     X2_train, X2_test, y2_train, y2_test = train_test_split(X2, Y2, test_size=0.2,
205         random_state=1)
206
207     X_train=np.concatenate((X1_train, X2_train), axis=0)
208     y_train=pd.concat([y1_train, y2_train])
209     X_test=np.concatenate((X1_test, X2_test), axis=0)
210     y_test=pd.concat([y1_test, y2_test])
211 elif Fault==3:
212     df = pd.read_csv("featureTableEnergies14"+type+".csv")
213     size=len(df["RMS"].to_numpy())
214
215     id1=1400
216     TimeCreation(df,size,id1)
217     array=[19,12,9]
218
219 if Fault!=2:
220     if safe==1:
221         df=remove_random_rows(df)
222     else:
223         df = df[df.Time != 0]
224         size=len(df["RMS"].to_numpy())
225         Y = df['Time']
226         df=df.iloc[:,array]
227         X= df
228     if window_size!=0:
229         Y=Y.iloc[window_size:]
230
231     scaler = StandardScaler()
232     scaler.fit(X)
233     X=scaler.transform(X)
234
235     if window_size!=0:
236         X= df_to_X_y2(X,window_size)
237     X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,
238         random_state=1)
239
240 #print(df)
241 LOG_DIR = './ApotelSmata13/'+str(Fault)+type+"_"+layertype+str(array)+"ws"+str(
242     window_size)+"p"+str(percentage)+"s"+str(percentage*safe)+"m"+str(typemodel)
243 fldcreation(LOG_DIR)
244
245 DF=pd.DataFrame()
246 DF = pd.DataFrame([{'LSTM': 0, 'Layer1': 0, 'MSE': 0, 'Skew': 0, 'MSEp': 0, 'TrMSE'
247     : 0, 'TrSkew': 0, 'TrMSEp': 0}])
248
249 scaler_path = os.path.join(LOG_DIR, 'scaler.pkl')
250 joblib.dump(scaler, scaler_path)
251
252 for i in range(cn):
253     print('-----')
254     print(str(i)+"/"+str(cn))
255     print('-----')
256
257 if typemodel==1:
258     model = Sequential()
259     model.add(InputLayer((window_size, len(array))))
260     model.add(LSTM(MATRIX[i][0]))
261     if MATRIX[i][1]!=0:
262         #model.add(Dropout(0.15))

```



```

261     model.add(Dense(MATRIX[i][1], 'relu'))
262     #model.add(Dropout(0.15))
263     model.add(Dense(1, 'sigmoid')) #Outup Layer
264 else:
265     model = Sequential()
266     model.add(InputLayer((window_size, len(array))))
267     model.add(LSTM(MATRIX[i][0]))
268     if MATRIX[i][1]!=0:
269         model.add(Dropout(0.15))
270         model.add(Dense(MATRIX[i][1], 'relu'))
271     model.add(Dropout(0.15))
272     model.add(Dense(1, 'sigmoid')) #Outup Layer
273
274
275 FLD2=LOG_DIR+"/Models/"+str(MATRIX[i][0])+" "+str(MATRIX[i][1])+"/model"
276 checkpoint_path = os.path.join(FLD2, 'model_checkpoint.h5')
277 checkpoint = ModelCheckpoint(checkpoint_path,
278                             save_weights_only=True,
279                             save_best_only=False,
280                             save_freq=1)
281
282
283
284 model.compile(optimizer=Adam(learning_rate=0.0005), loss=custom_loss)
285 history=model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=
    Nepoch, callbacks=[checkpoint], verbose=0)
286
287 predictions1 = model.predict(X_test).flatten()
288 error1=(y_test-predictions1).to_numpy()
289 predictions2 = model.predict(X_train).flatten()
290 error2=(y_train-predictions2).to_numpy()
291
292
293 mse1=0
294 mse2=0
295 for i1 in range(len(error1)):
296     if error1[i1]>0:
297         mse1 += error1[i1]**2
298     else:
299         mse1 += penalty*error1[i1]**2
300
301 for i2 in range(len(error2)):
302     if error2[i2]>0:
303         mse2 += error2[i2]**2
304     else:
305         mse2 += penalty*error2[i2]**2
306
307 mse1 /= len(error1)
308 mse2 /= len(error2)
309
310
311 DF.loc[len(DF.index)] = [MATRIX[i][0],MATRIX[i][1],mse(y_test, predictions1),
    skew(error1, axis=0, bias=True),mse1,mse(y_train, predictions2),skew(error2
    , axis=0, bias=True),mse2]
312 FLD=LOG_DIR+"/Models/"+str(MATRIX[i][0])+" "+str(MATRIX[i][1])
313 fldcreation(FLD)
314
315 plot_predictions1(model, X_test, y_test, X_train, y_train,FLD)
316 histogram(error1,error2,FLD)
317
318 train_loss = history.history['loss']
319 val_loss = history.history['val_loss']
320 epochplot(train_loss, val_loss,FLD)
321
322 #SAVE model
323 fldcreation(FLD2)
324
325
326 scaler_path = os.path.join(FLD2, 'scaler.pkl')
327 joblib.dump(scaler, scaler_path)

```

