ΠΑΝΕΠΙΣΤΉΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
“ΠΡΟΒΛΕΨΗ ΚΑΤΆΣΤΑΣΗΣ ROBOT ΜΕ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ”

ΣΥΓΓΡΑΦΕΑΣ:
ΚΟΜΗΤΗΣ ΧΡΥΣΟΒΑΛΑΝΤΗΣ
ΑΜ: **71446444**

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΝΙΚΟΛΆΟΥ ΓΡΗΓΟΡΙΟΣ

UNIVERSITY OF WEST ATTICA

SCHOOL OF ENGINEERING

DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION

DIPLOMA THESIS
"ROBOT'S STATE PREDICTION WITH MACHINE LEARNING"

AUTHOR:
KOMITIS CRYSOVALANTIS
REGISTRATION NUMBER: **71446444**

SUPERVISOR:
NIKOLAOU GRIGORIOS

Εξεταστική Επιτροπή

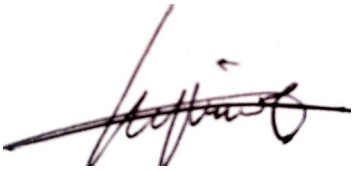
Νικολάου Γρηγόριος	Βασιλειάδου Σουλτάνα	Δρόσος Χρήστος
GRIGORIOS NIKOLAOU Digitally signed by GRIGORIOS NIKOLAOU Date: 2023.11.14 13:29:40 +02'00'	Soultana Vasileiadou Digitally signed by Soultana Vasileiadou Date: 2023.11.22 11:20:14 +02'00'	Christos Drosos Digitally signed by Christos Drosos Date: 2023.11.21 12:08:22 +02'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Κομήτης Χρυσοβαλάντης του Μόσχου, με αριθμό μητρώου 7146444 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο δηλών



Περιεχόμενα

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Θεό για την ψυχική δύναμη που μου έδωσε και τον πατέρα μου Κομήτη Μόσχο, την Μητέρα μου Πιτσιλή Μαρία καθώς και τον Αδερφό μου με την γυναίκα μου Παύλο και Στέλλα για την στήριξη, την εμπιστοσύνη και την υπομονή που δείξανε αυτά τα χρόνια για εμένα. Επιπλέον θα ήθελα να ευχαριστήσω τον καθηγητή μου Γρηγόριο Νικολάου για τις γνώσεις που μου έδωσε, την καθοδήγηση αλλά και την πίστη του σε εμένα και τον Aniketh Ramesh για την παροχή των δεδομένων και την συνεργασία του ώστε να ολοκληρωθεί το πείραμα. Τέλος την διπλωματική μου εργασία την αφιερώνω στην ανιψιά μου Μαρία Κομήτη και της εύχομαι να διαπρέψει και πετύχει όλους τους στόχους της.

Κομήτης Χρυσοβαλάντης

Περίληψη

Τα τελευταία χρόνια υπάρχει αύξηση της ζήτηση στον τομέα της τεχνητής νοημοσύνης και την μηχανικής μάθησης και αυτό διότι πλέον υπάρχουν μεγάλοι όγκοι δεδομένων που μπορούν να χρησιμοποιηθούν για την δημιουργία μοντέλων τεχνητής νοημοσύνης αλλά και μια πληθώρα εφαρμογών στις οποίες οι ερευνητές και επαγγελματίες του χώρου μπορούν να τα αξιοποιήσουν. Ένας από αυτούς του τομείς είναι η ανίχνευση ανωμαλιών, που βρίσκει εφαρμογή σε τομείς όπως είναι η ιατρική, η βιομηχανία και την ρομποτική, που ανιχνεύει ανωμαλίες σχετικά με την φυσιολογική λειτουργία των οργάνων, αν πρόκειται για έμβιο οργανισμό, ή εξαρτημάτων

Στην παρούσα διπλωματική θα γίνει έρευνα σχετικά με την ανίχνευση καινοτομιών στα δεδομένα του αισθητήρα ανίχνευσης θορύβου laser του ρομπότ Clearpath Husky. Στην εργασία αναφέρονται το τι είναι η μηχανική μάθηση, τι είναι η ανίχνευση ανωμαλιών και ποια η διαφορά με την ανίχνευση καινοτομιών ενώ θα γίνει αναφορά στο τι είναι τα Robot Vitals και Robot Health πάνω στα οποία γίνεται η ανίχνευση καινοτομιών για το πείραμα. Για το πειραματικό κομμάτι δημιουργήθηκαν τρία μοντέλα με σκοπό να συγκρίνουμε πιο είναι καταλληλότερο στην ανίχνευση των ακραίων τιμών στο dataset που έχουμε για το laser noise.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Ανίχνευση Ανωμαλίων, Ανίχνευση Ακραίων Τιμών, Ανίχνευση Καινοτομίας, Isolation Forest, One Class SVM, Local Outlier Factor, Robot Vitals, Robot Health.

Abstract

In recent years there has been an increase in demand in the field of artificial intelligence and machine learning and this is because there are now large volumes of data that can be used to create artificial intelligence models and a plethora of applications in which researchers and professionals in the field can use them. One of these areas is anomaly detection, which finds application in areas such as medicine, industry and robotics, which detects anomalies related to the physiological function of organs, if it is a living organism, or components

In this thesis, research will be done on detecting innovations in the data of the laser noise detection sensor of Clearpath Husky robot. The thesis will mention what machine learning is, what is anomaly detection and what is the difference with novelty detection and will mention what are Robot VItals and Robot Health on which novelty detection is done for the experiment. For the experimental part three models were created in order to compare which one is more suitable in detecting the outliers in the dataset we have for laser noise.

Key words

Machine Learning, Anomaly Detection, Extreme Value Detection, Innovation Detection, Isolation Forest, One Class SVM, Local Outlier Factor, Robot Vitals, Robot Health.

Κεφάλαιο 1

1. Μηχανική μάθηση

Η μάθηση όσον αφορά τον άνθρωπο είναι μία εμπειρική διαδικασία, όπου μέσο της επαγωγής προσπαθεί να κατανοήσει το περιβάλλον γύρω του. Ο άνθρωπος έχει τη δυνατότητα αυθόρμητα να οργανώνει και να συσχετίζει τις εμπειρίες και τις παρατηρήσεις του δημιουργώντας δομές που ονομάζονται πρότυπα, για παράδειγμα είναι σε θέση να διαβάσει ένα κείμενο ακόμα και αν ο βλέπει τον γραφικός χαρακτήρα του συγγραφέα για πρώτη φορά.

Η μηχανική μάθηση βασίστηκε πάνω στην ανθρώπινη μάθηση και με τον ορισμό κατά Mitchell (1997) μπορούμε να πούμε πως: "Ένα πρόγραμμα Η/Υ θεωρείται ότι μαθαίνει μέσω εμπειρίας Ε που αποκτά κάνοντας δραστηριότητες Τ και σε συνδυασμό με κάποια μετρική απόδοσης Ρ, αν οι επιδόσεις του στις δραστηριότητες Τ, όπως καταμετρώνται από την Π, βελτιώνονται με την εμπειρία Ε". Για να υπάρξει λοιπόν ένα σύστημα το οποίο μπορεί να μάθει πρέπει να υπάρχουν τρία βασικά συστατικά:

- Δεδομένα από το περιβάλλον της εργασίας που καλείτε να μάθει το σύστημα καθώς και το αποτέλεσμα των δεδομένων.
- Ένα κριτήριο που θα αξιολογεί την επίδοση του συστήματος.
- Μια συγκεκριμένη διαδικασία όπου το σύστημα θα καλείται να εκτελέσει.

Περιπτώσεις εργασιών που μπορεί ένα τέτοιο σύστημα να βελτιωθεί μέσο της μάθησης είναι:

- Η αναγνώριση προτύπων ή αντικειμένων, όπως για παράδειγμα ο γραφικός χαρακτήρας, εμβλήματα, έμβιοι οργανισμοί κλπ.
- Πρόβλεψη τιμών μετρήσιμων ποσοτήτων, όπως για παράδειγμα η τιμής μια μετοχής στο χρηματιστήριο, η θερμοκρασία του περιβάλλοντος.
- Ομαδοποίηση, μπορεί να ομαδοποιεί παρόμοια αντικείμενα ή και συμπεριφορές.

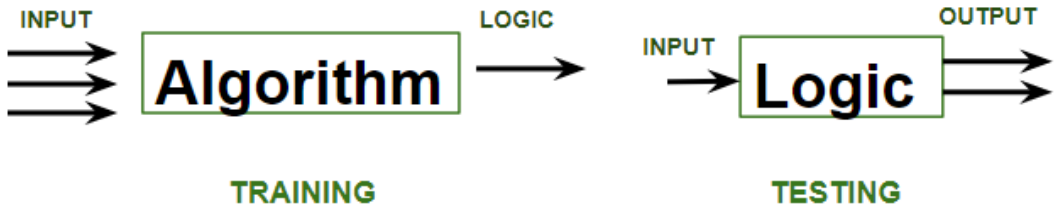
Οι τεχνικές μηχανικής μάθησης χωρίζονται σε τρεις κατηγορίες, τη μάθηση με επίβλεψη, τη μάθηση χωρίς επίβλεψη και την ενισχυτική μάθηση.

1.1 Τρόπος λειτουργίας

Για τη δημιουργία ενός μοντέλου μηχανικής μάθησης χρειάζεται να υπάρχουν ποιοτικά δεδομένα, χωρίς πολλές κενές τιμές και με αληθείς τιμές, στα οποία έχει γίνει η κατάλληλη επεξεργασία ώστε να γίνει η τροφοδοσία προς το μοντέλο, η ποιότητα των δεδομένων έχουν άμεση συσχέτιση με την ακρίβεια του μοντέλου καθώς ένα μοντέλο με αρκετές κενές τιμές δε θα έχει μεγάλη ακρίβεια ενώ ένα μοντέλο με δεδομένα τα οποία δε είναι αληθή δεν θα δίνει αποτελέσματα που θα ανταποκρίνονται στην πραγματικότητα. Για αυτό είναι σημαντική η σωστή ανάλυση,

επεξεργασία και εξαγωγή χαρακτηριστικών. Μέσα από την ανάλυση βγαίνουν συμπεράσματα όπως ο τύπος των δεδομένων, η κατανομή, τα χαρακτηριστικά, η συσχέτιση των δεδομένων κ.α. Κατά την επεξεργασία γίνεται ομαλοποίηση των δεδομένων, γέμισμα η αφαίρεση γραμμών με κενές τιμές, αφαίρεση θορύβου κ.α. Κατά την εξαγωγή χαρακτηριστικών δημιουργούνται στήλες με πληροφορίες που υπάρχουν μέσα από τα δεδομένα, για παράδειγμα αν υπάρχουν οι ημερομηνίες γέννησης ενός δείγματος ανθρώπων τότε μπορεί να βγει μια στήλη που τους κατηγοριοποιεί σε γενιές. Τα δεδομένα και ο ορισμός του προβλήματος θα μπορούν να δώσουν την απάντηση στον τύπο μάθησης που θα πρέπει να επιλεχθεί.

Εφόσον γίνει η προετοιμασία τα δεδομένα μοιράζονται σε δεδομένα εκπαίδευσης και δεδομένα εξάσκησης. Τα δεδομένα εκπαίδευσης εισάγονται στο μοντέλο που έχει επιλεχθεί, ανάλογα το είδος του προβλήματος τα δεδομένα μπορεί να έχουν ανακατευτεί τυχαία ή και να έχουν χωριστεί σε πακέτα δεδομένων. Μετά το πέρας της εκπαίδευσης ο αλγόριθμος χρησιμοποιεί τα δεδομένα εξάσκησης, τα οποία είναι δεδομένα που δεν έχει συναντήσει προηγουμένως, για να κάνει αξιολόγηση του μοντέλου. Υπάρχουν διάφοροι μετρητές της απόδοσης ενός μοντέλου μηχανικής μάθησης ανάλογα με ο πρόβλημα και τον αλγόριθμο. Για παράδειγμα, διαφορετικά μετράμε ένα μοντέλο ταξινόμησης με διακριτές κατηγορίες τιμών και διαφορετικά ένα μοντέλο παρεμβολής που προβλέπει μια συνεχής τιμή.

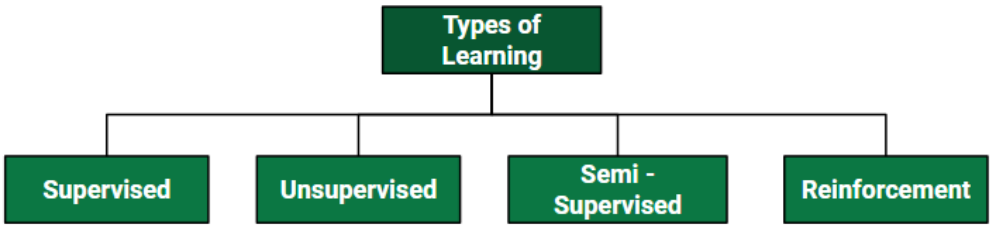


Εικόνα 1: Λειτουργία εκπαίδευσης μοντέλου μηχανικής μάθησης

Τέλος, αφού έχει γίνει η αξιολόγηση γίνεται η παραμετροποίηση του εκάστοτε αλγορίθμου, παραμετροποίηση μπορεί να γίνει και στην αρχή της εκπαίδευσης, αλλά μετά την αξιολόγηση υπάρχει πιο ξεκάθαρη εικόνα για την συμπεριφορά του μοντέλου.

1.2 Τύποι μηχανικής μάθησης

Οι τεχνικές μηχανικής μάθησης χωρίζονται σε τέσσερις κατηγορίες, τη μάθηση με επίβλεψη, τη μάθηση χωρίς επίβλεψη, την ημι-επιβλεπόμενη μάθηση και την ενισχυτική μάθηση.



Εικόνα 2: Τύποι μηχανικής μάθησης

1.2.1 Μάθηση με Επίβλεψη

Στη μάθηση με επίβλεψη το μοντέλο μαθαίνει μέσω επαγωγής τη συνάρτηση στόχου. Η συνάρτηση αυτή χρησιμοποιεί τα δεδομένα που έχουν οριστεί ως χαρακτηριστικά X (μεταβλητές εισόδου) για να προβλέψει την τιμή εξόδου Y. Τα χαρακτηριστικά χωρίζονται σε 2 κατηγορίες, τα δεδομένα εκπαίδευσης D, που χρησιμοποιούνται για την εκπαίδευση του μοντέλου, και τα δεδομένα δοκιμής, όπου το μοντέλο προβλέπει την τιμή εξόδου Y και στη συνέχεια συγκρίνει με την πραγματική τιμή.

Κατά τη διάρκεια της εκπαίδευσης γίνονται στο μοντέλο γνωστά τα X καθώς και τα Y. Τα Y είναι παράγωγα των X μέσα από μια άγνωστη συνάρτηση Y= f(X) όπου το μοντέλο καλείτε να αναπαραστήσει δημιουργώντας τη συνάρτηση στόχο που προσπαθεί να την προσεγγίσει εξετάζοντας κάποιες συναρτήσεις, Οι συναρτήσεις αυτές ονομάζονται υποθέσεις h και μπορούν να είναι είτε διακριτές h: X -> (c1,c2,c3...cn), είτε πραγματικές h: X -> R ανάλογα με τον τύπο του προβλήματος. Ο τύπος του προβλήματος μπορεί να γίνει αντιληπτός από την έξοδο, για παράδειγμα, έστω ότι έχουμε ένα ως είσοδο (χαρακτηριστικά) τρισδιάστατο διάνυσμα X με αριθμούς οι οποίοι αναπαριστούν το βάρος, την ηλικία και το ύψος και ως έξοδο Y τον δείκτη μάζα σώματος. Σε αυτή την περίπτωση το Y είναι μια συνεχής τιμή, όποτε οι συναρτήσεις h θα είναι πραγματικές. Αυτός ο τύπος προβλήματος ονομάζεται παρεμβολή. Στην περίπτωση που η έξοδος Y είναι το βιολογικό φύλο τότε έχουμε μια διακριτή τιμή όπου η h είναι διακριτή. Αυτός ο τύπος προβλήματος ονομάζεται ταξινόμηση.

1.2.2 Μη επιβλεπόμενη μάθηση

Η μη επιβλεπόμενη μάθηση είναι μια τεχνική μηχανικής μάθησης, όπου η ανάλυση των δεδομένων γίνεται χωρίς αυτά να έχουν ετικέτα, σε αντίθεση με την επιβλεπόμενη μάθηση. Στόχος του συστήματος είναι ανεύρεση συσχετίσεων και συσταδοποιήσεων βασιζόμενο μόνο στις ιδιότητες τους χωρίς να υπάρχει κάποια ετικέτα και χωρίς την παρέμβαση του ανθρώπου. Σαν αποτέλεσμα προκύπτουν πρότυπα πληροφόρησης, από τα οποία περιγράφονται τα μέρη από τα δεδομένα. Τέτοια πρότυπα είναι η συσταδοποίηση, μείωση διαστάσεων, οι κανόνες συσχέτισης και η ανίχνευση ανωμαλιών.

1.2.3 Ημι-επιβλεπόμενη μάθηση

Η ημι-επιβλεπόμενη μάθηση αποτελεί μια ιδική κατηγορία ταξινόμησης. Στην επιβλεπόμενη μάθηση χρησιμοποιούνται δεδομένα με ετικέτα για την εκπαίδευση του μοντέλου. Αυτού του είδους τα δεδομένα δεν είναι πάντα διαθέσιμο και απαιτούνται πόροι για να δημιουργηθούν. Στην ημι-επιβλεπόμενη μάθηση μπορεί γίνεται χρήση δεδομένων στα οποία κάποια από αυτά αποτελούνται από ζεύγη εισόδου-εξόδου, όπου η έξοδος είναι η ετικέτα αλλά και από δεδομένα που είναι μόνο είσοδοι και δεν έχουν την ετικέτα. Η ημι-επιβλεπόμενη μάθηση στοχεύει σε βελτιωμένη απόδοση και ικανότητα γενίκευσης του μοντέλου αξιοποιώντας τα δεδομένα που έχουν ετικέτα μαζί με τα μεγαλύτερου όγκου χωρίς ετικέτα δεδομένα. Η ιδέα πίσω από αυτή την προσέγγιση είναι πως τα δεδομένα χωρίς ετικέτα παρέχουν χρήσιμες πληροφορίες σχετικά με την υποκείμενη δομή της κατανομής τους, βοηθώντας έτσι το μοντέλο να μαθαίνει πιο αποτελεσματικά. Η ημι-επιβλεπόμενη μάθηση βρίσκει εφαρμογές σε διάφορους τομείς, όπως η αναγνώριση εικόνων, η επεξεργασία φυσικής γλώσσας και άλλα.

1.2.4 Ενισχυτική μάθηση

Η ενισχυτική μάθηση αφορά ένα σύνολο τεχνικών στις οποίες το σύστημα μαθαίνει μέσα από την αλληλεπίδραση που έχει με το περιβάλλον του. Βασίζεται πάνω μοντέλα επιβράβευσης και τιμωρίας που έχουν εφαρμοστεί σε έμβιους οργανισμού και σκοπός του συστήματος είναι η μεγιστοποίηση της συνάρτηση του αριθμητικού σήματος ενίσχυσης (συνάρτησης ανταμοιβής). Το σύστημα δε λαμβάνει εντολές όσον αφορά τις κινήσεις και τις ενέργειες που πρέπει να κάνει από τον χειριστεί ή κάποιο αλγόριθμο που δίνει συγκεκριμένες εντολές ανάλογα τις καταστάσεις που έχουν οριστεί αλλά ανακαλύπτει μόνο του τις ενέργειες που θα του επιφέρουν μεγαλύτερο κέρδος. Η διαφορά που έχει με τη μάθηση με επίβλεψη είναι πως το σύστημα μαθαίνει από τη δική του εμπειρία μέσω της δοκιμής και αποτυχίας καθώς δεν υπάρχει άμεσα κάποιος κανόνας επιθυμητής συμπεριφοράς.

1.3 Μέθοδοι συλλογικής μάθησης

Η μέθοδος συλλογικής μάθηση (ensemble learning) αποτελεί μία τεχνική συνδυασμού πολλαπλών μοντέλων με στόχο τη βελτιστοποίηση της απόδοσης του αλγορίθμου. Η μέθοδος αυτή σχεδόν πάντα βελτιώνει την απόδοση του αλγορίθμου όμως είναι αρκετά πολύπλοκος και δύσκολο να αναλυθούν οι παράγοντες που βοηθούν στην τελική απόφαση από τα συνδυασμένα μοντέλα. Οι μέθοδοι συλλογικής μάθησης αναλύονται παρακάτω.

1.3.1 Bagging

Η μέθοδος bagging (bootstrap aggregating) έχει εφαρμογή σε μοντέλα που εμφανίζουν μεγάλη διακύμανση στις προβλέψεις τους. Είναι δηλαδή ασταθής μοντέλα που η παραμικρή αλλαγή στα δεδομένα μπορούν να εμφανίσουν διαφορετικές αποφάσεις για κάποιες περιπτώσεις. Η λειτουργία της μεθόδους αυτής έχει ως εξής:

- Δημιουργούνται πολλά τυχαία υποσύνολα δεδομένων μέσα από τα αρχικό σύνολο με τη μέθοδο bootstraping. Η μέθοδος bootstraping δημιουργεί μέσα από ένα σύνολο δεδομένο μεγέθους N διαφορετικά σύνολα με πλήθος n και μεγέθους N μέσω της δειγματοληψίας με αντικατάσταση, όπου ένα σημείο που έχει επιλεχθεί τυχαία επανατοποθετείται στο αρχικό σύνολο, έτσι είναι δυνατόν κάποια σημεία να επιλεχθούν πάν από μια φορές ενώ κάποια άλλα καθόλου.
- Στη συνέχεια εφαρμόζεται ο αλγόριθμος σε όλα τα νέα σύνολα δεδομένων από όπου και παράγονται αντίστοιχα μοντέλα πρόβλεψης.
- Για την πρόβλεψη λαμβάνονται υπόψη οι αποφάσεις όλων των μοντέλων. Για την ταξινόμηση λαμβάνεται ως απόφαση η πλειοψηφία από τις εξόδους των μοντέλων ενώ για την παρεμβολή ο μέσος όρος των αριθμητικών προβλέψεων αποτελούν και την έξοδο.

1.3.2 Boosting

Η μέθοδος boosting αποτελεί μια επαναληπτική διαδικασία που στηρίζεται στην ανάθεση βαρών πάνω στα δεδομένων εκπαίδευσης. Με αυτόν τον τρόπο ο αλγόριθμος δίνει βάση στα δεδομένα που συνήθως ταξινομούνται λάθος και αποτελούν τις δύσκολες περιπτώσεις. Αν ο αλγόριθμος είναι ικανός να χειριστεί βάρη τότε δεν υπάρχει κάποιο πρόβλημα, σε αντίθετη περίπτωση όμως τα βάρη χρησιμοποιούνται για να καθοριστεί η κατανομή της δειγματοληψίας. Δημιουργείται ένα νέο σύνολο δεδομένων που εμφανίζεται με βάση τα βάρη, οπότε τα δεδομένα με μεγαλύτερο βάρος εμφανίζονται πιο πολλές φορές. Το κάθε μοντέλο που δημιουργείται βασίζεται στο προηγούμενο μοντέλο. Η διαδικασία έχει ως εξής:

- Αρχικοποίηση των βαρών, όπου δίνεται η τιμή 1/N στα N δεδομένα του συνόλου D.
- Δημιουργία συνόλου εκπαίδευσης, όπου δημιουργείται υποσύνολο εκπαίδευσης Di μέσα από το σύνολο D.
- Εκπαίδευση - Αξιολόγηση του μοντέλου, όπου παράγεται ένα μοντέλο ταξινομητή Ci και χρησιμοποιείται για να ταξινομηθούν τα δεδομένα από το αρχικό σύνολο D.
- Αναπροσαρμογή βαρών, όπου αυξάνεται το βάρος των δεδομένων που έχει γίνει λάθος πρόβλεψη σε αυτά.
- Επανάληψη από το βήμα 2 για τη μάθηση του επόμενου μοντέλου. Αυτό γίνεται μέχρι να επιτευχθεί το επιθυμητό αποτέλεσμα.

