ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης & Παραγωγής

Διπλωματική Εργασία

«Ανίχνευση ανωμαλιών με τη χρήση Μηχανικής Μάθησης»

του φοιτητή:

Τουτουντζάκη Άγγελου

ΑΜ:18389047

Επιβλέπων Καθηγητής:

Νικολάου Γρηγόριος

ΝΙΚΟΛΑΟΥ ΓΡΗΓΟΡΙΟΣ	ΒΑΣΙΛΕΙΑΔΟΥ ΣΟΥΛΤΑΝΑ	ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ
GRIGORIOS NIKOLAOU Digitally signed by GRIGORIOS NIKOLAOU Date: 2023.07.21 12:07:39 +03'00'	Soultana Vasileiadou Digitally signed by Soultana Vasileiadou Date: 2023.07.25 08:55:34 +03'00'	Christos Drosos Digitally signed by Christos Drosos Date: 2023.07.23 15:38:14 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Τουτουντζάκης Άγγελος** του **Ιωάννη**, με αριθμό μητρώου **18389047** 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

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω το Θεό και την οικογένεια μου που με στήριξαν κατά την διάρκεια της εκπόνησης της διπλωματικής αυτής εργασίας καθώς επίσης να πώ ένα ιδιαίτερο ευχαριστώ στον επιβλέποντα καθηγητή μου Νικολάου Γρηγόρη για την συνεχή καθοδήγηση και υπομονή που έδειξε μαζί μου όλα αυτά τα χρόνια. Εκτιμώ τόσο την συνεργασία μας όσο και τις γνώσεις που αποκόμισα από όλους τους καθηγητές μου κατά την διάρκεια των σπουδών μου.

ΤΟΥΤΟΥΝΤΑΚΗΣ ΑΓΓΕΛΟΣ

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια η μηχανική μάθηση κερδίζει όλο και περισσότερο έδαφος στην εργασιακή καθημερινότητα κάτι που οφείλεται σε μεγάλο βαθμό στην ραγδαία τεχνολογική εξέλιση στον τομέα της επιστήμης των υπολογιστών. Η μηχανική μάθηση βρίσκει χρήση σε μία πληθώρα εφαρμογών στις επιχειρήσεις, στην ιατρική, στην βιομηχανία ακόμα και στον τομέα του αθλητισμού, με τους επιστήμονες του χώρου να αξιοποιούν τις δυνατότητες της για στατιστικές μελέτες καθώς και για διάφορες προβλέψεις σε μεγάλα σύνολα δεδομένων.

Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι η ανίχνευση ανωμαλιών με την χρήση μηχανικής μάθησης. Στα πλαίσια της εργασίας αυτής, αναλύονται οι έννοιες της μηχανικής μάθησης, της ανίχνευσης ανωμαλιών καθώς επίσης κατασκευάστηκαν συνολικά 9 μοντέλα μηχανικής μάθησης για την αναγνώριση ακραίων τιμών και την πρόβλεψη νέων πιθανών ανωμαλιών στα θερμοκρασιακά δεδομένα ενός μεγάλου βιομηχανικού κινητήρα. Τα δεδομένα που χρησιμοποιήθηκαν προέρχονται από σύνολο δεδομένων Numenta Anomaly Benchmark που περιέχει πραγματικές καταγραφές δεδομένων από επιχειρήσεις και βιομηχανίες.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Μηχανική Μάθηση, Ανίχνευση Ανωμαλίων, Ανίχνευση Ακραίων Τιμών, Ανίχνευση Καινοτομίας, Isolation Forest, One Class SVM, Local Outlier Factor, Elliptic Envelope, Εκμάθηση Συνόλου

ABSTRACT

In recent years, machine learning has gained more and more ground in the work place, which is largely due to the rapid technological evolution in the field of computer science. Machine learning finds its use in a multitude of applications in business, medicine, industries and even sports, with machine learning scientists harnessing its potential for statistical studies as well as various predictions on large datasets.

The aim of this thesis is the detection of anomalies using machine learning on industry data. In the context of this thesis, the concepts of machine learning and anomaly detection are analyzed, as well as a total of 9 machine learning models were built to identify extreme values and predict new possible anomalies in the temperature data of a large industrial engine. The data used comes from Numenta Anomaly Benchmark dataset which contains real data records from businesses and industries.

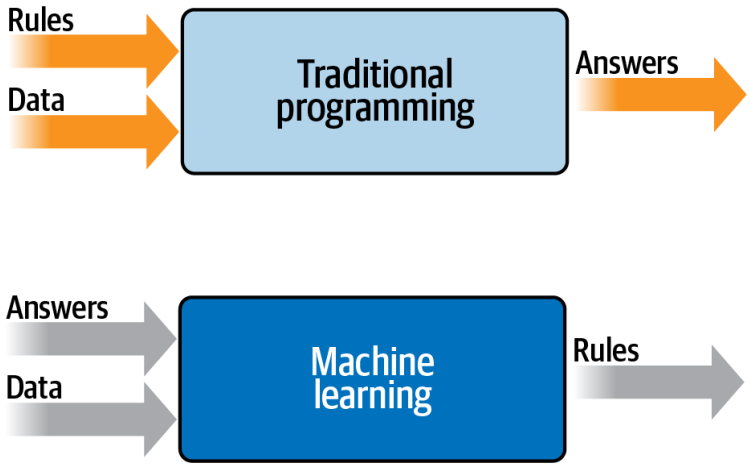
KEY WORDS

Machine Learning, Anomaly Detection, Outlier Detection, Novelty Detection, Isolation Forest, One Class SVM, Local Outlier Factor, Elliptic Envelope, Ensemble Machine Learning

ΚΕΦ. 1Ο ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

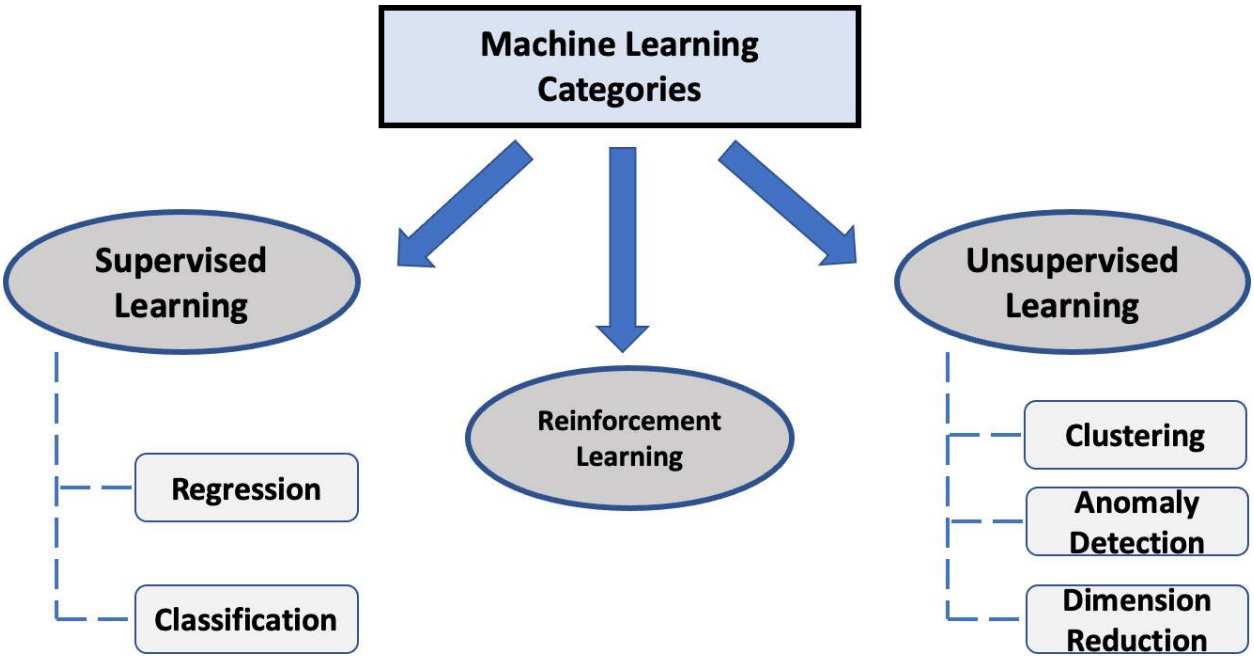
1.1 Τι είναι Μηχανική μάθηση

Η **Μηχανική Μάθηση** γνωστή και ως **Machine Learning** αποτελεί ένα κλάδο της τεχνητής νοημοσύνης (Artificial Iintelligence) και της επιστήμης των υπολογιστών που έχει ως κύρια βάση την χρήση δεδομένων και αλγορίθμων για τη μίμηση του τρόπου με τον οποίο μαθαίνουν οι άνθρωποι. Η μηχανική μάθηση είναι ένα σημαντικό και αναπόσπαστο κομμάτι της συνεχώς αναπτυσόμενης επιστήμης των δεδομένων (data science). Με την χρήση αλγορίθμων μηχανικής μάθησης και στατιστικών μεθόδων, έχουμε την ικανότητα να εκπαιδεύσουμε μοντέλα να κάνουν ταξινομήσεις ή προβλέψεις ή ακόμα και παρατηρήσεις πάνω στα μεγάλης κλίμακας δεδομένα μας. Κατά αυτό τον τρόπο αυτοματοποιείται η διαδικασία ανάλυσης των δεδομένων κάτι που είναι αναγκαίο όταν διαθέτουμε υπερβολικά μεγάλα σύνολα δεδομένων που σε διαφορετική περίπτωση θα ήταν αδύνατον να εκμεταλευτούμε με τις συμβατικές μεθόδους ανάλυσης. Τα αποτελέσματα των μοντέλων αυτών στη συνέχεια βοηθούν στην βελτιστοποίηση της λήψης αποφάσεων εντός των επιχειρήσεων (και όχι μόνο), επηρεάζοντας σημαντικά την ανάπτυξη τους.



Εικόνα 1

1.2 Κατηγορίες μηχανικής μάθησης



Εικόνα 2

1.2.1 Μάθηση με επίβλεψη (Supervised Learning)

Η εποπτευόμενη μηχανική μάθηση, γνωστή και ως supervised learning, αποτελεί μια υποκατηγορία της μηχανικής μάθησης και της εφαρμοσμένης τεχνητής νοημοσύνης. Ορίζεται από τη χρήση δεδομένων εισόδου και εξόδου με ετικέτα κατά τη φάση της εκπαίδευσης. Αυτά τα δεδομένα εκπαίδευσης (ονομάζονται και labeled datasets) συχνά επισημαίνονται από έναν επιστήμονα δεδομένων στη φάση προετοιμασίας, πριν χρησιμοποιηθούν για την εκπαίδευση και τη δοκιμή του μοντέλου. Μόλις το μοντέλο μάθει τη σχέση μεταξύ των δεδομένων εισόδου και εξόδου, μπορεί να χρησιμοποιηθεί για την ταξινόμηση νέων και μη ορατών συνόλων δεδομένων και την πρόβλεψη αποτελεσμάτων. Ο λόγος που ονομάζεται εποπτευόμενη μηχανική μάθηση είναι επειδή ένα σημαντικό μέρος αυτής της διαδικασίας απαιτεί ανθρώπινη επίβλεψη. Η συντριπτική πλειονότητα των διαθέσιμων δεδομένων είναι ακατέργαστα δεδομένα χωρίς ετικέτα. Για τον λόγο αυτό, η ανθρώπινη αλληλεπίδραση είναι απαραίτητη για την ακριβή επισήμανση δεδομένων έτσι ώστε να χρησιμοποιηθούν για μάθηση με επίβλεψη. Ωστόσο όπως είναι κατανοητό, αυτή η προσέγγιση μπορεί να είναι μια χρονοβόρα και απαιτητική διαδικασία, καθώς απαιτείται ένας μεγάλος αριθμός δεδομένων εκπαίδευσης με ακριβή σήμανση. Η μάθηση με επίβλεψη χρησιμοποιείται για την ταξινόμηση των καινούργιων δεδομένων σε καθιερωμένες

κατηγορίες (classification) καθώς και για την πρόβλεψη «τάσεων» (trends) ή μελλοντικών αλλαγών. Πιο συγκεκριμένα, ένα μοντέλο που εκπαιδεύτηκε με την βοήθεια της εποπτευόμενης μηχανικής μάθησης θα μάθει να αναγνωρίζει αντικείμενα και χαρακτηριστικά στα νέα δεδομένα εισόδου έτσι ώστε να κατηγοριοποιεί ανάλογα. Όσον αφορά τα προγνωστικά μοντέλα που εκπαιδεύονται με εποπτευόμενες τεχνικές μηχανικής μάθησης, η εκμάθηση μοτίβων μεταξύ δεδομένων εισόδου και εξόδου, τα καθιστά ικανά να προβλέψουν αποτελέσματα από καινούργια δεδομένα που δεν είχαν ξανασυναντήσει.

1.2.2 Μάθηση χωρίς επίβλεψη (Unsupervised Machine learning)

Έχοντας πλέον κατανοήσει την έννοια της μάθησης με επίβλεψη μπορούμε να χρησιμοποιήσουμε αυτή την γνώση για την κατανόηση της μη επιβλεπόμενης μάθησης. Σε ένα γενικότερο πλαίσιο η εποπτευόμενη και η μη εποπτευόμενη μηχανική εκμάθηση διαφέρουν ως προς την προσέγγιση της εκπαίδευσης και τα δεδομένα από τα οποία μαθαίνει το μοντέλο. Η ανάγκη για δεδομένα εκπαίδευσης με ετικέτα αποτελεί την κύρια και βασική διαφορά μεταξύ εποπτευόμενης (supervised) και μην εποπτευόμενης μηχανικής μάθησης (unsupervised). Όπως προαναφέραμε η μάθηση με επίβλεψη βασίζεται σε επισημασμένα δεδομένα εισόδου και εξόδου για την εκπαίδευση ενός μοντέλου. Στον αντίποδα, η μη εποπτευόμενη μηχανική μάθηση χρησιμοποιεί δεδομένα χωρίς ετικέτα ή ανεπεξέργαστα δεδομένα στην διαδικασία της εκπαίδευσης. Στην εποπτευόμενη μηχανική εκμάθηση το μοντέλο μαθαίνει τη σχέση μεταξύ των επισημασμένων δεδομένων εισόδου και εξόδου με τα μοντέλα να ρυθμίζονται με ακρίβεια έως ότου μπορούν να προβλέψουν τα αποτελέσματα των νέων δεδομένων. Από την άλλη μεριά, η μη εποπτευόμενη μηχανική εκμάθηση, μαθαίνει από μη επισημασμένα και ακατέργαστα δεδομένα εκπαίδευσης. Ένα μοντέλο αυτού του είδους έχει την δυνατότητα να μαθαίνει σχέσεις και μοτίβα μέσα σε αυτό το σύνολο δεδομένων χωρίς ετικέτα (unlabeled dataset). Για τον λόγο αυτό, χρησιμοποιείται συχνά για την ανακάλυψη εγγενών «τάσεων» (trends) και συμπεριφορών σε ένα δωσμένο σύνολο δεδομένων. Αυτό θα μπορούσε να είναι η ομαδοποίηση δεδομένων λόγω ομοιοτήτων ή διαφορών καθώς και ο εντοπισμός μοτίβων σε σύνολα δεδομένων όπως στην περίπτωση της ανίχνευσης ανωμαλιών.

1.2.3 Ημι-επιβλεπόμενη μηχανική μάθηση (Semi-supervised machine learning)

Ο συνδυασμός επιβλεπόμενης και μη επιβλεπόμενης μάθησης ονομάζεται ημι-επιβλεπόμενη μηχανική μάθηση. Σύμφωνα με αυτή την προσέγγιση, για την εκπαίδευση ενός μοντέλου ημι-επιπτευόμενης μηχανικής μάθησης, χρησιμοποιείται μια μικρή ποσότητα δεδομένων με ετικέτα και μια μεγάλη ποσότητα δεδομένων χωρίς ετικέτα. Κατα αυτόν τον τρόπο είναι δυνατή η αξιοποίηση των πλεονεκτημάτων τόσο της μη εποπτευόμενης όσο και της εποπτευόμενης μάθησης, ενώ παράληλλα αποφεύγεται η απαιτητική διαδικάσία της εύρεσης μεγάλου όγκου δεδομένων με ετικέτα. Ένα συνηθισμένο παράδειγμα μιας εφαρμογής ημι-επιβλεπόμενης μάθησης είναι ένας ταξινομητής εγγράφων κειμένου. Η εύρεση ενός μεγάλου αριθμού εγγράφων

κειμένου με ετικέτα είναι σχεδόν αδύνατη. Αυτό είναι ένα είδος προβλήματος όπου η ημι-εποπτευόμενη μηχανική μάθηση καλείται να λύσει. Αυτό συμβαίνει απλώς και μόνο επειδή είναι ανέφικτο και χρονοβόρο να έχει κάποιος διαβάσει ολόκληρα έγγραφα κειμένου μόνο και μόνο για να τους αναθέσει μια απλή ταξινόμηση.

1.2.4 Ενισχυτική μάθηση (Reinforcement Learning)

Η ενισχυτική μάθηση ή αλλιώς Reinforcement Learning είναι ένας τομέας της Μηχανικής Μάθησης που αποσκοπεύει στην λήψη κατάλληλων μέτρων για τη μεγιστοποίηση της ανταμοιβής σε μια συγκεκριμένη κατάσταση. Η ενισχυτική μάθηση αξιοποιεί αλγόριθμους που μαθαίνουν από τα αποτελέσματα και αποφασίζουν ποια ενέργεια θα ακολουθήσουν. Μετά από κάθε ενέργεια, ο αλγόριθμος λαμβάνει μία ανατροφοδότηση. Αυτή η ανατροφοδότηση είναι είτε αρνητική είτε θετική και σηματοδοτείται ως τιμωρία ή ανταμοιβή με στόχο, φυσικά, τη μεγιστοποίηση της συνάρτησης ανταμοιβής. Έτσι η RL μαθαίνει από τα λάθη της και προσφέρει στην τεχνητή νοημοσύνη την δυνατότητα να μιμείται τη φυσική νοημοσύνη όσο το δυνατόν περισσότερο. Είναι μια καλή τεχνική που χρησιμοποιείται για μία πληθώρα αυτοματοποιημένων συστημάτων που πρέπει να λαμβάνουν πολλές μικρές αποφάσεις χωρίς ανθρώπινη καθοδήγηση. Η κύρια διαφορά της ενισχυτική μάθηση από την εποπτευόμενη έγκειται στην διαφορά των δεδομένων εκπαίδευσης. Στην εποπτευόμενη μάθηση όπως προαναφέραμε τα δεδομένα εκπαίδευσης είναι κατηγοριοποιημένα και έτσι το μοντέλο εκπαιδεύεται με τη σωστή απάντηση, ενώ από την άλλη μεριά στην ενισχυτική μάθηση, δεν υπάρχει μία τέτοιου είδους «απάντηση», αλλά το μοντέλο εκπαιδεύεται μέσα απο την εμπειρία του, έτσι ώστε να αποφασίζει ποια ενέργεια να εκτελέσει για τη δεδομένη διεργασία. Επομένως θα μπορούσαμε να χαρακτηρίσουμε την ενισχυτική μάθηση ως ένα αυτόνομο, αυτοδιδακτικό σύστημα που ουσιαστικά μαθαίνει με δοκιμή και λάθος.

1.3 Τυπική διαδικασία και τρόπος λειτουργίας

Η τυπική διαδικασία που ακολουθούμε για την δημιουργία ενώς μοντέλου μηχανικής μάθησης μπορεί να περιγραφεί σε 7 γενικότερα βήματα.

1. **Συλλογή δεδομένων**
   Απαραίτητη προυπόθεση για την δημιουργία ενός μοντέλου μηχανικής μάθησης, είναι η συλλογή δεδομένων. Η ποιότητα των δεδομένων που θα χρησιμοποιηθούν στο μοντέλο θα καθορίσει πόσο ακριβές θα είναι το μοντέλο σας. Για τον λόγο αυτό είναι σημαντικό τα δεδομένα μας να είναι αξιόπιστα καθώς και να καλύπτουν σε ικανοποιητικό βαθμό όλους τους παράγωντες του εκάστοτε προβλήματος. Στην περίπτωση που τα δεδομένα είναι λανθασμένα ή παλιά, τα αποτελέσματα του μοντέλου θα είναι και αυτά εκτός πραγματικότητας.
2. **Προετοιμασία δεδομένων**
   Κατά την διαδικασία αυτή προετοιμάζουμε όλα τα δεδομένα που διαθέτουμε με σκοπό να μετατραπούν σε μία εκμεταλεύσιμή «είσοδο» για τον μοντέλο μηχανικής μάθησης.

Συγκεντρώνοντας και τυχαιοποιώντας τα δεδομένα διασφαλίζεται η ομοιόμορφη κατανομή τους. Ακόμα μία μέθοδος προετοιμασίας είναι ο καθαρισμός των δεδομένων για την κατάργηση ανεπιθύμητων τιμών, σειρών και στηλών που λείπουν ή είναι κενές. Επιπρόσθετα σημαντική είναι και η αναδιαμόρφωση του συνόλου δεδομένων με σκοπό την αλλαγή ολόκληρων γραμμών ή στήλων. Κατα την προετοιμασία των δεδομένων συνιστάται η οπτικοποίηση του συνόλου δεδομένων για την πληρέστερη κατανόηση της δομής και της σχέσης μεταξύ διαφόρων μεταβλητών και κλάσεων. Τέλος στο δεύτερο βήμα γίνεται και ο διαχωρισμός των καθαρισμένων δεδομένων σε δύο μικρότερα υποσύνολα, τα δεδομένα εκπαίδευσης από τα οποία μαθαίνει και εκπαιδεύεται το μοντέλο και τα δεδομένα δοκιμών ή τεστ που χρησιμοποιούνται για τον έλεγχο της ακρίβειας του μοντέλου μετά την εκπαίδευση.