```

328
329     model_path = os.path.join(FLD2, 'model.h5')
330     model.save(model_path)
331
332
333
334 DF = DF.drop(labels=0, axis=0)
335 print(DF)
336 DF = DF.sort_values(DF.columns[4])
337 filenm="results.xlsx"
338 DF.to_excel(f"{LOG_DIR}/{filenm}", index=False)
339
340 np.save(os.path.join(LOG_DIR, 'x_train.npy'), X_train)
341 np.save(os.path.join(LOG_DIR, 'y_train.npy'), y_train)
342 np.save(os.path.join(LOG_DIR, 'x_test.npy'), X_test)
343 np.save(os.path.join(LOG_DIR, 'y_test.npy'), y_test)

```

Listing 4: Κώδικας python επανεκπαίδευσης των ΝΔ ανύχνευσης βλάβης.

```

1 import tensorflow as tf
2 import os
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 from sklearn.model_selection import train_test_split
7 from sklearn.preprocessing import StandardScaler
8 from tensorflow import keras
9 from keras.models import Model
10 from keras.layers import Dense
11 from keras.models import Sequential
12 from keras.layers import *
13 from keras.callbacks import ModelCheckpoint
14 from keras.losses import MeanSquaredError
15 from keras.metrics import RootMeanSquaredError
16 from keras.optimizers import Adam
17 from sklearn.metrics import mean_squared_error as mse
18 import os
19 from scipy.stats import skew
20 from scipy.stats import kurtosis
21 import joblib
22 from keras.models import load_model
23 from keras.utils import custom_object_scope
24
25 Nepoch=2500
26 penalty=8
27 subplot_option="1"
28 load_folder="Apotelsmata13/1WS_LSTM[8,19,7,4]ws6p8s0.4m2"
29 folder=load_folder+"/BestResults"
30
31
32 def create_model_string(folder_path,row_number):
33     row_number=row_number+1
34     row_number=row_number+6
35     file_path = os.path.join(folder_path, 'results.xlsx')
36     df = pd.read_excel(file_path)
37
38     # Extract LSTM and Layer1 values based on the specified number of rows
39     lstm_values = df['LSTM'].head(row_number).tolist()
40     layer1_values = df['Layer1'].head(row_number).tolist()
41
42     lstm_value = df.loc[row_number - 1, 'LSTM']
43     layer1_value = df.loc[row_number - 1, 'Layer1']
44
45     model_string = f'Models/{lstm_value},{layer1_value}'
46     return model_string
47 def custom_loss(y_true, y_pred):
48     squared_diff_sum = tf.TensorArray(tf.float32, size=0, dynamic_size=True)
49
50     for i in range(len(y_true)):
51         if y_true[i]>y_pred[i]:

```