1.3.1 Stacking

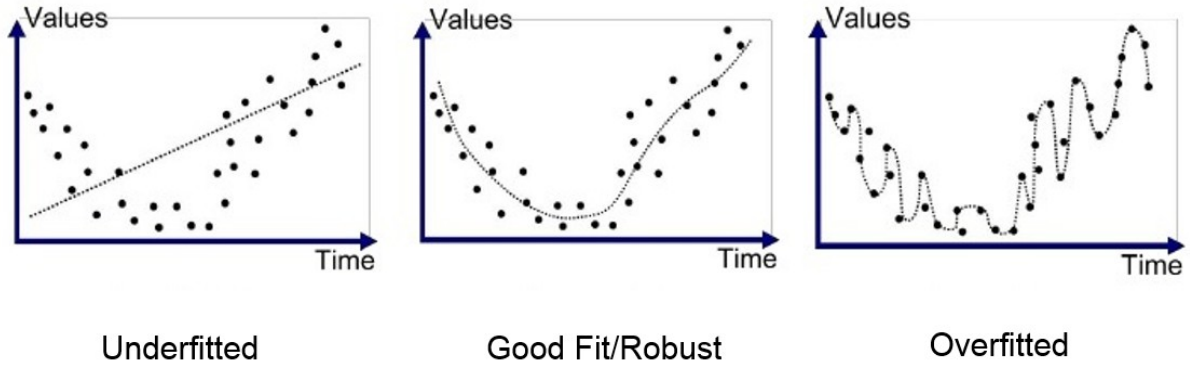
Το stacking είναι μία μέθοδος που συνδυάζονται μοντέλα πρόβλεψης που προέκυψαν από διαφορετικούς αλγόριθμους μάθησης. Στο stacking παράγονται αρχικά ένας αριθμός μοντέλων (επιπέδου 0) και στη συνέχεια από αυτά τα μοντέλα παράγεται ένα μετά-μοντέλο (επιπέδου 1) το οποίο εκπαιδεύεται ώστε να καταλαβαίνει ποια μοντέλα είναι πιο αξιόπιστα ώστε να δίνει με μεγάλη ακρίβεια σωστές απαντήσεις. Η διαδικασία έχει ως εξής:

- Δημιουργία 2 υποσυνόλων από τα δεδομένα, το υποσύνολο δεδομένων εκπαίδευσης DT και το υποσύνολο δεδομένων stacking DS (συνήθως το 10% του D)
- Εκπαίδευση των αρχικών αλγορίθμων με τα δεδομένα DT για την παραγωγή των μοντέλων επιπέδου 0
- Στη συνέχεια γίνεται η λήψη των σωστών αποφάσεων από το σύνολο DS μαζί με τις αποφάσεις από τα μοντέλα επιπέδου 0, για κάθε δεδομένων του συνόλου DS, για τη δημιουργία του συνόλου DM.
- Τέλος, από το σύνολο DM γίνεται η χρήση του κατάλληλου ανάλογα με το πρόβλημα αλγορίθμου για τη δημιουργία του μοντέλου επιπέδου 1.

1.4 Υπερπροσαρμογή - Υποπροσαρμογή

Η υπερπροσαρμογή (overfitting) και η υποπροσαρμογή (underfitting) είναι προβληματικές καταστάσεις που προκύπτουν συνήθως σε μοντέλα, από αλγορίθμους με ικανότητα μάθησης, όταν δεν έχει γίνει σωστά η εκπαίδευση και το καθιστούν μην ικανό να κάνει σωστές προβλέψεις.
Η υποπροσαρμογή αποτελεί ένα φαινόμενο όπου συνήθως το μοντέλο θεωρείται ατελής καθώς έχει δημιουργήσει μια απλοϊκή σχέση δίνοντας έτσι υψηλό σφάλμα, δηλαδή μεροληπτεί, αυτό μπορεί να οφείλεται σε ατελή εκπαίδευση, κακή ποιότητα δεδομένων ακόμα και κακή επιλογή μοντέλου.

Η υπερπροσαρμογή είναι μία κατάσταση όπου το μοντέλο τα πηγαίνει τέλεια, δίνοντας εξαιρετικά αποτελέσματα στα δεδομένα εκπαίδευσης αλλά αποτυγχάνει να δώσει αποτελέσματα στα δεδομένα δοκιμής. Αυτό συμβαίνει διότι το μοντέλο δημιουργεί μια πολύπλοκη και μεγάλου βαθμού πολυωνυμική σχέση με σκοπό να εκμηδενίσει το σφάλμα δημιουργώντας μια καμπύλη που περνάει πάνω, ή πολύ κοντά, από όλα τα σημεία. Αυτό μπορεί να δημιουργηθεί αν υπάρχει σε τεχνητό νευρωνικό δίκτυο κακή αναλογία νευρώνων και δεδομένων, θόρυβος στα δεδομένα ή εμμονή με την επίτευξη χαμηλού σφάλματος χωρίς άλλο έλεγχο. Αυτά μπορούν να αντιμετωπιστούν με τεχνικές όπως πρώιμο σταμάτημα, ομαλοποίηση βαρών και προσωρινή απόρριψη νευρώνων.



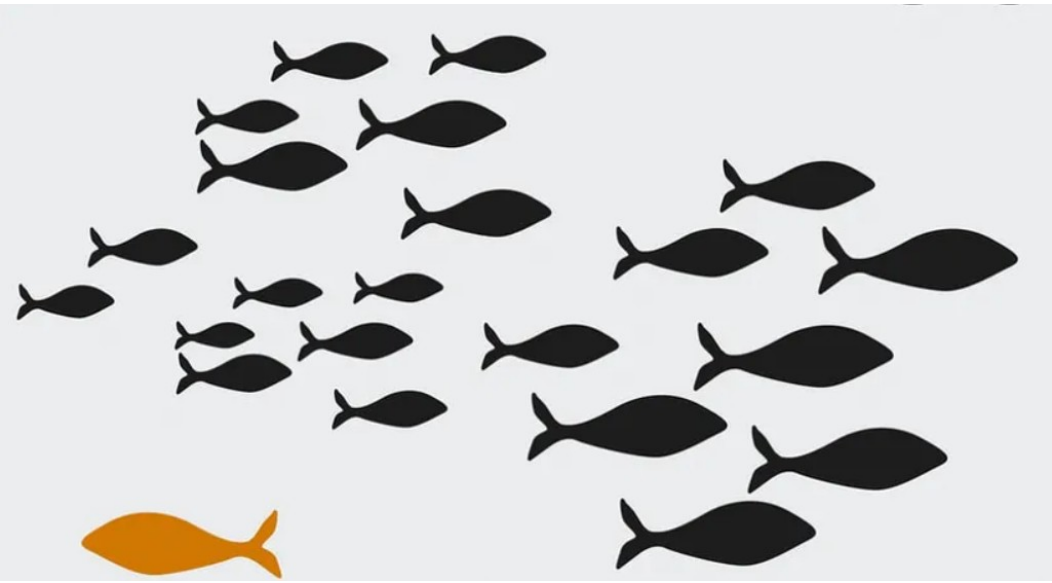
Εικόνα 3: Διαγράμματα προσαρμογής μοντέλου

Παραπάνω στην εικόνα διακρίνεται η απλοϊκή σχέση σε κατάσταση υποπροσαρμογής που αποτελεί μία ευθεία μέσα από τα δεδομένα, η πολύπλοκη σχέση σε κατάσταση υπερπροσαρμογής που προσπαθεί να περάσει πάνω από κάθε σημείο των δεδομένων και στη μέση έναν στόχο καλής προσαρμογής με βάση τα δεδομένα.

Κεφάλαιο 2

2. Ανίχνευση ανωμαλιών

Η ανίχνευση ανωμαλιών ή ανίχνευση ακραίων τιμών ή και ανίχνευση καινοτομιών αποτελεί μία από τις βασικές τεχνικές στη μηχανική μάθηση και την ανάλυση των δεδομένων. Αυτό αφορά τον εντοπισμό περιπτώσεων ή μοτίβων, τα οποία αυτά αποκλίνουν σημαντικά από το μεγαλύτερο μέρος των δεδομένων. Σκοπός της ανίχνευσης ανωμαλιών είναι η αποκάλυψη μη αναμενόμενων παρατηρήσεων οι οποίες μπορούν να σημαίνουν πιθανά σφάλματα ή ανωμαλίες. Η διαφορά ανάμεσα στην ανίχνευση ανωμαλιών και ανίχνευση καινοτομιών είναι πως η ανίχνευση καινοτομιών αφορά την ανίχνευση προηγουμένως αθέατων μοτίβων σημείων. Η ανίχνευση ανωμαλιών έχει αρκετές εφαρμογές σε τομείς όπως τα οικονομικά, η κυβερνοασφάλεια, η βιομηχανία καθώς και η υγεία.



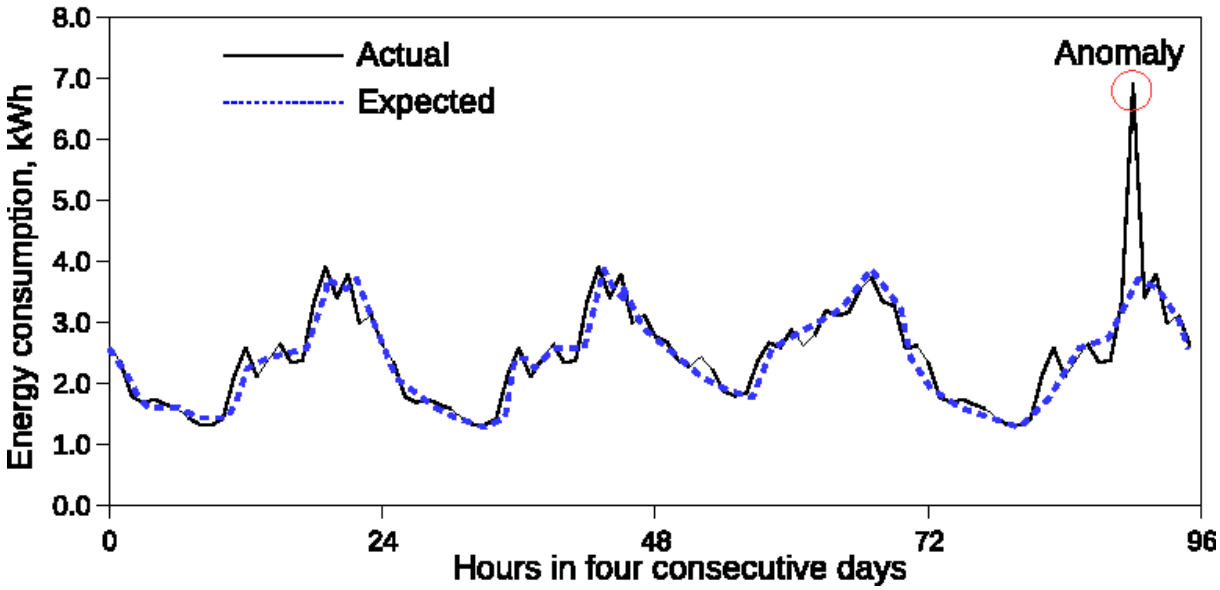
Εικόνα 4: Παράδειγμα ανωμαλίας σε εικόνα

Οι προηγμένοι αλγόριθμοι όπως οι one-class SVM, Isolation forrest, Local outliers factor στοχεύουν στο να δημιουργήσουν ένα μοντέλο φυσιολογικής συμπεριφοράς βασισμένα στην πλειοψηφία των δεδομένων μαρκάροντας στη συνέχεια τις παρατηρήσεις οι οποίες αποκλίνουν ως πιθανές ανωμαλίες ή καινοτομίες.

2.1 Τί είναι η ανωμαλία;

Οι ανωμαλίες, κάτι το μη ομαλό, είναι μοτίβα ή σημεία στα δεδομένα τα οποία δε συμμορφώνονται με μια σαφώς καθορισμένη έννοια της κανονικής-ομαλής

συμπεριφοράς. Από αυτό βγαίνει το συμπέρασμα πως σε ένα πλήθος δεδομένων η πλειοψηφία τείνει να συμπεριφέρεται με παρόμοιο τρόπο σύμφωνα με τα χαρακτηριστικά της. Για να γίνει η ανίχνευση ανωμαλιών σε ένα σύνολο δεδομένων-περιπτώσεων θα πρέπει πρώτα να γίνει η ανίχνευση προτύπων που ορίζουν την ομαλή συμπεριφορά αυτών. Για παράδειγμα, στο παρακάτω διάγραμμα η ομαλή συμπεριφορά όσων αφορά την κατανάλωση ενέργειας σε kWh κυμαίνεται μεταξύ των τιμών 1 και 4, τιμή 7 είναι μοναδική και είναι εκτός των ορίων όπου σύμφωνα με το μοτίβο που έχει δημιουργηθεί η τιμή θα έπρεπε να κυμαίνεται κοντά στο 4.



Εικόνα 5: Παράδειγμα ανωμαλίας σε χρονοσειρά

Στην επιστήμη των δεδομένων, αφού καθοριστεί ομαλή συμπεριφορά των δεδομένα, υπάρχουν 3 δημοφιλείς μέθοδοι για τον εντοπισμό των ασυνήθιστων δεδομένων:

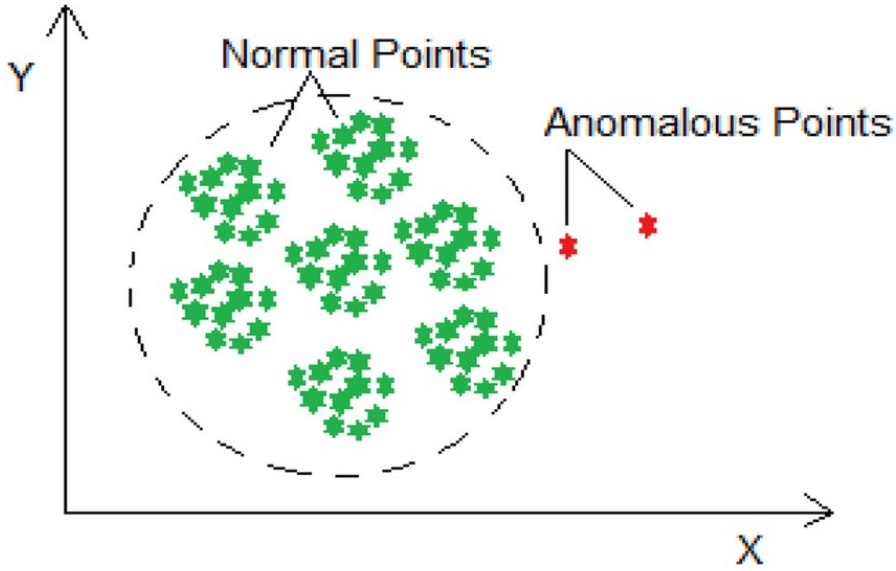
- **Χειροκίνητη ανίχνευση.**
  Τα δεδομένα επανεξετάζονται για να διαπιστωθεί τι είναι εκτός του κανονικού
- **Αυτόματη/στατιστική ανίχνευση.**
  Χρησιμοποιείται ένα σύστημα ειδοποίησης που βασίζεται σε κατώτατα όρια που ρυθμίζονται χειροκίνητα
- **Μηχανική μάθηση.**
  Χρησιμοποιεί αλγορίθμους που μαθαίνουν κανονικά μοτίβα στα δεδομένα για να ανιχνεύσουν τυχόν ανωμαλίες στα σημεία.

Η σύγκριση μεταξύ των παραπάνω μπορεί να γίνει με τους εξής παράγοντες ικανότητα κλιμάκωσης, ανίχνευση σε πραγματικό χρόνο, ακρίβεια, κόστος εφαρμογής, κόστος συντήρησης. Οι τεχνικές μηχανικής μάθησης αποτελεί την καλύτερη από τις τρεις αυτές επιλογές καθώς ο μοναδικός παράγοντας στον οποίο υστερεί είναι το κόστος εφαρμογής.

2.2 Τύποι ανωμαλιών

Υπάρχουν 3ις τύποι ανωμαλιών οι οποίες χωρίζονται με βάση τα υποκείμενα χαρακτηριστικά τους και στον τρόπο με τον οποίο αποκλίνουν από τα αναμενόμενα πρότυπα εντός ενός συνόλου δεδομένων.

1. **Ανωμαλία σημείου.**
   Με τον όρο αυτό εννοούμε ότι ένα σημείο από το σύνολο των δεδομένων έχει μεγάλη απόκλιση από τα υπόλοιπα. Αντιπροσωπεύουν ένα ακραίο σημείο, μια παρατυπία ή μια απόκλιση που εμφανίζεται τυχαία χωρίς καμία συσχέτιση με το κοινό μοτίβο στα δεδομένα. Όπως φαίνεται στην παρακάτω εικόνα τα πράσινα σημεία ανήκουν
   σε 6 διαφορετικές συστάδες ορισμένης απόστασης μεταξύ τους, τα 2 κόκκινα σημεία είναι εκτός των ορίων των συστάδων εμφανίζοντας μια ανωμαλία.



Εικόνα 6: Ανωμαλία σημείου

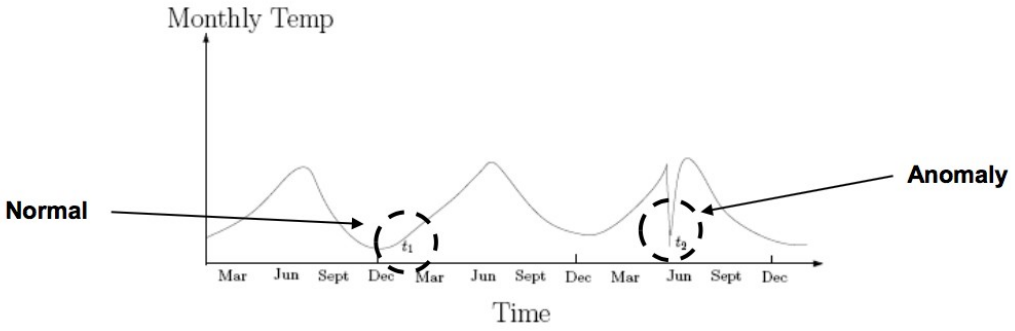
2. **Ανωμαλία πλαισίου.**
   Μια περίπτωση θεωρείται ανώμαλη από ένα συγκεκριμένο πλαίσιο, τότε λέγεται ότι είναι μια ανωμαλία του πλαισίου. Αυτό δε σημαίνει ότι αν υπάρξει ίδια τιμή στο σύνολο των δεδομένων θα είναι και αυτή ανώμαλη. Για παράδειγμα, στην περίπτωση δεδομένων χρονοσειρών, όπως οι καταγραφές μιας συγκεκριμένης ποσότητας σε βάθος χρόνου, το πλαίσιο είναι σχεδόν πάντα προσωρινό. Η έννοια του πλαισίου προκαλείται από τη δομή στο σύνολο δεδομένων και πρέπει να είναι προσδιορίζεται ως μέρος της διατύπωσης του προβλήματος. Κάθε περίπτωση δεδομένων ορίζεται χρησιμοποιώντας τα ακόλουθα δύο σύνολα χαρακτηριστικών:

   Συναφή χαρακτηριστικά.
   Τα Συναφή χαρακτηριστικά χρησιμοποιούνται για τον προσδιορισμό του πλαισίου (ή της γειτονιάς) για την εν λόγω περίπτωση. Για παράδειγμα, σε

σύνολα χωρικών δεδομένων, το γεωγραφικό μήκος και πλάτος μιας τοποθεσίας είναι τα χαρακτηριστικά πλαισίου. Στα δεδομένα χρονοσειρών, ο χρόνος είναι ένα συναφή χαρακτηριστικό που καθορίζει τη θέση μιας περίπτωσης σε ολόκληρη την ακολουθία.

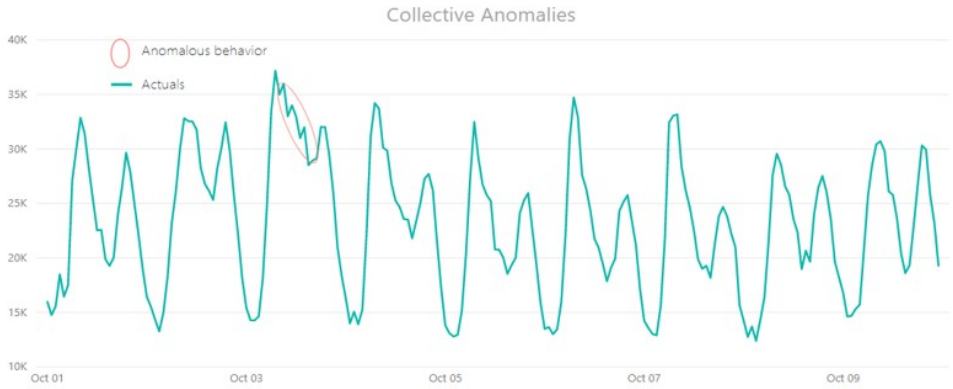
Χαρακτηριστικά συμπεριφοράς.
Τα χαρακτηριστικά συμπεριφοράς καθορίζουν τα μη πλαισιωμένα χαρακτηριστικά μιας περίπτωσης. Για παράδειγμα, σε ένα σύνολο χωρικών δεδομένων που περιγράφει το μέση βροχόπτωση ολόκληρου του κόσμου, η ποσότητα της βροχόπτωσης σε κάθε τοποθεσία είναι ένα χαρακτηριστικό συμπεριφοράς.

Η ανώμαλη συμπεριφορά προσδιορίζεται χρησιμοποιώντας τις τιμές των χαρακτηριστικών συμπεριφοράς σε ένα συγκεκριμένο πλαίσιο. Μία περίπτωση δεδομένων μπορεί να μη θεωρείται ανωμαλία σε ένα συγκεκριμένο πλαίσιο, ενώ σε ένα διαφορετικό να θεωρείται. Για παράδειγμα, στη φωτογραφία φαίνεται πως ενώ η t1=t2 παρατηρούμε ότι η t1 δεν είναι ανωμαλία καθώς είναι μέσα στην ομαλή λειτουργία του μοτίβου των δεδομένων για το πλαίσιο που βρίσκεται, σε αντίθεση με την t2 που είναι μια μη αναμενόμενη τιμή στο συγκεκριμένο πλαίσιο.



Εικόνα 7: Ανωμαλία πλαισίου

3. **Συλλογικές ανωμαλίες.**
Αν ένα πλήθος από παρόμοιες περιπτώσεις δεδομένων εμφανίζει ανωμαλία σε σχέση με το σύνολο των δεδομένων τότε αυτό το πλήθος ονομάζεται συλλογική ανωμαλία. Μια περίπτωση δεδομένων, η οποία ανήκει στο πλήθος της συλλογικής ανωμαλίας, μπορεί από μόνη της να μην εμφανίζει ανωμαλία. Στην παρακάτω εικόνα φαίνεται το παράδειγμα μιας συλλογικής ανωμαλίας, όπου το μοτίβο μεταξύ 3/10 και 5/10 φαίνεται ένα πλήθος των δεδομένων που συνθέτουν τη χρονοσειρά φαίνεται να ξεφεύγει από την ομαλή λειτουργία της.



Εικόνα 8: Συλλογική ανωμαλία

Η διαφορά με τις σημειακές ανωμαλίες είναι πως η συλλογικές ανωμαλίες εμφανίζονται σε σύνολα δεδομένων και όχι σε συλλογικές περιπτώσεις δεδομένων που σχετίζονται μεταξύ τους. Αντίθετα, η εμφάνιση ανωμαλιών πλαισίου βασίζεται στη διαθεσιμότητα χαρακτηριστικών πλαισίου στα δεδομένα. Επιπλέον, μπορεί μία σημειακή ή συλλογική ανωμαλία να εμφανιστεί ως ανωμαλία πλαισίου. Έτσι, ένα πρόβλημα ανίχνευσης σημειακών ανωμαλιών ή ένα πρόβλημα ανίχνευσης συλλογικών ανωμαλιών μπορεί να μετατραπεί σε πρόβλημα ανίχνευσης ανωμαλιών με βάση το πλαίσιο με την ενσωμάτωση της πληροφορίας πλαισίου.