3. **Επιλογή μοντέλου**

   Ένα μοντέλο μηχανικής μάθησης καθορίζει την έξοδο που λαμβάνεται μετά την εκτέλεση ενός αλγόριθμου μηχανικής μάθησης. Με την πάροδο του χρόνου, ειδικοί επιστήμονες και μηχανικοί σχεδίασαν διάφορα μοντέλα κατάλληλα για διαφορετικές εργασίες όπως για παράδειγμα αναγνώριση ομιλίας, αναγνώριση εικόνας, πραγματοποίηση προβλέψεων κ.λπ. Στην επιλογή του μοντέλου είναι σημαντικό να ελέγξουμε εάν το μοντέλο που θα επιλεγεί είνα κατάλληλο για αριθμητικά ή κατηγορικά δεδομένα.

4. **Εκπαίδευση του μοντέλου**

   Το τέταρτο βήμα είναι η εκπαίδευση του μοντέλου. Είναι το πιο σημαντικό βήμα στη διαδικασία της μηχανικής μάθησης. Κατά την εκπαίδευση, το μοντέλο μηχανικής εκμάθησης χρησιμοποιεί τα προετοιμασμένα δεδομένα (που είχαμε κρατήσει για την εκπαίδευση) για να εντοπίσει μοτίβα και να κάνει προβλέψεις. Κατα αυτό τον τρόπο το μοντέλο «μαθαίνει» από τα δεδομένα και είναι σε θέση να πραγματοποιήσει μία πρόβλεψη, μία αναγνώριση εικόνας ή οτιδήποτε άλλο κληθεί να κάνει όπως στην συγκεκριμένη διπλωματική εργασία, μία ανίχνευση ανωμαλιών.

5. **Αξιολόγηση του μοντέλου**

   Μετά το πέρας της εκπαίδευση, δοκιμάζεται η απόδοση του μοντέλου σε δεδομένα που δεν είχε συναντήση προηγουμένως. Τα δεδομένα αυτά είναι το σύνολο δοκιμών που δημιουργείται κατά την προετοιμασία των δεδομένων. Σε διαφορετική περίπτωση όπου η δοκιμή του μοντέλου πραγματοποιηθεί στα ίδια δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση του, η ακρίβεια της μέτρηση δεν θα είναι επαρκής, καθώς το μοντέλο θα είναι ήδη εκπαιδευμένο στα συγκεκριμένα δεδομένα και θα εντοπίζει ευκολότερα τα ίδια μοτίβα στο σύνολο αυτό. Κατα το φαινόμενο αυτό παρατηρείται μια δυσανάλογα υψηλή ακρίβεια αποτελεσμάτων γνωστή και ως overfitting.

6. **Ρύθμιση παραμέτρων**

   Το επώμενο βήμα είναι ο έλεγχος των παραμέτρων του μοντέλου με σκοπό την βελτιστοποίηση της ακρίβειας του. Παράμετροι ονομάζονται οι μεταβλητές του μοντέλου που επιρρεάζουν την ακρίβεια του και αποφασίζονται κυρίως από τον μηχανικό-χρήστη που θα το κατασκευάσει. Κάθε παράμετρος, σε μία συγκεκριμένη τιμή, θα εμφανίζει την μέγιστη δυνατή ακρίβεια. Ο συντονισμός των παραμέτρων σκοπό έχει την εύρεση αυτών των τιμών και ονομάζεται parameter tuning.

7. **Πραγματοποίηση προβλέψεων**
   Στο 7^{ο} και τελευταίο βήμα έχουμε την δυνατότητα να χρησιμοποιήσουμε το μοντέλο βάζοντας ως είσοδο εντελώς καινούργια δεδομένα με σκοπό να προβλέψουμε, να εντοπίσουμε ή να ανγνωρίσουμε με ακρίβεια.

1.4 Κίνδυνοι στην διαδικασία εκπαίδευσης του μοντέλου

Όπως αναφέραμε και παραπάνω το στάδιο της εκπαίδευσης είναι το πιο σημαντικό για την κατασκευη και δημιουργία ενός μοντέλου μηχανικής μάθησης. Χωρίς εκπαίδευση δεν υφίσταστε μηχανική μάθηση. Για τον λόγο αυτό είναι σημαντικό και απαραίτητο η διαδικασία αυτή να εκτελείται με ιδιαίτερη προσοχή έτσι ώστε να αποφευχθούν κάποιοι «αόρατοι» ή δύσκολα εντοπίσιμοι κινδύνοι όπως είναι η υπερπροσαρμογή (overfitting) και η υποπροσαρμογή (underfitting).

1.4.1 Overfitting

Η υπερπροσαρμογή ή όπως είναι ευρέως διαδεμένη με την αγγλική της ορολογία, **overfitting,** είναι μια ανεπιθύμητη συμπεριφορά μηχανικής μάθησης που εμφανίζεται όταν το μοντέλο μηχανικής εκμάθησης παρέχει ακριβείς προβλέψεις για δεδομένα εκπαίδευσης αλλά όχι για νέα δεδομένα. Επομένως όταν παρατηρείται overfitting παρατηρείται και μία δυσανάλογα υψηλή ακρίβεια αποτελεσμάτων που αγγίζει μέχρι και το 100%. Το αποτέλεσμα αυτό ακούγεται ιδανικό όμως απέχει πολύ από την πραγματική έννοια του όρου. Όταν οι ερευνητές, οι μηχανικοί και οι ειδικοί επιστήμωνες του χώρου των δεδομένων χρησιμοποιούν μοντέλα μηχανικής μάθησης για να κάνουν προβλέψεις, εκπαιδεύουν πρώτα το μοντέλο σε ένα γνωστό σύνολο δεδομένων που έχουν ξεχωρίσει για την εκπαίδευση. Ύστερα με βάση αυτές τις πληροφορίες, το μοντέλο προσπαθεί να πραγματοποιήσει προβλέψεις για νέα σύνολα δεδομένων τα οποία δεν έχει συναντήσει προηγουμένως (σύνολο δοκιμών). Ένα μοντέλο που έχει υποστεί υπερπροσαρμογή δεν μπορεί να προβλέψει με ακρίβεια σε νέα δεδομένα, ούτε να αποδώσει σε ικανοποιητικό βαθμό για όλους τους τύπους των νέων δεδομένων.

Υπάρχουν πολλές αιτίες που οδηγούν ένα μοντέλο μηχανικής μάθησης στην υπερπροσαρμογή. Πιο συγκεκριμένα, κάποιοι από τους λόγους που οδηγούν στο overfitting είναι όταν το διαθέσιμο σύνολο δεδομένων είναι πολύ μικρό και δεν περιέχει αρκετά δείγματα δεδομένων ώστε να αντιπροσωπεύουν με ακρίβεια όλες τις πιθανές τιμές δεδομένων εισόδου. Ένας ακόμα λόγος είναι ο μεγάλος αριθμός των άσχετων πληροφοριών, στα δεδομένα εκπαίδευσης, που ονομάζονται θορυβώδη δεδομένα. Επιπλέον όταν το μοντέλο εκπαιδεύεται για πολύ μεγάλο χρονικό διάστημα σε ένα μόνο δείγμα του συνόλου δεδομένων είναι σχεδόν βέβαιο ότι θα παρατηρηθεί υπερπροσαρμογή. Τέλος είναι πολύ πιθανό, η πολυπλοκότητα του μοντέλου να είναι αρκετά υψηλή, με αποτέλεσμα το μοντέλο να μαθαίνει τον θόρυβο στα δεδομένα εκπαίδευσης και ύστερα να χρησιμοποιεί τις παρατηρήσεις αυτές για νέες προβλέψεις, η οποίες θα είναι λανθασμένες.

Όσον αφορά την αντιμετώπιση της υπερπροσαρμογής σε ένα μοντέλο μηχανικής μάθησης, υπάρχουν αρκετά αντίμετρα.

Το **Cross-validation** είναι ένα ισχυρό προληπτικό μέτρο κατά του overfitting. Στην περίπτωση αυτή διαχωρίζουμε τα αρχικά δεδομένα εκπαίδευσης για να δημιουργηθούν πολλαπλά μικρότερα υποσύνολα (mini test splits). Σε μία τυπική διαδικασία k-fold cross-validation, τα δεδομένα χωρίζονται σε k υποσύνολα, που ονομάζονται folds. Στη συνέχεια, ο αλγόριθμος εκπαιδεύεται επαναλαμβανώμενα σε k-1 folds ενώ χρησιμοποιεί τα υπόλοιπα folds ως δοκιμαστικό σετ, που ονομάζεται "Holdout fold".

Ακόμα μία μέθοδος αντιμετώπισης του overfitting είναι η **εκπαίδευση σε περισσότερα δεδομένα**. Ωστόσο, αυτό δεν λειτουργεί πάντα. Εάν προσθεθούν απλώς περισσότερα θορυβώδη δεδομένα, αυτή η τεχνική δεν θα βελτιώσει το μοντέλο. Για αυτό θα πρέπει πάντα να διασφαλίζεται η αξιοπιστία και η πληρώτητα των δεδομένων.

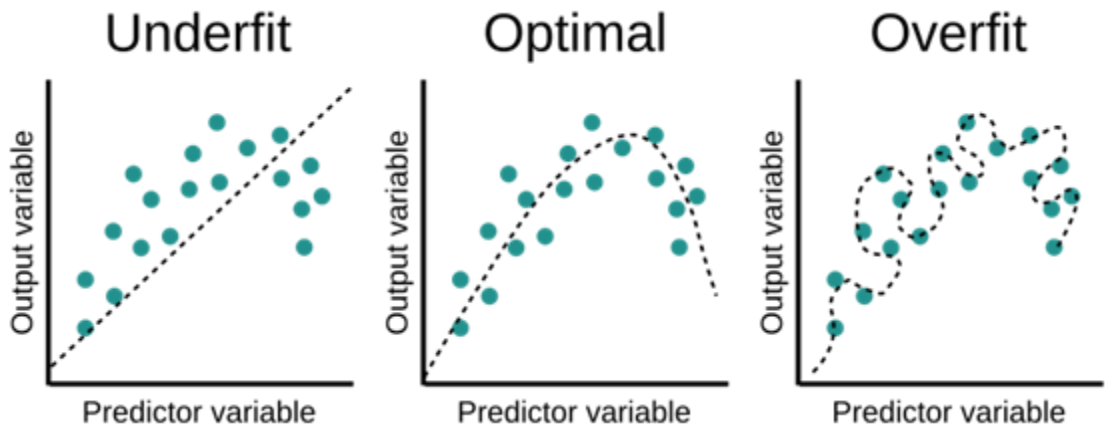
Η επόμενη μέθοδος είναι η **κατάργηση χαρακτηριστικών (feature remove)** στην οποία καταργώνται οι άσχετες πληροφορίες εισαγωγής οι οποίες είναι πιθανώς δυσνόητες.

Το **early stopping** αναφέρεται στη διακοπή της εκπαίδευσης ενός μοντέλου μηχανικής εκμάθησης πριν αυτό αρχίσει να παθαίνει overfitting. Κατά την επαναλαμβανώμενη εκπαίδευση ενώς αλγορίθμου μηχανικής μάθησης, έχουμε την δυνατότητα να μετρήσουμε την απόδοση του σε κάθε επανάληψη. Μέχρι έναν ορισμένο αριθμό επαναλήψεων, οι νέες επαναλήψεις βελτιώνουν το μοντέλο. Ωστόσο, η ικανότητα γενίκευσης του μοντέλου μπορεί να εξασθενήσει καθώς αρχίζει να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης.

Το **regularization** ή αλλιώς κανονικοποίηση, αναφέρεται στην εξαναγκασμένη απλούστευση του μοντέλου με ένα ευρύ φάσμα τεχνικών. Οι τεχνικές αυτές εξαρτόνται από τον τύπο του αλγορίθμου που χρησιμοποιείται κάθε φορά. Η αλλαγή του βάθους ενώς δέντρου απόφασης (decision tree), η εσκεμένη κατάργηση των νευρώνων (dropout) σε ένα νευρωνικό δίκτυο ή η αλλαγή του αριθμού των γειτώνων στον αλγόριθμο των K κοντινότερων γειτόνων αποτελούν παραδείγματα τέτοιων τεχνικών.

1.4.2 Underfitting

Η υποπροσαρμογή ή όπως είναι ευρύτερα διαδεδομένη με την αγγλικά ονομασία **underfitting** είναι ένας άλλος τύπος σφάλματος που εμφανίζεται όταν το μοντέλο δεν μπορεί να προσδιορίσει μια ουσιαστική σχέση μεταξύ των δεδομένων εισόδου και εξόδου. Τα μοντέλα που εμφανίζουν underfit δεν έχουν εκπαιδευτεί επαρκώς, για αρκετό χρονικό διάστημα σε ένα αξιόπιστο και μεγάλου μεγάθους σύνολο δεδομένων. Έτσι όπως είναι κατανοητό, το underfitting αντιμετωπίζεται ευκολότερα από το overfitting, απλώς αυξάνοντας τον χρόνο εκπαίδευσης, τα δεδομένα και την πολυπλοκότητα του μοντέλου.



Εικόνα 3

Στην Εικόνα 3 μπορούμε να παρατηρήσουμε την διαφορετική γραφική απεικόνιση της υποπροσαρμογής στα αριστερά και της υπερπροσαρμογής στα δεξιά. Στα διαγράμματα της εικόνας 3 οι μπλέ κουκίδες συμβολίζουν τις τιμές του συνόλου δεδομένων που χρησιμοποιούμε στο συγκεκριμένο παράδειγμα. Η διακεκομένη γραμμή αντιπροσωπεύει τις προβλεπόμενες τιμές του μοντέλου. Στο δεύτερο διάγραμμα (Optimal) παρατηρείται η ιδανικότερη και η πιο επιθυμητή κατανομή της σχέσης της προβλεπώμενης και της πραγματικής τιμής. Σε αυτή την περίπτωση και μόνο, έχουμε μία ομοιώμορφα κατανεμημένη, σε όλο το πλήθος των δεδομένων, «γραμμή» προβλέψεων.

1.5 Μέθοδοι Ensemble

Η μέθοδοι **Ensembling** συνδιάζουν προβλέψεις από πολλά ξεχωριστά μοντέλα. Οι τρείς πιο διαδεδομένες μέθοδοι ensembling είναι το bagging, το boosting και το stacking. Το **Bagging**, γνωστό και ως bootstrap agregation, είναι η μέθοδος εκμάθησης συνόλου που χρησιμοποιείται συνήθως για τη μείωση της διακύμανσης μέσα σε ένα θορυβώδες σύνολο δεδομένων. Στο bagging, ένα τυχαίο δείγμα δεδομένων σε ένα σετ εκπαίδευσης επιλέγεται με αντικατάσταση κάτι που σημαίνει ότι τα μεμονωμένα σημεία δεδομένων μπορούν να επιλεγούν περισσότερες από μία φορές. Αφού δημιουργηθούν πολλαπλά δείγματα δεδομένων, αυτά τα γενικά αδύναμα μοντέλα στη συνέχεια εκπαιδεύονται ανεξάρτητα και ανάλογα με τον τύπο της εργασίας που έχουν σκοπό να επιτελέσουν (regression, classification κλπ.). Ο μέσος όρος ή η πλειονότητα αυτών των προβλέψεων αποδίδουν μια πιο ακριβές και στοχευμένο αποτέλεσμα. Το **Boosting** ή αλλιώς ενίσχυση είναι μια μέθοδος που χρησιμοποιείται στη μηχανική μάθηση για τη μείωση των σφαλμάτων στην προγνωστική ανάλυση δεδομένων. Όπως και στο bagging έτσι και εδώ, το boosting εκπαιδεύει πολλαπλά μοντέλα διαδοχικά για να βελτιώσει την ακρίβεια του συνολικού συστήματος. Ενώ το bagging και το boosting είναι και οι δύο μέθοδοι συνόλου, διαθέτουν διαφορετικές προσεγγίσεις στην αντιμετώπιση του προβλήματος. Σε ένα γενικότερο πλαίσιο, το bagging χρησιμοποιεί πολύπλοκα βασικά μοντέλα και προσπαθεί να «εξομαλύνει» τις προβλέψεις τους, ενώ η ενίσχυση boosting χρησιμοποιεί απλά βασικά μοντέλα και προσπαθεί να

«ενισχύει» τη συνολική πολυπλοκότητά τους. Τέλος το **Stacking** είναι μια τεχνική εκμάθησης συνόλου που χρησιμοποιεί προβλέψεις για πολλούς κόμβους (όπως για παράδειγμα kNN, δέντρα αποφάσεων ή SVM) για τη δημιουργία ενός νέου μοντέλου. Αυτό το τελικό μοντέλο αποτελείται από μικρότερα μεμονωμένα μοντέλα και χρησιμοποιείται για την πραγματοποίηση προβλέψεων στο σύνολο δεδομένων δοκιμής. Τα μεμονωμένα μοντέλα εκπαιδεύονται σε διαφορετικά υποσύνολα δεδομένων χρησιμοποιώντας κάποιο τύπο τεχνικής διασταυρούμενης επικύρωσης, όπως η διασταυρούμενη επικύρωση σε k-fold, και στη συνέχεια οι προβλέψεις από κάθε μοντέλο συνδυάζονται για να γίνει η τελική πρόβλεψη. Αυτή η προσέγγιση μπορεί συχνά να οδηγήσει σε βελτιωμένη απόδοση, καθώς τα διαφορετικά μοντέλα μπορούν να μάθουν συμπληρωματικές πληροφορίες. Το Stacking είναι επίσης χρήσιμο για την αντιμετώπιση μη ισορροπημένων συνόλων δεδομένων, καθώς μπορεί να μειώσει τη διακύμανση των προβλέψεων. Επιπλέον σε αντίθεση με τις δύο προηγούμενες μεθόδους το stacking μπορεί να χρησιμοποιηθεί για τον συνδυασμό διαφορετικών τύπων μοντέλων, όπως δέντρα αποφάσεων και νευρωνικά δίκτυα και στην συνέχεια να χρησιμοποιεί τις προβλέψεις τους ως είσοδο σε ένα meta model το οποίο θα παράγει και την τελική πρόβλεψη όπως φαίνεται και στην εικόνα 4.

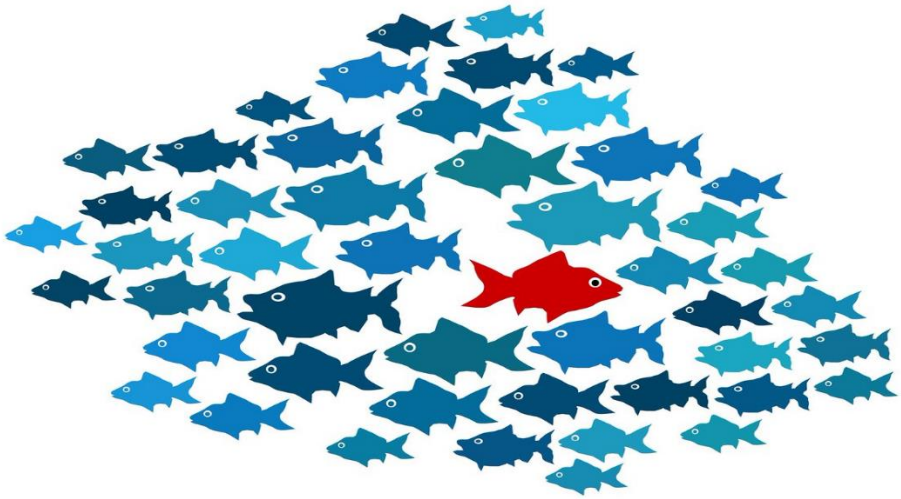
Εικόνα 4

ΚΕΦ 2^{Ο} ΑΝΙΧΝΕΥΣΗ ΑΝΩΜΑΛΙΩΝ / ANOMALY DETECTION

2.1 Τι είναι η ανίχνευση ανωμαλιών

Ανίχνευση ανωμαλιών αποκαλούμε την διαδικασία αναγνώρισης γεγονότων, στοιχείων ή παρατηρήσεων τα οποία συναντόνται σπάνια σε ένα σύνολο δεδομένων. Τα στοιχεία αυτά μπορεί να είναι μεμονωμένες τιμές ή ακόμα και «μοτίβα» συμπεριφορών τα οποία ξεχωρίζουν

σημαντικά από το υπόλοιπο dataset. Για το λόγο αυτό μπορούμε να πούμε ότι είναι ύποπτα επειδή διαφέρουν από τις τυπικές συμπεριφορές και τα πρότυπα που παρατηρούνται. Οι ανωμαλίες αυτές είναι επίσης γνωστές με τα εξής ονόματα: τυπικές αποκλίσεις, ακραίες τιμές, θόρυβος, καινοτομίες και εξαιρέσεις ή όπως αναφέρονται και με την Αγγλική τους ορολογία, standard deviations, outliers, noise, novelties, και exceptions αντίστοιχα. Σε αρκετές περιπτώσεις ανίχνευσης ανωμαλιών όπως για παράδειγμα στην ανίχνευση πιθανής εισβολής δικτύου (network intrusion) και στον εντοπισμού καταχρήσεων (abuse detection), τα ενδιαφέροντα και περίεργα συμβάντα συχνά δεν είναι σπάνια ωστόσο παραμένουν ασυνήθιστα και για αυτό τον λόγο ξεχωρίζουν από τα υπόλοιπα.