```

52     squared_diff = (y_true[i] - y_pred[i]) ** 2
53     else:
54         squared_diff = penalty*(y_true[i] - y_pred[i]) ** 2
55         squared_diff_sum = squared_diff_sum.write(i, squared_diff)
56
57     squared_diff_sum = squared_diff_sum.stack()
58     mse = tf.reduce_mean(squared_diff_sum)
59     return mse
60 def plot_predictions1(ax, model, X1, y1, X2, y2, label):
61     predictions1 = model.predict(X1).flatten()
62     df1 = pd.DataFrame(data={'Predictions':predictions1, 'Actuals':y1})
63
64     predictions2 = model.predict(X2).flatten()
65     df2 = pd.DataFrame(data={'Predictions':predictions2, 'Actuals':y2})
66
67     ax.scatter(y2, df2['Predictions'], color="blue", label="Train-Predictions-mse$
68         {:.5f}".format(mse(y2, predictions2)))
69     ax.scatter(y1, df1['Predictions'], color="red", label="Test-Predictions-mse$
70         {:.5f}".format(mse(y1, predictions1)))
71     ax.plot([0,1], [0,1], color="black", label="Actual")
72     ax.legend()
73     ax.set_xlabel('Real (RUL)')
74     ax.set_ylabel('Predicted (RUL)')
75     ax.set_title(label)
76 def histogram(ax1,X1,label,rank):
77     if label=="Train":
78         ax1.hist(X1, color='blue')
79     else:
80         ax1.hist(X1, color='red')
81     ax1.axvline(0, color='black')
82     ax1.set_xlabel('error of RUL')
83     ax1.set_ylabel('frequency')
84     ax1.set_title(rank)
85 def epochplot(train_loss, val_loss, fig, ax, rank):
86     ax.plot(range(1, len(train_loss) + 1), train_loss, color="blue", label='
87         Training Loss')
88     ax.plot(range(1, len(val_loss) + 1), val_loss, color="red", label='Test Loss')
89     ax.set_xlabel('Epoch')
90     ax.set_ylabel('Loss')
91     ax.set_title(rank)
92     ax.legend()
93 def fldcreation(fld):
94     # Check if the directory exists
95     if not os.path.exists(fld):
96         # If it doesn't exist, create it
97         os.makedirs(fld)
98
99 fldcreation(folder)
100 if subplot_option == '1':
101     # Initialize the subplot with a 1x3 configuration
102     numtrials=1
103     fig, axs = plt.subplots(1, 3, figsize=(12, 4))
104     fig1, axs1 = plt.subplots(1, 3, figsize=(12, 4))
105     fig2, axs2 = plt.subplots(1, 3, figsize=(12, 4))
106     fig3, axs3 = plt.subplots(1, 3, figsize=(12, 4))
107 elif subplot_option == '2':
108     # Initialize the subplot with a 2x3 configuration
109     numtrials=6
110     fig, axs = plt.subplots(2, 3, figsize=(12, 8))
111     fig1, axs1 = plt.subplots(1, 3, figsize=(12, 4))
112     fig2, axs2 = plt.subplots(2, 3, figsize=(12, 8))
113     fig3, axs3 = plt.subplots(2, 3, figsize=(12, 8))
114 else:
115     print("Invalid choice. Defaulting to 1x3 subplot configuration.")
116     # Default to 1x3 subplot configuration
117     numtrials=3
118     subplot_option = '1'
119     fig, axs = plt.subplots(1, 3, figsize=(12, 4))
120     fig1, axs1 = plt.subplots(1, 3, figsize=(12, 4))
121     fig2, axs2 = plt.subplots(1, 3, figsize=(12, 4))

```

```

119     fig3, axs3 = plt.subplots(1, 3, figsize=(12, 4))
120
121     x_train = np.load(os.path.join(load_folder, 'x_train.npy'))
122     y_train = np.load(os.path.join(load_folder, 'y_train.npy'))
123     x_test = np.load(os.path.join(load_folder, 'x_test.npy'))
124     y_test = np.load(os.path.join(load_folder, 'y_test.npy'))
125
126
127     for row_number in range(numtrials):
128         model_string = create_model_string(load_folder, row_number+2)
129         test_folder=load_folder+'/'+model_string
130         print(test_folder)
131
132         with custom_object_scope({'custom_loss': custom_loss}):
133             model = load_model(test_folder+'/model/model.h5')
134
135
136         # Continue training the model
137         model.compile(optimizer=Adam(learning_rate=0.0005), loss=custom_loss)
138         history=model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=
139             Nepoch)
140
141         predictions1 = model.predict(x_test).flatten()
142         error1=(y_test-predictions1)
143         predictions2 = model.predict(x_train).flatten()
144         error2=(y_train-predictions2)
145         train_loss = history.history['loss']
146         val_loss = history.history['val_loss']
147
148         rank='Rank' + str(row_number)
149
150         if subplot_option == '1':
151             plot_predictions1(axs[row_number], model, x_test, y_test, x_train,
152                 y_train, rank)
153             histogram(axs1[row_number], error1, 'Test', rank)
154             histogram(axs2[row_number], error2, 'Train', rank)
155             epochplot(train_loss, val_loss, fig3, axs3[row_number], rank)
156         elif subplot_option == '2':
157             plot_predictions1(axs[row_number // 3, row_number % 3], model, x_test,
158                 y_test, x_train, y_train, rank)
159             histogram(axs1[row_number // 3, row_number % 3], axs1[row_number // 3,
160                 row_number % 3], error1, rank)
161             histogram(axs2[row_number // 3, row_number % 3], axs2[row_number // 3,
162                 row_number % 3], error2, rank)
163             epochplot(train_loss, val_loss, fig3, axs3[row_number // 3, row_number
164                 % 3], rank)
165
166         strFLD="model"+str(row_number)+'.h5'
167         model_path = os.path.join(folder, strFLD)
168         model.save(model_path)
169
170
171
172     # Adjust spacing between subplots
173     fig.tight_layout()
174     fig1.tight_layout()
175     fig2.tight_layout()
176     fig3.tight_layout()
177
178     # Save the figures
179     fig.savefig(folder + '/predictions_subplot.png')
180     fig1.savefig(folder + '/Testerror_subplot.png')
181     fig2.savefig(folder + '/Traunerror_subplot.png')
182     fig3.savefig(folder + '/epoch_subplot.png')

```

Listing 5: Κώδικας python βελτιστοποίησης των ΝΔ εκτίμησης του RUL.