2.3 Οι έξοδοι της ανίχνευσης ανωμαλιών

Σημαντικό για τη δημιουργία ενός μοντέλου μηχανικής μάθησης για την ανίχνευση ανωμαλιών είναι οι έξοδοι που αυτό θα δώσει. Οι τύποι των εξόδων που μπορεί ένα μοντέλο μηχανικής μάθησης για ανίχνευση ανωμαλιών είναι 2 και συγκεκριμένα είναι οι:

- **Βαθμολογία**.
  Από αυτόν τον τύπο εξόδου το μοντέλο βγάζει μια βαθμολογία (score). Ορίζεται μια βαθμολογία στις περιπτώσεις των δεδομένων που έχουν οριστεί ως δοκιμαστικά δεδομένα ανάλογο στο κατά πόσο αυτά θεωρούνται ανωμαλίες. Η έξοδος με αυτόν τον τρόπο μπορεί να απεικονισθεί ως ένας ταξινομημένος κατάλογος ανωμαλιών. Ένας αναλυτής έχει την επιλογή να αναλύσει τις λίγες κορυφαίες ανωμαλίες ή να χρησιμοποιήσει ένα κατώφλι αποκοπής για την επιλογή των ανωμαλιών.
- **Ετικέτες**.
  Στον συγκεκριμένο τύπο εξόδου το μοντέλο δίνει μια ετικέτα (label) στα δεδομένα που έχουν οριστεί ως δεδομένα δοκιμής για το αν αυτά είναι ανωμαλίες ή όχι. Οι διαφορές με τον τύπο της βαθμολογίας είναι στο ότι οι ετικέτες δεν είναι ποσοτικές αλλά διακριτές, οπότε ο αναλυτής δεν μπορεί να χρησιμοποιήσει ένα κατώφλι ώστε να επιλέξει τις πιο σχετικές ανωμαλίες. Αυτό όμως μπορεί να ελεγχθεί έμμεσα μέσω επιλογών παραμέτρων εντός κάθε τεχνικής.

2.4 Εφαρμογές

Η ανίχνευση ανωμαλιών με μηχανική μάθηση έχει εφαρμογές και είναι ένα χρήσιμο εργαλείο σε αρκετούς τομείς καθώς διαθέτει μεγαλύτερη ταχύτητα και ακρίβεια από άλλες συμβατικές τεχνικές. Παρακάτω θα γίνει μια σύντομη αναφορά σε αυτούς τους τομείς.

- **Ανίχνευση εισβολών**.
  Ο συγκεκριμένος τομέας αφορά την κυβερνοασφάλεια αναφέρεται στην ανίχνευση κακόβουλης δραστηριότητας σε ένα υπολογιστικό σύστημα ή δίκτυο. Οι εισβολές σε ένα σύστημα δεν έχουν αποκλίσεις από την ομαλή συμπεριφορά αυτού με αποτέλεσμα οι τεχνικές ανίχνευσης ανωμαλιών να μπορούν να εφαρμοστούν πάνω σε ένα τέτοιο σύστημα. Η μεγαλύτερη πρόκληση για την ανίχνευση τέτοιων ανωμαλιών είναι ο πολύ όγκος δεδομένων. Τα δεδομένα σε αυτή την περίπτωση προέρχονται με ροή, και συνήθως σε πραγματικό χρόνο, απαιτώντας έτσι ανάλυση σε απευθείας σύνδεση. Επιπλέον, ένα ζήτημα που προκύπτει εξαιτίας του τεράστιου όγκου δεδομένων εισόδου είναι το ποσοστό ψευδών συναγερμών, καθώς ένα μόλις μικρό ποσοστό ψευδών συναγερμών μπορεί να μπορεί να είναι αρκετό ώστε να κάνει αρκετά δύσκολη την ανάλυση από τον αναλυτή. Στον συγκεκριμένο τομέα προτιμώνται μη επιβλεπόμενες ή ημι-επιβλεπόμενες τεχνικές ανίχνευσης ανωμαλιών καθώς ενώ τα οι ετικέτες δεδομένων ομαλής λειτουργίας είναι συνήθως διαθέσιμες αυτό δεν ισχύει για τις ετικέτες της εισβολής.
- **Ανίχνευση απάτης**.
  Ο τομέας αυτός αναφέρεται στην ανίχνευση εγκληματικών δραστηριοτήτων ή απάτης που συμβαίνουν σε εμπορικούς οργανισμούς. Η απάτη συμβαίνει όταν οι κακόβουλοι χρήστες, που μπορεί να είναι πελάτες του οργανισμού ή να παριστάνουν τους πελάτες, καταναλώνουν τους πόρους που παρέχει ο οργανισμός με μη εξουσιοδοτημένο τρόπο. Η ανίχνευση τέτοιων ανωμαλιών είναι σημαντική για τους οργανισμούς ώστε να υπάρξει η αποφυγή οικονομικών απωλειών. Οι Fawcett και Provost [1999] εισάγουν τον όρο παρακολούθηση δραστηριότητας ως γενική προσέγγιση για την ανίχνευση απάτης σε αυτούς τους τομείς. Η τυπική προσέγγιση των τεχνικών ανίχνευσης ανωμαλιών είναι η διατήρηση ενός προφίλ χρήσης για κάθε πελάτη και η παρακολούθηση των προφίλ για τον εντοπισμό τυχόν αποκλίσεων.
- **Υγεία**.
  Στον τομέα της υγείας η ανίχνευση ανωμαλιών εργάζεται με δεδομένα ασθενών όπου αυτά μπορεί να έχουν ανωμαλίες για διάφορους λόγους, όπως μη φυσιολογική κατάσταση του ασθενούς ή σφάλματα οργάνων ή σφάλματα καταγραφής. Από αυτό προκύπτει πως η ανίχνευση ανωμαλιών αποτελεί ένα πολύ κρίσιμο πρόβλημα σε αυτόν τον τομέα απαιτώντας τον βέλτιστο βαθμό ακρίβειας. Τα δεδομένα συνήθως αποτελούνται από πολλά διαφορετικά χαρακτηριστικών αλλά μπορούν επίσης να έχουν τόσο χρονική όσο και χωρική διάσταση. Συνήθως τα δεδομένα με ετικέτες ανήκουν σε υγιείς ασθενείς, με αποτέλεσμα οι περισσότερες τεχνικές χρησιμοποιούν ημι-

επιβλεπόμενη μάθηση. Μια άλλη μορφή δεδομένων είναι τα δεδομένα χρονοσειρών, όπως τα ηλεκτροκαρδιογραφήματα και τα ηλεκτροεγκεφαλογραφήματα στα οποία έχουν εφαρμοστεί συλλογικές τεχνικές ανίχνευσης ανωμαλιών σε τέτοια δεδομένα. Η μεγαλύτερη πρόκληση του προβλήματος ανίχνευσης ανωμαλιών σε αυτόν τον τομέα είναι ότι το κόστος της ταξινόμησης μιας ανωμαλίας ως φυσιολογικής μπορεί να είναι πολύ υψηλό.

- **Βιομηχανία**.
  Στον τομέα της βιομηχανίας οι βιομηχανικές μονάδες υφίστανται βλάβες λόγω της συνεχούς χρήσης αλλά και της φυσιολογικής φθοράς. Είναι σημαντικό οι βλάβες αυτές να εντοπίζονται το συντομότερο δυνατόν έτσι ώστε να μην υπάρξει κλιμάκωση του και να γίνει αποφυγή απωλειών. Τα δεδομένα από αυτόν τον τομέα λαμβάνονται συνήθως αισθητήρες, καθώς αυτά καταγράφονται με τη χρήση της βιομηχανικής μονάδας και συλλέγονται για ανάλυση. Στον συγκεκριμένο τομέα έχει υπάρξει μια εκτενής εφαρμογή τεχνικών ανίχνευσης ανωμαλιών. Η ταξινόμηση της ανίχνευση βιομηχανικών βλαβών γίνεται σε δύο τομείς, σε ελαττώματα μηχανικών εξαρτημάτων και σε ελαττώματα που αφορούν τις φυσικές δομές.
  Οι τεχνικές ανίχνευσης ανωμαλιών μηχανικές μονάδες παρακολουθούν την απόδοση βιομηχανικών εξαρτημάτων και ανιχνεύουν ελαττώματα που μπορεί να προκύψουν λόγω φθοράς ή άλλων απρόβλεπτων περιστάσεων. Τα δεδομένα σε αυτόν τον τομέα συνήθως έχουν τη μορφή χρονοσειρών δεδομένου ότι η παρακολούθηση ενός εξαρτήματος γίνεται σε βάθος χρόνου. Οι τύπο ανωμαλιών που συναντώνται αφορούν ως επί το πλείστων ανωμαλίες πλαισίου ή συλλογικές ανωμαλίες. Επειδή τα δεδομένα εξαρτημάτων χωρίς ελαττώματα είναι διαθέσιμα εφαρμόζονται τεχνικές ημι-επιβλεπώμενης μάθησης.
  Οι τεχνικές ανίχνευσης σε δεδομένα που αφορούν ελαττώματα ή βλάβες φυσικών δομών ανιχνεύουν δομικές ανωμαλίες στις κατασκευές. Τα δεδομένα που συλλέγονται σε αυτόν τον τομέα έχουν επίσης χρονική διάσταση. Οι τεχνικές ανίχνευσης ανωμαλιών είναι παρόμοιες με τις τεχνικές ανίχνευσης καινοτομιών ή ανίχνευσης σημείων αλλαγής, δεδομένου ότι προσπαθούν να ανιχνεύσουν αλλαγές στα δεδομένα που συλλέγονται από μια δομή. Τα κανονικά δεδομένα και συνεπώς τα μοντέλα που μαθαίνονται είναι συνήθως στατικά με την πάροδο του χρόνου. Τα δεδομένα ενδέχεται να έχουν χωρικές συσχετίσεις.
- **Αισθητήρες**.
  Αρκετά ενδιαφέρον είναι τα δεδομένα που συλλέγονται από αισθητήρες δεδομένου ότι έχουν αρκετά μοναδικά χαρακτηριστικά. Οι ανωμαλίες στα δεδομένα αυτά μπορεί είτε να σημαίνουν ότι ένας ή περισσότεροι αισθητήρες είναι ελαττωματικοί, είτε ότι ανιχνεύουν γεγονότα. Ένα ενιαίο δίκτυο αισθητήρων μπορεί να αποτελείται από αισθητήρες που συλλέγουν διαφορετικούς τύπους δεδομένων, όπως δυαδικά, διακριτά, συνεχή, ήχου, βίντεο κ.λπ. Ένα πρόβλημα των δικτύων αισθητήρων είναι πως το περιβάλλον στο οποίο λειτουργούν, καθώς και το κανάλι επικοινωνίας μπορούν να προκαλέσουν θόρυβο και χαμένες τιμές στα δεδομένα που

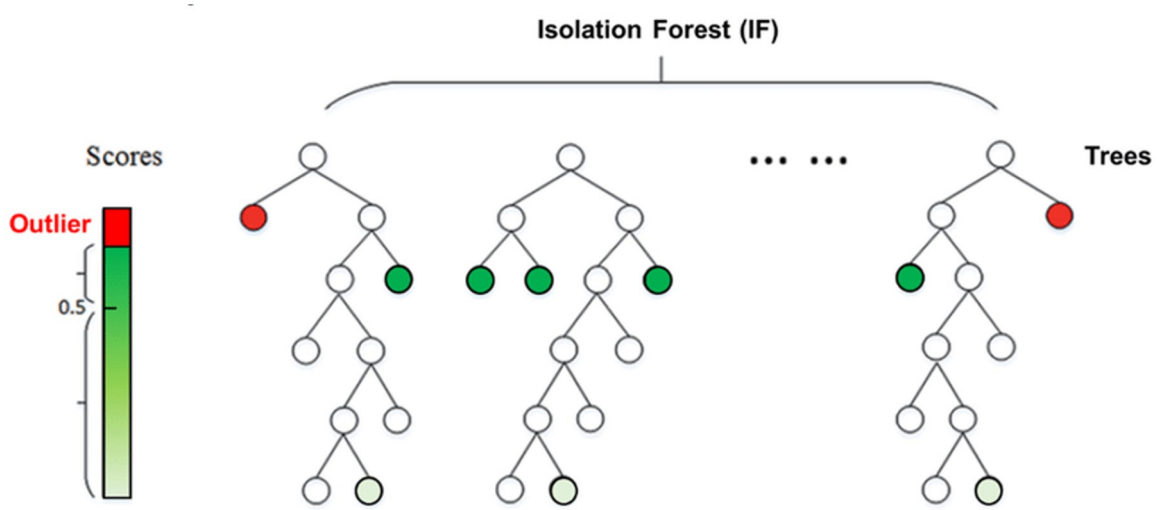
συλλέγονται. Η παρουσία θορύβου στα δεδομένα που συλλέγονται από τον αισθητήρα καθιστά την ανίχνευση ανωμαλιών πολύ δύσκολη, καθώς πρέπει πλέον να διακρίνει μεταξύ των ενδιαφερουσών ανωμαλιών και του ανεπιθύμητου θορύβου ή ελλιπών τιμών.

2.5 Αλγόριθμοι μηχανικής μάθησης για πρόβλεψη ανωμαλιών

Παρακάτω θα γίνει ανάλυση των αλγορίθμων μηχανικής μάθησης που χρησιμοποιήθηκαν για την υλοποίηση του πειράματος καθώς και κάποιον σημαντικών αλγορίθμων που χρησιμοποιούνται ευρέως.

2.5.1 Isolation Forest

Ο αλγόριθμος Isolation forest είναι ένας από τους πιο δυνατούς και αποδοτικούς αλγορίθμους μηχανικής μάθησης για την ανίχνευση ανωμαλιών. Προτάθηκε από τους Fei Tony Liu, Kai Ming Ting και Zhi-Hua Zhou το 2008 μέσο της δημοσίευσης της έρευνας με τίτλο “Isolation Forest”. Στην έρευνα αυτή προτείνεται μια μέθοδος βασισμένη στο μοντέλο που απομονώνει ρητά τις ανωμαλίες και μόνο. Για την επίτευξη αυτού του στόχου αυτή η μέθοδος εκμεταλλεύεται το γεγονός πως οι ανωμαλίες είναι “λίγες και διαφορετικές” πράγμα που τις κάνει πιο επιρρεπής στην απομόνωση. Λόγο αυτής τους της ευαισθησίας, οι ανωμαλίες, τείνουν να απομονώνονται κοντά στη ρίζα του δέντρου (Tree ή iTree) και αποτελεί τη βάση αυτής της μεθόδου.



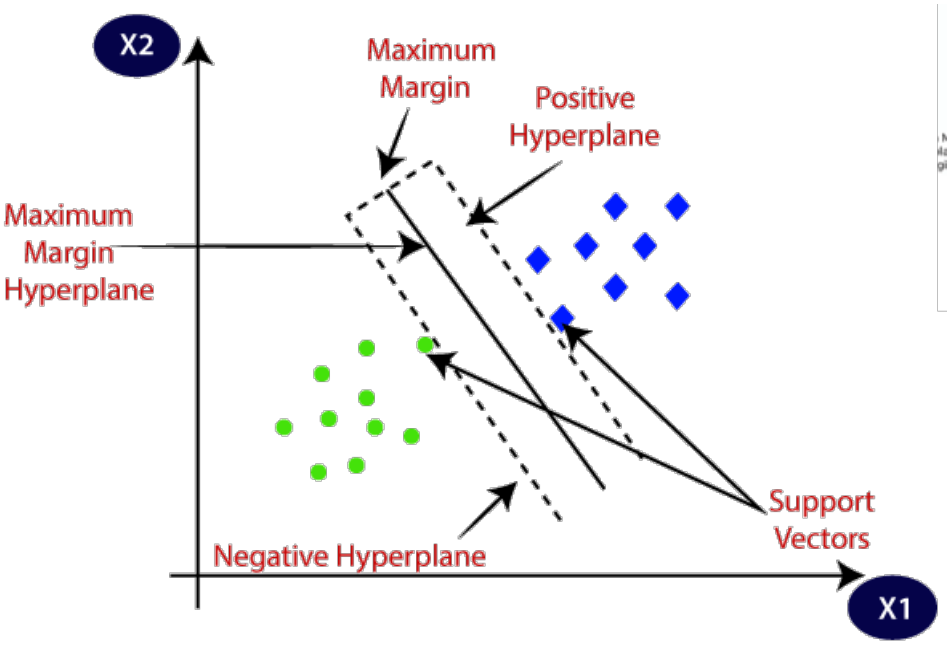
Εικόνα 9: Γραφική απεικόνιση λειτουργίας του Isolation Forrest

Ο αλγόριθμος για να λειτουργήσει δημιουργεί ένα σύνολο από τέτοια δέντρα για το εκάστοτε σύνολο των δεδομένων και θεωρεί ανωμαλίες τις περιπτώσεις που έχουν μικρό μέσο μήκος διαδρομής από τη ρίζα των δέντρων λαμβάνοντας υπόψη μόνο δύο παραμέτρους, των αριθμό των δέντρων που θα κατασκευαστούν και τον αριθμό υπο-δειγματοληψίας. Ο αλγόριθμος Isolation forest είναι ικανός να πετύχει υψηλή απόδοση ανίχνευσης με μεγάλη αποδοτικότητα χρησιμοποιώντας μικρό αριθμό δέντρων και δείγματος.

2.5.2 One Class – Support Vector Machine

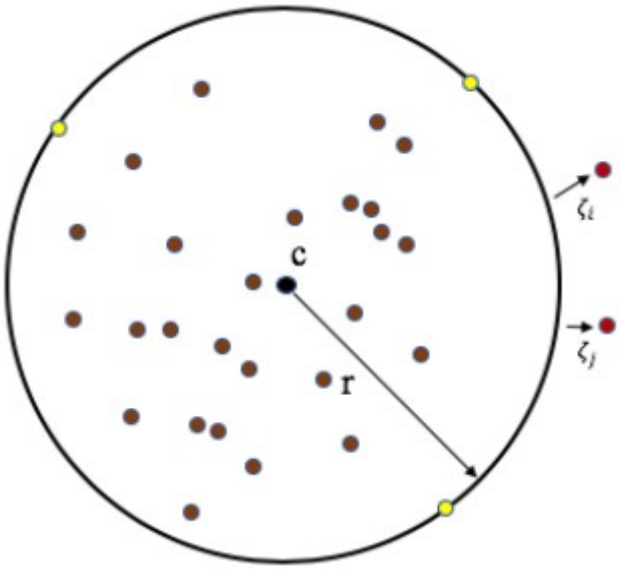
Ο αλγόριθμος One Class – Support Vector Machine αποτελεί έναν αλγόριθμο μη επιβλεπόμενης μάθησης που χρησιμοποιείται για την ανίχνευση ανωμαλιών και έχει εμπνευστεί από τον αλγόριθμο Support Vector Machine classifier που προτάθηκε από τους B. Scholkopf, J. Platt, J. Shawe-Taylor, A. J. Smola and RC Williamson, μέσα από την έρευνα που δημοσιεύθηκε με τίτλο “Estimating the support of a high-dimensional distribution”.

Είναι ένας συνδυασμός των αλγορίθμων One class Classifier, που βρίσκει ένα υπερεπίπεδο το οποίο διαχωρίζει τα δείγματα μιας συγκεκριμένης κλάσης με τη μάθηση από δείγματα μίας κλάσης κατά τη διάρκεια της εκπαίδευσης, και του Support Vector machines, που έχει την ικανότητα να σχεδιάσει το πιο κατάλληλα προσαρμοσμένο υπερεπίπεδο ώστε να διαχωρίσει τα σημεία της μία κλάσης από την άλλη, ευθεία που δημιουργεί ο SVM έχει σχεδόν ίση και μέγιστη απόσταση από τα ακραία σημεία των 2 κλάσεων (support vectors).



Εικόνα 10: Γραφική απεικόνιση του αλγορίθμου One Class - SVM

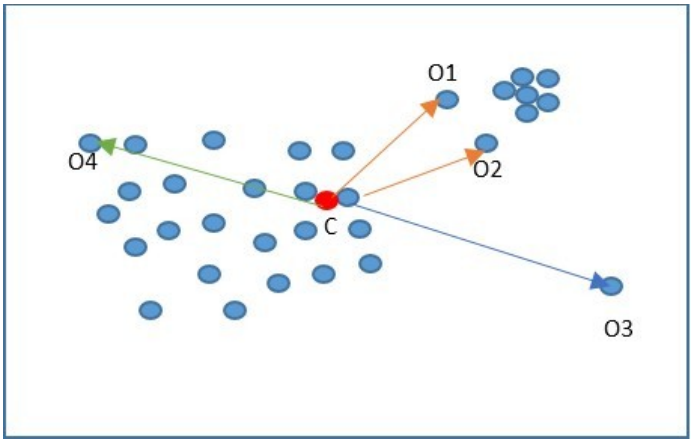
Ο αλγόριθμος One Class – Support Vector Machine δημιουργεί μια υπερσφαίρα με κέντρο c και ακτίνα r της μίας κατηγορίας παραδειγμάτων από τα δεδομένα εκπαιδεύσεις. Ο αλγόριθμος βρίσκει τη μικρότερη δυνατή απόσταση ελαχιστοποιώντας την ακτίνα r και θεωρεί ανωμαλίες όσα από τα σημεία δεν υπάγονται εντός των ορίων της υπερσφαίρας. Στην παρακάτω εικόνα φαίνεται η υπερσφαίρα με κέντρο c και ακτίνα r ενώ θεωρεί ανωμαλίες τα σημεία ζ.



Εικόνα 11: Δισδιάστατη απεικόνιση υπερσφέρας

2.5.3 Local Outlier Factor

Ο αλγόριθμος Local Outlier Factor (LOF) αποτελεί μια μη επιβλεπόμενη μέθοδο μηχανικής μάθησης για ανίχνευσης ανωμαλιών που υπολογίζει την τοπική διακύμανση της πυκνότητας κάθε αντικειμένου σε ένα σύνολο δεδομένων και προτάθηκε από τούς Breunig, M. M., Kriegel, H. P., Ng, R. T., και Sander, J. το 2000 μέσα από την έρευνα τους με όνομα “LOF: Identifying Density-Based Local Outliers”. Λειτουργεί κάνοντας εύρεση ακραίων τιμών σε ένα πολυδιάστατο χώρο όπου εισάγει μια τοπική ακραία τιμή (LOF) για κάθε αντικείμενο του συνόλου δεδομένων, η οποία υποδεικνύει τον βαθμό ακραίας τιμής του. Η συγκεκριμένη τιμή ποσοτικοποιεί το πόσο ακραίο είναι το εν λόγω αντικείμενο. Σε αντίθεση με τις σφαιρικές μεθόδους που εξετάζουν ολόκληρο το σύνολο των δεδομένων, ο LOF εστιάζει στη σχετική πυκνότητα των περιπτώσεων στη γειτονιά κάθε αντικειμένου.



Εικόνα 12: Απεικόνιση λειτουργίας Local Outlier Factor

Για παράδειγμα, στην παραπάνω εικόνα παρατηρούμε ότι τα αντικείμενο O1, O2, O3 ,O4 έχουν μικρότερη πυκνότητα από το αντικείμενο C λαμβάνοντας υπόψη έναν αριθμό γειτόνων x. Μπορούμε να λάβουμε για x = 5 και φαίνεται οπτικά πως και το μεσαίο αντικείμενο της πάνω δεξιάς συστάδας έχει μεγαλύτερη πυκνότητα από το O1. Ο αλγόριθμος υπολογίζει την πυκνότητα κάθε αντικειμένου και θεωρεί ανωμαλία τα αντικείμενα με χαμηλή πυκνότητα σε σχέση με τους γείτονές τους. Η μοναδική μεταβλητή που λαμβάνει ο αλγόριθμος είναι ο αριθμός των γειτόνων.

Κεφάλαιο 3

3. Robot Vitals

Τα Robot Vitals είναι ένα σύνολο μετρήσεων που υποδεικνύουν την υποβάθμιση της απόδοσης που αντιμετωπίζει ένα ρομπότ ανά πάσα στιγμή. Προτάθηκε από τον Aniketh Ramesh μέσα από την έρευνα του με τίτλο “Robot Vitals and Robot Health: Towards Systematically Quantifying Runtime Performance Degradation in Robots Under Adverse Conditions” το 2022. Κάθε robot vital αντιπροσωπεύει μία συγκεκριμένη πτυχή της συμπεριφοράς του ρομπότ κατά τη διάρκεια δυσμενών συνθηκών. Από μόνο κάποιο vital δεν μπορεί να δώσει σαφείς πληροφορίες για το αν ένα ρομπότ αποτυγχάνει, η κατάσταση όμως του συνόλου των vitals μπορεί να αποτελέσει έναν ισχυρό δείκτη για την κατανόηση της φύσης των δυσκολιών που αντιμετωπίζει το ρομπότ.

Ένα ρομπότ “υποφέρει” αν βιώνει υψηλή υποβάθμιση των επιδόσεων του και αυτό μπορεί να έχει ως αποτέλεσμα έως και την καταστροφή του. Το κάθε vital μας δίνει η την πιθανότητα ένα ρομπότ να βιώνει υψηλή υποβάθμιση των επιδόσεων (υποφέρει) για την συγκεκριμένη πτυχή. Τα vitals υπολογίζονται με φιλτράρισμα και δειγματοληψία δεδομένων πραγματικού χρόνου από τη λειτουργία του ρομπότ.