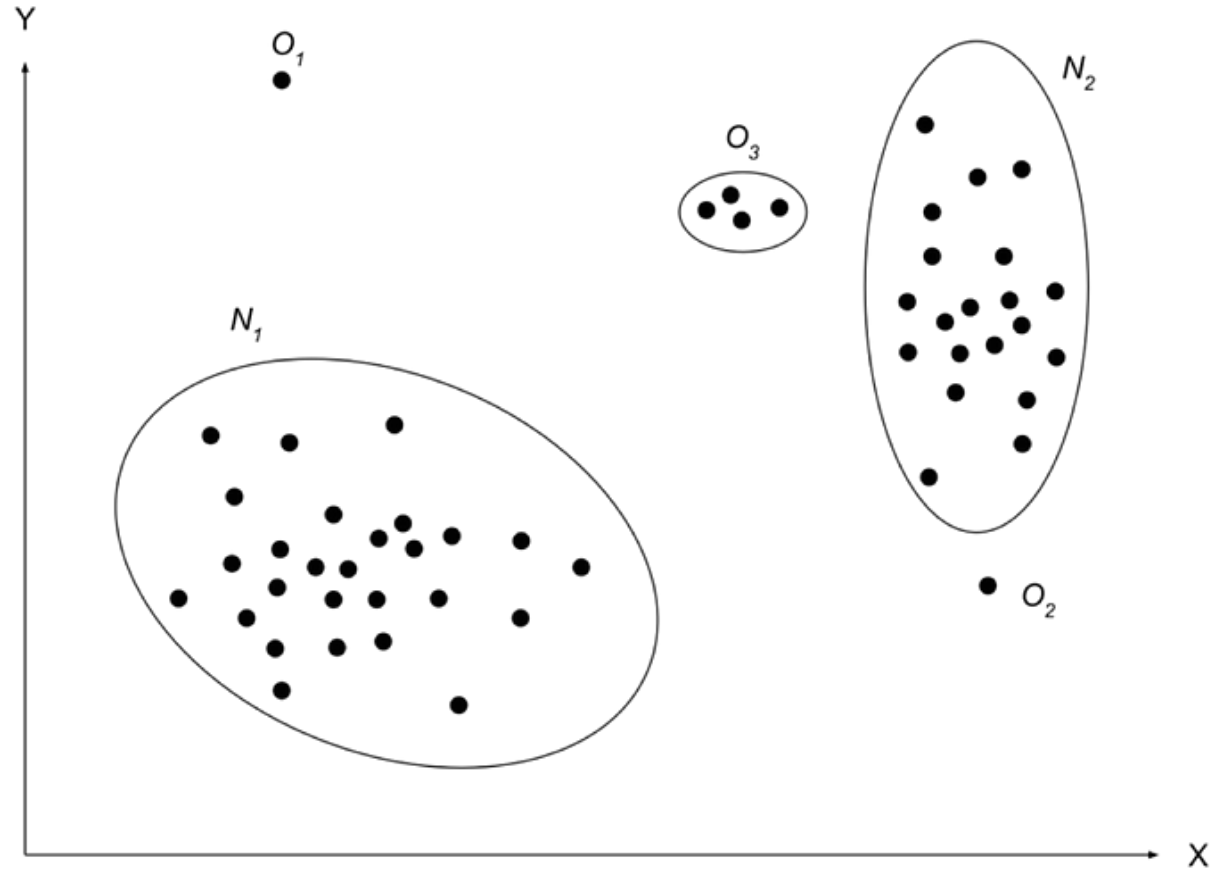


Εικόνα 5

2.2 Τι είναι οι ανωμαλίες

Ας ξεκινήσουμε την ανασκόπηση στην ανίχνευση ανωμαλιών (anomaly detection) κατανοώντας πρώτα την έννοια της «ανωμαλίας». Μια ανωμαλία είναι μια τιμή που αποκλίνει σημαντικά από τον κανόνα ώστε να θεωρείται ως σπάνια εξαίρεση. Με βάση λοιπόν αυτήν την υπόθεση αυτές οι ακραίες τιμές ή οι εξαιρέσεις θα πρέπει να ξεχωρίζουν από το σύνολο δεδομένων. Η διαδικασία ανίχνευσης προϋποθέτει πρώτα από όλα την καθιέρωση προτύπων και στη συνέχεια τον εντοπισμό των στοιχείων που παραβιάζουν αυτά τα πρότυπα.

Σύμφωνα με τα παραπάνω, ο ορισμός που μπορούμε να δώσουμε στον όρο «ανωμαλία» θα ήταν κάθε στοιχείο ή δεδομένο που δεν ταιριάζει με τον κανόνα, δηλαδή τις προσδοκίες κανονικής συμπεριφοράς ή τυπικής τιμής για ένα συγκεκριμένο σύνολο δεδομένων. Η μονάδα δεδομένων που θεωρείται ανωμαλία μπορεί να είναι πολύ μεγάλη ή πολύ μικρή σε σύγκριση με το μεγαλύτερο μέρος των δεδομένων, αποκλίνοντας έτσι από την μέση τιμή. Εάν το σύνολο δεδομένων οπτικοποιηθεί, μια ανώμαλη μονάδα δεδομένων θα απέχει σημαντικά από τις υπόλοιπα, πυκνότερα, τοποθετημένες, μονάδες δεδομένων.



Εικόνα 6

Η εικόνα 6 αποτελεί μία οπτική απεικόνηση ενός συνόλου δεδομένων σε έναν πίνακα δύο διαστάσεων. Παρατηρούμε ότι οι περισσότερες τιμές του συνόλου δεδομένων βρίσκονται στις περιοχές N1 και N2. Οι τιμές που βρίσκονται σε αυτές τις περιοχές ωρίζουν την φυσιολογική και αναμενώμενη συμπεριφορά του συνόλου δεδομένων. Ωστόσο στο παραπάνω σχήμα παρατηρούμε και κάποιες τιμές που βρίσκονται εκτός των περιοχών N1 και N2. Τα δεδομένα αυτά όπως φαίνεται και στο σχήμα είναι το O1, O2 και O3 και αποτελούν τις ανωμαλίες.

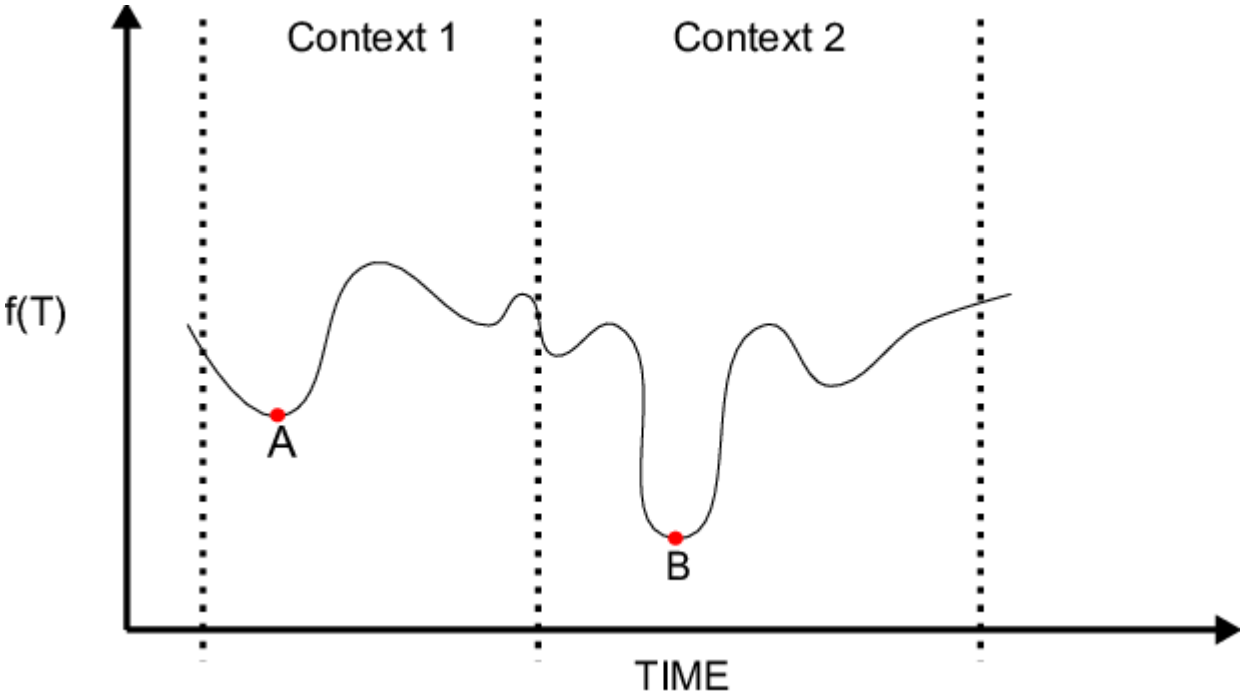
2.2.1 Κατηγορίες ανωμαλίων

Τώρα πλεόν που έχουμε μία γενικότερη εικόνα για την έννοια της «ανωμαλίας» σε ένα σύνολο δεδομένων, μπορούμε να τις ταξινομήσουμε στις ακόλουθες γενικότερες κατηγορίες:

Point Anomalies γνωστές και ως ανωμαλίες σημείου. Στην περίπτωση που ένα αντικείμενο παρατηρηθεί ότι ξεχωρίζει έναντι άλλων αντικειμένων ως ανωμαλία, τότε αυτό αποκαλείται σημειακή ανωμαλία. Η συγκεκριμένη κατηγορία αποτελεί και την απλούστερη κατηγορία ανωμαλιών με αρκετή βιβλιογραφεία και ερευνητικό έργο από διάφορους ερευνητές να

αφιερώνεται σε αυτή. Λαμβάνοντας υπόψη το παραπάνω παράδειγμα, τα σημεία O1 και O2 είναι ανωμαλίες σημείου καθώς περιλαμβάνουν μία και μόνο τιμή.

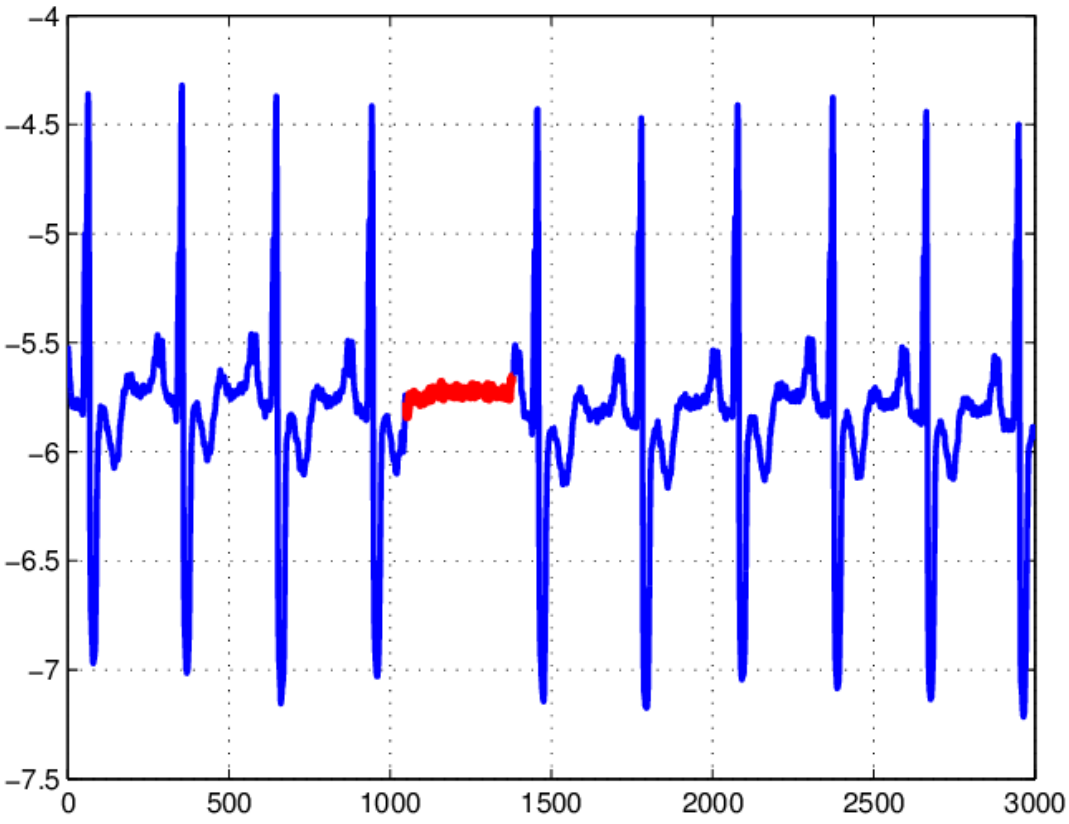
Contextual Anomalies γνωστές και ως **Conditional Anomalies** που μεταφράζονται ως συμφραζόμενες ανωμαλίες ή ανωμαλίες υπό όρους. Στην κατηγορία αυτή ανήκουν τα αντικείμενα που θεωρούνται ανωμαλίες μόνο σε κάποιο καθορισμένο πλαίσιο. Μόνο σε αυτή την περίπτωση πρόκειται για ανωμαλία συμφραζομένων. Το πλαίσιο αυτό για παράδειγμα θα μπορούσε να είναι ένα χρονικό περιθώριο και η απεικόνιση μίας ανωμαλίας αυτου του είδους θα είχε την εξής μορφή:



Εικόνα 7

Στο παραπάνω παραδειγμα της εικόνα 7 ανωμαλίες θεωρούνται οι κατώτερες τιμές του διαγράμματος. Πιο συγκεκριμένα παρατηρούμε ότι έχουμε χωρίσει τον οριζόντιο άξωνα του χρόνου σε δύο χρονικές περιοχές, Context 1 και Context 2. Στο πρώτο χρονικό περιθώριο (Context 1) η τιμή A αποτελεί την κατώτερη τιμή του υποσυνόλου και για αυτό τον λόγο θεωρείται contextual anomaly. Κατα παρόμοιο τρόπο στο δεύτερο χρονικό διάστημα (Context 2) η κατώτερη τιμή που αποτελεί και contextual ή αλλιώς conditional anomaly είναι η τιμή B. Παρόλο που η τιμή A φαίνεται να έχει μικρότερη απόκλιση από την τιμή B, σε σχέση με τις υπόλοιπες τιμές του συνόλου δεδομένων, και οι δύο αυτές τιμές όπως προαναφέραμε ξεχωρίζουν από τις υπόλοιπες μέσα στο υποσύνολο το οποίο ανήκουν.

Collective anomalies γνωστές και ως συλλογικές ανωμαλίες. Οι ανωμαλίες αυτές παρατηρούνται στην περίπτωση που ορισμένα συνδεδεμένα μεταξύ τους αντικείμενα μπορούν να παρατηρηθούν έναντι άλλων επίσης συνδεδεμένων αντικειμένων ως ανωμαλία. Στην κατηγορία αυτή δεν μπορούμε να έχουμε ένα μόνο μεμονωμένο αντικείμενο ως ανωμαλία παρά μόνο μία συλλογή-ομάδα αντικειμένων.



Εικόνα 8

Ένα παράδειγμα συλλογικής ανωμαλίας φαίνεται στο παράδειγμα της εικόνας 8. Το παράδειγμα αυτό περιλαμβάνει την έξοδο (output) ενός ανθρώπινου ηλεκτροκαρδιογραφήματος που απεικονίζει μία κολπική πρόωρη σύσπαση. Όπως μπορούμε να παρατηρήσουμε τα «μοτίβα» που επικρατούν στο μεγαλύτερο μέρος του καρδιογραφήματος είναι χρωματισμένα με μπλέ χρώμα και απεικονίζουν τους καρδιακούς παλμούς του ασθενή. Με κόκκινο χρώμα φαίνονται τα σημεία τα οποία εμφανίζουν διαφορετική συμπεριφορά σε σύγκριση με τα υπόλοιπα και για αυτό θεωρούνται ανωμαλίες με το σύνολο αυτών των κόκκινων σημείων, να ονομάζεται συλλογική ανωμαλία.

Οι ανωμαλίες διακρίνονται επίσης και με τον τρόπο απεικόνισης της εξόδου τους (δηλαδή του Output) σε **scoring-based anomalies** και σε **binary** ή **labeled anomalies**. Οι scoring-based τεχνικές ανίχνευσης ανωμαλιών εκχωρούν μία βαθμολογία ανωμαλίας σε κάθε ένα δείγμα του

συνόλου δεδομένων και στην συνέχεια οι βαθμολογίες αυτές κατατάσονται από έναν αναλυτή που επιλέγει τις ανωμαλίες ή χρησιμοποιεί κάποιο όριο για να τις επιλέξει. Από την άλλη μεριά οι binary τεχνικές ανίχνευσης ανωμαλιών κατηγοριοποιούν κάθε δείγμα του συνόλου δεδομένων με δυαδικό τρόπο, κατατάσοντας ως ανώμαλα ή ως φυσιολογικά. Οι binary τεχνικές που παρέχουν δυαδικές ετικέτες είναι κατά βάση υπολογιστικά αποτελεσματικότερες αφού κάθε στιγμιότυπο δεδομένων δεν χρειάζεται να παρέχει ή να έχει μία ξεχωριστή βαθμολογία ανωμαλίας.

2.3 Γιατί είναι σημαντική η ανίχνευση ανωμαλιών;

Σε μία κοινωνία βασισμένη στην πληροφόριση, η συνεχής ροή δεδομένων είναι απαραίτητη. Σε αυτό το πλαίσιο «βομβαρδισμού» πληροφοριών είναι σημαντικό να υπάρχει ένας έλεγχος για την ασφάλεια των προσωπικών μας δεδομένων και των πνευματικών μας δικαιωμάτων. Στα σύγχρονα λοιπόν συστήματα πληροφοριών όπως επίσης και στον τομέα της ασφάλειας πληροφοριών αναδεικνύεται περίτρανα η σημασία της ανίχνευσης ανωμαλιών. Πιο συγκεκριμένα είναι απαραίτητο να γνωρίζουμε πως ένα τυπικό σύστημα λειτουργεί εντός ορισμένων προκαθορισμένων περιορισμών. Όλα τα δεδομένα εισόδου και εξόδου, αυτού του συστήματος, θα πρέπει να συμβαδίζουν με τις τυπικές και αναμενώμενες τιμές. Οποιαδήποτε απόκλιση από αυτές τις τιμές μπορεί να εντοπιστεί με την βοήθεια τεχνικών ανίχνευσης ανωμαλιών, για να μελετηθεί περαιτέρω από το προσωπικό ασφαλείας και τους αναλυτές. Είναι επίσης σημαντικό να τονίσουμε ότι οι αποκλίσεις αυτές, δεν έχουν απόλυτα αρνητική έννοια, διότι εκτός από πιθανούς κινδύνους μπορεί να αποτελούν και μία μορφή ευκαιρίας. Πιο συγκεκριμένα παρακάτω παρατήθονται μερικά οφέλη της ανίχνευσης ανωμαλιών.

Έγκαιρη ανίχνευση προβλημάτων: Η ανίχνευση ανωμαλιών μπορεί να βοηθήσει στον εντοπισμό ασυνήθιστων συμβάντων που μπορεί να είναι ενδεικτικά ενός προβλήματος, όπως απάτη ή αστοχίες συστήματος. Ο έγκαιρος εντοπισμός αυτών των ζητημάτων μπορεί να αποτρέψει περαιτέρω ζημιά ή απώλεια.

Βελτιωμένη ασφάλεια: Ο εντοπισμός ανωμαλιών μπορεί να βοηθήσει στον ανίχνευση ασυνήθιστων μοτίβων που μπορεί να αποτελούν πιθανές παραβιάσεις ασφαλείας ή να είναι ανησυχητικές και πιθανός επικύνδυνες. Ενδεικτικά, μπορεί να χρησιμοποιηθεί για τον εντοπισμό ασυνήθιστης κίνησης δικτύου που μπορεί να είναι μία πιθανή επίθεση στον κυβερνοχώρο.

Βελτιωμένη απόδοση: Η ανίχνευση ανωμαλιών είναι ικανή να διευκολύνη τον εντοπισμό περιοχών ή πεδίων έρευνας, στις οποίες οι διαθέσιμοι πόροι μπορούν να κατανεμηθούν πιο αποτελεσματικά. Ένα παράδειγμα μίας τέτοιας περίπτωσης, μπορεί να είναι η ανίχνευση περιοχών όπου η κατανάλωση ενέργειας είναι ασυνήθιστα υψηλή, υποδεικνύοντας την ανάγκη βελτιστοποίησης και μείωσης της δαπανώμενης ποσότητας.

Βέλτιστη λήψη αποφάσεων: Η ανίχνευση ανωμαλιών μπορεί να προσφέρει πολύτιμες γνώσεις για μοτίβα και τάσεις που μπορεί να μην είναι εύκολα διακριτά. Κατα αυτό τον τρόπο μπορεί να οδηγήσει τους υπεύθυνους για την λήψη των αποφάσεων (σε μία επιχείριση ή έναν οργανισμό) σε λύσεις οι οποίες θα βασίζονται σε γνώσεις που προέρχονται από τεκμηριωμένα και αδιαμφισβήτα στοιχεία. Για παράδειγμα στις περισσότερες επιχειρήσεις, η ανάλυση KPI (Key

Performance Indicator) εξακολουθεί να είναι μια μη αυτόματη εργασία ταξινόμησης όλων των δεδομένων. Η ανάλυση KPI είναι η διαδικασία μέτρησης και αξιολόγησης βασικών δεικτών απόδοσης για την παρακολούθηση της προόδου προς την επίτευξη των οργανωτικών στόχων. Οι μετρήσεις αυτές, χρησιμοποιούνται για την παρακολούθηση και την αξιολόγηση της απόδοσης σε διαφορετικούς τομείς ενός οργανισμού, όπως για παράδειγμα οι πωλήσεις, το μάρκετινγκ και τα οικονομικά. Ανάλογα με το μέγεθος των δεδομένων που συλλέγει μία εταιρεία, αυτή η διαδικασία μπορεί να είναι μια απίστευτα χρονοβόρα εργασία. Παρόλα αυτα, με την χρήση συστημάτων ανίχνευσης ανωμαλιών, οι αλγόριθμοι τεχνητής νοημοσύνης είναι σε θέση να ελέγχουν συνεχώς όλα τα δεδομένα, αναλύοντας τις μετρήσεις αυτές αδιάκοπα. Αυτό σημαίνει ότι η διαδικασία ανάλυσης του KPI αυτοματοποιείται και έτσι δεν είναι πλέον αναγκαίος ο χειροκίνητος έλεγχος μέσα από εργαλεία BI, όπως το Google Analytics. Αντίθετα, η ανίχνευση ανωμαλιών θα ειδοποιεί αμέσως τους χρήστες όταν εντοπίζει οποιαδήποτε απόκλιση ή ασυνήθιστη συμπεριφορά στα δεδομένα, με αποτέλεσμα οι υπεύθυνοι λήψης αποφάσεων να μπορούν να χρησιμοποιήσουν αυτές τις πληροφορίες στη στρατηγική της επιχείρησης χωρίς καθυστέρησης.