```

1 import numpy as np
2 import pandas as pd
3 from sklearn.model_selection import train_test_split

```

```

4 from sklearn.preprocessing import StandardScaler, LabelEncoder
5 import matplotlib.pyplot as plt
6 from LabelCreation import *
7 from keras.models import Sequential
8 from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense
9 from keras.utils import np_utils
10 from sklearn.metrics import confusion_matrix
11 import seaborn as sns
12 import tensorflow as tf
13 from keras.callbacks import ModelCheckpoint
14 import os
15 import joblib
16 from keras.models import load_model
17 from keras.utils import custom_object_scope
18
19
20 penalty2=4
21 def TimeCreation(df,size,in1):
22     timecolumn=np.zeros(size)
23     for i in range(size):
24         if i <= in1:
25             timecolumn[i]=1.0
26         else:
27             timecolumn[i]=1-(i-in1)/(size-in1-1)
28     df["Time"] = timecolumn.tolist()
29 def FaultLabel(df,size,in1,type):
30     classcolumn=np.zeros(size)
31
32
33
34     for i in range(size):
35         if i <= in1:
36             classcolumn[i]=0
37         else:
38             classcolumn[i]=type
39
40     df["Fault"] = classcolumn.tolist()
41 def df_to_X_y2(df, window_size):
42     df_as_np = df
43     X = []
44     for i in range(len(df_as_np)-window_size+1):
45         row = [r for r in df_as_np[i:i+window_size]]
46         X.append(row)
47     return np.array(X)
48 def custom_loss(y_true, y_pred):
49     squared_diff_sum = tf.TensorArray(tf.float32, size=0, dynamic_size=True)
50
51     for i in range(len(y_true)):
52         if y_true[i]>y_pred[i]:
53             squared_diff = (y_true[i] - y_pred[i]) ** 2
54         else:
55             squared_diff = penalty2*(y_true[i] - y_pred[i]) ** 2
56         squared_diff_sum = squared_diff_sum.write(i, squared_diff)
57
58     squared_diff_sum = squared_diff_sum.stack()
59     mse = tf.reduce_mean(squared_diff_sum)
60     return mse
61
62 ft1 = pd.read_csv("featureTableEnergies13WS.csv")
63 size1=len(ft1["RMS"].to_numpy())
64 ft2 = pd.read_csv("featureTableEnergies14WS.csv")
65 size2=len(ft2["RMS"].to_numpy())
66 ft3 = pd.read_csv("featureTableEnergies24WS.csv")
67 size3=len(ft3["RMS"].to_numpy())
68 ft4 = pd.read_csv("featureTableEnergies33WS.csv")
69 size4=len(ft4["RMS"].to_numpy())
70
71 id1=1808
72 id2=1400
73 id3=586

```