3.1 Rate of Change of Distance from navigational goal ($\dot{d}g$)

Το συγκεκριμένο vital “mag_distFromGoal_roc” υποδεικνύει καταστάσεις στις οποίες οι παράγοντες μειώνουν την απόδοση του ρομπότ το εμποδίζουν στα να κινηθεί προς τον στόχο. Η θέση του ρομπότ λαμβάνεται μετά τη συγχώνευση αισθητήρων του φίλτρου ExtendedKalman και ο στόχος δίνεται από τον χειριστή ή τον αλγόριθμο πλοήγησης. Η απόσταση από τον στόχο dg (“mag_distFromGoal”) υπολογίζεται χρησιμοποιώντας την ευκλείδεια απόσταση ενώ το ρομπότ κινείται προς τον στόχο με ομοιόμορφη ταχύτητα όταν έχει ιδανική συμπεριφορά. Το vital έχει που βγαίνει από την θέση του ρομπότ έχει τρεις καταστάσεις.

Η πρώτη κατάσταση αφορά την ιδανική συμπεριφορά και έχει τιμές $\dot{d}g < 0$ εκτός από κάποιες διακυμάνσεις, τη δεύτερη αφορά το $\dot{d}g \approx 0$ που δείχνει ότι το ρομπότ έχει μικρή ομοιότητα με την ιδανική συμπεριφορά και αφορά την κατάσταση του ρομπότ που δεν μπορεί να κινηθεί και την τρίτη $\dot{d}g > 0$ όπου το ρομπότ δεν έχει ομοιότητα με την ιδανική συμπεριφορά και η υποβάθμιση της απόδοσης του έχει ως αποτέλεσμα το ρομπότ να απομακρύνεται από τον στόχο.

3.2 Jerk along Axis of Motion (a˙z)

Το συγκεκριμένο vital ανιχνεύουν καταστάσεις μείωσης απόδοσης της λειτουργίας που αφορούν το ανώμαλο έδαφος. Οι ξαφνικές αλλαγές, κυρίως του z άξονα, όπως τρεμούλιασμα, τράνταγμα οι βυθίσεις υψόμετρο του εδάφους μπορούν να ανατρέψουν το ρομπότ και να οδηγήσουν σε μεγάλες βλάβες. Στα πειράματα της έρευνας παρατηρήθηκε πως ξαφνικά τραντάγματα ± 30 μοιρών (≈ ±0,5 ακτίνια) ή και άνω είναι ικανά να ανατρέψουν το ρομπότ οπότε η πιθανότητα να υποστεί το ρομπότ τέτοιου είδους αναταράξεις είναι υψηλή όταν το |a˙z|≈ ±0,5 ακτίνια, και χαμηλή αν |a˙z|≈ ±0.

Το μέγεθος του τραντάγματος κατά μήκος του άξονα κίνησης υπολογίζεται χρησιμοποιώντας τον ρυθμό μεταβολής της γραμμικής επιτάχυνσης κατά μήκος του άξονα Z ˙az. Η μέτρηση az (“imu_data__linear_acceleration_z”) λαμβάνετε από τον τη μονάδα IMU. Επειδή οι μετρήσεις της μονάδας IMU συνήθως είναι θορυβώδεις εξομαλύνονται με τη χρήση μέσου όρου κυλιόμενου παραθύρου και στη συνέχεια υποδειγματοληπτούνται σε μία μέτρηση ανά δευτερόλεπτο πριν από τον υπολογισμό του ˙az (“imu_linAcc_z_roc”).

3.3 RoC of Localisation Error (˙δloc)

Το συγκεκριμένο vital αναφέρετε σε καταστάσεις υποβάθμισης της απόδοσης του ρομπότ από περιπτώσεις όπου το ρομπότ έχει κολλήσει, λόγο ανώμαλου εδάφους για παράδειγμα, αλλά οι ρόδες του συνεχίζουν να κινούνται ελεύθερα. Σε μί9α τέτοια κατάσταση η ακατέργαστη οδομετρία του ρομπότ (χ1) συνεχώς αλλάζει ενώ η οπτική οδομετρία του ρομπότ ή η συγχώνευση αισθητήρων EKF (χ2) παραμένει σταθερή δημιουργώντας σφάλμα εντοπισμού δloc = χ1 - χ2 (“odom_posErr”). Διαφορετικοί αλγόριθμοι SLAM είναι ανθεκτικοί σε διαφορετικά επίπεδα σφαλμάτων εντοπισμού (δloc). Ωστόσο, η απόδοση ενός ρομπότ επιδεινώνεται μετά από παρατεταμένες περιόδους υψηλών δloc. Σε περιόδους χαμηλής υποβάθμισης των επιδόσεων, το σφάλμα εντοπισμού είναι κοντά στο 0, εκτός από μικρές διακυμάνσεις. Σε περιόδους υψηλής υποβάθμισης των επιδόσεων, το σφάλμα εντοπισμού αυξάνεται σταθερά. Για την ανίχνευση τέτοιων καταστάσεων, γίνεται η μέτρηση του αριθμού των περιόδων που το βήμα tevent = t | ˙δloc (“odom_posErr_roc”) | παίρνει συνεχώς μια μη μηδενική τιμή.

3.4 Robot Velocity (x)˙

Το συγκεκριμένο vital αφορά τις καταστάσεις υποβάθμισης της απόδοσης του ρομπότ που σχετίζονται με την ταχύτητα του. Η ταχύτητα ενός ρομπότ παραμένει σταθερή σε περιόδους ιδανικής συμπεριφοράς εκτός και αν πρέπει να στρίψει. Απότομη πτώση της ταχύτητας, μπορεί να προκληθεί από τα σφάλματα πλοήγησης, οι περιορισμοί του αλγορίθμου SLAM και τα προβλήματα υλικού. Αντίθετα,

δυσλειτουργίες του κινητήρα και προβλήματα πέδησης προκαλούν την επιτάχυνση ενός ρομπότ για μεγάλα χρονικά διαστήματα.

Η ταχύτητα του ρομπότ υπολογίζεται με τη διαφοροποίηση διαδοχικών εκτιμήσεων θέσης του ρομπότ με συγχώνευση EKF (“velocity_fromOdom”) . Η τιμή που μας δίνει την πιθανότητα να εμφανιστεί βλάβη βγαίνει από τη μέτρηση του χρονικού βήματος t (“velocity_event_count”) όπου το ρομπότ έχει μηδενική ταχύτητα ή ταχύτητα που υπερβαίνει το όριο φυσιολογικής ταχύτητας του ρομπότ (“velocity_event_count”). Στην έρευνα φαίνεται το ρομπότ να αντιμετωπίζει κατάσταση χαμηλής υποβάθμισης όταν το “velocity_event_count” είναι μεγαλύτερο από 4 δευτερόλεπτα.

3.5 Laser Scanner Noise Variance(σ2 noise)

Αυτό το vital αφορά καταστάσεις υποβάθμισης της απόδοσης του ρομπότ όπου το ρομπότ αδυνατεί να προηγηθεί, να αντιληφθεί ή να χαρτογραφήσεις τον χώρο που βρίσκεται λόγω του θορύβου στον σαρωτή laser που ανακριβείς αναπαραστάσεις του περιβάλλοντος ενός ρομπότ, αυξάνοντας έτσι την πιθανότητα συγκρούσεων, μη βέλτιστου σχεδιασμού διαδρομής και αποτυχίας του ρομπότ. Η συστοιχία μετρήσεων του σαρωτή λέιζερ αναδιατάσσεται αρχικά ως τετραγωνική εικόνα κλίμακας του γκρι. Χρησιμοποιείτε η διακύμανση του θορύβου, σ2 noise, της εικόνας ως εκτίμηση του συνολικού θορύβου του σαρωτή laser. Η τιμή σ2 noise υπολογίζεται στη συνέχεια με τη σύμπτυξη της εικόνας με μια μάσκα 3x3 και την εφαρμογή αθροισμάτων στο προκύπτοντα πίνακα (“psnr_laserScan__data”). Τα πειράματα της έρευνας έδειξαν πως το χαμηλό σ2 noise ≈ 0,7 δεν είχε σχεδόν καθόλου επίδραση στο σύστημα του ρομπότ όμως η πιθανότητα του ρομπότ να αποτύχει αυξάνεται όσο το σ2 noise αυξανόταν.

3.6 Robot Health

Η υγεία του ρομπότ είναι μια κλιμακωτή εκτίμηση της ικανότητας ενός ρομπότ να εκτελεί τα καθήκοντά του με τον βέλτιστο τρόπο χωρίς οι δυνατότητές του να επηρεάζονται από παράγοντες που μειώνουν την απόδοση, δηλαδή την ικανότητα του να εκτελεί της εργασίες τους. Για την τιμή του “robot_health” πρώτα υπολογίζεται η πιθανότητα του ρομπότ να υποφέρει το ρομπότ σε συνάρτηση με όλα τα vital και στη συνέχεια η εντροπία της πληροφορίας από την πιθανότητα αυτή για δέκα βήματα. Για τη διαισθητική σύνδεση της υψηλή και της χαμηλής “υγείας” με υψηλή και χαμηλή υποβάθμιση των επιδόσεων αντίστοιχα, χρησιμοποιούμε το προσθετικό αντίστροφο της εντροπίας πληροφοριών ως το ρομπότ υγεία του ρομπότ, όσο απομακρύνεται η τιμή από το 0 τόσο μεγαλύτερη και η υποβάθμιση των επιδόσεων.

Κεφάλαιο 4

4. Προετοιμασία

Στο παρακάτω κεφάλαιο θα γίνει αναφορά στην προετοιμασία που έγινε πριν τη διεξαγωγή του πειράματος. Αυτό αφορά τα δεδομένα που μας δόθηκαν, το περιβάλλον ανάπτυξης αλλά και οι απαραίτητες βιβλιοθήκες που χρησιμοποιήθηκαν για τη διεξαγωγή του πειράματος.

4.1 Περιβάλλον ανάπτυξης

4.1.1 Google Colabs

Το Colaboratory, ή εν συντομία "Colab", είναι ένα δωρεάν προϊόν της Google Research, το οποίο αφορά ένα περιβάλλον σημειωματάριου Jupyter που βασίζεται στο cloud και δεν απαιτεί καμία εγκατάσταση για να χρησιμοποιηθεί, ενώ παρέχει δωρεάν πρόσβαση σε υπολογιστικούς πόρους, μέσα στους οποίους συμπεριλαμβάνονται TPU και GPU. Μπορεί οποιοσδήποτε να συνδεθεί με τον λογαριασμό του στην Google, να γράφει και να εκτελεί αυθαίρετο κώδικα Python μέσω ενός προγράμματος περιήγησης.

Το Google Colab είναι ιδιαίτερα κατάλληλο για μηχανική μάθηση, ανάλυση δεδομένων και εκπαίδευση. Πέρα της σύνταξης και εκτέλεσης του κώδικα Python το Colab, να γίνεται διαμοιρασμός και ταυτόχρονη επεξεργασία του κώδικά με άλλους χρήστες, αποθήκευσης μοντέλων και notebooks στο cloud της Google. Είναι δυνατή η χρήση συνόλων δεδομένων από εξωτερικές πηγές. Έχει δυνατότητα ενσωμάτωσης με το GitHub. Το Colab έχει προεγκατεστημένες τις περισσότερες βιβλιοθήκες που χρησιμοποιούνται. Επιπλέον, υποστηρίζει την χρήση κειμένων, διαγραμμάτων, εικόνων, HTML και LaTeX για την σωστή τεκμηρίωση του υλικού μέσα σε ένα σημειωματάριο. Δεδομένου του ότι το Colab έχει βασιστεί στο Jupyter, είναι ικανό να επεξεργαστεί χωρίς κάποια μετατροπή αρχεία Jupyter.

Οι πόροι που παρέχει το Colab δεν είναι εγγυημένοι ούτε απεριόριστοι ενώ τα όριο χρήσης αυξομειώνονται. Αυτό είναι σημαντικό για να λειτουργεί δωρεάν το Colab. Το Colab μπορεί να παρέχει το Colab Pro επι πληρωμή που δεν έχει αυτά τα προβλήματα πόρων.
Το Google Colab επιλέχθηκε λόγω της παροχής καλύτερων υπολογιστικών πόρων από το τοπικό σύστημα για την εκπαίδευσης αλγορίθμων αλλά και την ανάλυση των δεδομένων.

4.1.2 Jupyter Notebooks

Το Jupyter Notebooks είναι ένα διαδραστικό και διαδικτυακό υπολογιστικό περιβάλλον που επιτρέπει στους χρήστες τη δημιουργία, τον διαμοιρασμό και την εκτέλεση κώδικα σε διάφορες γλώσσες προγραμματισμού, όπως η Python, η R και η Julia. Συνδυάζει την εκτέλεση κώδικα, επεξηγήσεις κειμένου, οπτικοποιήσεις και μαθηματικές εξισώσεις σε ένα ενιαίο σημειωματάριο, καθιστώντας το ιδανικό εργαλείο για την ανάλυση δεδομένων, στην έρευνα, στη διδασκαλία και την εξερεύνηση. Τα σημειωματάρια Jupyter προσφέρουν ένα διαδραστικό περιβάλλον όπου ο κώδικας μπορεί να εκτελεστεί σε μεμονωμένα κελιά, επιτρέποντας στους χρήστες να αναπτύσσουν και να δοκιμάζουν επαναληπτικά τον κώδικα, διατηρώντας παράλληλα πλούσια τεκμηρίωση. Αυτός ο διαδραστικός και ευέλικτος χαρακτήρας έχει καταστήσει τα Jupyter Notebooks μια ευρέως χρησιμοποιούμενη πλατφόρμα για την επιστήμη δεδομένων, τη μηχανική μάθηση και την επιστημονική έρευνα. Το Jupyter Notebooks χρησιμοποιήθηκε για εξαγωγή δεδομένων από τα αρχικά μη επεξεργασμένα δεδομένα και για τη δημιουργία διαδραστικών γραφημάτων μέσο της βιβλιοθήκης HoloViews όπου το Google Colabs αδυνατούσε να δημιουργήσει.

4.2 Βιβλιοθήκες

Η ανάπτυξη του πειράματος έγινε σε γλώσσα Python. Η Python προτιμάται ιδιαίτερα για έργα επιστήμης δεδομένων λόγω των πολύπλευρων πλεονεκτημάτων της. Το εκτεταμένο οικοσύστημα βιβλιοθηκών και εργαλείων της, παρέχει μια ολοκληρωμένη εργαλειοθήκη για χειρισμό δεδομένων, ανάλυση, οπτικοποίηση και μηχανική μάθηση. Η απλότητα και η αναγνωσιμότητα της Python αποτελούν βασικά πλεονεκτήματα, καθιστώντας την προσιτή τόσο σε αρχάριους όσο και σε έμπειρους προγραμματιστές. Επιπλέον, η ακμάζουσα κοινότητα της Python εξασφαλίζει συνεχή υποστήριξη, προάγει την ανταλλαγή γνώσεων και προσφέρει λύσεις σε διάφορες προκλήσεις της επιστήμης των δεδομένων. Η ευελιξία της Python επεκτείνεται στην επεκτασιμότητα, καθώς μπορεί να χειριστεί πειράματα μικρής κλίμακας καθώς και επεξεργασία και ανάλυση δεδομένων μεγάλης κλίμακας, καθιστώντας την ιδανική επιλογή για ένα ευρύ φάσμα έργων επιστήμης δεδομένων.

4.2.1 Pickle

Η βιβλιοθήκη pickle της Python αποτελεί ένα ισχυρό εργαλείο για τη σειριοποίηση και την απόσειριοποίηση αντικειμένων της Python, επιτρέποντάς τους να αποθηκεύονται ως δυαδικά δεδομένα στο δίσκο ή να μεταφέρονται μεταξύ διαφορετικών συστημάτων. Πήρε το όνομα της από το pickling στη μαγειρική όπου γίνεται η επεξεργασία των τροφίμων με σκοπό τη συντήρηση τους και την αποθήκευσης τους. Το pickling ενός αντικειμένου παράγει μια ροή bytes που μπορεί να αποθηκευτεί. Το unpickling μετατρέπει τη ροή των bytes πίσω στο αρχικό αντικείμενο. Με τη χρήση της βιβλιοθήκης pickle, σύνθετες δομές δεδομένων όπως λεξικά, λίστες και προσαρμοσμένα αντικείμενα μπορούν να αποθηκευτούν σε μη σειριακή μορφή,

διατηρώντας την κατάσταση και την ιεραρχία τους. Αυτό είναι ιδιαίτερα χρήσιμο κατά την εργασία με μοντέλα μηχανικής μάθησης, την προσωρινή αποθήκευση υπολογισμένων αποτελεσμάτων ή την κοινή χρήση δεδομένων μεταξύ διαφορετικών τμημάτων ενός προγράμματος. Τα αρχεία pickle είναι σε μην αναγνωρίσιμη μορφή για τον άνθρωπο και μπορούν να διαβαστούν μόνο από το σύστημα. Επιπλέον, είναι σημαντικό να σημειωθεί ότι η βιβλιοθήκη pickle ενέχει κάποιους κινδύνους ασφαλείας όταν φορτώνει δεδομένα από μη αξιόπιστες πηγές, καθώς μπορεί να εκτελέσει αυθαίρετο κώδικα κατά την αποσειριοποίηση. Επομένως, συνιστάται η χρήση της βιβλιοθήκης με προσοχή και μόνο σε αξιόπιστες πηγές δεδομένων.

4.2.2 Pandas

Η βιβλιοθήκη pandas είναι πολύ δυνατό και χρήσιμο εργαλείο όσων αφορά την ανάλυση και επεξεργασία δεδομένων παρέχοντας πλούσιες δομές δεδομένων και συναρτήσεις που έχουν σχεδιαστεί για να κάνουν την εργασία με δομημένα δεδομένα γρήγορη, εύκολη και εκφραστική. Η pandas λειτουργεί με 2 δομές δεδομένων, series και dataframes, καθιστώντας την αποτελεσματική σε εργασία με δομημένα δεδομένα. Παρέχει εξελιγμένη λειτουργικότητα ευρετηρίασης για να διευκολύνει την αναδιαμόρφωση, το τεμαχισμό, την εκτέλεση αθροίσεων και την επιλογή υποσυνόλων δεδομένων ενώ αντιμετωπίζει την πρόκληση του εύχρηστου χειρισμού ελλειπόντων δεδομένων, επιτρέποντας στους χρήστες να διαχειρίζονται αποτελεσματικά τα κενά κατά την ανάλυση. Επιπλέον, η βιβλιοθήκη pandas συνδυάζει των βιβλιοθηκών NumPy, για αριθμητικούς υπολογισμούς, και Matplotlib ,για οπτικοποίηση δεδομένων, καθιστώντας το έτσι ένα ολοκληρωμένο εργαλείο για ροές εργασίας δεδομένων.

4.2.3 NumPy

Η βιβλιοθήκη NumPy για την Python είναι ένα απαραίτητο εργαλείο για επιστημονικούς και αριθμητικούς υπολογισμούς. Παρέχει υποστήριξη για μεγάλους πίνακες και μήτρες, καθώς και για μαθηματικές συναρτήσεις πάνω σε αυτούς. Το NumPy είναι το θεμέλιο για πολλές άλλες επιστημονικές και αναλυτικές βιβλιοθήκες στην Python, με τη δυνατότητα αποδοτικής χρήσης μνήμης και δυνατότητες πραγματικού χρόνου. Η βασική δομή δεδομένων του NumPy είναι ο ndarray (n-διάστατος πίνακας), που είναι σχεδιασμένος για αποτελεσματικές μαθηματικές πράξεις. Αυτός ο πίνακας επιτρέπει γρήγορες πράξεις σε ολόκληρους πίνακες, σε αντίθεση με τις παραδοσιακές λίστες της Python. Επιπλέον, το NumPy περιλαμβάνει ποικίλες μαθηματικές συναρτήσεις για την επεξεργασία πινάκων, γραμμική άλγεβρα, στατιστική κλπ. Συνολικά, η ευελιξία και η απόδοση του καθιστούν το NumPy μια ανεκτίμητη βιβλιοθήκη για επιστήμονες δεδομένων, ερευνητές, μηχανικούς και οποιονδήποτε ασχολείται με αριθμητικά δεδομένα και υπολογισμούς στην Python.

4.2.4 SciKit-Learn

Η Scikit-learn, συχνά αναφερόμενη ως sklearn, είναι μια ισχυρή βιβλιοθήκη Python ανοικτού κώδικα για μηχανική μάθηση και επιστήμη δεδομένων. Αναπτύχθηκε από μια συνεργατική κοινότητα ερευνητών και έχει γίνει ακρογωνιαίος λίθος του οικοσυστήματος της Python λόγο του ολοκληρωμένου συνόλου εργαλείων και αλγορίθμων που έχουν σχεδιαστεί για την απλοποίηση και την ευελιξία των εργασιών μηχανικής μάθησης. Η Scikit-learn παρέχει υποστήριξη για διάφορες εργασίες μηχανικής μάθησης, όπως ταξινόμηση, παλινδρόμηση, ομαδοποίηση και μείωση διαστάσεων, καθιστώντας την πολύτιμη τόσο για αρχάριους όσο και για ειδικούς στον τομέα. Η φιλοσοφία σχεδιασμού της δίνει έμφαση στη φιλικότητα προς το χρήστη και τις επιδόσεις, προσφέροντας ένα συνεπές API για ένα ευρύ φάσμα αλγορίθμων μηχανικής μάθησης και ενσωμάτωση με δημοφιλείς βιβλιοθήκες όπως η NumPy και η SciPy. Η εκτεταμένη τεκμηρίωση, τα σεμινάρια και η πληθώρα διαδικτυακών πόρων του Scikit-learn συμβάλλουν στην προσβασιμότητά του και το έχουν καθιερώσει ως επιλογή για τους επαγγελματίες της μηχανικής μάθησης.
Η Scikit-learn προσφέρει πρόσθετες ενότητες και υποπακέτα για να φιλοξενήσει ένα ευρύ φάσμα εργασιών και τεχνικών μηχανικής μάθησης. Αυτές οι πρόσθετες ενότητες παρέχονται για να παρέχουν στους χρήστες ευελιξία, επεκτασιμότητα και εξειδικευμένα εργαλεία για διάφορες πτυχές της μηχανικής μάθησης και της ανάλυσης δεδομένων. Παρακάτω θα γίνει αναφορά στα υποπακέτα που χρησιμοποιήθηκαν στο πείραμα:

1. **Sklearn.ensable.**
   Το υποπακέτο "ensemble" περιέχει διάφορες μεθόδους μάθησης ensemble, οι οποίες είναι τεχνικές που συνδυάζουν πολλαπλά μοντέλα μηχανικής μάθησης για τη βελτίωση της προβλεπτικής απόδοσης και της γενίκευσης. Εντός της ενότητας "ensemble" του scikit-learn, υπάρχουν κλάσεις για δημοφιλείς τεχνικές ensemble, όπως Random Forests, Gradient Boosting, AdaBoost, Bagging και άλλες. Αυτές οι μέθοδοι ensemble χρησιμοποιούνται για εργασίες όπως η ταξινόμηση, η παλινδρόμηση και η ανίχνευση ανωμαλιών. Από το υποπακέτο "ensemble" χρησιμοποιήθηκε ο αλγόριθμος "Isolation Forest".
2. **Sklearn.neighbors.**
   Το υποπακέτο "neighbors" αποτελεί ένα βασικό κομμάτι της εργαλειοθήκης της Scikit-learn για τεχνικές μηχανικής μάθησης που βασισμένες στην θεωρία των πλησιέστερων γειτόνων. Προσφέρει μια σειρά αλγορίθμων και βοηθητικών προγραμμάτων για εργασίες επιβλεπόμενης και μη επιβλεπόμενης μάθησης που βασίζονται στην εγγύτητα των σημείων δεδομένων στο χώρο των χαρακτηριστικών. Ο πιο γνωστός αλγόριθμος στο πλαίσιο αυτής της ενότητας είναι ο αλγόριθμος k-κοντινότεροι γείτονες (KNN). Επιπλέον, το "sklearn.neighbors" παρέχει αποτελεσματικούς αλγορίθμους για την εύρεση πλησιέστερων γειτόνων και υποστηρίζει διάφορες μετρικές απόστασης, καθιστώντας το απαραίτητο πόρο για εργασίες που αφορούν την απόσταση των δεδομένων μεταξύ τους. Από το υποπακέτο "neighbors" χρησιμοποιήθηκε ο αλγόριθμος "Local Outlier Factor".