Συνολικά, ο εντοπισμός ανωμαλιών είναι μία χρήσιμη μέθοδος που βρίσκει χρήση σε μία πληθώρα εφαρμογών και πεδίων, συμπεριλαμβανομένων των οικονομικών, της υγειονομικής περίθαλψης, της ασφάλειας στον κυβερνοχώρο καθώς και της βιομηχανικής παραγωγής. Ανιχνεύοντας ασυνήθιστα μοτίβα ή γεγονότα, οι οργανισμοί μπορούν να βελτιώσουν την ασφάλεια, την αποτελεσματικότητα και τη λήψη αποφάσεων, ενώ παράλληλα μειώνουν τον κίνδυνο σφαλμάτων και αποφεύγουν πιθανές απώλειες.

2.4 OUTLIER DETECTION

Η ανίχνευση ακραίων σημείων η οποία είναι ευρέως γνωστή με την αγγλική ορολογία, outlier detection, αποτελεί ένα βασικό στοιχείο για την σχεδίαση και την ανάπτυξη αλγορίθμων μηχανικής μάθησης και σκοπό έχει τον εντοπισμό ανώμαλων τιμών ή στοιχείων σε ένα δοσμένο σύνολο δεδομένων. Τα μοντέλα που χρησιμοποιούνται για την εκτέλεση του outlier detection, βασίζονται σε μεγάλα σύνολα δεδομένων για τη λειτουργία τους. Η οικονομική μοντελοποίηση, η χρηματοοικονομική πρόβλεψη, η επιστημονική έρευνα και οι καμπάνιες ηλεκτρονικού εμπορίου είναι μερικοί από τους ποικίλους τομείς στους οποίους χρησιμοποιείται ο εντοπισμός ακραίων τιμών με την χρήση μηχανικής μάθησης. Τα μοντέλα μηχανικής μάθησης μαθαίνουν από δεδομένα για να κατανοούν τις τάσεις και τη σχέση μεταξύ των σημείων δεδομένων. Βασικό συστατικό για την επίτευξη υψηλού επιπέδου ακρίβειας στην ανάπτυξη ενός μοντέλου είναι η εκπαίδευση σε μεγάλες σειρές δεδομένων. Σε ένα τέτοιο «πλούσιο» σε δεδομένα περιβάλλον, είναι αναμενώμενο ότι θα υπάρχουν ακραία δεδομένα τα οποία θα πρέπει να εντοπιστούν και να μελετηθούν. Οι ακραίες τιμές μπορεί να αλλοιώσουν τα αποτελέσματα και οι ανωμαλίες στα δεδομένα εκπαίδευσης μπορούν να επηρεάσουν τη συνολική αποτελεσματικότητα του μοντέλου. Για τον λόγο αυτό το outlier detection είναι ένα βασικό εργαλείο για τη διασφάλιση της ποιότητας των δεδομένων, καθώς τα ανώμαλα δεδομένα και τα σφάλματα μπορούν να αφαιρεθούν ή να αναλυθούν μόλις εντοπιστούν. Επιπλέον, η ανίχνευση ακραίων τιμών είναι πολύ σημαντική διότι μπορεί να χρησιμοποιηθεί σε κάθε στάδιο της διαδικασίας μηχανικής εκμάθησης. Πριν την εκπαίδευση διασφαλίζει την εγκυρότητα και την

αξιοπιστία του συνόλου δεδομένων έτσι ώστε να καθαριστεί από ανακριβή δεδομένα και ανωμαλίες αλλά και μετά την ανάπτυξη του μοντέλου για να διατηρηθεί η αποτελεσματικότητα του.

2.5 NOVELTY DETECTION

Η ανίχνευση καινοτομίας (Novelty Detection) όπως υποδηλώνει και το όνομα της, είναι η αναγνώριση πρωτότυπων ή ασυνήθιστων δεδομένων μέσα από ένα σύνολο δεδομένων. Η ανίχνευση καινοτομίας, είναι μια στατιστική μέθοδος που χρησιμοποιείται για τον προσδιορισμό νέων ή άγνωστων δεδομένων καθώς και την ταξινόμιση τους σε δεδομένα που βρίσκονται εντός του γενικού πλαισίου τιμών (τα οποία και αποκαλούμε inlier) ή εκτός αυτού (outlier). Μερικές φορές, ωστόσο, οι αλγόριθμοι ανίχνευσης καινοτομίας πρέπει να συντονιστούν για να αναζητήσουν όχι μόνο μεμονωμένα περιστατικά ασυνήθιστων δεδομένων, αλλά μάλλον ομάδες ή μοτίβα ασυνήθιστων πληροφοριών. Αυτή η εναλλακτική λύση ονομάζεται ανάλυση συμπλέγματος και είναι μια κοινή τεχνική στους αλγόριθμους τραπεζικής απάτης για την παρακολούθηση μοτίβων ύποπτης δραστηριότητας. Η ανίχνευση καινοτομίας είναι μία από τις θεμελιώδεις απαιτήσεις για ένα σωστό σύστημα ταξινόμησης και στη μηχανική μάθηση. Επιπλέον στα συστήματα μηχανικής μάθησης, πολλές φορές δεν μπορούν να συμπεριληφθούν όλοι οι πιθανοί συνδιασμοί δεδομένων κατά τη διάρκεια της εκπαίδευσης. Επομένως θα υπάρχουν πάντα νέα είδη δεδομένων και συνδιασμοί που δεν θα έχουν παρατηρηθεί προηγούμενα. Στην ανίχνευση σφαλμάτων και απάτης, για παράδειγμα, το σύστημα είναι εκπαιδευμένο να ανιχνεύει δεδομένα που έχουν υποεκπροσωπηθεί ή δεν έχουν εμφανιστεί καθόλου, καθώς πρόκειται για πιθανά σφάλματα. Για παράδειγμα ένα τέτοιο «σφάλμα» στα συστήματα ιατρικών δεδομένων, αυτό θα μπορούσε να αντιπροσωπεύει μία πιθανή ασθένεια. Για αμιγώς συστήματα ανίχνευσης καινοτομίας, το δίκτυο εκπαιδεύεται στα αρνητικά παραδείγματα και στη συνέχεια εντοπίζει μόνο εισόδους που δεν ταιριάζουν σε αυτό το μοντέλο ως νέα κατηγορία. Η αναγνώριση ότι μια είσοδος διαφέρει από τις προηγούμενες εισροές είναι μια πολύ σημαντική και χρήσιμη ικανότητα για τα συστήματα μάθησης. Αυτό συνεπάγεται ότι το σύστημα έχει την δυνατότητα να μαθαίνει πραγματικά και όχι απλώς να αντιδρά σε προηγούμενες εισόδους στις οποίες είχε προηγουμένως εκπαιδευτεί. Στην περίπτωση των ζώων και των ανθρώπων, ασκούμε ανίχνευση καινοτομίας συνεχώς, καθόλη την διάρκεια της ζωής μας. Πιο συγκεκριμένα η ανίχνευση καινοτομίας που ασκούμε καθημερινά προσδιορίζεται στην ικανότητα διάκρισης αντικειμένων από άλλα αντικείμενα. Για παράδειγμα, βλέπουμε έναν απλό λευκό τοίχο ενώ παράλληλα παρατηρούμε μια κηλίδα να κινείται στην επιφάνειά του. Αμέσως την διαχωρίζουμε από τον τοίχο αναγνωρίζοντας ότι αποτελεί ένα διαφορετικό αντικείμενο, πιθανώς ένα έντομο.

Εν κατακλείδι, σε ένα γενικότερο πλαίσιο, θα μπορούσαμε να πούμε ότι η ανίχνευση ακραίων τιμών (outlier detection) και η ανίχνευση καινοτομίας (novelty detection) είναι δύο τεχνικές που χρησιμοποιούνται και οι δύο για την ανίχνευση ανωμαλιών, οπουδήποτε απαιτηθεί η ανίχνευση μη φυσιολογικών ή ασυνήθιστων τιμών και παρατηρήσεων. Η ανίχνευση ακραίων τιμών είναι επίσης γνωστή ως ανίχνευση ανωμαλίας χωρίς επίβλεψη και η ανίχνευση καινοτομίας ως ημι-εποπτευόμενη ανίχνευση ανωμαλιών.

2.6 ΒΙΟΜΗΧΑΝΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

2.6.1 Σημασία της βιομηχανίας

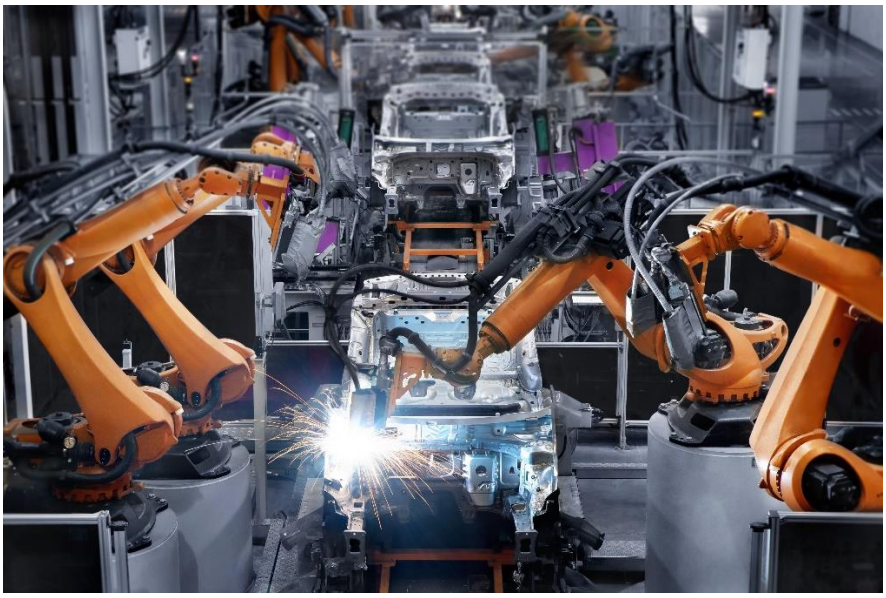
Η βιομηχανία είναι ένας θεμελιώδης παράγοντας στην οικονομία κάθε χώρας και είναι υπεύθυνη για την επεξεργασία και τη μετατροπή των φυσικών πόρων σε άλλα προϊόντα. Η σημασία της βιομηχανίας έγινε ιδιαίτερα αισθητή μετά τη Βιομηχανική Επανάσταση που έλαβε χώρα στην Αγγλία τον 18ο αιώνα χάρη στην εφεύρεση νέων μηχανών που πραγματοποιούσαν τις εργασίες που εκτελούσαν προηγουμένως άνθρωποι με μεγαλύτερη αποτελεσματικότητα. Η εκβιομηχάνιση υπήρξε η κινητήρια δύναμη πίσω από την άνθιση της οικονομίας. Αυτό συμβαίνει διότι η μαζική παραγωγή δημιουργεί οικονομίες κλίμακας. Πιο συγκεκριμένα όσο περισσότερες μονάδες παράγονται, τόσο χαμηλότερο είναι το κόστος ανά μονάδα και επομένως αυξάνεται η αξία των εκροών ανά εισροή. Επιπλέον η βιομηχανία τείνει να έχει ισχυρούς δεσμούς με άλλα μέρη της οικονομίας, δημιουργώντας ζήτηση για νέες δεξιότητες στον τομέα αυτό. Αυτό σημαίνει ότι η ανάπτυξη της βιομηχανίας ενισχύει παράλληλα και την ανάπτυξη σε ένα ευρύτερο σύνολο δραστηριοτήτων, συμπεριλαμβανομένου του τομέα των υπηρεσιών. Τέλος οι περισσότερες καινοτομίες και τεχνολογικές εξελίξεις προέρχονται κατά βάση από τον βιομηχανικό τομέα, ο οποίος μπορεί στη συνέχεια να τροφοδοτήσει άλλους οικονομικούς και κοινωνικούς τομείς μίας χώρας, καθιστώντας τους επίσης πιο παραγωγικούς.

2.6.2 Condition Monitoring

Σε ένα σύγχρονο βιομηχανικό περιβάλλον, η μηχανική μάθηση βρίσκει εφαρμογή σε πολλούς τομείς. Μία εξαιρετικά χρήσιμη και συνηθισμένη εφαρμογή της είναι η παρακολούθηση συνθηκών ή αλλιώς condition monitoring. Πρόκειται για μία διαδικασία μέτρησης των παραμέτρων των μηχανών όπως θερμοκρασίες, κραδασμοί, πιέσεις , κατανάλωση ρεύματος κ.λπ, προκειμένου να εντοπιστούν και να αποτραπούν αστοχίες (ή αλλιώς ανωμαλίες). Ωστόσο, η μέχρι πρότεινος εφαρμογή της παρακολούθησης κατάστασης ενός μηχανήματος δεν ήταν ικανή να διαχειριστεί τον κατακλυσμό δεδομένων στον σημερινό κόσμο. Για να βρεθούν αυτές οι ανωμαλίες στην λειτουργία ενός βιομηχανικού μηχανήματος και να βελτιωθεί η παραγωγικότητα ολόκληρης της γραμμής παραγωγής, η παραδοσιακή παρακολούθηση της κατάστασης πρέπει να συνδυαστεί με την επιστήμη δεδομένων και τη μηχανική μάθηση. Μια μελέτη της McKinsey εκτίμησε ότι η κατάλληλη χρήση τεχνικών και διεργασιών που βασίζονται στην εμπεριστατωμένη χρήση δεδομένων, συνήθως μειώνει το χρόνο διακοπής λειτουργίας του μηχανήματος κατά 30 έως 50 % και αυξάνει τη διάρκεια ζωής του μηχανήματος κατά 20 έως 40 %. Κατα αυτόν τον τρόπο φανερώνεται περίτρανα ότι με τη διασταύρωση της παραδοσιακής βιομηχανίας, της επιστήμης δεδομένων και της μηχανικής μάθησης, είμαστε σε θέση να πετύχουμε το μέγιστο κέρδος.

2.6.3 Σημασία συντήρησης βιομηχανικού εξοπλισμού

Τα βιομηχανικά μηχανήματα επιρρεάζουν άμεσα ολόκληρη την λειτουργία της παραγωγικής διαδικασίας από την αποτελεσματικότητα της έως και το κέρδος της επιχείρησης. Σε πολλές περιπτώσεις μάλιστα τα βιομηχανικά μηχανήματα λειτουργούν αδιάκοπα καθημερινά με αποτέλεσμα να φθείρονται και να καταπονούνται σε μεγαλύτερο βαθμό. Ως εκ τούτου, το όφελός κάθε επιχειρηματία-βιομήχανου είναι να διασφαλίσει με κάθε μέσο τη λειτουργικότητα του βιομηχανικού εξοπλισμού που διαθέτει, καθώς και να αποτρέψει τυχόν προβλήματα που μπορεί να εμφανιστούν με την χρήση του conditional monitoring. Για την επίτευξη αυτών των στόχων η συντήρηση των βιομηχανικών μηχανημάτων καθιστάται υψίστης σημασίας.



Εικόνα 9

Αναλυτικότερα, μερικά από τα ωφέλη της συντήρησης του βιομηχανικού αναφαίρονται παρακάτω:

Αύξηση της παραγωγικότητας

Όταν ο βιομηχανικός εξοπλισμός είναι σε βέλτιστη κατάσταση, οι πιθανότητες καθυστερήσεων των προϊόντων ή εμφάνισης βλαβών, μειώνονται σημαντικά. Επιπρόσθετα η συντήρηση έχει ως στόχο την αντιμετώπιση των προβλημάτων μόλις εμφανιστούν και προτού κλιμακωθούν, γεγονός που μπορεί να βελτιώσει την ποιότητα των βιομηχανικών διαδικασιών και της παραγωγής. Επιπλέον τα μηχανήματα που λειτουργούν σωστά μειώνουν το βάρος της εργασίας για ολόκληρη την εγκατάσταση.

Οικονομική ανάπτυξη

Η συντήρηση του βιομηχανικού εξοπλισμού περιορίζει κατά έναν σημαντικό βαθμό τα βαριά έξοδα από συνεχείς επισκευές. Το αρχικό κόστος για τη σύνταξη ενός σχεδίου συντήρησης

μπορεί να είναι αρκετά σημαντικό, ωστόσο το κόστος αυτό θα αντισταθμιστεί στην πορεία από την εξοικονόμηση χρόνου και χρημάτων από την άριστη λειτουργία μιας εγκατάστασης με πλήρως λειτουργικό και συντηρημένο εξοπλισμό.

Βελτίωση της Ασφάλειας

Η συντήρηση των βιομηχανικών μηχανημάτων βοηθά στην αποφυγή ανεπιθύμητων ατυχημάτων στις εγκαταστάσεις, βελτιστοποιόντας έτσι την ασφάλεια στον εργασιακό χώρο. Κατά αυτόν τον τρόπο διασφαλίζεται η υγεία και η ευημερία των εργαζομένων, καθώς επίσης διατηρείται σε μεγάλο βαθμό η ποιότητα της εργασίας. Σύμφωνα με τα στατιστικά στοιχεία, υπολογίζεται ότι περίπου το 15-20% των ατυχημάτων που συμβαίνουν σε βιομηχανικές εγκαταστάσεις συνδέονται με την έλλειψη σωστής συντήρησης.

ΚΕΦ. 3ο Προετοιμασία και υλοποίηση

3.1 Dataset

Όπως προαναφέραμε η συγκεκριμένη πτυχιακή εργασία σκοπό έχει την ανίχνευση ανωμαλιών πάνω στις τιμές της θερμοκρασία ενός βιομηχανικού μηχανήματος. Για την υλοποίηση της ανίχνευσης αυτής γίνεται χρήση του Numenta Anomaly Benchmark (NAB). Το NAB αποτελείται από μία πληθώρα dataset που βρίσκουν εφαρμογή σε ένα τεράστιο εύρος πεδίων όπως η μηχανική, η πωλήσεις κλπ. Όλα τα σύνολα δεδομένων που περιλαμβάνονται στο Numenta Anomaly Benchmark είναι ειδικά σχεδιασμένα για την ανιχνευση ανωμαλιών καθώς περιέχουν πραγματικά δεδομένα από βιομηχανιές και επιχειρήσεις που συνοδεύονται με επιβεβαιωμένες και καταγραμένες ανωμαλίες. Πιο συγκεκριμένα πρόκειται για ένα πρωτοποριακό dataset για την αξιολόγηση αλγορίθμων στην ανίχνευση ανωμαλιών σε συνεχείς διαδικτυακές εφαρμογές. Αποτελείται από περισσότερα από 50 επισημασμένα αρχεία δεδομένων πραγματικού κόσμου και τεχνητών χρονοσειρών συν έναν νέο μηχανισμό βαθμολόγησης που έχει σχεδιαστεί για εφαρμογές σε πραγματικό χρόνο. Επιπρόσθετα το Numenta Anomaly Benchmark διαθέτει και ένα github account στο οποίο αναγράφονται χρήσιμες πληροφορίες για κάθε αρχείο δεδομένων που παρέχεται. Σήμα κατατεθέν του NAB είναι οι χρονοσειρές καθώς όλα τα διαθέσιμα αρχεία δεδομένων παρουσιάζονται σε συνάρτηση με τον χρόνο. Μεγάλο μέρος των παγκόσμιων δεδομένων είναι ροή δεδομένων χρονοσειρών, όπου οι ανωμαλίες δίνουν σημαντικές πληροφορίες σε κρίσιμες καταστάσεις. Τα παραδείγματα αφθονούν σε τομείς όπως τα οικονομικά, η πληροφορική, η ασφάλεια, η ιατρική και η ενέργεια. Ωστόσο, ο εντοπισμός ανωμαλιών στη ροή δεδομένων είναι μια δύσκολη εργασία, καθώς απαιτεί από τους ανιχνευτές να επεξεργάζονται δεδομένα σε πραγματικό χρόνο και να μαθαίνουν κάνοντας ταυτόχρονα προβλέψεις. Δεν υπάρχουν σημεία αναφοράς για την επαρκή δοκιμή και αξιολόγηση της αποτελεσματικότητας των μοντέλων ανίχνευσης ανωμαλιών σε πραγματικό χρόνο. Την ανάγκη αυτή έρχεται να καλύψει το Numenta Anomaly Benchmark, το

οποίο επιχειρεί να παρέχει ένα ελεγχόμενο και επαναλαμβανόμενο περιβάλλον εργαλείων ανοιχτού κώδικα για τη δοκιμή και τη μέτρηση αλγορίθμων ανίχνευσης ανωμαλιών σε δεδομένα ροής. Ένας ιδανικός ανιχνευτής πρέπει να είναι σε θέση να ανιχνεύει όλες τις ανωμαλίες το συντομότερο δυνατό, χωρίς να προκαλεί ψευδείς συναγερμούς, να δουλεύει με δεδομένα χρονοσειρών του πραγματικού κόσμου σε μια ποικιλία τομέων και να προσαρμόζεται αυτόματα στις μεταβαλλόμενες στατιστικές. Η επιβράβευση αυτών των χαρακτηριστικών διευκολύνεται στο NAB, χρησιμοποιώντας έναν αλγόριθμο βαθμολόγησης που έχει σχεδιαστεί για ροή δεδομένων. Το Numenta Anomaly Benchmark αξιολογεί τους ανιχνευτές με βάση ένα σύνολο δεδομένων αναφοράς με επισημασμένα δεδομένα σε χρονοσειρές πραγματικού κόσμου. Τα στοιχεία αυτά δίνουν αποτελέσματα και αναλύσεις για διάφορους αλγόριθμους ανοιχτού κώδικα. Ο στόχος για το NAB είναι να παρέχει ένα πρότυπο πλαίσιο ανοιχτού κώδικα με το οποίο η ερευνητική κοινότητα μπορεί να συγκρίνει και να αξιολογήσει διαφορετικούς αλγόριθμους για τον εντοπισμό ανωμαλιών στη ροή δεδομένων.