```

74 id4=5956
75
76 TimeCreation(ft1,size1,id1)
77 TimeCreation(ft2,size2,id2)
78 TimeCreation(ft3,size3,id3)
79 TimeCreation(ft4,size4,id4)
80 FaultLabel(ft1,size1,id1,1)
81 FaultLabel(ft2,size2,id2,3)
82 FaultLabel(ft3,size3,id3,2)
83 FaultLabel(ft4,size4,id4,2)
84
85 df = pd.concat([ft1, ft2, ft3, ft4])
86
87
88 df.reset_index(inplace=True, drop=True)
89 size=len(df["RMS"].to_numpy())
90 print(df)
91
92
93 #DitECTION
94 iFault=0
95 FLD_Fault="Apotelsmata15\\FaultDitECTION_[4,5,8,9,19,12,16,17,18]"
96 FLD_FaultModel=FLD_Fault+"\BestResults\model"+str(iFault)+".h5"
97 arrayFault=[4, 5, 8, 9, 19, 12, 16, 17, 18]
98
99 scalerf = joblib.load(os.path.join(FLD_Fault, 'scaler.pkl'))
100 modelf = load_model(FLD_FaultModel)
101
102
103 #Fault1
104 i1=2
105 ws1=6
106 FLD_1="Apotelsmata13/1WS_LSTM[8,19,7,4]ws6p4s0.4m2"
107 FLD_1Model=FLD_1+"\BestResults\model"+str(i1)+".h5"
108 array1=[8, 19, 7, 4]
109
110 with custom_object_scope({'custom_loss': custom_loss}):
111     model1 = load_model(FLD_1Model)
112
113 scaler1 = joblib.load(os.path.join(FLD_1, 'scaler.pkl'))
114
115 #Fault2
116 # i2=0
117 # ws2=6
118 # FLD_2="Apotelsmata3/2WS_LSTM[12, 19, 5, 10]ws6p4s1"
119 i2=0
120 ws2=6
121 FLD_2="Apotelsmata13/2WS_LSTM[5,8,10,19,12,16]ws6p4s0.4m2"
122 FLD_2Model=FLD_2+"\BestResults\model"+str(i2)+".h5"
123 array2=[5, 8, 10, 19, 12, 16]
124
125 with custom_object_scope({'custom_loss': custom_loss}):
126     model2 = load_model(FLD_2Model)
127
128 scaler2 = joblib.load(os.path.join(FLD_2, 'scaler.pkl'))
129
130
131 #Fault3
132 #i3=1
133 #ws3=5
134 #FLD_3="Apotelsmata3/3WS_LSTM[19, 12, 9]ws5p4s0.3"
135 i3=1
136 ws3=6
137 FLD_3="Apotelsmata12/3WS_LSTM[19,12,9]ws6p4s0.4m2"
138 FLD_3Model=FLD_3+"\BestResults\model"+str(i3)+".h5"
139 array3=[19, 12, 9]
140
141 with custom_object_scope({'custom_loss': custom_loss}):
142     model3 = load_model(FLD_3Model)
143

```

```

144 scaler3 = joblib.load(os.path.join(FLD_3, 'scaler.pkl'))
145
146
147 ids=[]
148 cn=0
149 for i in range(ws1,size1):
150     ids.append(i)
151     cn=cn+1
152 for i in range(size1+ws3,size1+size2):
153     ids.append(i)
154     cn=cn+1
155 for i in range(size1+size2+ws2,size1+size2+size3):
156     ids.append(i)
157     cn=cn+1
158 for i in range(size1+size2+size3+ws2,size1+size2+size3+size4):
159     ids.append(i)
160     cn=cn+1
161
162 print(cn)
163 print(size1+size2+size3+size4)
164 RULpred=[]
165 yc=[]
166 yR=[]
167 errorh=[]
168 cn2=0
169 errn=0
170 errp=0
171 Nn=0
172 Np=0
173 for j in ids:
174
175     print(str(cn2)+"/"+str(cn))
176     print('-----')
177     Xf=df.iloc[j,arrayFault]
178     yf=df.iloc[j,-1]
179     yRUL=df.iloc[j,-2]
180
181     scaled_rowf = scalerf.transform(Xf.values.reshape(1,-1))
182     predictionsf = np.argmax(modelf.predict(scaled_rowf, verbose=0), axis=1)
183     pr=predictionsf[0]
184
185     if pr==0:
186         predictions=1.0
187     elif pr==1:
188         X=(df.iloc[j-ws1:j,array1])
189         scaledX=scaler1.transform(X)
190         scaled_row=df_to_X_y2(scaledX, ws1)
191         predictions=model1.predict(scaled_row)
192     elif pr==2:
193         X=(df.iloc[j-ws2:j,array2])
194         scaledX=scaler2.transform(X)
195         scaled_row=df_to_X_y2(scaledX, ws2)
196         predictions=model2.predict(scaled_row)
197     elif pr==3:
198         X=(df.iloc[j-ws3:j,array3])
199         scaledX=scaler3.transform(X)
200         scaled_row=df_to_X_y2(scaledX, ws3)
201         predictions=model3.predict(scaled_row)
202
203     RULpred.append(predictions)
204     yc.append(pr)
205     yR.append(yRUL)
206     cn2=cn2+1
207
208     err=yRUL-predictions
209     if err!=0.0:
210         err=err.item()
211     # if err>0:
212     #     Np=Np+1
213     #     errp=errp+err

```