3. **Sklearn.SVM.**
   Το υποπακέτο "SVM" διαθέτει μια πληθώρα εργαλείων για την υλοποίηση των Μηχανών Διανυσμάτων Υποστήριξης (SVM) για εργασίες επιβλεπόμενης και μη επιβλεπόμενης μηχανικής μάθησης. Οι SVM είναι γνωστές για την αποτελεσματικότητά τους στην εύρεση βέλτιστων ορίων απόφασης, καθιστώντας τις πολύτιμα εργαλεία στη μηχανική μάθηση. Το υποπακέτο "sklearn.svm" της Scikit-learn περιλαμβάνει διάφορους ταξινομητές και παλινδρομείς βασισμένους σε SVM, όπως SVC (Support Vector Classification), SVR (Support Vector Regression) και OneClass-SVM, επιτρέποντας στους χρήστες να εφαρμόζουν τις SVM για ένα ευρύ φάσμα προβλημάτων. Από το υποπακέτο "SVM" χρησιμοποιήθηκε ο αλγόριθμος " OneClass-SVM".

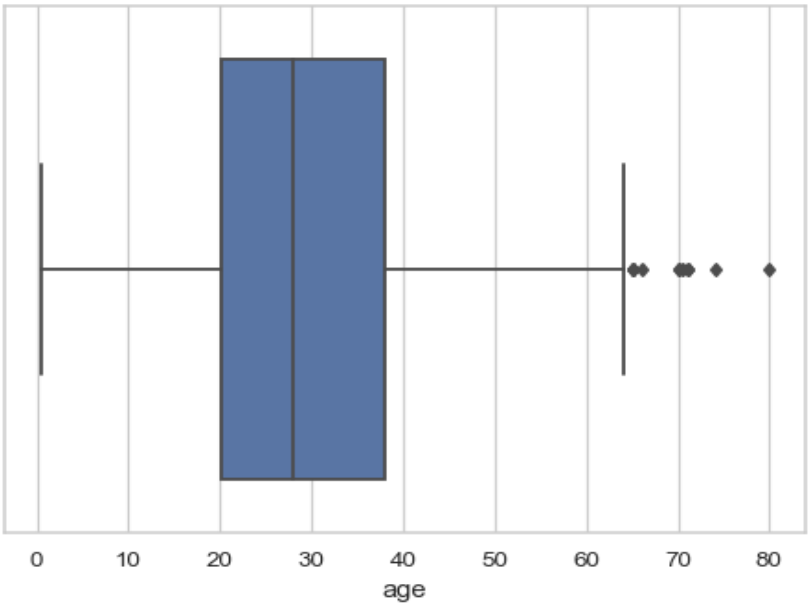
4.2.5 Matplotlib

Η Matplotlib αποτελεί απαραίτητη βιβλιοθήκη για την οπτικοποίηση των δεδομένων παρέχοντας ολοκληρωμένα λύσεις και ευρεία γκάμα εργαλείων για τη δημιουργία γραφικών παραστάσεων, διαγραμμάτων και γραφημάτων. Είναι συμβατή βιβλιοθήκες της Python, όπως η NumPy για χειρισμό δεδομένων και η SciPy για επιστημονικούς υπολογισμούς, πράγμα που την καθιστά χρήσιμη σε εργασίες που βασίζονται σε δεδομένα. Ένα μεγάλο θετικό της βιβλιοθήκης είναι η απλότητά της, επιτρέποντας ακόμα και σε αρχάριους χρήστες να δημιουργούν εύκολα κατατοπιστικές απεικονίσεις. Επιπλέον, υποστηρίζει διάφορους τύπους γραφικών παραστάσεων, όπως γραμμικά διαγράμματα, ραβδογράμματα, boxplots, διαγράμματα διασποράς, ιστογράμματα, heatmaps και άλλα, καθιστώντας την κατάλληλη για ένα ευρύ φάσμα αναγκών ανάλυσης και παρουσίασης δεδομένων. Επιπλέον, δίνει τη δυνατότητα προσαρμογής των γραφημάτων, ο χρήστης έχει τη δυνατότητα να προσθέτει τίτλους, ετικέτες, λεζάντες, σχόλια, αλλά επίσης να αλλάζει χρώματα, τύπους γραμμών, διαστάσεις, σχήματα δεικτών δημιουργώντας έτσι ένα μια ξεκάθαρη και ευανάγνωστη απεικόνιση. Η βιβλιοθήκη Matplotlib δίνει τη δυνατότητα στο χρήστη να αποθηκεύσει τα γραφήματα σε διάφορες μορφές αρχείων όπως png, pdf και svg ή να τις εμφανίζει διαδραστικά σε γραφικές διεπαφές, λόγο της υποστήριξής της σε πολλαπλά backends. Τέλος, η πλήρης βιβλιογραφία της είναι διαθέσιμη online όπου περιλαμβάνει λεπτομερή σεμινάρια, παραδείγματα και μια γκαλερί που παρουσιάζει ένα ευρύ φάσμα οπτικοποιήσεων.

4.2.6 Seaborn

Η βιβλιοθήκη Seaborn είναι ένα εργαλείο οπτικοποίησης δεδομένων σε Python που βασίζεται στη βιβλιοθήκη Matplotlib και έχει σχεδιαστεί ειδικά για τη δημιουργία οπτικά ελκυστικών και κατατοπιστικών γραφικών. Παρέχει προκαθορισμένα θέματα, στυλ και παλέτες χρωμάτων για εύκολη δημιουργία σύνθετων οπτικοποιήσεων. Μπορεί να χρησιμοποιηθεί για στατιστικά γραφήματα όπως διασποράς, ράβδων, ιστογράμματα και heatmaps, καθώς και εξειδικευμένους τύπους γραφημάτων, όπως

γραφήματα βιολιού, γραφήματα πλαισίου και γραφήματα ζεύγους, τα οποία είναι χρήσιμα για την οπτικοποίηση κατανομών, τη σύγκριση πολλαπλών μεταβλητών και τον εντοπισμό συσχετίσεων στα δεδομένα Ένα από τα δυνατά σημεία της Seaborn είναι η ικανότητά του να συνεργάζεται με τη βιβλιοθήκη Pandas, επιτρέποντας την απευθείας οπτικοποίηση δεδομένων χωρίς πολλαπλές επεξεργασίες. Επίσης, παρέχει υποστήριξη για στατιστική εκτίμηση και οπτικοποίηση χρησιμοποιώντας διαγράμματα παλινδρόμησης και εκτίμησης πυκνότητας πυρήνα. Η Seaborn είναι ένα πολύτιμο εργαλείο για επιστήμονες δεδομένων, αναλυτές και ερευνητές για τη δημιουργία ελκυστικών και κατατοπιστικών οπτικοποιήσεων δεδομένων.



Εικόνα 13: BoxPlot σε περιβάλλον Seaborn

4.2.9 Scipy stats

Το Scipy stats είναι ένα υποπακέτο της βιβλιοθήκης Scipy, μιας ολοκληρωμένης βιβλιοθήκης Python για επιστημονικούς και τεχνικούς υπολογισμούς. Το Scipy.stats προσφέρει ένα ευρύ φάσμα στατιστικών συναρτήσεων και εργαλείων, καθιστώντας το μια πηγή για ερευνητές, επιστήμονες δεδομένων και στατιστικολόγους. Το υποπακέτο αυτό χρησιμοποιήθηκε για την εκμετάλλευση της συνάρτησης gaussian_kde, η οποία σημαίνει Gaussian Kernel Density Estimation. Η Gaussian KDE είναι μια ισχυρή στατιστική τεχνική που χρησιμοποιείται για την εκτίμηση συναρτήσεων πυκνότητας πιθανότητας από δεδομένα. Χρησιμοποιεί γκαουσιανούς (κανονικούς) πυρήνες με κέντρο κάθε σημείο δεδομένων για να παρέχει μια ομαλή, συνεχή αναπαράσταση της υποκείμενης κατανομής δεδομένων. Η λειτουργικότητα

Gaussian KDE του Scipy είναι ανεκτίμητη για τη διερευνητική ανάλυση δεδομένων, την οπτικοποίηση και την κατανόηση των χαρακτηριστικών των δεδομένων. Χρησιμοποιείται ευρέως σε τομείς όπως η μηχανική μάθηση, η ανάλυση δεδομένων και η στατιστική για τη μοντελοποίηση και την οπτικοποίηση κατανομών δεδομένων με μη παραμετρικό τρόπο.

4.3 Τα δεδομένα

Τα δεδομένα του πειράματος αφορούν τις μετρήσεις του αισθητήρα laser του “Clearpath Husky Robot” κατά την πλοήγηση του. Το σύνολο των δεδομένων αποτελείται από τις μετρήσεις των παραμέτρων του robot και τις τιμές των vitals για τις ανάγκες της έρευνας του Aniketh Ramesh αυτή όπως αυτή αναφέρθηκε στο κεφάλαιο 3. Τα δεδομένα λήφθηκαν σε μορφή pickle και είναι σε δύο διαφορετικές εκδόσεις. Η μία αφορά τα raw δεδομένα όπως τα κατέγραψαν οι αισθητήρες του ρομπότ ενώ η άλλη έκδοση τα δεδομένα είναι σε πιο απλοποιημένη μορφή και επεξεργασμένα από τον πάροχο τους.

4.3.1 Raw Data

Τα raw δεδομένα αποτελούνται από 180 στήλες και κατά μέσο όρο 6000 χιλιάδες γραμμές. Η κάθε γραμμή αποτελεί τη μέτρηση για κάθε εκατοστό του δευτερολέπτου που λαμβάνουν οι αισθητήρες που ρομπότ για τις μετρήσεις στο περιβάλλον του, οπότε ο όγκος τού κάθε dataset εξαρτάται από την ώρα χρειάστηκε για να ολοκληρωθεί το εκάστοτε πείραμα. Η ονομασία των φακέλων έγινε με βάση περιβάλλον κατά τη διάρκεια διεξαγωγής πειράματος. Για παράδειγμα, το αρχείο pickle με όνομα “real_roughTerrainLaserNoise_4” αναφέρετε στο τέταρτο πείραμα πραγματικών συνθηκών, όπου υπάρχει δύσβατο έδαφος (roughTerrain) και εφαρμόζεται laser noise. Τα πειράματα χωρίζονται σε 3 διαφορετικά επίπεδα, το επίπεδο 1 δεν εφαρμόζεται laser noise ενώ η πλοήγηση γίνεται σε ομαλό έδαφος, στο επίπεδο 2, δεν εφαρμόζεται laser noise και η πλοήγηση γίνεται σε ανώμαλο έδαφος, ενώ στο επίπεδο 3 εφαρμόζεται laser noise και η πλοήγηση γίνεται σε ανώμαλο έδαφος. Από τα raw δεδομένα μπορεί να γίνει απευθείας εξαγωγή του psnr_laser_scan_data, στην οποία μέτρηση θα εφαρμοστούν οι τεχνικές εύρεσης ανωμαλιών, όμως για διερευνητικούς λόγους χρειάζεται η εξαγωγή των τιμών από τα vitals.

4.3.2 Simplified Data

Τα απλοποιημένα δεδομένα αποτελούνται από τα raw data των αισθητήρων που εκμεταλλεύτηκε ο συγγραφέας της έρευνας για την εξαγωγή των vitals, τα ίδια τα vitals, τις μετρήσεις “pf_total” και “robot_health”, της πληροφορίες διεξαγωγής πειράματος (δηλαδή αν υπάρχει ή όχι laser noise, πιο κατά σειρά πείραμα είναι κλπ), καθώς και όλες τις ενδιάμεσες τιμές από τα μαθηματικά μοντέλα που χρειάστηκαν

για την εξαγωγή τους. Ο όγκος τους το 1/100, δηλαδή μία μέτρηση ανά δευτερόλεπτο. Ο λόγος που ονομάζονται απλοποιημένα είναι διότι υπάρχουν μέσα σε αυτά οι μετρήσεις ανά δευτερόλεπτο μόνο από τους αισθητήρες οι οποίοι έχουν εκμεταλλευτεί για τη δημιουργία των μετρητών απόδοσης Robot Vitals και όχι όλες οι μετρήσεις από κάθε αισθητήρα στο ρομπότ. Η ονομασία κάθε dataset είναι ίδια με τα raw με μόνη διαφορά τη λέξη simplified στο τέλος, π.χ. real_roughTerrainLaserNoise_4_simplified.

Η ιδέα στη συνέχεια ήταν να ενωθούν όλα τα πειράματα σε ένα συνεχόμενο πείραμα ώστε να τροφοδοτηθούν στους αλγορίθμους μάθησης ως μία χρονοσειρά, διότι ήταν ένα επαναλαμβανόμενο πείραμα οπότε συγκεντρωτικά οι λίγες ανωμαλίες θα ήταν ακόμα πιο έυκολο εμφανείς.

Κεφάλαιο 5

5. Πείραμα

Το πείραμα αφορά την ανίχνευση καινοτομιών από δεδομένα που έχουν εξαχθεί μέσω δοκιμών στο μη επανδρωμένο όχημα Clearpath Husky Robot. Θα γίνουν αναφορές στα παρακάτω κεφάλαια για το περιβάλλον ανάπτυξης του αλγορίθμου για την ανίχνευση ανωμαλιών, των δεδομένων και της ανάλυσης τους και τέλος θα γίνει ανάλυση τις εξαγωγής μοντέλων που προέκυψαν το πείραμα.

5.1 Εξαγωγή δεδομένων και δημιουργία dataset

Το θέμα που υπήρχε με τα δεδομένα ήταν πως χρειαζόμασταν τα δεδομένα που υπήρχαν στα simplified αρχεία pickle, δηλαδή τα δεδομένα από τα raw data των αισθητήρων που εκμεταλλεύτηκε ο συγγραφέας της έρευνας για την εξαγωγή των vitals, τα ίδια τα vitals, τις μετρήσεις “pf_total” και “robot_health”, καθώς και όλες τις ενδιάμεσες τιμές από τα μαθηματικά μοντέλα που χρειάστηκαν για την εξαγωγή τους αλλά στον όγκο των raw data, δηλαδή μία μέτρητη ανα εκατοστό του δευτερολέπτου.

Για να γίνει αυτό χρησιμοποιήθηκε ο κώδικας με τα μαθηματικά μοντέλα για την εξαγωγή των simplified που έχει αναρτήσει ο Aniketh στο Github (https://github.com/anikethramesh/robotVitals/blob/main/DataAnalysis/RV_RealExp.ipynb). Με βάση αυτών τον κώδικα έγινε αλλαγή στο ρυθμό δειγματοληψίας (sampling rate) από 100 σε 1 ώστε να γίνεται η επεξεργασία για κάθε γραμμή των δεδομένων και έτσι δημιουργήθηκαν τα simplified data με τον απαραίτητο όγκο δεδομένων για την εκπαίδευσή του αλγορίθμου.

Η ιδέα στη συνέχεια ήταν να ενωθούν όλα τα πειράματα σε ένα συνεχόμενο πείραμα ώστε να τροφοδοτηθούν στους αλγορίθμους μάθησης ως μία χρονοσειρά, διότι ήταν ένα επαναλαμβανόμενο πείραμα οπότε συγκεντρωτικά οι λίγες ανωμαλίες θα ήταν ακόμα πιο εύκολο εμφανείς. Αρχικά, εφόσον η μορφή του dataframe θα έπρεπε να είναι ίδια με τα simplified data, δημιουργήθηκε ένα κενό pandas dataframe με όλες τις στήλες που περιέχονται σε αυτά. Στη συνέχεια με 2 επαναλήψεις η μία εμφωλευμένη στην άλλη, λαμβάνονται τα δεδομένα από όλα τα πειράματα με σειρά σε ένα καινούργιο pandas dataframe από τα οποία αφαιρείται το index, και ενώνονται χρησιμοποιώντας μέθοδο concat με το pandas dataframe που δημιουργήθηκε με τα columns απο τα simplified data. Τέλος, αποθηκεύονται στο σύστημα ώστε αρχείο pickle.

5.2 Εξερεύνηση δεδομένων

Το συνολικό dataset με όλα τα πειράματα αποτελείται από 29 στήλες και 294215 γραμμές. Αφού το γίνει το unpickling, γίνεται μετατροπή σε pandas dataframe για την

εξερεύνηση των δεδομένων. Για τις ανάγκες του πειράματος θα χρειαστούν οι στήλες:

- **“psnr_laserScan__data”**, οι τιμές που λαμβάνει ο αισθητήρας laser από το περιβάλλον. Στην τιμή αυτή θα εφαρμοστεί το μοντέλο ανεύρεσης καινοτομιών. Οι τιμές κυμαίνονται μεταξύ ο και 0.932440, έχει μέση τιμή 0.368277 και είναι τύπου float64 ενώ δεν έχει καμία κενή τιμή.
- **“Difficulty”**, οι τιμές αυτές είναι η πληροφορία για το περιβάλλον που κινείται το ρομπότ. Αυτή η τιμή χρειάζεται ώστε να διαχωριστούν τα δεδομένα σε normal και anomalous ώστε να μελετηθούν και να τροφοδοτηθούν στους αλγορίθμους. Είναι τύπου object και έχει 3ις συνολικά διαφορετικές τιμές, “obstacles”, “roughTerrain” και “roughTerrainLaserNoise” ενώ δεν έχει καμία κενή τιμή.
- **“rv_snr_event”**, αποτελεί το vital που βγαίνει από τις τιμές του αισθητήρα laser. Χρησιμοποιήθηκε για οπτικοποίηση και μελέτη του dataset, επιπλέον είναι ενδιαφέρον να δειχθεί πως οι ανωμαλίες στον αισθητήρα επηρεάζουν το vital. Οι τιμές κυμαίνονται μεταξύ 0 και 0.416344, έχει μέση τιμή 0.049804 και είναι τύπου float64 ενώ δεν έχει καμία κενή τιμή.
- **“robot_health”**, αποτελεί μια κλιμακωτή εκτίμηση της ικανότητας ενός ρομπότ να εκτελεί τα καθήκοντά του. Εφόσον το “robot_health” αποτελεί παράγωγό και συνολικό δείκτη όλων των vitals είναι ενδιαφέρον να δειχθεί πως οι ανωμαλίες στον αισθητήρα επηρεάζουν το vital. Οι τιμές κυμαίνονται μεταξύ ο και 0.932440, έχει μέση τιμή 0.368277 και είναι τύπου float64 ενώ έχει 378 κενές τιμές οι οποίες είναι οι 9 αρχικές από κάθε dataset.

```
[ ]  df1.describe()
```

	psnr_laserScan__data	rv_snr_event	robot_health
count	294215.000000	294215.000000	293837.000000
mean	0.368277	0.049804	-1.481057
std	0.116486	0.056949	0.231217
min	0.000000	0.006693	-1.597618
25%	0.312799	0.031189	-1.589644
50%	0.352421	0.037764	-1.582531
75%	0.404256	0.048397	-1.561927
max	0.932440	0.416344	-0.558095

```
[ ]  df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 294215 entries, 0 to 294214
Data columns (total 4 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   psnr_laserScan__data  294215 non-null  float64
 1   difficulty           294215 non-null  object 
 2   rv_snr_event         294215 non-null  float64
 3   robot_health         293837 non-null  float64
dtypes: float64(3), object(1)
memory usage: 9.0+ MB
```

Παρακάτω φαίνονται οι πρώτες 5 σειρές του dataframe καθώς και οι 5 τελευταίες.

```
df1 = df_main[['psnr_laserScan__data', 'difficulty','rv_snr_event','robot_health']]
```

```
[ ] df1.head()
```

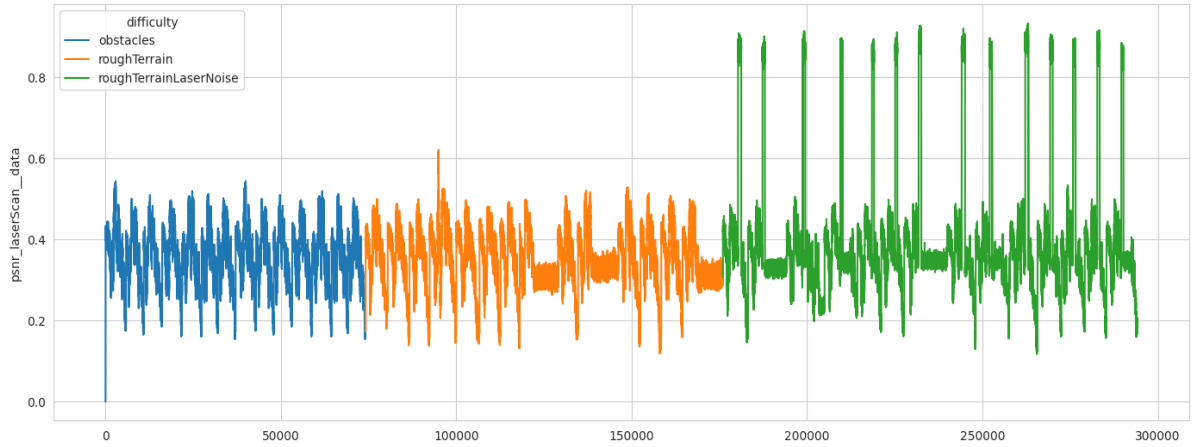
	psnr_laserScan__data	difficulty	rv_snr_event	robot_health
0	0.0	obstacles	0.006693	NaN
1	0.0	obstacles	0.006693	NaN
2	0.0	obstacles	0.006693	NaN
3	0.0	obstacles	0.006693	NaN
4	0.0	obstacles	0.006693	NaN

```
[ ] df1.tail()
```

	psnr_laserScan__data	difficulty	rv_snr_event	robot_health
294210	0.197808	roughTerrainLaserNoise	0.017794	-1.542372
294211	0.197808	roughTerrainLaserNoise	0.017794	-1.534890
294212	0.197808	roughTerrainLaserNoise	0.017794	-1.527407
294213	0.197808	roughTerrainLaserNoise	0.017794	-1.519925
294214	0.197808	roughTerrainLaserNoise	0.017794	-1.512442

5.3 Οπτικοποίηση δεδομένων

Για το πρώτο γράφημα επιλέχθηκε από τη βιβλιοθήκη seaborn ένα line plot με ενεργοποιημένο whitegrid ώστε να εμφανίζονται οι κάθετες και παράλληλες γραμμές στις τιμές κάνοντας τη μελέτη του διαγράμματος πιο εύκολη και απεικονίζει την εξέλιξη της τιμής “psnr_laserScan__data” κατά τη διάρκεια των πειραμάτων χωρισμένο στις καταστάσεις του “difficulty”, οπότε για δεδομένα θέσαμε τη στήλη “psnr_laserScan__data” και δόθηκε ως hue η στήλη “difficulty” για την χρωματική διαφορά της παραμέτρου αυτής. Από αυτό το γράφημα λαμβάνεται η πληροφορία του πως αλλάζει η τιμή σχετικά με κάθε κατάσταση ώστε να εμφανίσουμε τις διαφορές μεταξύ τους.

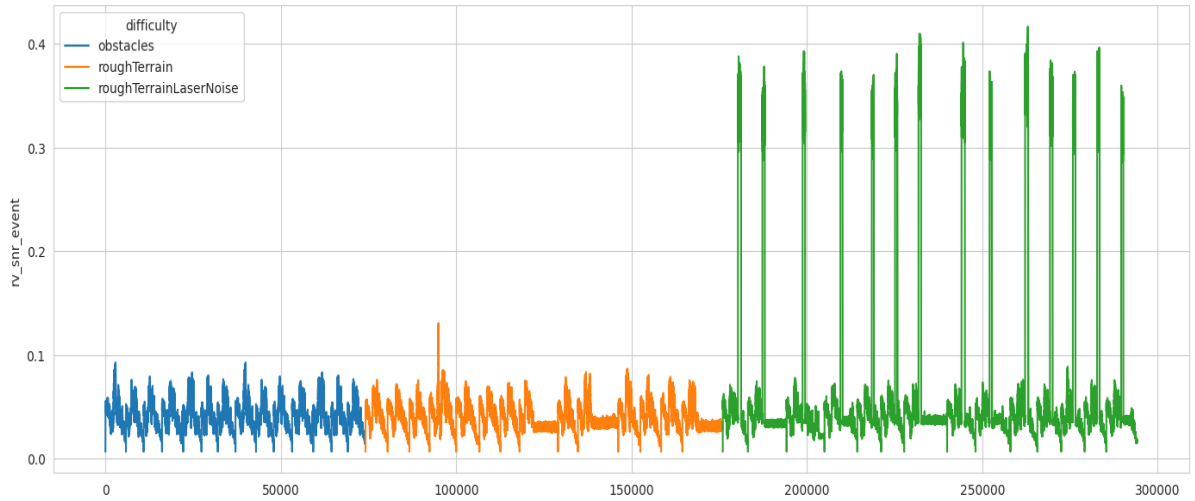


Σχήμα 1: Η εξέλιξη της τιμής psnr_laserScan__data

Στο παραπάνω γράφημα φαίνεται με πράσινο χρώμα η περιοχή που εφαρμόζεται laser noise αλλά και υπάρχει ανώμαλο έδαφος, ενώ αντίστοιχα με πορτοκαλί η περιοχή που υπάρχει ανώμαλο έδαφος και μπλε η περιοχή που τίποτα από τα 2 δεν εφαρμόζεται. Στην μπλε περιοχή παρατηρούμε πως ο αισθητήρας λειτουργεί αλλά λαμβάνει χαμηλές τιμές που κυμαίνονται μεταξύ 0,18 και 0,5, επιπλέον παρατηρούμε πως σε φυσιολογικό περιβάλλον οι τιμές δεν είναι ποτέ μηδέν ενώ εμφανίζουν ένα επαναλαμβανόμενο μοτίβο, εφόσον η εξέλιξη της τιμής είναι όλα τα πειράματα ενωμένα τότε σε κάθε πείραμα οι τιμές μοιάζουν μεταξύ τους ενώ ο αισθητήρας δείχνει να επηρεάζεται και να υπάρχει μια διακύμανση κατά τη διάρκεια του κάθε πειράματος. Δεδομένου σε αυτή τη φάση του πειράματος του ρομπότ δεν εφαρμόζεται κάποιο laser noise θεωρούμε τις τιμές αυτές του αισθητήρα ως φυσιολογικές.