Πιο συγκεκριμένα το αρχείο δεδομένων που χρησιμοποιείται στην παρούσα διπλωματική εργασία ονομάζεται «machine temperature system failure» και ανήκει στην κατηγορία «realKnownCase» του NAB. Πρόκειται για ένα σύνολο δεδομένων που αποτελείται απο αληθινά δεδομένα τα οποία προήλθαν από μία βιομηχανία. Περιλαμβάνει τις τιμές της θερμοκρασίας ενός μεγάλου βιομηχανικού μηχανήματος σε συνάρτηση με τον χρόνο. Κάθε θερμοκρασιακή μέτρηση απέχει από την προηγούμενη χρονικό διάστημα 5 λεπτών. Οι μετρήσεις αυτές ξεκινούν στις 2 Δεκεμβρίου του 2013 και τελειώνουν στις 19 Φεβρουαρίου του 2014. Σε αυτό το χρονικό διάστημα των 3 μηνών συγκεντρώνονται συνολικά 22.695 τιμές θερμοκρασίας. Επιπρόσθετα στην Github ιστοσελίδα του Numenta Anomaly Benchmark αναγράφονται τέσσερις περιπτώσεις στις οποιές καταγράφηκε μία σίγουρη θερμοκρασιακή ανωμαλία. Πιο συγκεκριμένα οι περιπτώσεις αυτές είναι 4 διαφορετικά χρονικά διαστήματα. Η πρώτη ανωμαλία (από τις 10 έως τις 12 Δεκεμβρίου του 2013) φανερώνει μία απότομή και σχεδόν κατακόρυφη αύξηση της θερμοκρασίας του βιομηχανικού μηχανήματος. Η δεύτερη ανωμαλία παρατηρήθηκε από τις 15 έως και τις 17 Δεκεμβρίου του 2013 και ήταν μία προγραμματισμένη διακοπή λειτουργίας του μηχανήματος στην οποία είναι αναμενώμενη η πτώση της θερμοκρασίας. Η τρίτη ανωμαλία έλαβε χώρα από τις 27 έως και τις 29 Ιανουαρίου του 2014 και ήταν δύσκολο να εντοπιστεί. Η ανωμαλία αυτή ήταν και η αιτία που οδήγησε άμεσα στην τέταρτη και τελευταία καταγεγραμένη ανωμαλία, μια καταστροφική βλάβη του μηχανήματος (από τις 7 έως τις 9 Φεβρουαρίου του 2014).

3.2 Περιβάλλον υλοποίησης

3.2.1 Anaconda

Βασική προυπόθεση για την σωστή λειτουργία του κώδικα που δημιουργήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας είναι η κατασκευή ενός ψηφιακού περιβάλλοντος (virtual

environment) με την χρήση του Anaconda. Το Anaconda Navigator είναι μια γραφική διεπαφή για την εκκίνηση κοινών προγραμμάτων Python χωρίς να είναι απαραίτητη η χρήση εντολών στην γραμμή εντολών, για την εγκατάσταση πακέτων και τη διαχείριση των περιβαλλόντων. Επιπρόσθετα το Anaconda Navigator επιτρέπει την εκκίνηση εφαρμογών και την ευκολότερη διαχείρηση πακέτων και περιβάλλοντων, παρακάμπτοντας πάλι την γραμμή εντολών. Το Navigator μπορεί να αναζητήσει πακέτα στο Anaconda Cloud ή σε ένα τοπικό αποθετήριο Anaconda και είναι διαθέσιμο για μία πληθώρα λειτουργικών συστημάτων όπως Windows, macOS και Linux.Συμπερασματικά το Anaconda Navigator είναι ένας διαδραστικός και φιλικός προς τον χρήστη τρόπος για εργασία με πακέτα και ψηφιακά περιβάλλοντα χωρίς να χρειάζεται η χρήση εντολών conda σε ένα κάποιο terminal. Τέλος παρέχετε και η δυνατότητα εύρεσης νέων πακέτων για την εγκατάσταση τους, την εκτέλεση τους και την ενήμερωση τους σε ένα εικονικό περιβάλλον, πάντα με την χρήση του Navigator.

Για τους σκοπούς της άσκησης δημιουργήθηκε ένα virtual environment με την ονομασία «gpu_env», με σκοπό να αποτελέσει ένα φιλικό περιβάλλον για την ορθή λειτουργία του προγράμματος. Πρόκειται για ένα περιβάλλον σχεδιασμένο να λειτουργήσει με την γλώσσα προγραμματισμού python στο οποίο εγκαταστήθηκαν ορισμένες βιβλιοθήκες. Μεταξύ των άλλων το gpu_env, περιλαμβάνει τις βιβλιοθήκες: Pandas, Numpy, Scikit-learn, Holoviews και Matplotlib οι οποίες θα αναλυθούν περαιτέρω στην συνέχεια.

3.2.2 Jupyter Notebook

Για την υλοποίηση του πρακτικού μέρους της διπλωματικής αυτής εργασίας έγινε χρήση του Jupyter Notebook. Το notebook επεκτείνει την προσέγγιση που βασίζεται στον διαδραστικό προγραμματισμό σε μια ποιοτικά νέα κατεύθυνση, παρέχοντας μια διαδικτυακή εφαρμογή κατάλληλη για την αποτύπωση ολόκληρης της διαδικασίας υπολογισμού: ανάπτυξη, τεκμηρίωση και εκτέλεση κώδικα, καθώς και επικοινωνία των αποτελεσμάτων. Ένα Jupyter notebook αποτελείται από δύο μέρη. Το πρώτο μέρος είναι ένα web application που είναι ένα εργαλείο το οποίο βασίζεται σε ένα πρόγραμμα περιήγησης για την διαδραστική σύνταξη εγγράφων που συνδυάζουν επεξηγηματικό κείμενο, μαθηματικά, υπολογισμούς και την παραγωγή εμπλουτισμένων μέσων. Το δεύτερο μέρος ονομάζεται notebook document και αποτελεί μια αναπαράσταση όλου του περιεχομένου που είναι ορατό στο web application, συμπεριλαμβανομένων των εισόδων και των εξόδων των υπολογισμών, του επεξηγηματικού κειμένου, των μαθηματικών, των εικόνων και των αναπαραστάσεων με εμπλουτισμένα μέσα αντικειμένων.

3.3 Βιβλιοθήκες

3.3.1 Pandas

Η βιβλιοθήκη pandas είναι ένα πακέτο Python που παρέχει γρήγορες, ευέλικτες και διαδραστικές δομές δεδομένων που έχουν σχεδιαστεί για να κάνουν την εργασία με "συνεχή" ή "κατηγορηματικά" δεδομένα τοσό εύκολη όσο και γρήγορη. Σκοπός της βιβλιοθήκης pandas είναι να αποτελέσει το θεμελιώδες δομικό στοιχείο υψηλού επιπέδου για την πραγματοποίηση πρακτικής ανάλυσης δεδομένων πραγματικού κόσμου με την χρήση της ευρέως διαδεδομένης γλώσσας προγραμματισμού Python. Επιπλέον, έχει τον ευρύτερο στόχο να γίνει το πιο ισχυρό και ευέλικτο εργαλείο ανάλυσης και χειρισμού δεδομένων ανοιχτού κώδικα διαθέσιμο σε οποιαδήποτε γλώσσα. Μεταξύ των άλλων η Pandas προσφέρει εύκολος χειρισμός δεδομένων που λείπουν (τα οποία αντιπροσωπεύονται είτε ως NaN, NA ή NaT) σε δεδομένα κινητής υποδιαστολής καθώς και σε δεδομένα μη κινητής υποδιαστολής. Ακόμα ένα πλεονέκτημα της είναι η μεαβλητότητα που προσδίδει στο μεγέθους ενός DataFrame, με την δυνατότητα αντικατάστασης, προσθήκης ή και διαγραφής στηλών ή γραμμών. Επιπρόσθετα η αυτόματη και ρητή στοίχιση των δεδομένων επιτρέπει στα αντικείμενα να ευθυγραμμιστούν σε ένα σύνολο ετικετών ή ο χρήστης με την σειρά του μπορεί απλώς να αγνοήσει τις ετικέτες και να αφήσει το Series, το DataFrame κ.λπ. να ευθυγραμμίσουν αυτόματα τα δεδομένα. Η pandas έχει ακόμα την δυνατότητα να εκτελέσει λειτουργίες split-apply-combine σε σύνολα δεδομένων, τόσο για τη συγκέντρωση όσο και για τη μετατροπή δεδομένων. Επιπλέον κάνει ευκολότερη τη μετατροπή δεδομένων με διαφορετικό ευρετήριο (differently-indexed data) από άλλες δομές δεδομένων Python και Numpy σε αντικείμενα DataFrame. Η συγκεκριμένη βιβλιοθήκη δίνει την δυνατότητα συγχώνευσης και ένωσης, αναδιαμόρφωσης και περιστροφής συνόλων δεδομένων καθώς επιτρέπει ακόμα και την ιεραρχική επισήμανση των αξόνων τους. Η Pandas περιέχει ισχυρά εργαλεία IO για φόρτωση δεδομένων από αρχεία CSV, αρχεία Excel, βάσεις δεδομένων και αποθήκευση/φόρτωση δεδομένων από την εξαιρετικά γρήγορη μορφή HDF5. Τέλος προσφέρει και πληθώρα εργαλείων για την μετατροπή, ανάλυση και αξιοποίηση των χρονοσειρών σε ένα αντίστοιχο σύνολο δεδομένων.

3.3.2 NumPy

Το NumPy είναι το θεμελιώδες πακέτο για την επιστημονικούς υπολογισμούς στην Python. Είναι μια βιβλιοθήκη Python που παρέχει πολυδιάστατους πίνακες αντικειμένων (multidimensional array), διάφορα παράγωγα αντικείμενα και μια ποικιλία γρήγορων λειτουργιών σε πίνακες, συμπεριλαμβανομένων μαθηματικών πράξεων, διακριτών μετασχηματισμών Fourier, βασικών αρχών γραμμικής άλγεβρας, βασικών στατιστικών λειτουργιών και πολλά άλλα. Στον πυρήνα του πακέτου NumPy, βρίσκεται το αντικείμενο ndarray. Το ndarray έχει ως στόχο την ενσωμάτωση πολυδιάστατων συστοιχείων ομοιογενών τύπων δεδομένων, με πολλές λειτουργίες να εκτελούνται σε μεταγλωττισμένο κώδικα για βελτιωμένη απόδοση. Επιπρόσθετα οι συστοιχίες NumPy έχουν σταθερό μέγεθος κατά τη

δημιουργία τους, σε αντίθεση με τις λίστες Python (οι οποίες μπορούν να αναπτυχθούν δυναμικά). Η αλλαγή του μεγέθους ενός ndarray θα δημιουργήσει έναν νέο πίνακα και θα διαγράψει τον πρωτότυπο. Τα στοιχεία σε έναν αριθμητικό πίνακα NumPy απαιτούνται όλα να είναι του ίδιου τύπου δεδομένων έτσι ώστε να έχουν το ίδιο μέγεθος στη μνήμη. Ωστόσο μπορεί κανείς να έχει συστοιχίες (Python, συμπεριλαμβανομένων των NumPy) αντικειμένων, επιτρέποντας έτσι την ύπαρξη στοιχείων διαφορετικού μεγέθους σε έναν πίνακα. Επιπλέον οι NumPy συστοιχίες διευκολύνουν την εφαρμογή προηγμένων μαθηματικών και άλλων τύπων λειτουργιών σε ένα μεγάλο αριθμό δεδομένων. Κατα κανόνα, αυτές οι λειτουργίες εκτελούνται πιο αποτελεσματικά με την χρήση του NumPy και φυσικά απαιτούν λιγότερο κώδικα από ότι οι ενσωματωμένες ακολουθίες της Python. Μια αυξανόμενη πληθώρα επιστημονικών και μαθηματικών πακέτων που βασίζονται σε Python χρησιμοποιούν NumPy arrays. Τέτοια πακέτα συνήθως υποστηρίζουν είσοδο ακολουθίας Python, ωστόσο αρκετά από αυτά μετατρέπουν τέτοιες εισόδους σε NumPy arrays πριν από την επεξεργασία και συχνά δίνουν ως έξοδο NumPy arrays. Με άλλα λόγια, για την αποτελεσματική χρήση πολλών λειτουργιών από το σημερινό επιστημονικό και μαθηματικό λογισμικό (που βασίζεται στην Python), είναι ανεπαρκής η απλή γνώση των ενσωματωμένων τύπων ακολουθίας της Python καθώς η γνώση της χρήσης των NumPy arrays καθιστάται απαραίτητη.

3.3.3 Scikit-learn

Το Scikit-learn είναι ίσως μία από τις πιο χρήσιμες βιβλιοθήκες για μηχανική μάθηση με την χρήση της Python. Η βιβλιοθήκη Scikit-learn γνωστή και ως sklearn περιέχει πολλά αποτελεσματικά εργαλεία για μηχανική μάθηση και στατιστική μοντελοποίηση, συμπεριλαμβανομένης της ταξινόμησης, της παλινδρόμησης, της ομαδοποίησης και της μείωσης των διαστάσεων. Μεταξύ των άλλων η βιβλιοθήκη Scikit-learn περιλαμβάνει αλγόριθμους εποπτευόμενης μάθησης. Ξεκινώντας από γενικευμένα γραμμικά μοντέλα (π.χ. Γραμμική Παλινδρόμηση), Support Vector Machines (SVM), δέντρα αποφάσεων έως μεθόδους Bayesian – όλα αυτά αποτελούν μέρος της εργαλειοθήκης scikit-learn. Η εξάπλωση των αλγορίθμων μηχανικής μάθησης είναι ένας από τους σημαντικότερους λόγους για την υψηλή ζήτηση και χρήση του scikit-learn. Επιπλέον περιλαμβάνει αλγορίθμους για Cross-validation με μία ποικιλία μεθόδων για τον έλεγχο της ακρίβειας των μοντέλων μηχανικής μάθησης με επίβλεψη σε νέα δεδομένα. Από το sklearn δεν λείπουν επίσης και οι αλγόριθμοι μή επιβλεπόμενης μάθηση. Όπως και στην μάθηση με επίβλεψη έτσι και στην μάθηση χωρίς επίβλεψη υπάρχει μεγάλη ποικιλία αλγορίθμων μηχανικής μάθησης στην συγκεκριμένη βιβλιοθήκη. Από την ομαδοποίηση (clustering), την ανάλυση παραγόντων, την ανάλυση κύριων συστατικών (principal component analysis) έως και στα μη εποπτευόμενα νευρωνικά δίκτυα. Επιπρόσθετα η Scikit-learn περιέχει διάφορα μικρά σύνολα δεδομένων. Αυτά τα σύνολα δεδομένων είναι χρήσιμα για την γρήγορη απεικόνιση της συμπεριφοράς των διαφόρων αλγορίθμων που εφαρμόζονται στο scikit-learn και αποκαλούνται “toy datasets”. Ωστόσο, συχνά είναι πολύ μικρά για να είναι αντιπροσωπευτικά των εργασιών μηχανικής εκμάθησης του πραγματικού κόσμου. Τέλος το sklearn περιλαμβάνει και μία λειτουργία εξαγωγής χαρακτηριστικών από εικόνες και κείμενο.

3.3.4 HoloViews

Το HoloViews είναι μια βιβλιοθήκη Python ανοιχτού κώδικα (open-source) που έχει σχεδιαστεί με κύριο σκοπό να κάνει την ανάλυση και την απεικόνιση των μεγάλων συνόλων δεδομένων απρόσκοπτη και απλή. Η συγκριμένη βιβλιοθήκη είναι φιλική προς τον χρήστη δίνοντας του την δυνατότητα να εξευρευνήσει τα δεδομένα που διαθέτει μέσα σε λίγες γραμμές κώδικα και χωρίς να εστιάζει άσκοπα στην διαδικασία σχεδίασης. Το HoloViews δεν αποτελεί μία βιβλιοθήκη σχεδίασης, αλλά συνδέει τα δεδομένα του dataset με κώδικα σχεδίασης που εφαρμόζεται σε άλλα πακέτα, όπως το matplotlib ή το Bokeh. Επίσης, το HoloViews έχει σχεδιαστεί για να μορφοποιεί τα διαθέσιμα δεδομένα ώστε να τα καθιστά ευκόλως ορατά και διαδραστικά. Κατά την προσθήκη νέων πληροφορίων στα δεδομένα με το μεγαλύτερο ενδιαφέρον, το HoloViews επιτρέπει την αποθήκευση, την ταξινόμηση, τον τεμαχίσμό, την ανάλυση, την μειώση, την σύνθεση, την οπτικοποίηση και την μετακίνηση των δεδομένων όσο το δυνατόν πιο φυσικά. Επιπρόσθετα το HoloViews «ζωντανεύει» την αναπαράσταση των αριθμητικών δεδομένων, αποκαλύπτοντας τα εύκολα και χωρίς εκτενή κωδικοποίηση. Επιπλέον τα στοιχεία δεδομένων του HoloViews έχουν ελάχιστες απαιτούμενες προϋποθέσεις (τις Numpy και Param, καμία εκ των οποίων δεν απαιτεί προηγούμενα πακέτα και βιβλιοθήκες για την λειτουργία τους). Οι μορφές δεδομένων του HoloViews μπορούν να ενσωματωθούν απευθείας στον κώδικα έρευνας ή ανάπτυξης, για μέγιστη ευκολία και ευελιξία. Προς το παρόν στο HoloViews παρέχονται υλοποιήσεις σχεδίασης για το matplotlib και το Bokeh. Ακόμα, το HoloViews παρέχει ισχυρή υποστήριξη για τη διεπαφή φορητού υπολογιστή IPython/Jupyter δημιουργόντας διαδραστικές ροές εργασίας. Για τον λόγο αυτό, το HoloViews έχει σχεδιαστεί για να ταιριάζει στην υπάρχουσα ροή εργασίας του χρηστή, χωρίς να προσθέτει περίπλοκες προϋποθέσεις λειτουργίας.

Η ταχύτητα του HoloViews στην επεξεργασία και απεικόνιση μεγάλων συνόλων δεδομένων ήταν ένας από τους βασικούς λόγους που με ώθησε στην επιλογή της συγκεκριμένης βιβλιοθήκης Python. Τέλος σημαντικό ρόλο έπαιξε και η διαδραστικότητα που παρέχει το HoloViews σε συνδιασμό με την επέκταση Bokeh, στην απεικόνιση των δεδομένων επιτρέποντας στον χρήστη να περιηγηθεί στο πλήθος των δεδομένων και να αλληλεπιδράσει με αυτά κάνοντας zoom in και zoom out.

3.3.5 Bokeh

Μία από τις κύριες αρχές σχεδίασης του HoloViews είναι η δήλωση των δεδομένων είναι εντελώς ανεξάρτητη από την υλοποίηση της γραφικής παράστασης. Η Bokeh παρέχει μια ισχυρή πλατφόρμα για τη δημιουργία διαδραστικών γραφημάτων χρησιμοποιώντας καμβά HTML5 και WebGL, και είναι ιδανική για διαδραστική εξερεύνηση δεδομένων. Συνδυάζοντας την ευκολία δημιουργίας διαδραστικών, πολυδιάστατων οπτικοποιήσεων με τα διαδραστικά γραφικά στοιχεία και τη γρήγορη απόδοση που παρέχει η Bokeh, το HoloViews γίνεται ένα ακόμη πιο ισχυρό και εύχρηστο εργαλείο.

3.3.6 Matplotlib

Το Matplotlib είναι μια βιβλιοθήκη που περιέχει πολλαπλές πλατφόρμες, οπτικοποίησης δεδομένων και γραφικών παραστάσεων για την Python και την αριθμητική της επέκταση, NumPy. Ιστογράμματα, διαγράμματα διασποράς, γραφήματα ράβδων και πολλές άλλες γραφικές απεικονίσεις εμπλουτίζουν το περιεχόμενο της βιβλιοθήκης Matplotlib. Ως εκ τούτου, προσφέρει μια βιώσιμη και εναλλακτική λύση ανοιχτού κώδικα στο MATLAB. Το Matplotlib είναι επίσης ιδανικό για εργασία με data frames και πίνακες. Επιπλέον μέσω της χρήσης των API (Application Programming Interfaces) του matplotlib είναι δυνατή η ενσωμάτωση γραφικών σε εφαρμογές GUI (Graphical User Interface). Ένα από τα πιο δημοφιλή API της συγκεκριμένης βιβιοθήκης Python είναι το pyplot. Το pyplot API περιέχει μια βολική και φιλική προς τον χρήστη, διεπαφή κατάστασης τύπου MATLAB. Στην πραγματικότητα, η βιβλιοθήκη matplotlib Python γράφτηκε με πρωταρχικό σκοπό να αποτελέσει μία εναλλακτική λύση ανοιχτού κώδικα για την ευρέως γνωστή εφαρμογή MATLAB. Το Matplotlib παρέχει επίσης ένα OO API (Object-Oriented API) και η διεπαφή του είναι πιο προσαρμόσιμη και ισχυρή από το pyplot. Ωστόσο θεωρείται πιο δύσκολο στη χρήση με αποτέλεσμα, η διεπαφή pyplot να χρησιμοποιείται πιο συχνά από διάφορους ερευνητές και επιστήμονες του χώρου.