```

214     # else if err<0:
215     #     Nn=Nn+1
216     #     errn=errn+err
217     if yf!=0:
218         errorh.append(err)
219
220 # File path to save the array
221 file_path = 'array_data.txt'
222
223 # Save the array to a text file
224 np.savetxt(file_path, errorh, fmt='%.5f')
225 # print(Np)
226 # print(errp/Np)
227
228 # print(Nn)
229 # print(errn/Nn)
230 print(errorh)
231 # yc=np.array(yc)
232 # Flattening the errorh list
233 plt.hist(errorh, bins=10)
234 plt.xlabel('Error')
235 plt.ylabel('Frequency')
236 plt.title('Histogram of Errors')
237 plt.show()
238 # colors = ['red', 'green', 'blue', 'yellow']
239 # # labels = ['Safe', 'Inner', 'Outer', 'Roller']
240 # # yc_list = yc.tolist()
241
242 # colorsarray = []
243 # sizesarray = []
244 # for i in range(len(ids)):
245 #     if yc[i] == 0:
246 #         colorsarray.append(colors[0])
247 #         sizesarray.append(20)
248 #     elif yc[i] == 1:
249 #         colorsarray.append(colors[1])
250 #         sizesarray.append(50)
251 #     elif yc[i] == 2:
252 #         colorsarray.append(colors[2])
253 #         sizesarray.append(50)
254 #     else:
255 #         colorsarray.append(colors[3])
256 #         sizesarray.append(50)
257
258 # # Add 6 to the first and last elements of 'ids'
259 # ids_modified = [value for value in range(6, len(ids) + 6)]
260
261 # print(sizesarray)
262
263 # scatter = plt.scatter(ids_modified, RULpred, c=colorsarray, s=sizesarray)
264 # plt.plot(ids_modified, yR, color='black', linewidth=2)
265
266 # # Set x and y labels bigger
267 # plt.xlabel('ids', fontsize=30)
268 # plt.ylabel('RUL prediction', fontsize=30)
269
270 # # Create a legend with bigger font size
271 # legend_elements = [
272 #     plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='red', markersize
273 #               =8, label='Prediction: Safe'),
274 #     plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='green',
275 #               markersize=8, label='Prediction: Inner Race'),
276 #     plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='blue',
277 #               markersize=8, label='Prediction: Outer Race'),
278 #     plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='yellow',
279 #               markersize=8, label='Prediction: Roller Elements'),
280 #     plt.Line2D([0], [0], color='black', linewidth=2, label='Real RUL')
281 # ]
282 # plt.xticks(fontsize=20) # Adjust the fontsize as needed for the x-axis
283 # plt.yticks(fontsize=20)

```



```

280
281 # plt.legend(handles=legend_elements, fontsize=20)
282
283 # # Create a grid
284 # plt.grid(True)
285 # plt.show()

```

Listing 6: Κώδικας python επανεκπαίδευσης των ΝΔ εκτίμησης του RUL.

```

1 import numpy as np
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler, LabelEncoder
5 import matplotlib.pyplot as plt
6 from LabelCreation import *
7 from keras.models import Sequential
8 from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense
9 from keras.utils import np_utils
10 from sklearn.metrics import confusion_matrix
11 import seaborn as sns
12 import tensorflow as tf
13 from keras.callbacks import ModelCheckpoint
14 import os
15 import joblib
16 from keras.models import load_model
17 from keras.utils import custom_object_scope
18
19
20 penalty2=4
21 def TimeCreation(df,size,in1):
22     timecolumn=np.zeros(size)
23     for i in range(size):
24         if i <= in1:
25             timecolumn[i]=1.0
26         else:
27             timecolumn[i]=1-(i-in1)/(size-in1-1)
28     df["Time"] = timecolumn.tolist()
29 def FaultLabel(df,size,in1,type):
30     classcolumn=np.zeros(size)
31
32
33
34     for i in range(size):
35         if i <= in1:
36             classcolumn[i]=0
37         else:
38             classcolumn[i]=type
39
40     df["Fault"] = classcolumn.tolist()
41 def df_to_X_y2(df, window_size):
42     df_as_np = df
43     X = []
44     for i in range(len(df_as_np)-window_size+1):
45         row = [r for r in df_as_np[i:i+window_size]]
46         X.append(row)
47     return np.array(X)
48 def custom_loss(y_true, y_pred):
49     squared_diff_sum = tf.TensorArray(tf.float32, size=0, dynamic_size=True)
50
51     for i in range(len(y_true)):
52         if y_true[i]>y_pred[i]:
53             squared_diff = (y_true[i] - y_pred[i]) ** 2
54         else:
55             squared_diff = penalty2*(y_true[i] - y_pred[i]) ** 2
56         squared_diff_sum = squared_diff_sum.write(i, squared_diff)
57
58     squared_diff_sum = squared_diff_sum.stack()
59     mse = tf.reduce_mean(squared_diff_sum)
60     return mse
61

```