Η πορτοκαλί περιοχή είναι παρόμοια με την μπλε, πράγμα αναμενόμενο αφού και εδώ σύμφωνα με το πρακτικό κομμάτι της έρευνας ούτε εδώ εφαρμόζεται laser noise, όμως διακρίνεται μια αλλαγή στο μοτίβο κατά τα τέλος τριών πειραμάτων, τα πειράματα αυτά αφορούσαν περιπτώσεις που το ρομπότ αντιμετώπιζε δυσκολία προς την ολοκλήρωση του στόχου, άρα μπορούμε να συμπεράνουμε πως και τα υπόλοιπα vitals επηρεάζουν έστω και σε μικρό βαθμό τις μετρήσεις του αισθητήρα, τέλος εμφανίζεται μια κορυφή που ξεπερνάει το 0,6. Εφόσον ούτε εδώ εφαρμόζεται laser noise θα θεωρήσουμε και αυτό τμήμα τιμών φυσιολογικό

Στην πράσινη περιοχή που γίνεται εμφανές το πότε εφαρμόζεται το laser noise. Οι τιμές κατά τη διάρκεια εφαρμογής του laser noise ξεπερνούν το 0.8 ενώ παραμένουν στα φυσιολογικά όρια όλες τις φορές που δεν υπάρχει εφαρμογή του laser noise. Επιπλέον, και εδώ παρατηρούμε πως το μοτίβο σπάει όταν το ρομπότ αντιμετωπίζει δυσκολία προς την ολοκλήρωση του στόχου. Η πράσινη περιοχή θα χρησιμοποιηθεί ως περιοχή με ανωμαλίες καθώς είναι η περιοχή που εφαρμόζεται το laser noise.



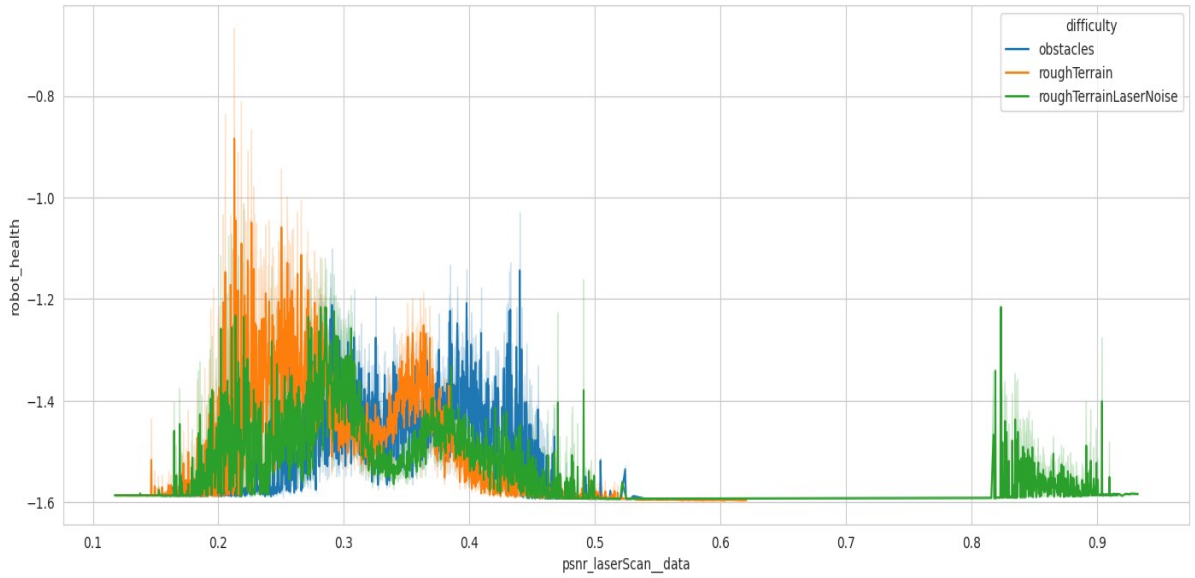
Σχήμα 2: Η εξέλιξη της τιμής του rv_snr_event

Στο παραπάνω σχήμα φαίνεται η εξέλιξη της τιμής του rv_snr_event. Διακρίνεται ότι το vital είναι ίδιο με την πηγαία τιμή “psnr_laserScan__data”, το οποίο ήταν αναμενόμενο σύμφωνα με την εξίσωση που χρησιμοποιήθηκε για την εξαγωγή του vital:

$$P\left(suffering\left|\sigma^2_{noise}\right|\right)=\frac{1}{1+\exp\left(\left(-a\cdot t_{event}+a\cdot b\right)\right)}$$

Όπου α = 5 και b = 1 και σ^2noise = “psnr_laserScan__data”. Η διαφορά είναι ότι το vital έχει αναλογικά μεγαλύτερη διακύμανση από την πηγαία τιμή με αποτέλεσμα να είναι πιο έντονες οι ακραίες τιμές. Αυτό ίσως να φανεί χρήσιμο στην εύρεση ανωμαλιών, αλλά για την πρώτη φάση της δημιουργίας ενός τέτοιου μοντέλου θα χρησιμοποιηθεί η πηγαία τιμή.

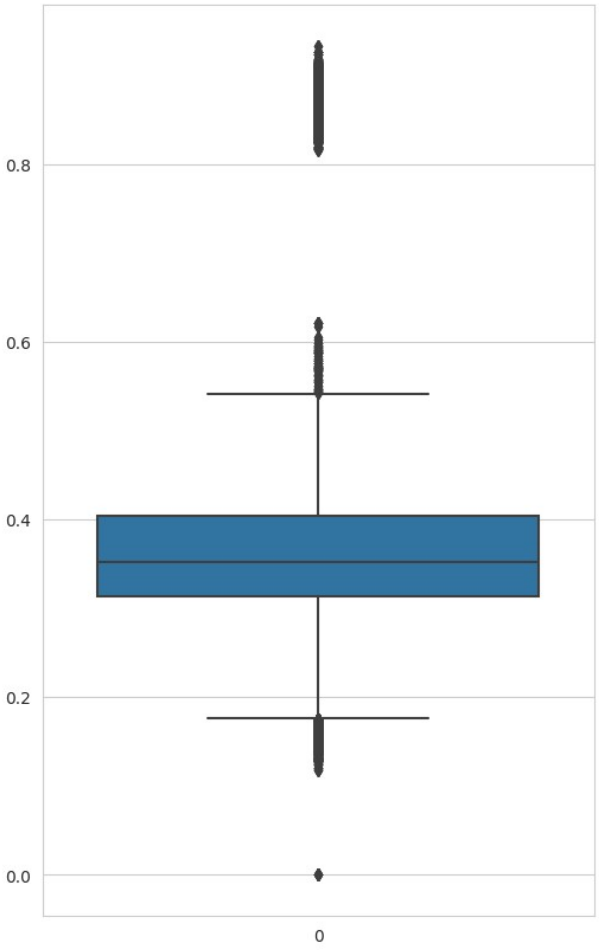
Στο παρακάτω σχήμα, το οποίο είναι πάλι ένα line plot που δημιουργήθηκε με τη βιβλιοθήκη seaborn, βλέπουμε τη σχέση τιμής μεταξύ “psnr_laserScan__data” και “robot_health” για κάθε κατάσταση του “difficulty”, από το οποίο θέλουμε να δούμε αν επηρεάζεται η τιμή του “robo_health” κατά τη διάρκεια εφαρμογής του laser noise.



Σχήμα 3: psnr_laserScan__Data σε συνάρτηση με το robot_health

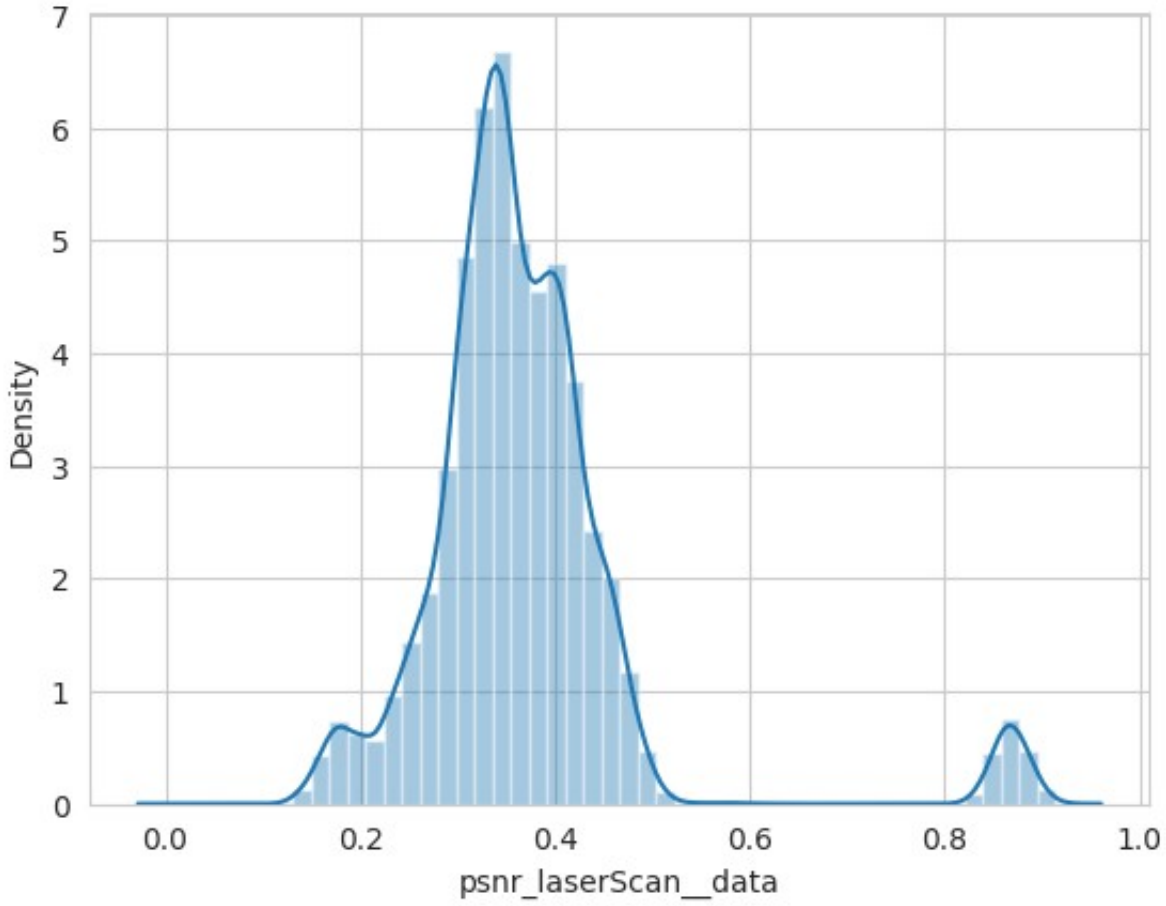
Είναι εμφανές ότι οι ακραίες τιμές στην περιοχή που θεωρούμε ότι υπάρχουν ανωμαλίες, την πράσινη, έχουν επιρροή στο “robot_health”. Δεδομένου πως όσο απομακρύνεται η τιμή του “robot_health” από το 0 τόσο μεγαλύτερη η αδυναμία του robot να ολοκληρώσει τον στόχο του. Φαίνεται πως όσο οι τιμές είναι στα φυσιολογικά όρια η τιμή του “robot_health” κυμαίνεται σε επίπεδα που είναι πιο κοντά στο 0. Ειδικά στην πορτοκαλί περιοχή που το robot δεν έχει να αντιμετωπίσει καμία δυσκολία στο περιβάλλον του. Αντίθετος λίγο πιο μακριά από το 0 φαίνονται οι άλλες 2 καταστάσεις του “difficulty”, πράγμα αναμενόμενο καθώς το robot αντιμετωπίζει δύσβατο έδαφος και laser noise οπότε αυξάνεται η πιθανότητα να υπάρχει δυσκολία επίτευξης στόχου. Αυτό που είναι πιο εμφανές βέβαια είναι η επιρροή του “psnr_laserScan__data” στις υψηλές τιμές. Στις τιμές άνω του 0.8 φαίνεται ότι το “robot_health” απομακρύνεται από το 0, οπότε βγαίνει το συμπέρασμα ότι η εφαρμογή laser noise δημιουργεί πρόβλημα στην ομαλή λειτουργία του ρομπότ.

Στη συνέχεια, από τη βιβλιοθήκη seaborn, θα δημιουργήσουμε ένα boxplot με ενεργοποιημένο το whitegrid για την εμφάνιση οριζόντιων γραμμών για το χαρακτηριστικό “psnr_laserScan__data”. Η επιλογή του boxplot έγινε διότι με ένα σχήμα μπορεί να δώσει αρκετές πληροφορίες σχετικά με το κάθε χαρακτηριστικό.



Σχήμα 4: BoxPlot για το psnr_laserScan__data

Το σχήμα 1 μας δίνει ήδη τις πληροφορίες της διαμέσου και των τεταρτημορίων, όμως μας δίνει μια πληροφορία που δεν υπάρχει στο σχήμα 1. Από το boxplot μπορούμε να διακρίνουμε το στατιστικό όριο στο οποίο ξεκινάνε οι ακραίες τιμές, ενώ στο σχήμα 1 βλέπαμε μόνο τη μέγιστη και ελάχιστη τιμή. Παρατηρούμε λοιπόν ότι υπάρχουν ακραίες τιμές και στις ελάχιστες τιμές, κάτω από το 1.8, ενώ στις άνω ακραίες τιμές φαίνονται οι τιμές που εφαρμόζεται το laser noise οι οποίες είναι όλες πάνω από 0.8 όμως φαίνονται σαν ακραίες τιμές και κάποιες τιμές οι οποίες δεν οφείλονται στο laser noise οι οποίες ξεπερνούν το 0.5, θεωρούμε ότι αυτές οι τιμές μπορεί να είναι θόρυβος στο περιβάλλον ανεξαρτήτος του αν εφαρμόστηκε χειροκίνητα laser noise.



Σχήμα 5: DistPlot για το psnr_laserScan__data

Τέλος, από τη βιβλιοθήκη seaborn θα χρησιμοποιήσουμε ένα distplot, για να δημιουργήσουμε ένα διάγραμμα πυκνότητας του “psnr_laserScan__data” και να παρατηρήσουμε την κατανομή της τιμής. Η τιμή ακολουθεί μια σχεδόν κανονική (γκαουσιανή) κατανομή.

Η γκαουσιανή κατανομή, που ονομάζεται επίσης καμπύλη καμπάνας, μας βοηθά να κατανοήσουμε τα δεδομένα. Μας λέει πού βρίσκονται τα περισσότερα σημεία δεδομένων και πόσο διασκορπισμένα είναι. Η κορυφή δείχνει που είναι ο μέσος όρος και οι πλευρές δείχνουν πού μπορεί να βρίσκονται τα δεδομένα. Αν λοιπόν δεν κάνει κορυφή αλλά πλατειάζει, τα δεδομένα είναι διασκορπισμένα, αν τα δεδομένα δημιουργούν ένα μυτερό λόφο, τα δεδομένα είναι κοντά στο μέσο όρο. Αυτό μας βοηθά να βλέπουμε μοτίβα και να λαμβάνουμε αποφάσεις με βάση τα δεδομένα.

Στο δικό μας σχήμα διακρίνεται εύκολα ότι υπάρχουν τιμές που βρίσκονται εκτός της κατανομής, όπως ο λοφίσκος στις τιμές άνω του 0.8 όμως επίσης παρατηρούμε ένα σημείο μεταξύ 1,5 και 0.2 όπου σπάει η κατανομή αλλά και μία κορυφή στο 0.4. Οι τιμές στο 0.4 θεωρούμε ότι είναι η στιγμές που έσπαγε το μοτίβο. Όμως για την πρώτη φάση του πειράματος θα προσπαθήσουμε να κάνουμε ανίχνευση των ακραίων τιμών. Το distplot θα βοηθήσει στην επιλογή του αλγόριθμου καθώς υπάρχουν αλγόριθμοι που λειτουργούν πιο αποδοτικά με δεδομένα που ακολουθούν

γκασουιανή κατανομή.

5.4 Διαχωρισμός δεδομένων

Στη συνέχεια προχωράμε στον διαχωρισμό των δεδομένων σε δεδομένα εκπαίδευσης και δεδομένα ελέγχου. Έχουμε επιλέξει ότι θα γίνει έλεγχος των ακραίων τιμών του dataset οπότε και ο διαχωρισμός θα γίνει καθαρά και μόνο με βάση το “difficulty” και δε θα λάβουμε υπόψη τυχόν αποκλίσεις από το μοτίβο που ακολουθεί η εξέλιξη της τιμής του “psnr_laserScan__data”. Σκοπός είναι να εκπαιδευτούν οι αλγόριθμοι σε καταστάσεις φυσιολογικές και χωρίς πολλές ακραίες τιμές, όπως υπάρχουν όταν εφαρμόζεται laser noise ώστε να δημιουργηθεί ένα μοντέλο κατάλληλο για ανίχνευση καινοτομιών. Για να γίνει αυτό χρησιμοποιήθηκε η groupby για τη δημιουργία ενός αντικειμένου με το όνομα groups όπως σε αυτό αποθηκεύτηκαν οι ξεχωριστές τιμές που υπάρχουν στο “difficulty”.

Στη συνέχεια χρησιμοποιείται η get_group για να γίνει η εξαγωγή των δεδομένων για το group “roughTerrainLaserNoise”, τα οποία αναθέτονται στο καινούργιο dataframe με το όνομα df_anom που πλέον περιέχει τα δεδομένα που εφαρμόζεται laser noise και θα χρησιμοποιηθεί για τον έλεγχο της ανίχνευσης καινοτομιών.

Για τη δημιουργία του dataframe εκπαίδευσης χρησιμοποιείται η ίδια τεχνική με τη διαφορά πως αναθέτουμε σε καινούργιο dataframe με το όνομα df τα δεδομένα από τα υπόλοιπα 2 group “obstacles” και “roughTerrain” τα οποία θα χρησιμοποιηθούν για την εκπαίδευση του αλγορίθμου. Με αυτόν τον τρόπο δημιουργήθηκαν 2 διαφορετικά dataset, το df με 175979 εγγραφές και το df_Anom με 118236 εγγραφές.

```
df.head()
```

	psnr_laserScan__data	difficulty	rv_snr_event	robot_health
0	0.0	obstacles	0.006693	NaN
1	0.0	obstacles	0.006693	NaN
2	0.0	obstacles	0.006693	NaN
3	0.0	obstacles	0.006693	NaN
4	0.0	obstacles	0.006693	NaN

```
[ ] df.tail()
```

	psnr_laserScan__data	difficulty	rv_snr_event	robot_health
175974	0.311608	roughTerrain	0.031009	-1.545981
175975	0.311608	roughTerrain	0.031009	-1.538860
175976	0.297904	roughTerrain	0.029016	-1.531641
175977	0.307437	roughTerrain	0.030389	-1.524493
175978	0.307437	roughTerrain	0.030389	-1.517345

Σχήμα 6: Οι πρώτες 5 και οι τελευταίες 5 γραμμές του df

```
df_Anom.head()
```

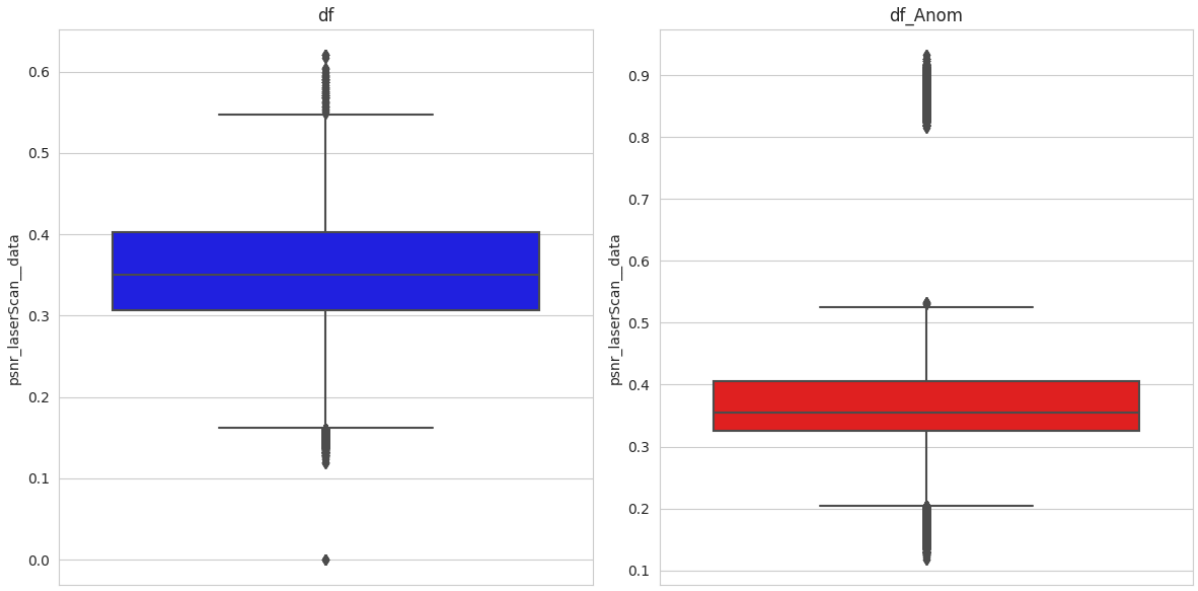
	psnr_laserScan__data	difficulty	rv_snr_event	robot_health
175979	0.307437	roughTerrainLaserNoise	0.006693	NaN
175980	0.307437	roughTerrainLaserNoise	0.006693	NaN
175981	0.307437	roughTerrainLaserNoise	0.006693	NaN
175982	0.307437	roughTerrainLaserNoise	0.006693	NaN
175983	0.307437	roughTerrainLaserNoise	0.006693	NaN

```
[ ] df_Anom.tail()
```

	psnr_laserScan__data	difficulty	rv_snr_event	robot_health
294210	0.197808	roughTerrainLaserNoise	0.017794	-1.542372
294211	0.197808	roughTerrainLaserNoise	0.017794	-1.534890
294212	0.197808	roughTerrainLaserNoise	0.017794	-1.527407
294213	0.197808	roughTerrainLaserNoise	0.017794	-1.519925
294214	0.197808	roughTerrainLaserNoise	0.017794	-1.512442

Σχήμα 7: Οι πρώτες 5 και οι τελευταίες 5 γραμμές του df_Anom

Στα παρακάτω σχήματα φαίνονται οι στατιστικές πληροφορίες που αφορούν τα 2 dataset. Για τις ανάγκες αυτές δημιουργήθηκαν 2 boxplots, ένα για το κάθε dataset και 2 distplot, επίσης ένα για το κάθε dataset, για τη σύγκριση των τιμών του “psnr_laserScan__data” μεταξυ τους.