3.3.7 MLxtend

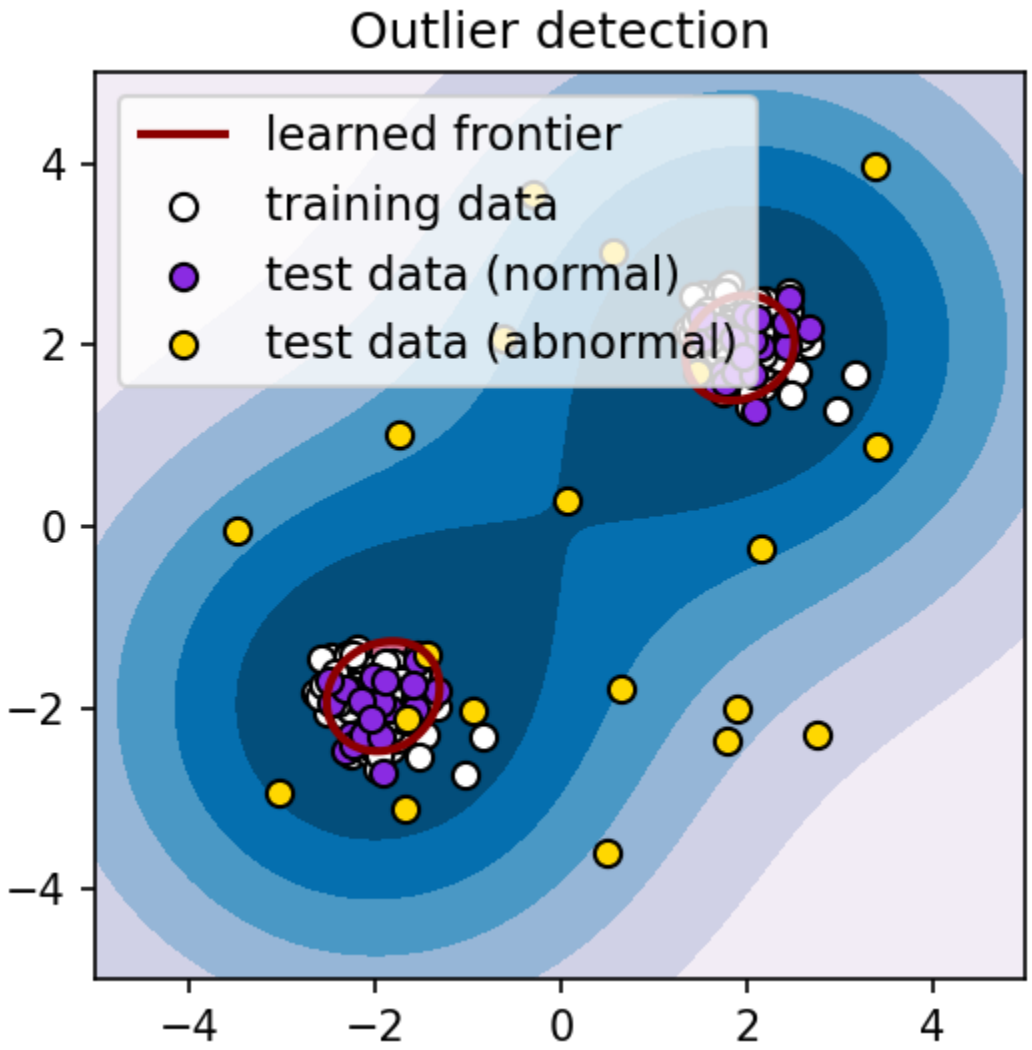
Η βιβλιοθήκη MLxtend αναπτύχθηκε από τον Sebastian Raschka (καθηγητή στατιστικής στο Πανεπιστήμιο του Wisconsin-Madison). Η συγκεκριμένη βιβλιοθήκη διαθέτει ένα όμορφα τεκμηριωμένο και δομημένο API καθώς και πολλά παραδείγματα. Η βιβλιοθήκη MLxtend αποτελεί ένα πακέτο επεκτάσεις μηχανικής μάθησης και έχει πολλές ενδιαφέρουσες λειτουργίες για καθημερινή ανάλυση δεδομένων και εργασίες μηχανικής μάθησης. Αν και υπάρχουν πολλές βιβλιοθήκες μηχανικής μάθησης διαθέσιμες για την Python όπως scikit-learn, TensorFlow, Keras, PyTorch κ.λπ., το MLxtend προσφέρει πρόσθετες λειτουργίες και μπορεί να είναι μια πολύτιμη προσθήκη στην εργαλειοθήκη του χρήστη. Από την συγκεκριμένη βιβλιοθήκη θα εκμεταλευτούμε τον Stacking Classifier, για την μετέπειτα κατασκευή του Stacking μοντέλου.

3.4 Αλγόριθμοι-Algorithms

3.4.1 One-Class Support Vector Machine (One Class SVM)

Το One-Class SVM αποτελεί έναν αλγόριθμό μη επιβλεπόμενης μηχανικής μάθησης για την διαφοροποίηση των δειγμάτων μιας συγκεκριμένης κλάσσης. Πρόκειται για έναν συνδιασμό του One-Class Classification (OCC) και των Support Vector Machines (SVM), δύο αλγόριθμοι

που θα αναλυθούν παρακάτω. Είναι μία από τις πιο δημοφιλής μεθόδους για την προσέγγιση προβλημάτων για την ανίχνευση ανωμαλιών. Η αρχή λειτουργίας του One-Class SVM βασίζεται στην ελαχιστοποίησης της υπερσφαίρας μίας μεμονωμένης κατηγορίας παραδειγμάτων των δεδομένων εκπαίδευσης και θεωρεί ότι όλα τα άλλα δείγματα εκτός της υπερσφαίρας είναι ακραία ή εκτός των δεδομένων εκπαίδευσης. Μία τέτοια υπερσφαίρα παρουσιάζεται στο παρακάτω σχήμα (Εικόνα 7) που σχεδιάστηκε από το One-Class SVM για τον εντοπισμό ακραίων τιμών.



Εικόνα 10

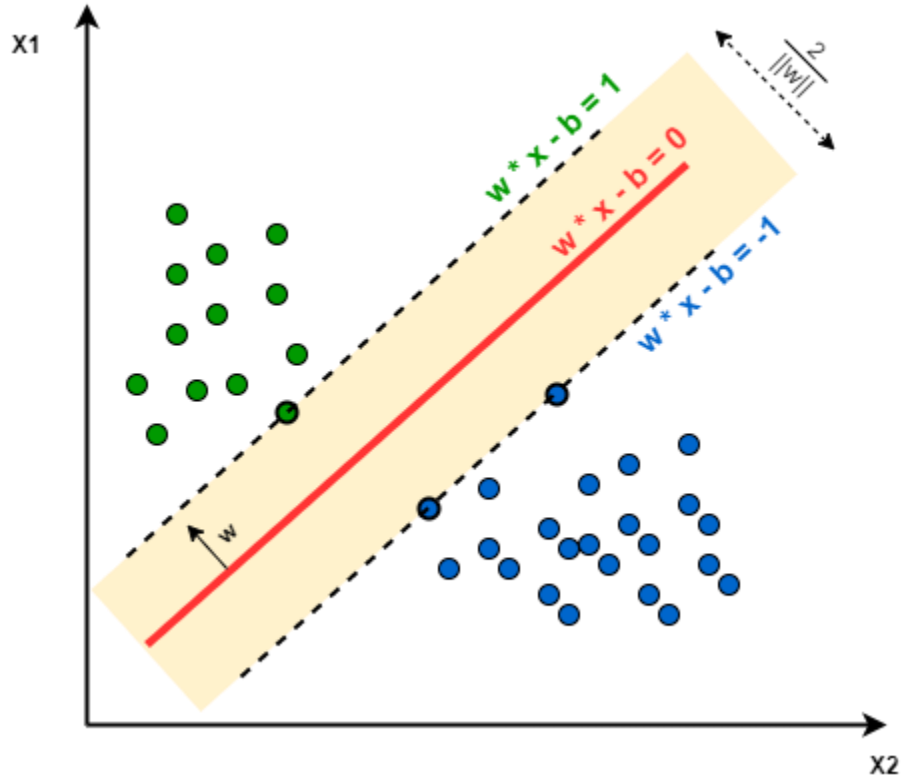
Στο σχήμα της εικόνας 10 τα πολύχρωμα και άτακτα τοποθετημένα σημεία συμβολίζουν τις τιμές των δειγμάτων μίας κλάσσης ενός συνόλου δεδομένων. Τα λευκά σημεία είναι τα δείγματα που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου. Τα μώβ και τα κίτρινα σημεία απεικονίζουν τα νέα δεδομένα που χρησιμοποιήθηκαν για την αξιολόγηση του μοντέλου με τις κίτρινες κουκίδες να αποτελούν τις ακραίες τιμές.

One Class Classification (OCC)

Το One Class Classification (OCC) είναι ένας τύπος μηχανικής μάθησης όπου σκοπός του είναι να μάθει ένα μοντέλο να διακρίνει μεταξύ κανονικών σημείων δεδομένων και ακραίων τιμών, χωρίς να έχει πρόσβαση σε κανένα παράδειγμα των ακραίων τιμών κατά τη διάρκεια της εκπαίδευσης. Με άλλα λόγια, στοχεύει στην διαφοροποίηση των δειγμάτων μιας συγκεκριμένης τάξης, μαθαίνοντας από διάφορα δείγματα πάλι μιας μεμονωμένης τάξης. Είναι μια από τις πιο δημοφιλής προσεγγίσεις για την επίλυση του Anamoly Detection (AD). Το OCC ονομάζεται επίσης unary classification ή class-modelling.

Support Vector Machines (SVM)

Τα Δίκτυα Διανυσμάτων Υποστήριξης ή αλλιώς Support Vector Machines (SVM) είναι ένας από τους πιο ισχυρούς στατιστικούς αλγόριθμους που χρησιμοποιείται για εργασίες ταξινόμησης και παλινδρόμησης. Πρόκειται για ένα αποτελεσματικό και αποδοτικό αλγόριθμος εποπτευόμενης μηχανικής μάθησης που μπορεί να χρησιμοποιηθεί και σε χώρους υψηλών διαστάσεων. Η εκπαίδευση ενός ταξινομητή SVM περιλαμβάνει τη λήψη απόφασης για ένα όριο διαχωρισμού μεταξύ των κλάσεων. Αυτό το όριο είναι γνωστό ότι έχει τη μέγιστη απόσταση από το πλησιέστερο σημείο σε κάθε κατηγορία δεδομένων. Λόγω αυτής της ιδιότητας, το SVM αναφέρεται επίσης ως ταξινομητής μέγιστου περιθωρίου (maximum-margin classifier).



Εικόνα 11

Για μία πιο διαισθητική κατανόηση των SVM πρέπει να εξετάσουμε ένα σύνολο δεδομένων θετικών (σημεία με μπλέ χρώμα) και αρνητικών παραδειγμάτων (σημεία με πράσινο χρώμα). Όπως φαίνεται και στην εικόνα 11, ο στόχος των SVM είναι να σχεδιάσουν τη γραμμή καλύτερης προσαρμογής (κόκκινη γραμμή) η οποία θα διαχωρίζει τα θετικά παραδείγματα από τα αρνητικά παραδείγματα. Σε αντίθεση με τους γραμμικούς ταξινομητές (linear classifiers), το SVM εγγυάται ότι η απόσταση μεταξύ των ακραίων σημείων και των δύο κατηγοριών από την γραμμή καλύτερης προσαρμογής θα είναι σχεδόν ίση και μέγιστη, τονίζοντας έτσι τις διαφορές των θετικών και αρνητικών παραδειγμάτων. Τα SVM έχουν ήδη εφαρμοστεί στην βιβλιοθήκη scikit-learn απευθείας από το libsvm και είναι εύκολα στην χρήση.

Η συνάρτηση που μεγιστοποιείται για να διασφαλιστεί ότι η βέλτιστη γραμμή προσαρμογής είναι 2/|w| όπου w είναι ένα διάνυσμα τυχαίων βαρών έτσι ώστε η συνάρτηση να μεγιστοποιεί το διάστημα μεταξύ των διανυσμάτων υποστήριξης (διακεκομένες γραμμές). Η μεγιστοποίηση του 2/|w| είναι παρόμοια με την ελαχιστοποίηση της συνάρτησης 1/2*(|w|^2). Εάν η συνάρτηση ταξινομήσει εσφαλμένα οποιοδήποτε δείγμα, εφαρμόζεται ο πολλαπλασιαστής Lagranges. Η εφαρμογή του πολλαπλασιαστή Lagranges δίνει αύξηση σε μια μιγαδική εξίσωση, της οποίας το w δίνεται από την ακόλουθη σχέση:

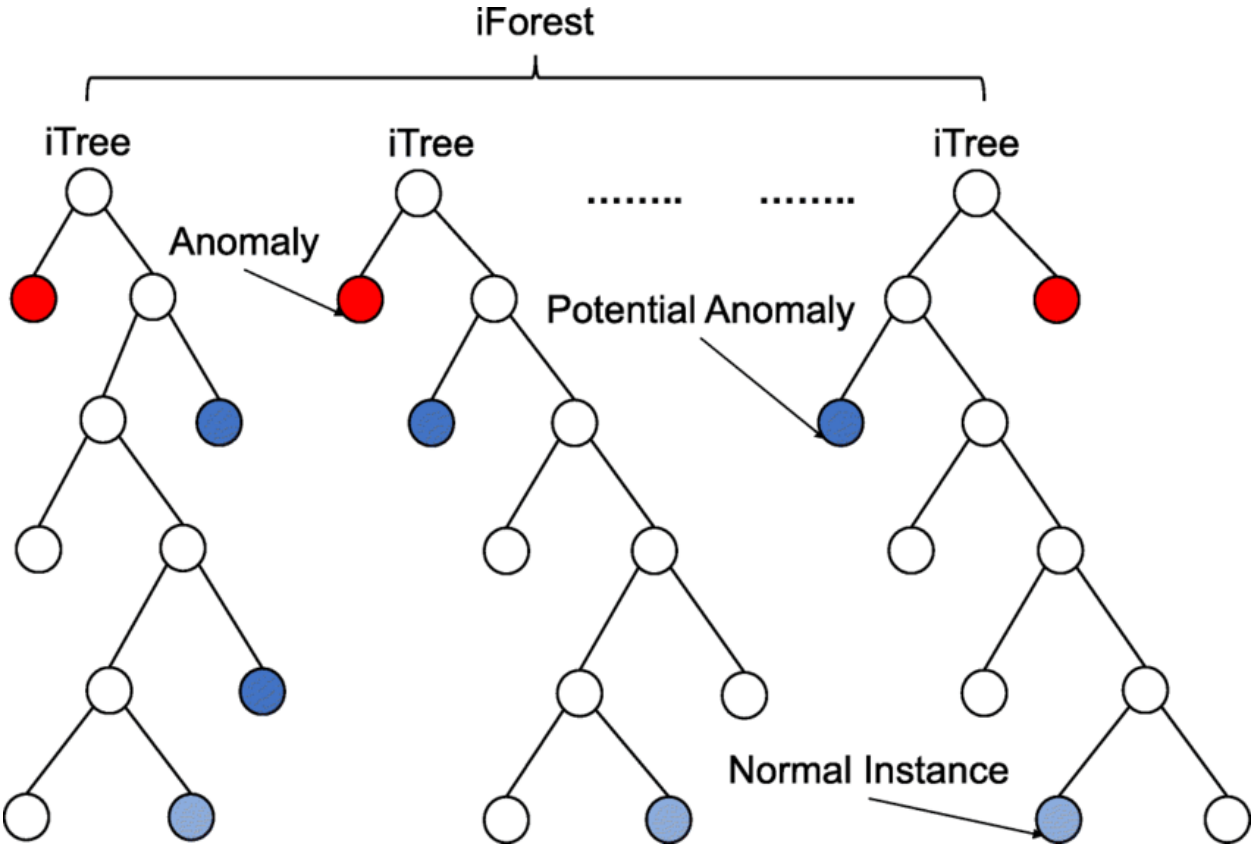
$$|\mathrm{w}| = \Sigma \alpha_i y_i x_i$$

όπου α είναι ο πολλαπλασιαστής Lagrange, το y υποδηλώνει είτε +1 είτε -1, (δηλαδή την κλάση του δείγματος) και το x υποδηλώνει τα δείγματα του συνόλου δεδομένων.

3.4.2 Isolation Forest

Το Isolation Forest είναι μια τεχνική για τον εντοπισμό ακραίων τιμών στα δεδομένα και δημοσιεύτηκε για πρώτη φορά από τους Fei Tony Liu και Zhi-Hua Zhou το 2008. Η προσέγγιση χρησιμοποιεί δυαδικά δέντρα για την ανίχνευση ανωμαλιών, με αποτέλεσμα να επιτυγχάνει μια γραμμική πολυπλοκότητα χρόνου και μια λύση που δεν απαιτεί μεγάλα επίπεδα μνήμης με αποτέλεσμα να είναι κατάλληλη για την επεξεργασία μεγάλων συνόλων δεδομένων.Το Isolation Forest είναι ένας αρκετά διαδεδομένος και αξιόπιστος αλγόριθμος που χρησιμοποιείται για μία πληθώρα εφαρμογών στον τομέα της ανίχνευση ανωμαλιών (ασφάλεια στον κυβερνοχώρο, οικονομικά, ιατρική έρευνα κ.α.). Πρόκειται για έναν αλγόριθμο μη επιβλεπόμενης μηχανικής μάθησης ο οποίο έχει την δυνατότητα να εντοπίζει τις ανωμαλίες απομονώνοντας τις ακραίες τιμές στα δεδομένα. Το Isolation Forest βασίζεται στον αλγόριθμο Decision Tree. Πιο συγκεκριμένα, επιλέγει τυχαία ένα χαρακτηριστικό από το δεδομένο σύνολο χαρακτηριστικών και στη συνέχεια, επιλέγει τυχαία μια διαίρεση μεταξύ των μέγιστων και ελάχιστων τιμών αυτού του χαρακτηριστικού απομονώνοντας έτσι τις ακραίες τιμές. Αυτή η τυχαία επιλογή χαρακτηριστικών θα παράγει μικρότερα paths στα δέντρα απόφασης για τα ανώμαλα σημεία δεδομένων, διακρίνοντάς τα έτσι από τα υπόλοιπα δεδομένα. Στην ανίχνευση ανωμαλιών ένα σύνηθες βήμα που συχνά παρατηρείται είναι η δημιουργία ενός συνόλου που περιέχει κάποια παραδείγματα τα οποία θα ορίζουν το φυσιολογικό. Ωστόσο στην περίπτωση του Isolation

Forest αυτό το βήμα δεν είναι απαραίτητο καθώς δεν είναι αναγκαίο οριστεί πρώτα μία «κανονική» συμπεριφορά. Ο αλγόριθμος Isolation Forest βασίζεται στην αρχή ότι οι ανωμαλίες είναι παρατηρήσεις που είναι λίγες και διαφορετικές, γεγονός που θα πρέπει να διευκολύνει τον εντοπισμό τους.