```

62 ft1 = pd.read_csv("featureTableEnergies13WS.csv")
63 size1=len(ft1["RMS"].to_numpy())
64 ft2 = pd.read_csv("featureTableEnergies14WS.csv")
65 size2=len(ft2["RMS"].to_numpy())
66 ft3 = pd.read_csv("featureTableEnergies24WS.csv")
67 size3=len(ft3["RMS"].to_numpy())
68 ft4 = pd.read_csv("featureTableEnergies33WS.csv")
69 size4=len(ft4["RMS"].to_numpy())
70
71 id1=1808
72 id2=1400
73 id3=586
74 id4=5956
75
76 TimeCreation(ft1,size1,id1)
77 TimeCreation(ft2,size2,id2)
78 TimeCreation(ft3,size3,id3)
79 TimeCreation(ft4,size4,id4)
80 FaultLabel(ft1,size1,id1,1)
81 FaultLabel(ft2,size2,id2,3)
82 FaultLabel(ft3,size3,id3,2)
83 FaultLabel(ft4,size4,id4,2)
84
85 df = pd.concat([ft1, ft2, ft3, ft4])
86
87
88 df.reset_index(inplace=True, drop=True)
89 size=len(df["RMS"].to_numpy())
90 print(df)
91
92
93 #DitECTION
94 iFault=0
95 FLD_Fault="Apotelsmata15\\FaultDitECTION_[4,5,8,9,19,12,16,17,18]"
96 FLD_FaultModel=FLD_Fault+"\BestResults\model"+str(iFault)+".h5"
97 arrayFault=[4, 5, 8, 9, 19, 12, 16, 17, 18]
98
99 scalerf = joblib.load(os.path.join(FLD_Fault, 'scaler.pkl'))
100 modelf = load_model(FLD_FaultModel)
101
102
103 #Fault1
104 i1=2
105 ws1=6
106 FLD_1="Apotelsmata13/1WS_LSTM[8,19,7,4]ws6p4s0.4m2"
107 FLD_1Model=FLD_1+"\BestResults\model"+str(i1)+".h5"
108 array1=[8, 19, 7, 4]
109
110 with custom_object_scope({'custom_loss': custom_loss}):
111     model1 = load_model(FLD_1Model)
112
113 scaler1 = joblib.load(os.path.join(FLD_1, 'scaler.pkl'))
114
115 #Fault2
116 # i2=0
117 # ws2=6
118 # FLD_2="Apotelsmata3/2WS_LSTM[12, 19, 5, 10]ws6p4s1"
119 i2=0
120 ws2=6
121 FLD_2="Apotelsmata13/2WS_LSTM[5,8,10,19,12,16]ws6p4s0.4m2"
122 FLD_2Model=FLD_2+"\BestResults\model"+str(i2)+".h5"
123 array2=[5, 8, 10, 19, 12, 16]
124
125 with custom_object_scope({'custom_loss': custom_loss}):
126     model2 = load_model(FLD_2Model)
127
128 scaler2 = joblib.load(os.path.join(FLD_2, 'scaler.pkl'))
129
130
131 #Fault3

```