Σχήμα 8: BoxPlots για το df και το df_Anom

Στα παραπάνω boxplots παίρνουμε μια ιδέα για τις στατιστικές ακραίες τιμές που υπάρχουν στο df. Βλέπουμε πως αυτές ξεκινάνε από 0.55 για τις άνω ακραίες ενώ για τις κάτω από 0.18. Αυτό που θα κάνουμε είναι να βρούμε το ποσοστό αυτών των ακραίων τιμών. Αυτό θα βοηθήσει στην εκπαίδευση των αλγορίθμων καθώς θα παραμετροποιηθεί με βάση το ποσό των outliers που υπάρχουν στο set εκπαίδευση df.

```
filtered_values = df[(df['psnr_laserScan__data'] > 0.55) | (df['psnr_laserScan__data'] < 0.18)]
count_filtered = len(filtered_values)
total_percentage = (count_filtered / len(df)) * 100
print(f"Count of df outliers: {count_filtered}")
print(f"Total Percentage: {total_percentage:.2f}%")
```

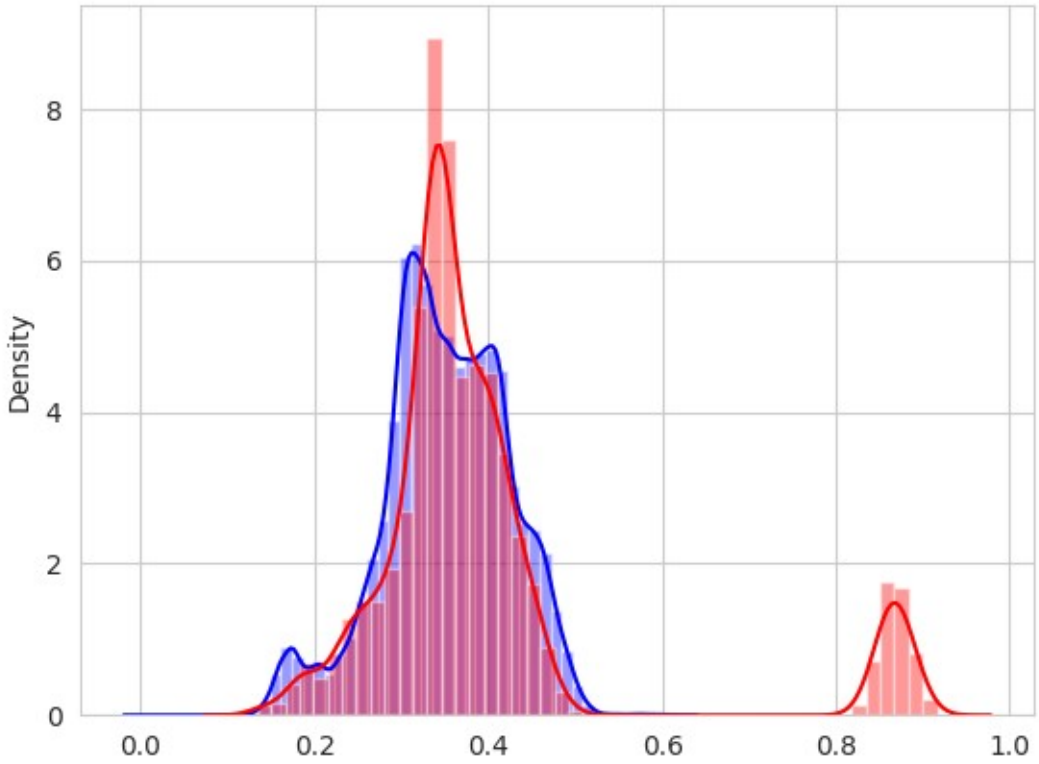
```
Count of df outliers: 4696
Total Percentage: 2.67%
```

Το ποσοστό αυτό είναι ανέρχεται στο 2.67% και συνολικά υπάρχουν 4696 ακραίες τιμές. Το ίδιο θα κάνουμε και για το df_Anom ώστε να συγκρίνουμε με τα αποτελέσματα που θα έχουν τα μοντέλα.

```
filtered_values = df_Anom[(df_Anom['psnr_laserScan__data'] > 0.52) | (df_Anom['psnr_laserScan__data'] < 0.21)]
count_filtered = len(filtered_values)
total_percentage = (count_filtered / len(df)) * 100
print(f"Count of df_Anom outliers: {count_filtered}")
print(f"Total Percentage: {total_percentage:.2f}%")
```

```
Count of df_Anom outliers: 13746
Total Percentage: 7.81%
```

Το ποσοστό αυτό ανέρχεται στο 7.81% και συνολικά υπάρχουν 13746 ακραίες τιμές.



Σχήμα 9: DistPlot για το df και το df_Anom

Στη σύγκριση των 2 distplot παρατηρούμε όπως ήταν αναμενόμενο από τα boxplots πως δεν υπάρχουν τιμές στην περιοχή πάνω από 0.55 για το df, επιπλέον παρατηρούμε πως το df_Anom για την τιμή “psnr_laserScan__data” ακολουθεί κανονική κατανομή όμως φαίνεται να ξεφεύγει προς τα πάνω στη μέση τιμή. Ενώ στο df ότι ακολουθεί μια σχεδόν κανονική κατανομή από τις περιπτώσεις μετά το 0.4 και πριν το 0.2. Θεωρούμε πως αυτές μπορεί να έχουν επίδραση στο μοντέλο.

Τέλος, για να τροφοδοτηθούν τα δεδομένα στους αλγορίθμους, θα πρέπει να είναι σε συγκεκριμένη μορφή. Τα μοντέλα scikit-learn απαιτούν τα δεδομένα εισόδου να έχουν τη μορφή ενός δισδιάστατου πίνακα όπου κάθε γραμμή αναπαριστά ένα σημείο δεδομένων και κάθε στήλη αναπαριστά ένα χαρακτηριστικό. Για να γίνει αυτό έγινε εξαγωγή του χαρακτηριστικού “psnr_laserScan__data” από τα εκάστοτε dataframes, μετατράπηκαν σε πίνακα numpy και αναδιαμορφώθηκαν από μονοδιάστατο σε πίνακα σε δισδιάστατο πίνακα με μία στήλη. Οι πίνακες αυτοί ονομάστηκαν normal_data για τα δεδομένα χωρίς εφαρμογή laser noise και anomalous_data για τα δεδομένα με εφαρμογή laser noise.

```
normal_data = df['psnr_laserScan__data'].values.reshape(-1, 1)
anomalous_data = df_Anom['psnr_laserScan__data'].values.reshape(-1, 1)
```

5.5 Ανίχνευση καινοτομιών

Παρακάτω θα γίνει η ανίχνευση καινοτομιών. Για την ανίχνευση καινοτομιών επιλέχθηκαν τρεις αλγόριθμοι προς εκπαίδευση:

1. Isolation forrest,
   Διότι φαίνεται πως οι ανωμαλίες είναι λίγες και διαφέρουν από το τη φυσιολογική συμπεριφορά.
2. One-Class SVM,
   Λόγο της γκαουσιανής κατανομής που ακολουθούν τα δεδομένα.
3. Local Outlier Factor,
   Επειδή είναι ένας αλγόριθμος που λαμβάνει υπόψη την απόσταση για την ανίχνευση ανωμαλιών.

5.5.1 Isolation forest

Δημιουργήθηκε μοντέλο με το όνομα model_IF στο οποίο μοναδική παράμετρος υπήρξε στο contamination, δηλαδή στο ποσοστό ανωμαλιών που αναμένει χονδρικά να ανιχνεύσει ο αλγόριθμος, το οποίο ορίστηκε στο 0.027, δηλαδή 2.7% που είναι το ποσοστό ακραίων τιμών που υπολογίσαμε για τα δεδομένα εκπαίδευσης στα οποία στη συνέχεια ο αλγόριθμος εκπαιδεύτηκε.

```
model_IF = IsolationForest(contamination=0.027)
model_IF.fit(normal_data)
```

```
IsolationForest(contamination=0.027)
```

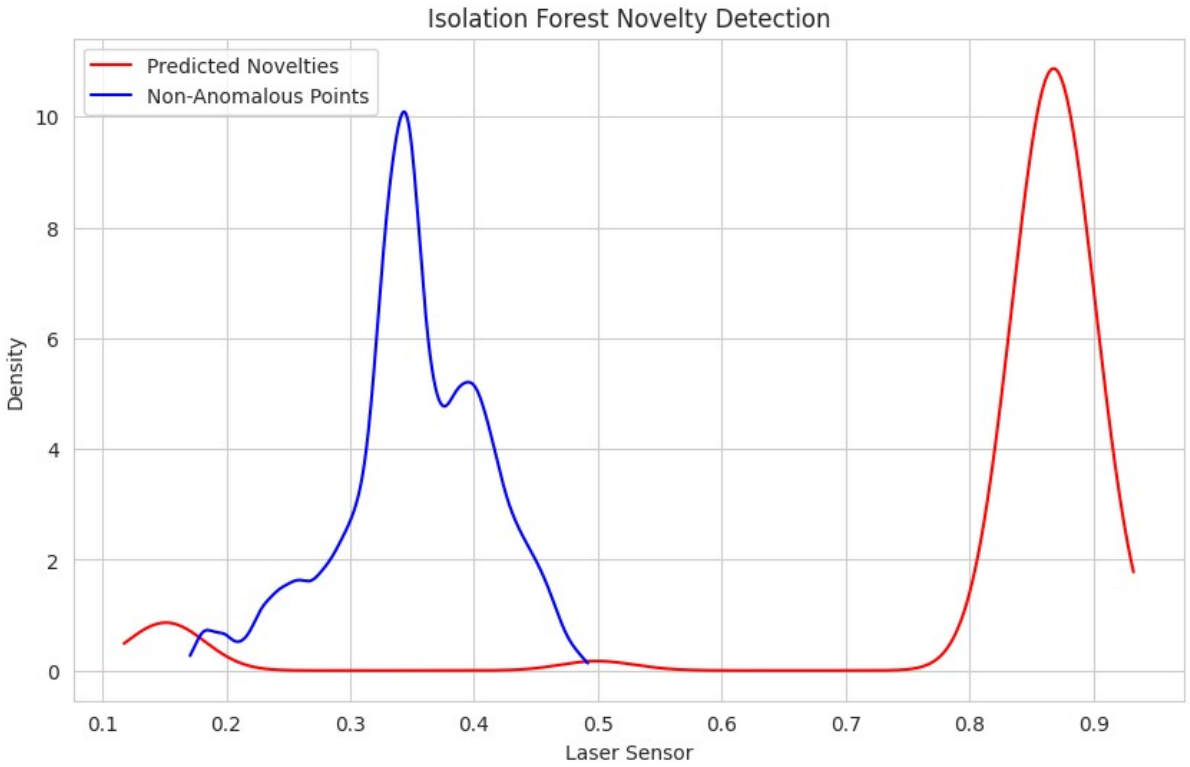
Στη συνέχεια χρησιμοποιήθηκε το μοντέλο αυτό για ανίχνευση καινοτομιών σε δεδομένα που δεν είχε εκπαιδευτεί με τη μέθοδο predict, τα οποία είναι τα anomalous_data. Η πρόβλεψη για το αν τα νέα αυτά δεδομένα αποτελούν ανωμαλίες ή όχι αναθέτονται στον numpy πίνακα y_pred_IF. Τα αποτελέσματα που πήραμε είναι:

```
Novelties: 11181
Non-Anomalous points: 107055
Total Percentage: 9.46%
```

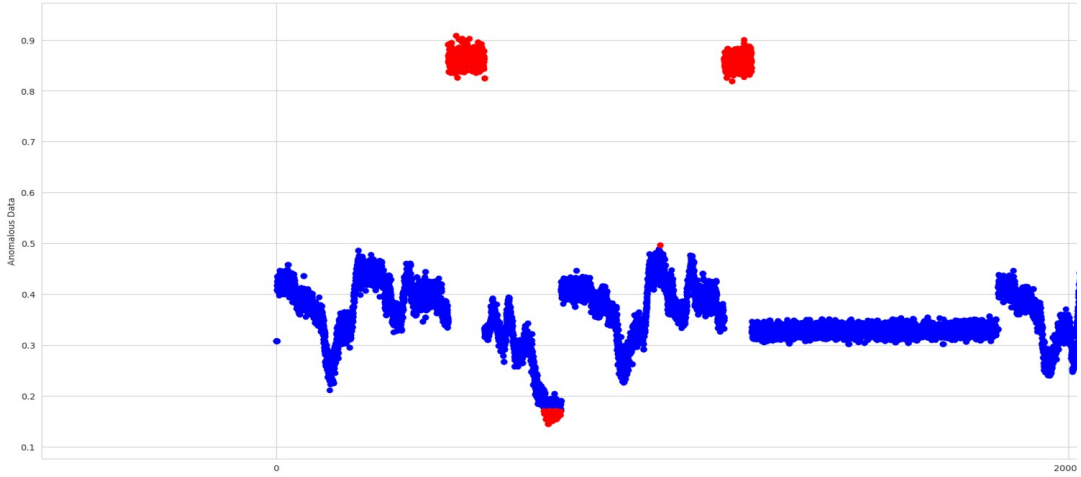
Παρατηρούμε ότι το μοντέλο εμφανίζει ποσοστό μεγαλύτερο από τα ποσοστά που εξήχθησαν από το boxplot, ανιχνεύοντας σαν καινοτομίες το 9,46%.

Για τη γραφική απεικόνιση των αποτελεσμάτων επιλέχθηκαν 3 γραφήματα. Το πρώτο γράφημα αφορά την πυκνότητα των τιμών για τις τιμές που προβλέφθηκαν ως καινοτομίες και τις τιμές που προβλέφθηκαν ως φυσιολογικές. Το δεύτερο γράφημα αφορά την εξέλιξη της τιμής του “psnr_laserScan__data” στο οποίο υπάρχουν επισημασμένες οι καινοτομίες με κόκκινο χρώμα ενώ ο φυσιολογικές με

μπλε και τέλος το 3 αφορά την εξέλιξη της τιμής του “robot_health” που με κόκκινο είναι οι τιμές στις οποίες ανιχνεύονται καινοτομίες στο “psnr_laserScan__data”.



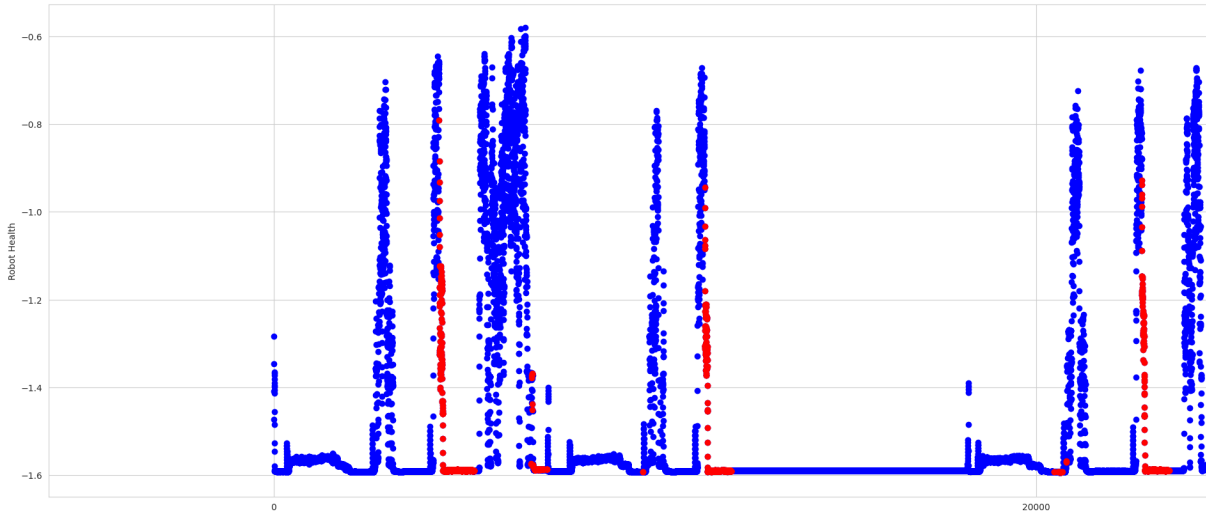
Σχήμα 10: Πυκνότητα τιμών για καινοτομίες και φυσιολογικά σημεία



Σχήμα 11: Ανίχνευση καινοτομιών του psnr_laserScan__Data

Στο πρώτο γράφημα λοιπόν διακρίνουμε ότι η πλειοψηφία των καινοτομιών που ανιχνεύει το μοντέλο βρίσκονται ψηλά και συγκεκριμένα μετά το 7.5. Αυτό ήταν πλήρως αναμενόμενο για τον συγκεκριμένο αλγόριθμο και τον τρόπο λειτουργίας του. Επιπλέον, βλέπουμε ακόμα 2 δύο περιοχές όπου ανιχνεύονται καινοτομίες στο dataset και είναι κοντά στις περιοχές με τιμή 0.5 και κάτω από 0.2 Παρατηρούμε δηλαδή ότι θέτει ένα άνω κατώφλι για τις άνω ακραίες τιμές το οποίο είναι πιο χαμηλά από αυτό που είδα στο boxplot κατά 0.05 περίπου ενώ το κάτω κατώφλι είναι εκεί που περιμέναμε δηλαδή στο 0.21. Μπορούμε να βγάλουμε καλύτερα συμπεράσματα παρατηρώντας και το δεύτερο γράφημα.

Για να είναι πιο ευπαρουσίαστο το γράφημα, λόγο όγκου, θα απεικονιστεί το πρώτο κομμάτι το οποίο αποτελεί μέρος του γραφήματος και έχει τις απαραίτητες πληροφορίες που χρειάζεται για να βγουν συμπεράσματα. Φαίνεται λοιπόν πως τον συγκεκριμένο αλγόριθμο μπορούμε να δημιουργήσουμε άνω και κάτω φράγματα για την ανίχνευση καινοτομιών σε μια χρονοσειρά με μεγάλη επιτυχία. Τα δεδομένα από τα συγκεκριμένα γράφημα πέρασαν για πρώτη φορά μέσα από το μοντέλο και με επιτυχία κατάφερε να ανιχνεύσει ανώτατες αλλά και κατώτατες τιμές ενώ με μεγάλη επιτυχία ανιχνεύει το laser noise που εφαρμόστηκε στο ρομπότ. Επιπλέον, όμως βλέπουμε πως δεν είναι ικανό να ανιχνεύσει αλλαγή στο μοτίβο με το οποίο αλλάζει η τιμή του “psnr_laserScan__data”.



Σχήμα 12: Οι καινοτομίες του psnr_laserScan_Data στην εξέλιξη τιμής του robot_health

Στο τελευταίο γράφημα που θα απεικονίσουμε θέλουμε να δούμε αν οι καινοτομίες που ανιχνεύονται επηρεάζουν την τιμή του “robot_health”.

Επίσης και για αυτό το γράφημα για τις ανάγκες ανάγνωσης του θα μελετήσουμε το αρχικό μέρος το οποίο περιέχει τις απαραίτητες πληροφορίες για να βγουν συμπεράσματα. Φαίνεται λοιπόν πως στις μαρκαρισμένες με κόκκινο χρώμα περιοχές η τιμή του “robot_health” απομακρύνεται από 0 ή διατηρούν την τιμή μακριά από το 0 με τιμή μεγαλύτερης απομάκρυνσης τη -1,6. Οπότε φαίνεται πως οι τιμές που ανιχνεύονται ως καινοτομίες έχουν επίδραση πάνω στο “robot_health” πράγμα που σημαίνει ότι όταν υπάρχουν τέτοιες τιμές το ρομπότ αδυνατεί να ολοκληρώσει με επιτυχία τον στόχο του.

5.5.2 One Class – SVM

Με παρόμοιο τρόπο έγινε και η εκπαίδευση των άλλον δυο αλγορίθμων. Από τον αλγόριθμο One Class – SVM δημιουργήθηκε μοντέλο με το όνομα ocsvm_model στον οποίο για την παράμετρο kernel επιλέχθηκε ο Radial Basis Function (RBF) που αποτελέι μια λογική επιλογή όταν έχουμε να κάνουμε με δεδομένα που ακολουθούν μια γκαουσιανή κατανομή και δεν είναι γραμμικά, ενώ στο contamination, δηλαδή στο ποσοστό ανωμαλιών που αναμένει χονδρικά να ανιχνεύσει ο αλγόριθμος, το οποίο ορίστηκε στο 0.027, δηλαδή 2.7% που είναι το ποσοστό ακραίων τιμών που υπολογίσαμε για τα δεδομένα εκπαίδευσης στα οποία στη συνέχεια ο αλγόριθμος εκπαιδεύτηκε.

Στη συνέχεια ακολουθήθηκε παρόμοια διαδικασία για την ανίχνευση καινοτομιών σε δεδομένα που δεν είχε εκπαιδευτεί τα οποία είναι τα anomalous_data, χρησιμοποιώντας τη μέθοδο predict. Η πρόβλεψη για το αν τα νέα αυτά δεδομένα αποτελούν ανωμαλίες ή όχι αναθέτονται στον numpy πίνακα y_pred_OCSVM. Τα αποτελέσματα που πήραμε είναι:

```
[26] ocsvm_model = OneClassSVM(kernel='rbf', nu=0.027)
```

```
[27] ocsvm_model.fit(normal_data)
```

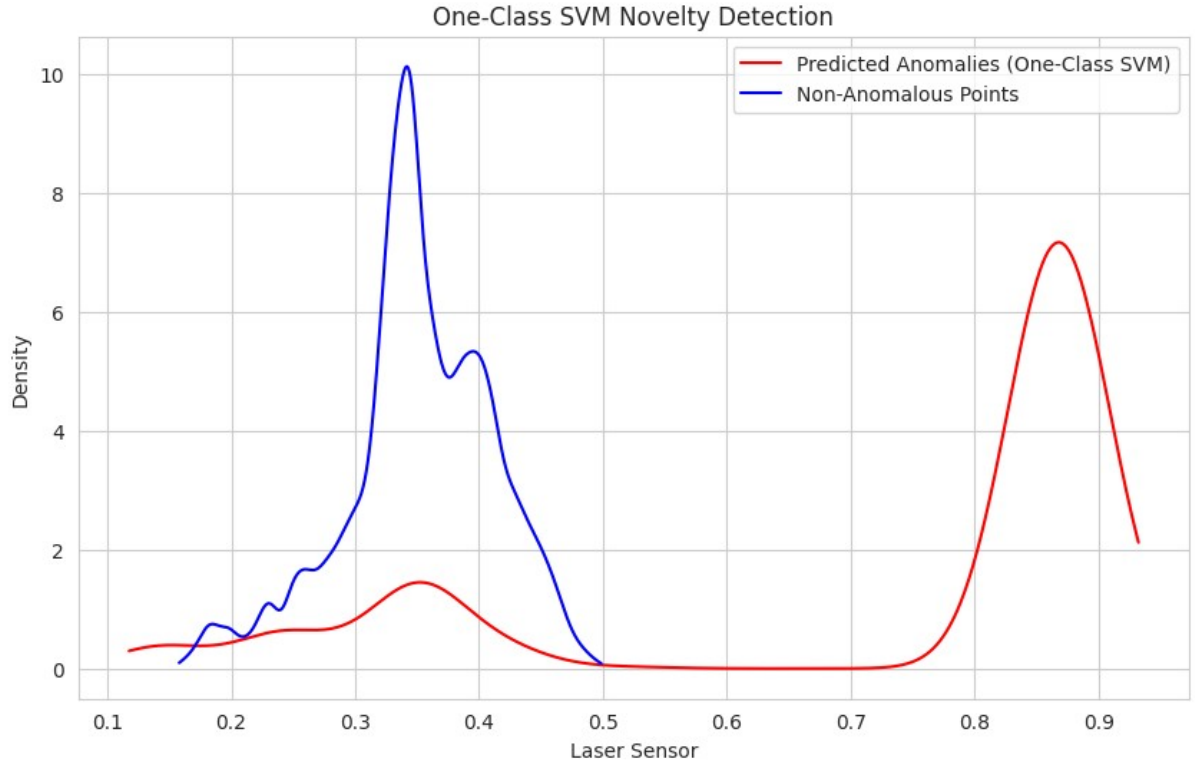
```
OneClassSVM
OneClassSVM(nu=0.027)
```

```
[28] y_pred_OCSVM = ocsvm_model.predict(anomalous_data)
```

Παρατηρούμε ότι το μοντέλο εμφανίζει ποσοστό μεγαλύτερο από τα ποσοστά που εξήχθησαν από το boxplot, ανίχνευοντας σαν καινοτομίες το 11,76%. Επιπλέον, όμως φαίνεται σε σύγκριση με το προηγούμενο μοντέλο ότι ανιχνεύει περισσότερες καινοτομίες από το Isolation Forrest. Τα παρακάτω γραφήματα θα δώσουν περισσότερες εξηγήσεις για τις καινοτομίες αυτές.