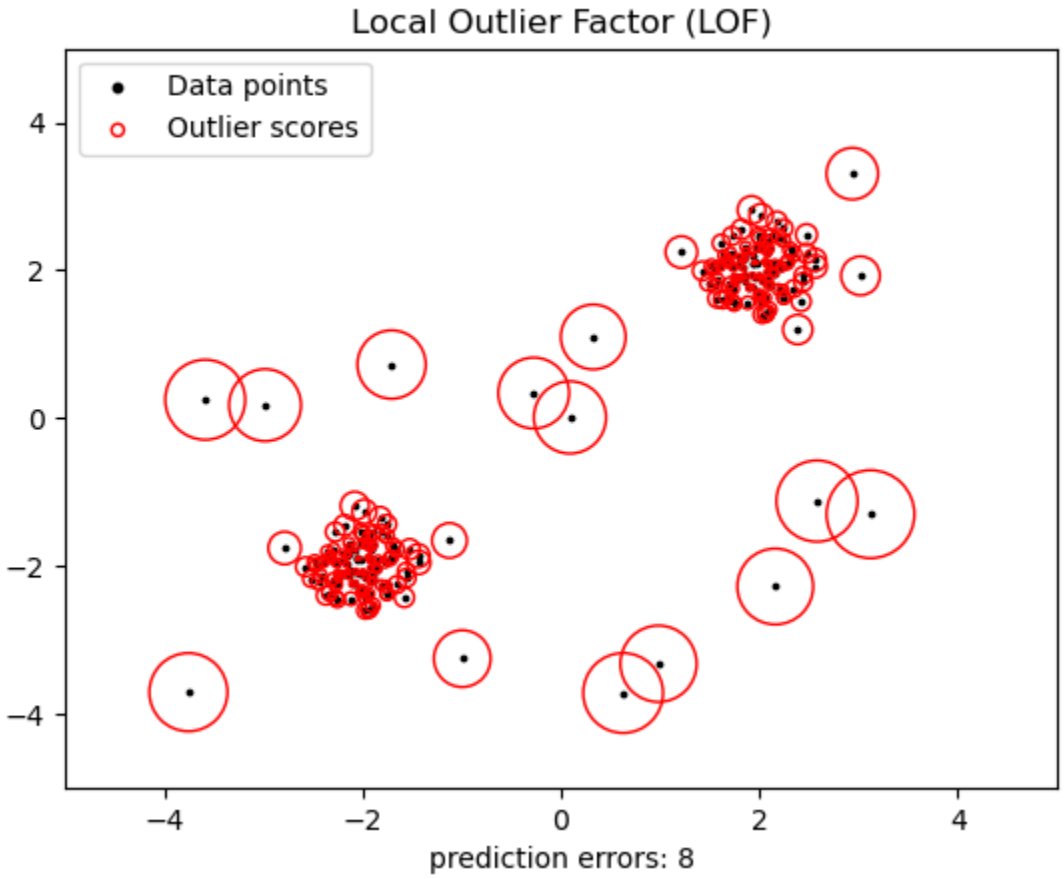
Εικόνα 12

Στο σχήμα της εικόνας 12 φαίνεται μία τυπική διάταξη ενός αλγορίθμου Isolation Forest. Όπως παρατηρούμε το Isolation Forest χρησιμοποιεί ένα σύνολο δέντρων (Isolation Trees) για να απομονώσει τις ανωμαλίες. Επιπλέον όπως φαίνεται και στο σχήμα, τα δείγματα που ταξιδεύουν «βαθύτερα» μέσα στο δέντρο (μπλέ σημεία), είναι λιγότερο πιθανό να είναι ανωμαλίες καθώς χρειάζονταν περισσότερες τομές για να απομονωθούν. Ομοίως, τα δείγματα που καταλήγουν σε μικρότερα κλαδιά αποτελούν ανωμαλίες (κόκκινα σημεία) καθώς ήταν ευκολότερο για το δέντρο να τα διαχωρίσει από άλλες παρατηρήσεις. Χρησιμοποιώντας το Isolation Forest, μπορούμε όχι μόνο να ανιχνεύσουμε ανωμαλίες πιο γρήγορα, αλλά χρειαζόμαστε επίσης λιγότερη μνήμη σε σύγκριση με άλλους αλγόριθμους.

3.4.3 Local Outlier Factor (LOF)

Ο αλγόριθμος Local Outlier Factor είναι ίσως ο πιο ευρέως διαδεδομένος αλγόριθμος για προβλήματα εντοπισμού τοπικών ανωμαλιών (local anomalies). Η βασική αρχή λειτουργίας του

LOF είναι παρόμοια με αυτή των πλησιέστερων γειτόνων. Πρόκειται για μια μέθοδο ανίχνευσης ανωμαλιών χωρίς επίβλεψη που υπολογίζει την τοπική απόκλιση πυκνότητας ενός σημείου δεδομένων σε σχέση με τα γειτονικά του σημεία. Η τοπική πυκνότητα προσδιορίζεται με την εκτίμηση των αποστάσεων μεταξύ σημείων δεδομένων που είναι γειτονικά όπως και στην περίπτωση των k-πλησιέστερων γειτόνων. Έτσι, για κάθε σημείο δεδομένων, μπορεί να υπολογιστεί η τοπική πυκνότητα. Κατα αυτόν τον τρόπο μπορούμε να εντοπίσουμε ποια σημεία δεδομένων έχουν παρόμοιες πυκνότητες και ποια έχουν μικρότερη πυκνότητα από τα γειτονικά τους. Ο αλγόριθμός Local Outlier Factor θεωρεί ως ακραία (ή αλλιώς ανωμαλίες) τα δείγματα που έχουν σημαντικά μικρότερη πυκνότητα από τα γειτονικά τους.



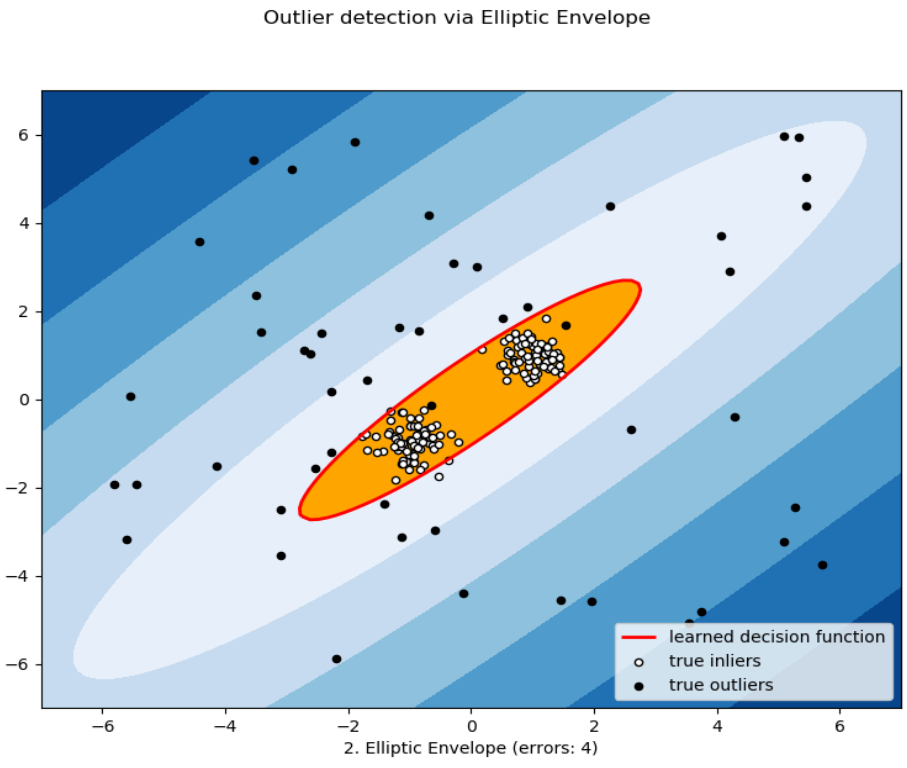
Εικόνα 13

Στο σχήμα της εικόνας 13 φαίνεται η γραφική απεικόνιση ενός αλγορίθμου Local Outlier Factor. Οι μαύρες κουκίδες συμβολίζουν την κατανομή των διαθέσιμων σημείων δεδομένων στον πίνακα αυτό, ενώ οι κόκκινοι κύκλοι που περιβάλλουν τα σημεία αυτά φανερώνουν το εύρος της τοπικής πυκνότητας γύρω από τα σημεία αυτά. Όπως είναι φανερό τα πιο αραιά τοποθετημένα σημεία του σχήματος αποτελούν ανωμαλίες καθώς μέσα στους κύκλους που τα περιβάλλουν δεν περιλαμβάνονται άλλα σημεία δεδομένων. Η κύρια διαφορά του LOF από τον αλγόριθμο KNN είναι ότι πραγματοποιεί τον καθορισμό ακραίων τιμών κάνοντας βαθμολόγηση με βάση την

πυκνότητα ενώ στην περίπωση του KNN αναζητούμε τον πλησιέστερο γείτονα. Με άλλα λόγια ο KNN αλγόριθμος εντοπίζει και κατηγοριοποιεί παρατηρήσεις που να είναι κοντά μεταξύ τους, αλλά ο αλγόριθμος Local Outlier Factor έχει την ικανότητα να εντοπίσει παρατηρήσεις που δεν είναι όμοιες με τις άλλες.

3.4.4 Elliptic Envelope

Η ανίχνευση ανωμαλιών με την χρήση του Elliptic Envelope είναι μια στατιστική τεχνική ανάλυσης που χρησιμοποιείται για τον εντοπισμό πιθανών ακραίων τιμών (outlier detection) σε διάφορα σύνολα δεδομένων. Πρόκειται για έναν αλγόριθμο μη επιβλεπόμενης μηχανικής μάθησης που λειτουργεί προσδίδοντας μια κατανομή πιθανότητας στα διαθέσιμα σημεία δεδομένων. Συνήθως η κατανομή αυτή είναι ελλειπτικού σχήματος όπως μια γκαουσιανή κατανομή (Gaussian). Στη συνέχεια ο αλγόριθμος Elliptic Envelope έχει την δυνατότητα να υπολογίζει πόσες τυπικές αποκλίσεις μακριά από την αναμενόμενη μέση τιμή πέφτουν τα πραγματικά σημεία δεδομένων. Κατα αυτό τον τρόπο τα σημεία που υπερβαίνουν τον προβλεπόμενο αριθμό τυπικών αποκλίσεων θα αντιμετωπιστούν ως ανώμαλα ή ακραίες τιμές. Ο συγκεκριμένος αλγόριθμος μπορεί να αποδώσει καλύτερα σε σύνολα δεδομένων με χαμηλή μόλυνση ή όταν τα ακραία σημεία απέχουν πολύ από τις φυσιολογικές τιμές.



Εικόνα 14

Στο σχήμα της εικόνας 14 απεικονίζεται γραφικά ο τρόπος λειτουργείας του αλγόριθμου Elliptic Envelope. Με κόκκινο χρώμα βλέπουμε το ελλειπτικό περίβλημα που προσαρμόζει ο αλγόριθμος αυτός στα δεδομένα. Όπως παρατηρούμε τα στοιχεία που βρίσκονται εκτός του ελλειπτικού αυτού σχήματος θεωρούνται outliers.

3.5 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΡΕΥΝΑ

Με την ραγδαία αύξηση των πηγών δεδομένων σε πραγματικό χρόνο η ανίχνευση ανωμαλιών σε δεδομένα χρονοσειρών αποκτάει όλο και περισσότερη σημασία. Πιο συγκεκριμένα βρίσκει χρήση σε μία πληθώρα εφαρμογών όπως την προληπτική συντήρηση, τον εντοπισμό και την αντιμετώπιση απάτης καθώς και την παρακολούθηση στατιστικών στοιχείων. Η ανίχνευση ανωμαλιών σε χρονοσειρές μπορεί να εφαρμοστεί στους κλάδους της οικονομίας, της βιομηχανίας, της ιατρικής, της επιστήμης των υπολογιστών, της ενέργιας, του ηλεκτρονικού εμπορίου ακόμα και στα μέσα κοινωνικής δικτύωσης.

3.5.1 Ερευνα για το dataset

Στον τομέα της ανίχνευσης ανωμαλιών η σύγκριση των μέχρι τώρα αλγοριθμών ήταν μία ιδιαίτερα δύσκολη διαδικασία καθώς δεν υπήρχε μία κοινή βάση σύγκρισης. Το 2015 οι Lavin Alexander και Ahmad Subutai δημοσίευσαν την έρευνα τους με τίτλο «Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark» στην οποία και παρουσίασαν το Numenta Anomaly Benchmark στην επιστημονική κοινότητα. Το NAB αποτελείται από δύο κύρια μέρη. Το πρώτο μέρος είναι το NAB scoring system το οποίο λειτουργεί με βάση ένα σύνολο προκαθορισμένων κανόνων με σκοπό τον υπολογισμό της συνολικής ποιότητας της ανίχνευσης ανωμαλιών. Το δεύτερό κομμάτι είναι τα 58 σύνολα δεδομένων που διαθέτει τα οποία και καλύπτουν ένα μεγάλο εύρος πεδίων (οικονομία, βιομηχανία, ιατρική, IT κ.λπ.). Επιπρόσθετα στην συγκεκριμένη έρευνα γίνεται σύγκριση μερικών αλγορίθμων στα δεδομένα του NAB μεταξύ των οποίων είναι ο Numenta HTM anomaly detector, ο Etsy Skyline, ο AnomalyDetectionTs και ο AnomalyDetectionVec με τον αλγόριθμο HTM να παρουσιάζει τα βέλτιστα αποτελέσματα.

Ακόμα μία σημαντική ερευνητική συνεισφορά στην οποία αναφέρεται το σύνολο δεδομένων Numenta Anomaly Benchmark πραγματοποιήθηκε το 2017 από τους Nidhi Singh και Craig Olinsky. Η έρευνα τους είχε σκοπό να αναλύσει το NAB scoring system και να εντοπίσει ελατώμματα μέσα σε αυτό. Επιπρόσθετα εστιάζει στις δυσκολίες στην χρήση του Numenta Anomaly Benchmark και τέλος δοκίμαζει πέντε αλγορίθμους ανίχνευσης ανωμαλιών σε κάθε

ένα από τα 58 σύνολα δεδομένων του NAB. Για τους σκοπούς της έρευνας χρησιμοποιούνται οι αλγόριθμοι Contextual Anomaly Detection (ContextOSE), AnomalyDetectionVec, Etsy Skyline και 2 ακόμα παραλλαγές του Numenta HTM. Τα αποτελέσμα της έρευνας ήταν ότι οι συγκεκριμένοι αλγόριθμοι δεν είναι ικανοί να προβλέψουν επιτυχώς τις ανωμαλίες σε αυτά τα σύνολα δεδομένων καθώς καταφέρνουν αν εντοπίσουν μόνο 1-7 ανωμαλίες από τις 150-1558 καταγεγραμμένες ανωμαλίες. Ωστόσο παρατηρήθηκε ότι το NAB scoring system εξακολουθεί να αποδίδει υψηλές βαθμολογίες σε αυτούς τους αλγόριθμους για ορισμένα από τα σύνολα δεδομένων. Το NAB scoring system έχει σχεδιαστεί για να ανταμείβει την έγκαιρη ανίχνευση ανωμαλιών και όχι το ολικό ποσοστό ανίχνευσης όπως το precision και το recall. Τέλος η έρευνα καταλήγει στο γεγονός ότι τα σύνολα δεδομένων του NAB δεν είναι ιδανικά για χρήση σε ένα επαγγελματικό περιβάλλον ωστόσο αποτελούν μία καινοτόμο λύση για την αξιολόγηση των διάφορων αλγορίθμων μηχανικής μάθησης για τους σκοπούς της ανίχνευσης ανωμαλιών.

3.5.2 Time-based feature engineering

Η ανάλυση μίας χρονόσειρας μπορεί να αποβεί μία χρονοβόρα και απαιτητική διαδικασία καθώς εξαρτάται σε μεγάλο βαθμό με τις προκλίσεις του προβλήματος που επιχειρούμε να αντιμετωπίσουμε. Κατα παρόμοιο τρόπο και στον τομέα της ανίχνευσης ανωμαλιών είναι σημαντική η ανάλυση μία χρονοσειράς σε δεδομένα με την μεγαλύτερη χρησιμότητα αναφορικά με το εκάστωτε πρόβλημα.

Μία παρόμοια διεργασία εξόριξης δεδομένων από χρονοσειρά ακολούθησαν το 2018 στην έρευνα τους οι Wei Mao, Xiu Cao, Qinhua zhou, Tong Yan και Yongkang Zhang. Η συγκεκριμένη έρευνα είχε τίτλο «Anomaly Detection for Power Consumption Data based on Isolation Forest». Σκόπος της ήταν η ανίχνευση ανωμαλιών στην ενεργειακή κατανάλωση των χρηστών ενός δικτύου. Επειδή η κατανάλωση ενέργειας συνήθως παρουσιάζει μία κυκλική επαναληπτική τάση, οι συγκεκριμένοι ερευνητές φρόντισαν να μετρήσουν την μέση καθημερινή κατανάλωση μίας εργάσιμης ημέρας και την μέση καθημερινή κατανάλωση του Σαββατοκύριακου. Κατα αυτόν τον τρόπο υπολόγισαν την μέση κατανάλωση κάθε ημέρας της εβδομάδας και διαχώρισαν τα δεδομένα σε 7 στήλες. Στην συνέχεια με την χρήση του αλγορίθμου isolation forest εντόπισαν τις ανώμαλες περιπτώσεις στο σύνολο των χρηστών του δικτύου και έπειτα μία ομάδα εμπειρογνωμώνων τα αξιολόγησε. Τέλος το μοντέλο εκπαιδεύτηκε ξανά με τις μεθόδους PCA και Autoencoders για καλύτερα αποτελέσματα με το PCA αποδίδει καλύτερα.

Ακόμα μία σημαντική έρευνα αναφορικά με την εξαγωγή δεδομένων από μια μονομετάβλητη χρονοσειρά (univariate time series) για την ανίχνευση ανωμαλιών, διεξύχθει τον Απρίλιο του 2020 από τους Mohammed Braei και Dr.-Ing. Sebastian Wagner. Μια μονομεταβλητή χρονοσειρά αναφέρεται σε έναν τύπο δεδομένων χρονοσειράς όπου υπάρχει μόνο μία μεταβλητή ενδιαφέροντος που καταγράφεται σε μια ακολουθία χρονικών σημείων. Πιο αναλυτικά,

περιλαμβάνει μια ενιαία ροή σημείων δεδομένων που συλλέγονται σε τακτά χρονικά διαστήματα, με κάθε σημείο δεδομένων να αντιπροσωπεύει την τιμή της μεταβλητής σε μια συγκεκριμένη χρονική στιγμή. Η έρευνας τους με τίτλο «Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-art» αναλύει 20 διαφορετικές μεθόδους ανίχνευσης ανωμαλιών σε univariate time series, συγκρίνοντας στατικές μεθόδους, μεθόδους μηχανικής μάθησης και βαθιάς μάθησης. Για την σύγκριση των μεθόδων αυτών χρησιμοποιήθηκαν 5 συνόλα δεδομένων μεταξύ των οποίων και το Numenta Anomaly Benchmark. Τα αποτελέσματα της συγκεκριμένης έρευνας φανερώνουν ότι η προσέγγιση που είναι πιο γρήγορη, πιο αποτελεσματική και λίγοτερο απαιτητική σε υπολογιστική ισχύ, είναι η στατιστική ανάλυση. Ωστόσο η στατιστικές αυτές μέθοδοι αντιμετωπίζουν μεγάλη δυσκολία στον εντοπισμό των contextual anomalies σε αντίθεση με την μηχανική μάθηση και την βαθιά μάθηση. Τέλος η προσέγγιση που φαίνεται ότι αποδίδει χειρότερα αποτελέσματα από τις υπόλοιπες σε univariate time series είναι αυτή της βαθιάς μάθησης.

ΚΕΦ 4ο ΥΛΟΠΟΙΗΣΗ

4.1 Εισαγωγή βιβλιοθηκών

Το πρώτο και απαραίτητο βήμα για την υλοποίηση του κώδικα σε python είναι η εισαγωγή όλων των βιβλιοθηκών και πακέτων που έγινε αναφορά στο υποκεφάλαιο «Βιβλιοθήκες».

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import holoviews as hv
from holoviews import opts
hv.extension('bokeh')


from sklearn.svm import OneClassSVM
from sklearn.neighbors import LocalOutlierFactor
from sklearn.neighbors import NearestNeighbors
from sklearn.covariance import EllipticEnvelope
from sklearn.ensemble import IsolationForest
from sklearn.linear_model import LogisticRegression
from mlxtend.classifier import StackingClassifier
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
```

4.2 Προετοιμασία δεδομένων

Το στάδιο της προετοιμασία των δεδομένων σκοπό έχει την ανάλυση των δεδομένων που θα χρησιμοποιηθούν καθώς μάλιστα και την συλλογή χρήσιμων πληροφοριών για αυτά. Το πρώτο στάδιο της προετοιμασίας των δεδομένων είναι η εισαγωγή του dataset.

Εισαγωγή του συνόλου δεδομένων machine_temperature_system_failure από το NAB σε csv μορφή μέσα από την βιβλιοθήκη Pandas:

```
df = pd.read_csv(r"C:\Users\atout\OneDrive\Έγγραφα\MACHINE
LEARNING\data\NAB\realKnownCause\realKnownCause\machine_temperature_system_failur
e.csv")
```

Για την πληρέστερη κατανόηση των χαρακτηριστικών του dataset εμφανίζουμε τα 5 πρώτα δείγματα με την εντολή:

```
df.head()
```

τα 5 αυτά δείγματα είναι τα εξής:

	timestamp	value
0	2013-12-02 21:15:00	73.967322
1	2013-12-02 21:20:00	74.935882
2	2013-12-02 21:25:00	76.124162
3	2013-12-02 21:30:00	78.140707
4	2013-12-02 21:35:00	79.329836

Όπως παρατηρούμε το σύνολο δεδομένων που θα χρησιμοποιήσουμε αρχικά αποτελείται από 2 στήλες. Η πρώτη στήλη είναι μία χρονοσειρά που εκτείνεται σε βάθος τριών μηνών και η δεύτερη αποτελείται από τις θερμοκρασιακές μετρήσεις ενός μεγάλου βιομηχανικού μηχανήματος σε κάθε πέντε λεπτά.

Με την εντολή shape είμαστε σε θέση να γνωρίζουμε το ακριβές μέγεθος του dataset το οποίο εμπεριέχει 22.695 δείγματα και 2 στήλες.

```
df.shape
```

Το επόμενο βήμα που πρέπει να ελέγξουμε είναι το αν υπάρχουν κενά στοιχεία μέσα στο σύνολο δεδομένων μας. Στην προκειμένη περίπτωση το σύνολο δεδομένων είναι πλήρες και δεν λείπει καμία τιμή.