```

132 #i3=1
133 #ws3=5
134 #FLD_3="Apotelsmata3/3WS_LSTM[19, 12, 9]ws5p4s0.3"
135 i3=1
136 ws3=6
137 FLD_3="Apotelsmata12/3WS_LSTM[19, 12, 9]ws6p4s0.4m2"
138 FLD_3Model=FLD_3+"\BestResults\model"+str(i3)+".h5"
139 array3=[19, 12, 9]
140
141 with custom_object_scope({'custom_loss': custom_loss}):
142     model3 = load_model(FLD_3Model)
143
144 scaler3 = joblib.load(os.path.join(FLD_3, 'scaler.pkl'))
145
146
147 ids=[]
148 cn=0
149 for i in range(ws1,size1):
150     ids.append(i)
151     cn=cn+1
152 for i in range(size1+ws3,size1+size2):
153     ids.append(i)
154     cn=cn+1
155 for i in range(size1+size2+ws2,size1+size2+size3):
156     ids.append(i)
157     cn=cn+1
158 for i in range(size1+size2+size3+ws2,size1+size2+size3+size4):
159     ids.append(i)
160     cn=cn+1
161
162 print(cn)
163 print(size1+size2+size3+size4)
164 RULpred=[]
165 yc=[]
166 yR=[]
167 errorh=[]
168 cn2=0
169 errn=0
170 errp=0
171 Nn=0
172 Np=0
173 for j in ids:
174
175     print(str(cn2)+"-"+str(cn))
176     print('-----')
177     Xf=df.iloc[j,arrayFault]
178     yf=df.iloc[j, -1]
179     yRUL=df.iloc[j, -2]
180
181     scaled_rowf = scalerf.transform(Xf.values.reshape(1, -1))
182     predictionsf = np.argmax(modelf.predict(scaled_rowf, verbose=0), axis=1)
183     pr=predictionsf[0]
184
185     if pr==0:
186         predictions=1.0
187     elif pr==1:
188         X=(df.iloc[j-ws1:j,array1])
189         scaledX=scaler1.transform(X)
190         scaled_row=df_to_X_y2(scaledX, ws1)
191         predictions=model1.predict(scaled_row)
192     elif pr==2:
193         X=(df.iloc[j-ws2:j,array2])
194         scaledX=scaler2.transform(X)
195         scaled_row=df_to_X_y2(scaledX, ws2)
196         predictions=model2.predict(scaled_row)
197     elif pr==3:
198         X=(df.iloc[j-ws3:j,array3])
199         scaledX=scaler3.transform(X)
200         scaled_row=df_to_X_y2(scaledX, ws3)
201         predictions=model3.predict(scaled_row)

```

```

202
203     RULpred.append(predictions)
204     yc.append(pr)
205     yR.append(yRUL)
206     cn2=cn2+1
207
208     err=yRUL-predictions
209     if err!=0.0:
210         err=err.item()
211         # if err>0:
212         #     Np=Np+1
213         #     errp=errp+err
214         # else if err<0:
215         #     Nn=Nn+1
216         #     errn=errn+err
217     if yf!=0:
218         errorh.append(err)
219
220 # File path to save the array
221 file_path = 'array_data.txt'
222
223 # Save the array to a text file
224 np.savetxt(file_path, errorh, fmt='%.5f')
225 # print(Np)
226 # print(errp/Np)
227
228 # print(Nn)
229 # print(errn/Nn)
230 print(errorh)
231 # yc=np.array(yc)
232 # Flattening the errorh list
233 plt.hist(errorh, bins=10)
234 plt.xlabel('Error')
235 plt.ylabel('Frequency')
236 plt.title('Histogram of Errors')
237 plt.show()
238 # colors = ['red', 'green', 'blue', 'yellow']
239 # # labels = ['Safe', 'Inner', 'Outer', 'Roller']
240 # # yc_list = yc.tolist()
241
242 # colorsarray = []
243 # sizesarray = []
244 # for i in range(len(ids)):
245 #     if yc[i] == 0:
246 #         colorsarray.append(colors[0])
247 #         sizesarray.append(20)
248 #     elif yc[i] == 1:
249 #         colorsarray.append(colors[1])
250 #         sizesarray.append(50)
251 #     elif yc[i] == 2:
252 #         colorsarray.append(colors[2])
253 #         sizesarray.append(50)
254 #     else:
255 #         colorsarray.append(colors[3])
256 #         sizesarray.append(50)
257
258 # # Add 6 to the first and last elements of 'ids'
259 # ids_modified = [value for value in range(6, len(ids) + 6)]
260
261 # print(sizesarray)
262
263 # scatter = plt.scatter(ids_modified, RULpred, c=colorsarray, s=sizesarray)
264 # plt.plot(ids_modified, yR, color='black', linewidth=2)
265
266 # # Set x and y labels bigger
267 # plt.xlabel('ids', fontsize=30)
268 # plt.ylabel('RUL prediction', fontsize=30)
269
270 # # Create a legend with bigger font size
271 # legend_elements = [

```

```

272 # plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='red', markersize
    =8, label='Prediction: Safe'),
273 # plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='green',
    markersize=8, label='Prediction: Inner Race'),
274 # plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='blue',
    markersize=8, label='Prediction: Outer Race'),
275 # plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='yellow',
    markersize=8, label='Prediction: Roller Elements'),
276 # plt.Line2D([0], [0], color='black', linewidth=2, label='Real RUL')
277 # ]
278 # plt.xticks(fontsize=20) # Adjust the fontsize as needed for the x-axis
279 # plt.yticks(fontsize=20)
280
281 # plt.legend(handles=legend_elements, fontsize=20)
282
283 # # Create a grid
284 # plt.grid(True)
285 # plt.show()

```

Listing 7: Κώδικας python τελικής αξιολόγησης των μετρήσεων.