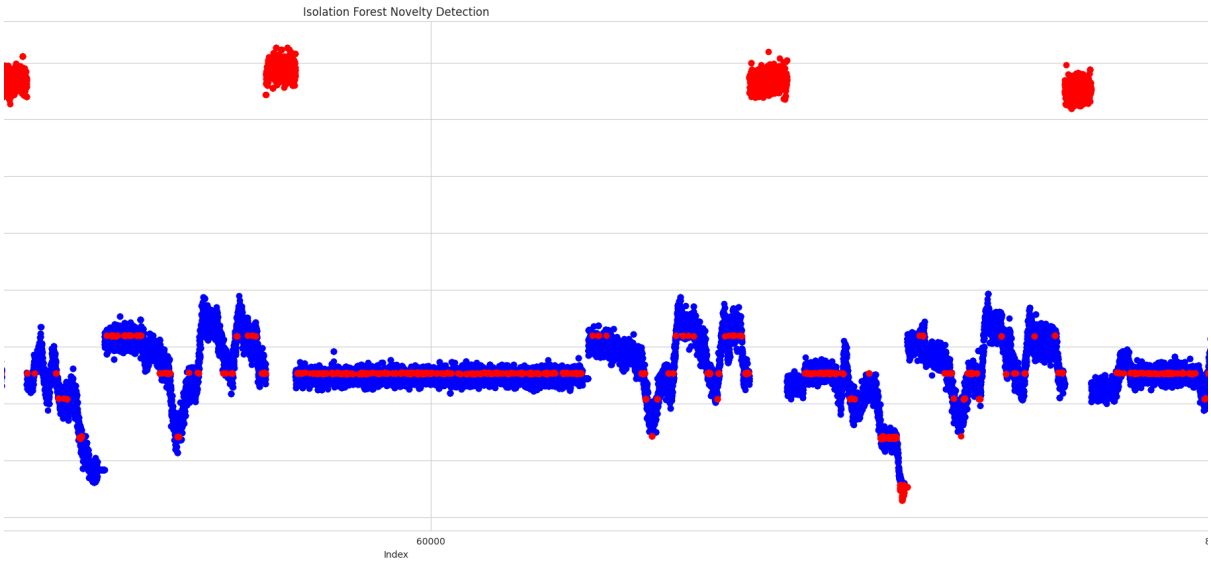
```
Novelties: 13903
Non-Anomalous points: 104333
Total Percentage: 11.76%
```

Φαίνεται λοιπόν πως στο πρώτο γράφημα πυκνότητας των τιμών για τις τιμές που προβλέφθηκαν ως καινοτομίες και τις τιμές που προβλέφθηκαν ως φυσιολογικές, ότι η πλειοψηφία των καινοτομιών που ανιχνεύει το μοντέλο βρίσκονται ψηλά και συγκεκριμένα μετά το 7.5, αναμενόμενο καθώς περιμέναμε και από αυτό το μοντέλο να ανιχνεύσει το πότε εφαρμόζεται laser noise καθώς είναι ένας αλγόριθμος που δουλεύει καλά στις κατανομές που έχουν τα δεδομένα. Αυτό που παρουσιάζει ενδιαφέρον όμως είναι το κάτω φράγμα και συγκεκριμένα η περιοχή πριν από το 0.4 που όπως φαίνεται και στο σχήμα 13 ξεφεύγει από την κανονική κατανομή. Τη συγκεκριμένη περιοχή δεν την ανίχνευσε σαν καινοτομία το Isolation Forest. Επιπλέον, το κάτω κατώφλι φαίνεται να ξεκινάει λίγο μετά το 0.4 σε σχέση με το 0.21 που έβγαλε το Isolation forest.



Σχήμα 13: Πυκνότητα τιμών για καινοτομίες και φυσιολογικά σημεία

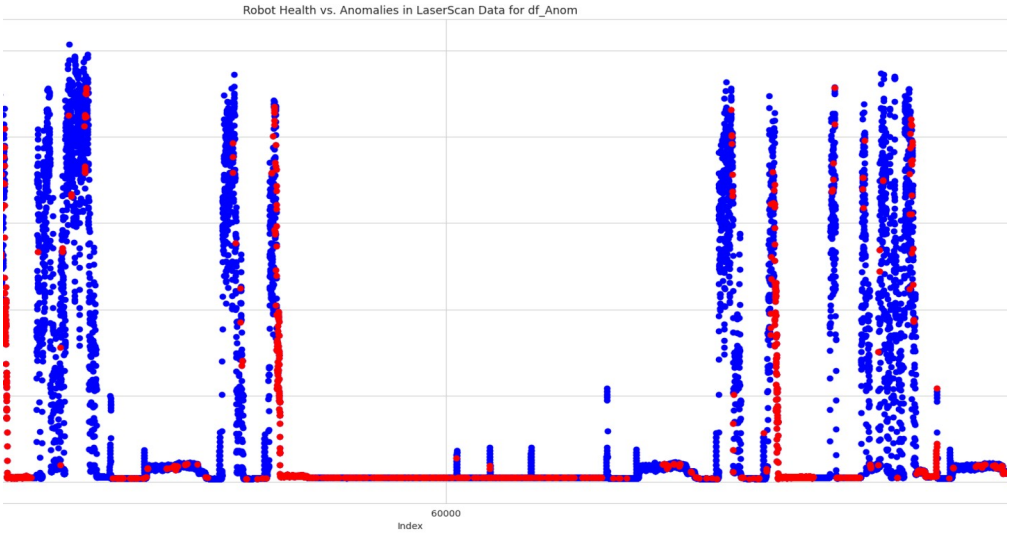
Στο παρακάτω σχήμα, του οποίου επίσης θα μελετήσουμε μέρος του, θα δούμε και πού ακριβώς είναι αυτές οι καινοτομίες που ανίχνευσε το One Class – SVM.



Σχήμα 14: Ανίχνευση καινοτομιών του psnr_laserScan__Data

Όπως ήταν αναμενόμενο και από το παραπάνω σχήμα όλες τις φορές που εφαρμόστηκε laser noise το μοντέλο το ανίχνευσε ως καινοτομία. Το ενδιαφέρον είναι όμως πως το μοντέλο ανιχνεύει καινοτομίες σε συγκεκριμένες τιμές του laser noise επίσης. Αυτές είναι τιμές που δεν ανταποκρίνονται στην κατανομή των dataset.

Αυτό κάνει το μοντέλο ικανό να ανιχνεύσει τιμές που αποκλίνουν από το μοτίβο που ακολουθούν τα normal_data όπως φαίνεται και στο σχήμα 21, ενώ φαίνεται να έχει και ένα κατώφλι περίπου στο 0.15 όπου από εκεί και κάτω θεωρεί όλες τις τιμές ως καινοτομίες, το οποίο είναι πιο χαμηλά από αυτό του πρώτου μοντέλου.



Σχήμα 15: Οι καινοτομίες του psnr_laserScan_Data στην εξέλιξη τιμής

Φαίνεται λοιπόν οι καινοτομίες που ανιχνεύει το μοντέλο επηρεάζουν το “robot_health”. Οι περισσότερες καινοτομίες που ανιχνεύει είναι καινοτομίες που όταν υπάρχουν όπως φαίνεται δημιουργούν πρόβλημα στη λειτουργία του ρομπότ όπως και το μοντέλο του αλγορίθμου Isolation Forrest, με διαφορά πως εδώ εντόπισε την ανωμαλία στην αλλαγή του μοτίβου που φαίνεται να διατηρεί το “robot_health” μακριά από το 0 για μεγάλο χρονικό διάστημα.

5.5.3 Local Outlier Factor

Για τον τρίτο και τελευταίο αλγόριθμος επιλέχθηκε ο Local Outlier Factor από τον οποίο και δημιουργήθηκε το μοντέλο με όνομα model_LOF. Για τη δημιουργία του μοντέλου επιλέχθηκε ο μονός αριθμός για n_neighbors = 11, το novelty ως True καθώς για να ανιχνεύσει καινοτομίες εφόσον θα του δίναμε ένα dataset που δεν έχει ξαναδεί και τέλος το contamination ορίστηκε στο 0.027, δηλαδή 2.7% όπως και στους άλλους δύο.

```
[71] model_LOF = LocalOutlierFactor(n_neighbors=11, novelty=True, contamination=0.027)
     model_LOF.fit(normal_data)
```

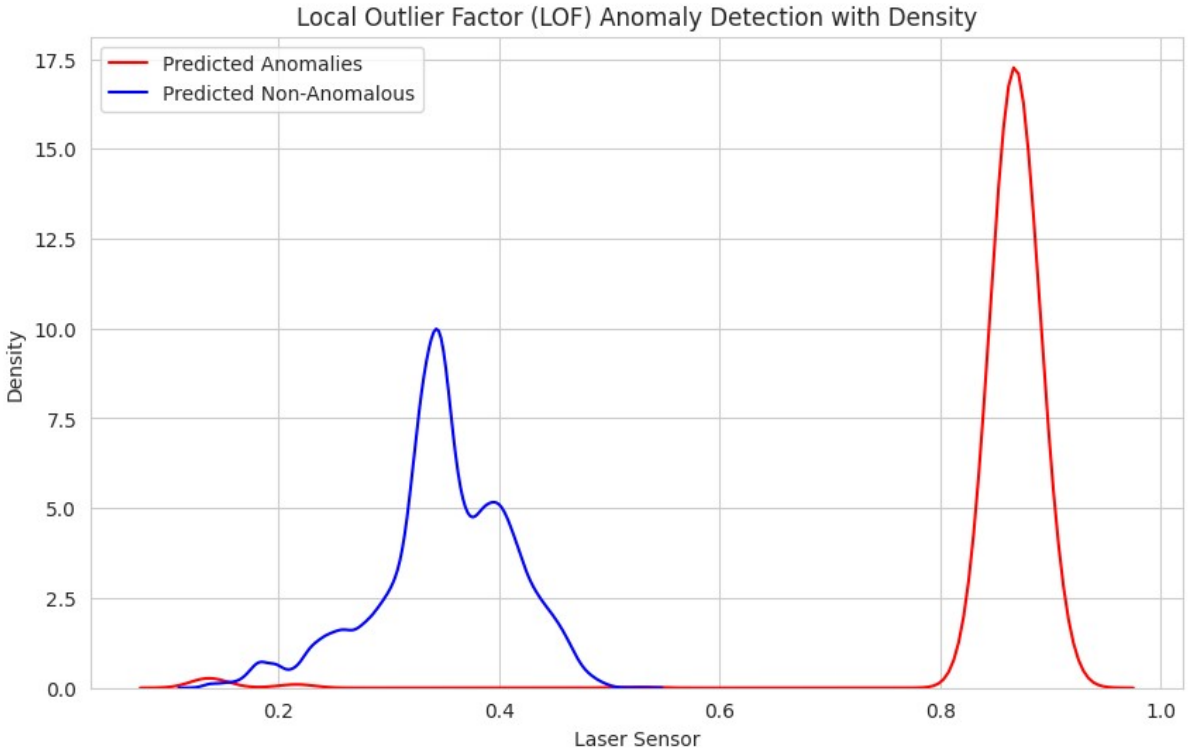
```
LocalOutlierFactor
LocalOutlierFactor(contamination=0.027, n_neighbors=11, novelty=True)
```

```
[72] # Predict the anomalies on the anomalous data
     y_pred_LOF = model_LOF.predict(anomalous_data)
```

Στην εκπαιδεύσαμε μοντέλο με τα normal_data χρησιμοποιώντας τη μέθοδο predict. Η πρόβλεψη για το αν τα νέα αυτά δεδομένα αποτελούν ανωμαλίες ή όχι αναθέτονται στον numpy πίνακα y_pred_LOF. Τα αποτελέσματα που πήραμε είναι:

```
Novelties: 10467
Non-Anomalous points: 107769
Total Percentage: 8.85%
```

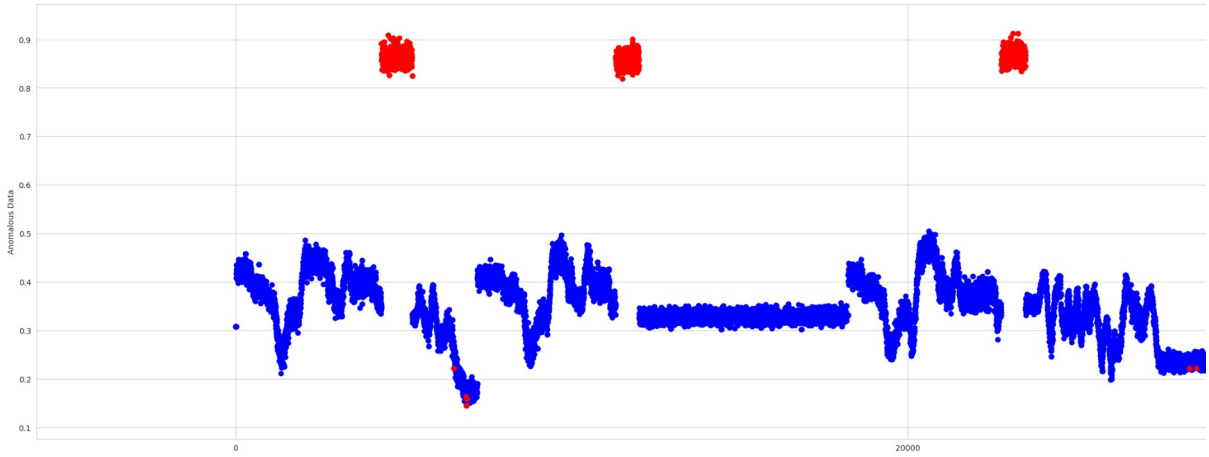
Το model_LOF δίνει το μικρότερο ποσοστό καινοτομιών στο dataset το οποίο είναι 8.85% πιο κάτω από το modeil_IF αλλά και πάλι δείχνει να ανιχνεύει περισσότερες τιμές από ότι οι ακραίες τιμές που έδωσε το boxplot.



Σχήμα 16: Πυκνότητα τιμών για καινοτομίες και φυσιολογικά σημεία

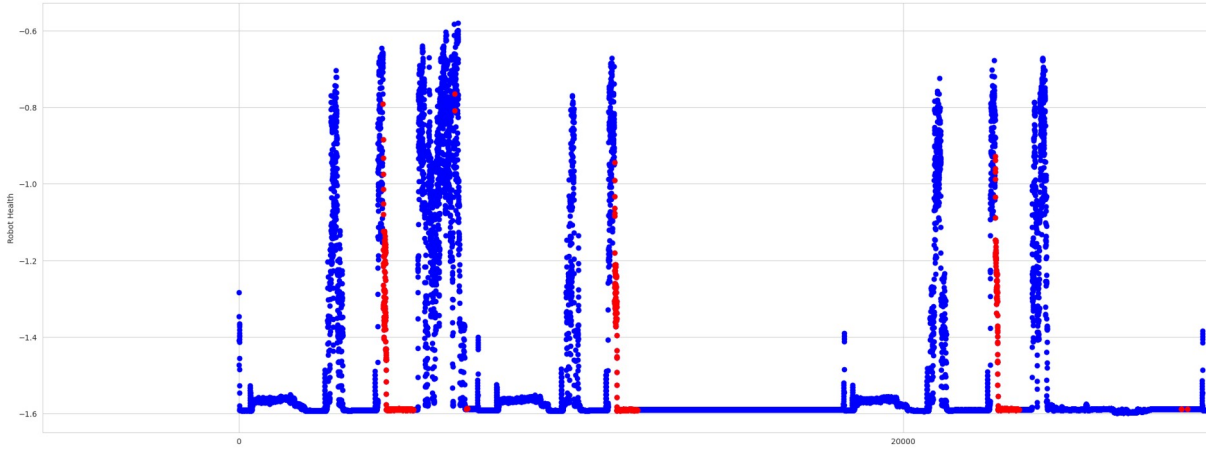
Στο γράφημα πυκνότητας των τιμών για τις τιμές που προβλέφθηκαν ως καινοτομίες και τις τιμές που προβλέφθηκαν ως φυσιολογικές, επίσης φαίνεται ότι η πλειοψηφία των καινοτομιών που ανιχνεύει το μοντέλο βρίσκονται ψηλά και συγκεκριμένα μετά το 7.5, πράγμα και αυτό αναμενόμενο καθώς περιμέναμε και από αυτό το μοντέλο να ανιχνεύσει το πότε εφαρμόζεται laser noise διότι επεξεργάζεται τις αποστάσεις μεταξύ των δεδομένων. Φαίνεται ότι δεν ανιχνεύει τη διαφορά στο μοτίβο που έχουν τα anomalous_data με τα normal_data ενώ είναι λίγο θολά τα σημεία των άνω και κάτω κατωφλίων.

Παρακάτω φαίνονται τα σημείο που ανίχνευσε το μοντέλο ως καινοτομίες.



Σχήμα 17: Ανίχνευση καινοτομιών του psnr_laserScan__Data

Φαίνεται πως και αυτό το μοντέλο είναι ικανό να ανιχνεύσει το πότε εφαρμόζεται laser noise στο ρομπότ. Δε φαίνεται να υπάρχει εμφανές άνω φράγμα από πιο σημείο και πάνω μια τιμή θεωρείται καινοτομία ενώ είναι θολό και από αυτό το γράφημα το κάτω φράγμα. Τέλος, δεν είναι ικανό να ανιχνεύσει κάποια αλλαγή στο μοτίβο. Όπως και ο Isolation Forest έτσι και αυτός ο αλγόριθμος φαίνεται να ανιχνεύουν μόνο ανώτερες και κατώτερες τιμές.



Σχήμα 18: Οι καινοτομίες του psnr_laserScan_Data στην εξέλιξη τιμής του robot_health

Στο τελευταίο σχήμα βλέπουμε πως το μοντέλο ανιχνεύει ανωμαλίες που είναι όλες επιβλαβείς για τη λειτουργία του ρομπότ καθώς όλα τα σημεία είναι σημεία που το “robot_health” απομακρύνεται από το 0, όπως συμπεράναμε και από τα παραπάνω σχήματα η συμπεριφορά του είναι πιο κοντά σε αυτή του Isolation Forest.

Κεφάλαιο 6

6. Συμπεράσματα

Μετά και το τέλος του τρίτου μοντέλου το συμπέρασμα που βγάζουμε είναι πως το ιδανικότερο μοντέλο για να προβλέψουμε τις ακραίες τιμές ως καινοτομίες από αυτά τα 3 είναι το Isolation Forest και αυτό διότι είναι καλύτερο με το να θέτει άνω και κάτω φράγματα για τις ακραίες τιμές. Με μεγάλη επιτυχία προέβλεψε όλες τις τιμές που υπερβαίνουν το όριο θέτει σε αντίθεση με το Local Outlier Factor στο οποίο τα όριο μεταξύ των τιμών ήταν θολά σχετικά με το πότε μια τιμή είναι καινοτομία και πότε όχι, επιπλέον εντόπισε τις λιγότερες καινοτομίες σε σχέση με τα άλλα 2 μοντέλα. Αυτό μπορεί και να οφείλεται στον τρόπο λειτουργία του αλγορίθμου καθώς προέβλεψε με επιτυχία το πότε υπάρχει εφαρμογή του laser noise, όταν τα δεδομένα ομαδοποιούνται με μεγάλες αποστάσεις από τον πυρήνα των φυσιολογικών δεδομένων ενώ όταν αυτά βρίσκονταν κοντά πολλές φορές φάνηκε μη ικανό.

Το One Class – SVM φάνηκε εξίσου ικανό στην πρόβλεψη των τιμών που εφαρμόζεται το laser noise αλλά δεν ήταν ικανό να δημιουργήσει άνω και κάτω φράγματα. Αυτό συνέβαινε καθώς μπορούσε με επιτυχία να ανιχνεύσει αλλαγές στο μοτίβο της τιμής οπότε συχνά τιμές οι οποίες ήταν μέσα στα φυσιολογικά όρια που έθετε και η θεωρία της έρευνας που βασιστήκαμε ανιχνευόντουσαν ως καινοτομίες, αυτός ήταν και ο λόγος που ανίχνευσε περισσότερες καινοτομίες από τους άλλου αλγορίθμους, πράγμα αναμενόμενο για έναν αλγόριθμου που βασίζει την ανίχνευση σε μεγάλο βαθμό στην κατανομή των δεδομένων.

Κατάλογος Σχημάτων

Κατάλογος εικόνων

Πηγές

1.Ramesh, Aniketh & Chiou, Manolis & Stolkin, Rustam. (2021). Robot Vitals and Robot Health: An Intuitive Approach to Quantifying and Communicating Predicted Robot Performance Degradation in Human-Robot Teams. 303-307. 10.1145/3434074.3447181.

2. Russell, S. J. 1., Norvig, P., & Davis, E. (2010). Artificial intelligence: a modern approach. 3rd ed. Upper Saddle River, NJ, Prentice Hall.

3. Murphy, K. P. (2013). Machine learning : a probabilistic perspective. Cambridge, Mass. [u.a.]: MIT Press. ISBN: 9780262018029 0262018020

4. Müller Andreas C & Guido S. (2017). Introduction to machine learning with python : a guide for data scientists (First). O'Reilly Media. Retrieved September 20 2023 from http://www.dawsonera.com/depp/reader/protected/external/AbstractView/S9781449369903.

5. van Engelen, J.E., Hoos, H.H. A survey on semi-supervised learning. Mach Learn 109, 373–440 (2020). https://doi.org/10.1007/s10994-019-05855-6

6. Zhu, X. (2005). Semi-Supervised Learning Literature Survey (1530). Computer Sciences, University of Wisconsin-Madison .

7. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Comput. Surv., 41, 15:1-15:58.

8. Pimentel, M.A., Clifton, D.A., Clifton, L.A., & Tarassenko, L. (2014). A review of novelty detection. Signal Process., 99, 215-249.

9. F. T. Liu, K. M. Ting and Z. -H. Zhou, "Isolation Forest," 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.

10. Bhattacharyya, D.K., & Kalita, J.K. (2013). Network Anomaly Detection: A Machine Learning Perspective (1st ed.). Chapman and Hall/CRC.

11. Nassif, A.B., Talib, M.A., Nasir, Q., & Dakalbab, F.M. (2021). Machine Learning for Anomaly Detection: A Systematic Review. IEEE Access, 9, 78658-78700.

12. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural computation, 13(7), 1443–1471. https://doi.org/10.1162/089976601750264965

13. Wang, Yanxin & Wong, Johnny & Miner, Andrew. (2004). Anomaly intrusion detection using one class SVM. 358 - 364. 10.1109/IAW.2004.1437839.

14. Perdisci, Roberto & Gu, Guofei & Lee, Wenke. (2006). Using an Ensemble of One-Class SVM Classifiers to Harden Payload-based Anomaly Detection Systems. Proceedings of the IEEE International Conference on Data Mining (ICDM 2006). 488-498. 10.1109/ICDM.2006.165.

15. McKinney W. (2017). Python for data analysis : data wrangling with pandas numpy and ipython (Second). O'Reilly Media. Retrieved September 20 2023 from https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1605925.

16. VanderPlas, J. (2016). Python data science handbook : essential tools for working with data. Sebastopol, CA: O'Reilly Media, Inc. ISBN: 978-1491912058

17. Breunig, M. M., Kriegel, H.-P., Ng, R. T. & Sander, Jö. (2000). LOF: identifying density-based local outliers. ACM sigmod record (p./pp. 93--104), .

18. Kriegel, HP., Kröger, P., Schubert, E., Zimek, A. (2009). Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, TB.

(eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2009. Lecture Notes in Computer Science(), vol 5476. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01307-2_86
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. & others (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825--2830.
20. https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76
21. https://towardsdatascience.com/a-friendly-intro-to-semi-supervised-learning-3783c0146744
22. https://www.ibm.com/topics/unsupervised
23. https://medium.com/@jeremiascampos3/types-of-anomaly-detection-1965b41587a1
24.https://www.anodot.com/blog/what-is-anomaly-detection/
25. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
26. https://www.analyticsvidhya.com/blog/2022/06/one-class-classification-using-support-vector-machines/
27. https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html
28. https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html
30. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html
31. https://research.google.com/colaboratory/faq.html
32. https://www.androidpolice.com/google-colab-explainer/
33. https://docs.python.org/3/library/pickle.html#module-pickle
34. https://favtutor.com/blogs/pickle-python
35. https://numpy.org/doc/stable/user/whatisnumpy.html
36. https://scikit-learn.org/stable/modules/ensemble.html
37. https://scikit-learn.org/stable/modules/neighbors.html
38. https://scikit-learn.org/stable/modules/svm.html
39. https://holoviews.org
40. https://notebook.community/vascotenner/holoviews/doc/Tutorials/Introduction
41. https://holoviews.org/user_guide/Plotting_with_Bokeh.html
42. https://docs.bokeh.org/en/latest/
43. https://www.simplilearn.com/tutorials/python-tutorial/python-bokeh#what_is_bokeh