```
df.isnull().any()
```

Για να εμβαθύνουμε στην κατανόηση του dataset δημιουργούμε μία σύνοψη διαφόρων στατιστικών μέτρων με την εντολή:

```
df.describe()
```

Η παραπάνω εντολή μας δίνει το εξής output:

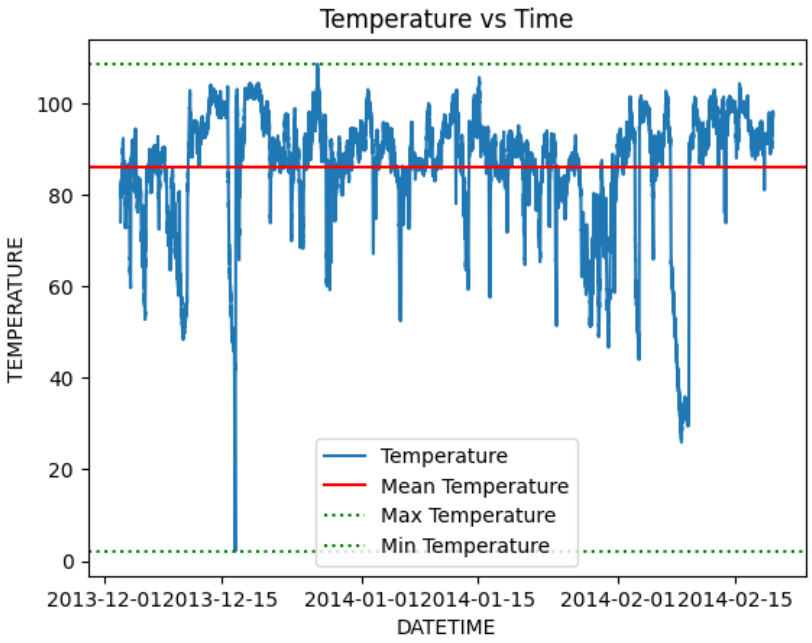
	value
count	22695.000000
mean	85.926498
std	13.746912
min	2.084721
25%	83.080078
50%	89.408246
75%	94.016252
max	108.510543

Παρατηρούμε ότι η στήλη με το όνομα “value” περιέχει 22.695 δείγματα με την μέση τιμή των δειγμάτων να αποτελεί το 85,92. Αυτή η μέση τιμή αποτελεί και την μέση τιμή της θερμοκρασίας του βιομηχανικού μηχανήματος την οποία και μπορούμε να θεωρήσουμε φυσιολογική. Το std υποδεικνύει την τυπική απόκλιση των θερμοκρασιακών τιμών. Επιπλέον το min και το max αντιπροσωπεύουν την ελάχιστη και την μέγιστη τιμή αντίστοιχα. Τέλος οι τιμές 25%, 50% και 75% υποδεικνύουν την τιμή του 25ου του 50ου και του 75ου εκατοστημορίου της στήλης «value».

Στην συνέχεια βρίσκουμε την μέση τιμή της θερμοκρασίας και την απεικονίζουμε και γραφικά στο σύνολο των δεδομένων μας:

```
plt.plot(pd.to_datetime(df['timestamp']),df['value'],label='Temperature')
plt.axhline(y = df['value'].mean(), color = 'r', linestyle = '-',label='Mean
Temperature')
plt.axhline(y = df['value'].max(), color = 'g', linestyle = 'dotted',label='Max
Temperature')
plt.axhline(y = df['value'].min(), color = 'g', linestyle = 'dotted',label='Min
Temperature')
plt.xlabel('DATETIME')
plt.ylabel('TEMPERATURE')
plt.legend()
plt.title('Temperature vs Time')
```

Η παραπάνω εντολή θα μας δώσει την εξής γραφική παράσταση:



Διάγραμμα 1

Όπως παρατηρούμε ο πίνακας αυτός αποτελείται από 2 κάθετους άξωνες. Στον οριζόντιο άξωνα φανερώνονται τα δεδομένα χρόνου που διαθέτουμε ενώ στον κάθετο άξωνα οι θερμοκρασίες του μηχανήματος την δεδομένη στιγμή. Η κόκκινη γραμμή που εκτείνεται στο πλήθος όλων των δεδομένων φανερώνει την μέση τιμή της θερμοκρασίας του μηχανήματος που όπως αναμέναμε είναι κοντά στου 89 βαθμούς κελσίου. Τέλος οι δύο πράσινες διακεκομένες γραμμές συμβολίζουν την θέση της μέγιστης και της ελάχιστης τιμής του dataset.

Έπειτα αναλύουμε την χρονοσειρά σε 5 ξεχωριστές στηλές για την βέλτιστη εκμετάλευση των δεδομένων του dataset. Οι πέντες αυτές στήλες θα ονομάζονται “year”, “month”, “day”, “hour” και “minute”.

```
df['timestamp'] = pd.to_datetime(df['timestamp'])
df['year'] = df['timestamp'].apply(lambda x : x.year)
df['month'] = df['timestamp'].apply(lambda x : x.month)
df['day'] = df['timestamp'].apply(lambda x : x.day)
df['hour'] = df['timestamp'].apply(lambda x : x.hour)
df['minute'] = df['timestamp'].apply(lambda x : x.minute)
df.head()
```

Μετά την παρέμβαση αυτή το dataset πλεόν θα έχει την εξής μορφή:

	timestamp	value	year	month	day	hour	minute
0	2013-12-02 21:15:00	73.967322	2013	12	2	21	15
1	2013-12-02 21:20:00	74.935882	2013	12	2	21	20
2	2013-12-02 21:25:00	76.124162	2013	12	2	21	25
3	2013-12-02 21:30:00	78.140707	2013	12	2	21	30
4	2013-12-02 21:35:00	79.329836	2013	12	2	21	35

Το σύνολο δεδομένων μας αποτελείται πλεόν από 7 στήλες και 22.695 δείγματα.

4.3 Outlier detection (υλοποίηση)

Για την ανίχνευση ακραίων τιμών είναι εξαιρετικά χρήσιμη και θεμιτή η ύπαρξη μίας επιπλέον στήλης που να κατηγοριοποιεί με σιγουριά τα δείγματα του διαθέσιμου dataset σε ανωμαλίες και σε φυσιολογικές τιμές, έτσι ώστε να αξιοποιηθεί ως μέτρο σύγκρισης για την αξιολόγηση των προβλέψεων του μοντέλου που θα κατασκευαστεί. Στην προκειμένη περίπτωση όμως, όπως είναι φανερό η στήλη αυτή απουσιάζει. Ωστόσο αυτό είναι ένα σύνηθες φαινόμενο στον εργασιακό χώρο καθώς αρκετές είναι οι φορές που απουσιάζουν βασικά στοιχεία μέσα σε ένα dataset.

Σε αυτήν την περίπτωση θα αξιοποιήσουμε τις δωσμένες από την βιομηχανία ημερομηνίες στις οποίες οι μετρήσεις που πήραμε αναμένουμε να είναι πέραν του φυσιολογικού δηλαδή ανωμαλίες. Τις ημερομηνίες αυτές τις πήραμε από την Github ιστοσελίδα του Numenta Anomaly Benchmark για το machine_temperature_system_failure dataset.

Το πρώτο βήμα είναι να δημιουργήσουμε μία λίστα με το όνομα “real_anomalies” που θα περιέχει τις ημερομηνίες αυτές.

```
real_anomalies = [
        ["2013-12-10 06:25:00.000000","2013-12-12 05:35:00.000000"],
        ["2013-12-15 17:50:00.000000","2013-12-17 17:00:00.000000"],
        ["2014-01-27 14:20:00.000000","2014-01-29 13:30:00.000000"],
        ["2014-02-07 14:55:00.000000","2014-02-09 14:05:00.000000"]
]
```

Όπως παρατηρούμε στην λίστα αυτή αναγράφονται 4 διαφορετικές χρονικές στιγμές στις οποίες οι μετρήσεις μας αναμένετε να είναι ανώμαλες. Η πρώτη ανωμαλία είναι ένα σχεδιασμένο κλείσιμο του μηχανήματος. Η δεύτερη ανωμαλία είναι δύσκολο να ανιχνευτεί και οδηγεί στην

τρίτη ανωμαλία (πιθανότατα βλάβες του μηχανήματος). Η 4η και τελευταία ανωμαλία σηματοδοτεί μία καταστροφική βλάβη του βιομηχανικού μηχανήματος.

Δημιουργούμε μία νέα στήλη με το όνομα “anomalies” της οποίας όλα τα στοιχεία αρχικά θα είναι 1 (δηλαδή όχι ανωμαλίες). Η στήλη αυτή θα περιέχει τις ανωμαλίες που θα ορίσουμε από τις παραπάνω ημερομηνίες. Ορίζουμε ως ανωμαλίες τις θερμοκρασιακές τιμές που βρίσκονται ανάμεσα σε αυτά τα χρονικά περιθώρια και τις προσθέτουμε στην στήλη “anomalies” με την τιμή -1 (που συμβολίζει τις ανωμαλίες):

```
df['anomalies'] = 1
for start, end in real_anomalies:
    df.loc[((df['timestamp'] >= start) & (df['timestamp'] <= end)), 'anomalies'] = -1
df.head()
```

Το DataFrame θα έχει πλέον την εξής μορφή:

	timestamp	value	year	month	day	hour	minute	anomalies
0	2013-12-02 21:15:00	73.967322	2013	12	2	21	15	1
1	2013-12-02 21:20:00	74.935882	2013	12	2	21	20	1
2	2013-12-02 21:25:00	76.124162	2013	12	2	21	25	1
3	2013-12-02 21:30:00	78.140707	2013	12	2	21	30	1
4	2013-12-02 21:35:00	79.329836	2013	12	2	21	35	1

Για να βεβαιωθούμε ότι στην στήλη “anomalies” υπάρχουν και δείγματα με τιμή -1 χρησιμοποιούμε την παρακάτω εντολή:

```
unique_values = df['anomalies'].unique()
print(unique_values)
```

η οποία μας δίνει αποτέλεσμα:

```
[ 1 -1]
```

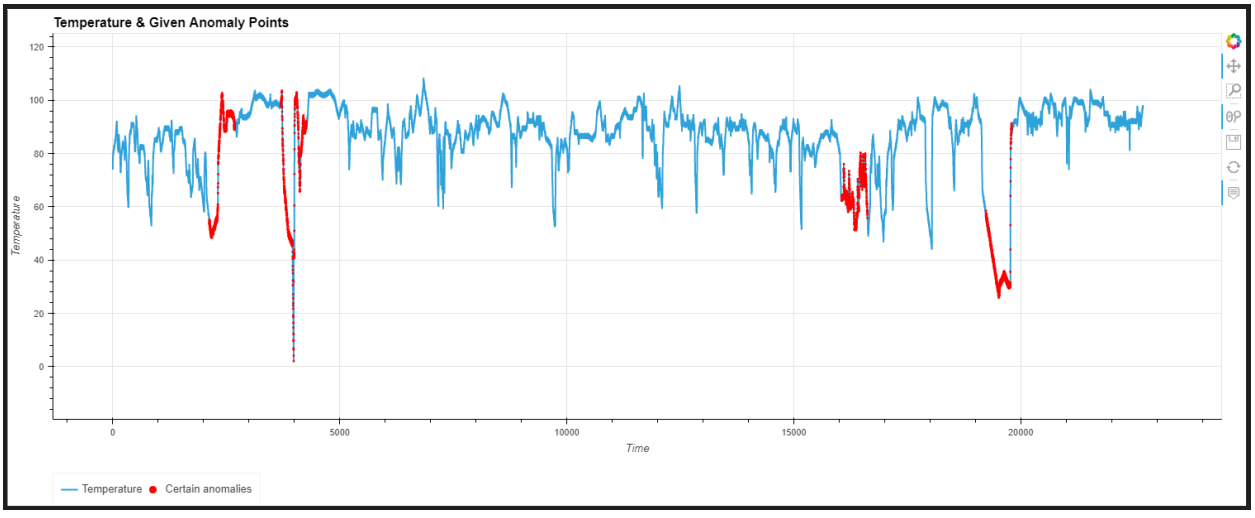
Δημιουργούμε μία λίστα με το όνομα “anomalies” η οποία εντοπίζει τα δείγματα του df όπου η στήλη “anomalies” έχει τιμή -1 και επιλέγει τις αντίστοιχες τιμές ευρετηρίου και της στήλης “values”. Κατα αυτό τον τρόπο ξεχωρίζουμε τις τιμές που αποτελούν ανωμαλία σύμφωνα με τα στοιχεία από το NAB.

```
anomalies = [[ind, value] for ind, value in zip(df[df['anomalies']==-1].index,
df.loc[df['anomalies']==-1,'value'])]
```

Στην συνέχεια απεικονίζουμε γραφικά με την χρήση του Holoviews τις 4 αυτές περιπτώσεις ανωμαλίας σε σχέση με τον χρόνο προσδίδοντας κόκκινο χρώμα στα δείγματα που εμπεριέχονται στην λίστα “anomalies”:

```
(hv.Curve(df['value'], label="Temperature") * hv.Points(anomalies,
label="Certain anomalies").opts(color='red', legend_position='bottom',
size=2, title="Temperature & Given Anomaly Points"))\
    .opts(opts.Curve(xlabel="Time", ylabel="Temperature", width=1500,
height=600,tools=['hover'],show_grid=True))
```

Η γραφική απεικόνιση των καταγεγραμμένων ανωμαλιών όπως φανερώνονται στο Holoviews.



Διάγραμμα 2

Ορίζουμε την στήλη “timestamp” (η οποία περιέχει την χρονοσείρά) ως ευρετήριο του DataFrame. Κατα αυτό τον τρόπο θα λειτουργεί ως indexing και θα μας διευκολύνει κατα την διάρκεια της εκπαίδευσης του μοντέλου.

```
df.set_index('timestamp',inplace = True)
df.head()
```

Επομένως το DataFrame έχει πλέον την εξής μορφή:

	value	year	month	day	hour	minute	anomalies
timestamp							
2013-12-02 21:15:00	73.967322	2013	12	2	21	15	1
2013-12-02 21:20:00	74.935882	2013	12	2	21	20	1
2013-12-02 21:25:00	76.124162	2013	12	2	21	25	1
2013-12-02 21:30:00	78.140707	2013	12	2	21	30	1
2013-12-02 21:35:00	79.329836	2013	12	2	21	35	1

Δεύτερη προσέγγιση

Σε αυτό το σημείο αξίζει να σημειωθεί πως αυτός ο τρόπος δεν αποτελεί την μοναδική προσέγγιση του προβλήματος. Ακόμα ένας τρόπος για τον έλεγχο των προβλέψεων του μοντέλου σε δεδομένα που γνωρίζουμε με σιγουριά πως είναι ανώμαλα είναι η μέθοδος της κατασκευής τεχνητών ανωμαλίων. Σύμφωνα με την μέθοδο αυτή αξιοποιούμε τα δεδομένα με την μεγαλύτερη βαρύτητα (στην συγκεκριμένη περίπτωση τις θερμοκρασιακές τιμές της στήλης value) και σε κάθε ν σειρά της στήλης value προσθέτουμε στην ήδη υπάρχουσα τιμή, 0.5 φορές την αρχική τιμή του εαυτού της ενώ παράλληλα τις κατηγοριοποιούμε ως ανώμαλες. Κατα αυτόν τον τρόπο για κάθε ν σειρά του συνόλου δεδομένων θα βρίσκεται μία τεχνητή ανωμαλία και έτσι θα μπορούμε να αξιοποιήσουμε το τεχνητό αυτό σύνολο δεδομένων για την αξιολόγηση των μοντέλων.

4.3.1 ISOLATION FOREST MODEL (OUTLIER DETECTION)

Για το outlier detection θα αξιοποιήσω ολόκληρο το σύνολο δεδομένων για την εκπαίδευση του μοντέλου και πιο συγκεκριμένα την στήλη value που εμπεριέχει τις θερμοκρασιακές τιμές του

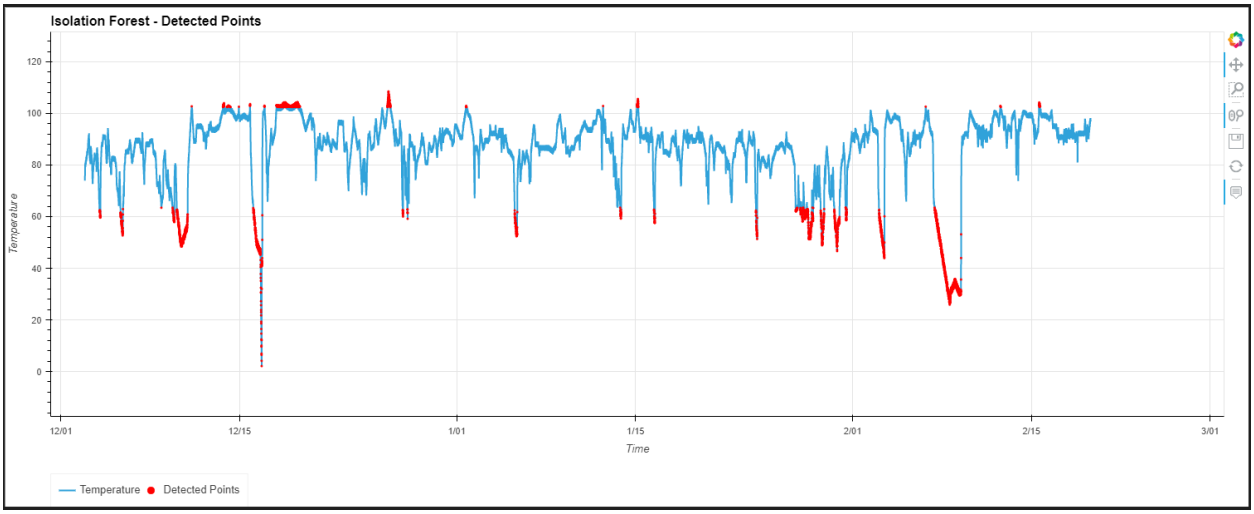
μηχανήματος. Ο πρώτος αλγόριθμος που χρησιμοποίησα είναι ο Isolation Forest. Για την εφαρμογή του αλγορίθμου αυτού, δημιουργούμε ένα αντίγραφο του παραπάνω DataFrame με το όνομα “isf_df” έτσι ώστε τα δεδομένα μας να είναι εύκολα κατανοητά και διαχωρισμένα για τον χρήστη. Έπειτα δημιουργούμε έναν Isolation Forest Classifier όπου το “n_estimators” υποδικνύει τον αριθμό των βασικών εκτιμητών (δηλαδή των δέντρων απόφασης) που θα χρησιμοποιήσουμε και το “contamination” καθορίζει το αναμενώμενο ποσοστό ανωμαλιών στο σύνολο δεδομένων. Μετά από αρκετές δοκιμές ο συνδιασμός που μου έδωσε το μεγαλύτερο ποσοστό επιτυχίας ήταν για n_estimators = 300 και contamination = 0.1. Τέλος δημιουργούμε μία νέα στήλη στο isf_df με το όνομα “prediction” στην οποία θα τοποθετήσουμε όλες τις τιμές τις οποίες ο αλγόριθμος θα προβλέψει ως ακραίες τιμές ή ανωμαλίες με την εντολή “fit_predict()”.

```
isf_df = df.copy()
isf_model = IsolationForest(n_estimators = 300, contamination= .1)
isf_df.loc[:,'prediction'] = isf_model.fit_predict(isf_df[['value']].values)
```

Για να απεικονίσουμε γραφικά τα ακραία σημεία που εντόπισε ο αλγόριθμος isolation forest στο σύνολο της στήλης value, ακολουθούμε την ίδια διαδικασία, δημιουργόντας μία λίστα με το όνομα “outliers” που θα εντοπίζει τα δείγματα της στήλης “prediction” με τιμή -1 και θα τα απεικονίζει με κόκκινο χρώμα μέσω από ένα Holoviews curve object:

```
outliers = [[ind, value] for ind, value in zip(isf_df[isf_df['prediction']==-
1].index, isf_df.loc[isf_df['prediction']==-1,'value'])]
(hv.Curve(isf_df['value'], label="Temperature") * hv.Points(outliers,
label="Detected Points").opts(color='red', legend_position='bottom', size=2,
title="Isolation Forest - Detected Points"))\
    .opts(opts.Curve(xlabel="Time", ylabel="Temperature", width=1500,
height=600,tools=['hover'],show_grid=True))
```

Τα ακραία σημεία ή outliers που εντόπισε ο αλγόριθμος Isolation Forest:



Διάγραμμα 3

Παρατηρούμε ότι ο αλγόριθμος Isolation Forest εντοπίζει επιτυχώς εκείνα τα σημεία που αναμέναμε να αποτελούν ανωμαλίες. Πιο συγκεκριμένα βλέπουμε ότι στις θερμοκρασίες κάτω των 60 βαθμών Celsius και άνω των 100 βαθμών τις αναγνωρίζει ως outliers. Συμφωνα με τις προβλέψεις αυτές συμπεραίνουμε ότι οι ιδανικές συνθήκες λειτουργίας του μεγάλου αυτού βιομηχανικού μηχανήματος είναι μεταξύ 60-100 βαθμούς κελσίου.

Για την μέτρηση της αποτελεσματικότητας του isolation forest μοντέλου για outlier detection χρησιμοποιούμε το f1_score διότι ενδείκνεται για binary classification εργασίες. Για τον υπολογισμό του f1_score συγκρίνουμε τα αποτελέσματα της στήλης “anomalies” του αρχικού μας df με τα αποτελέσματα της στήλης “prediction” του isf_df που μόλις δημιουργήσαμε.

```
isf_f1 = f1_score(df['anomalies'], isf_df['prediction'])
print(f'Isolation Forest F1 Score : {isf_f1}')
```

Το f1_score του isolation forest για ανίχνευση ακραίων τιμών (outlier detection) σε θερμοκρασιακά δεδομένα αγγίζει το **94,85%**.

4.3.2 ONE CLASS SVM MODEL (OUTLIER DETECTION)

Για την εφαρμογή του αλγορίθμου αυτού, δημιουργούμε ένα αντίγραφο του αρχικού DataFrame με το όνομα “ocsvm_df”. Στην συνέχεια δημιουργούμε έναν One Class SVM Classifier όπου το

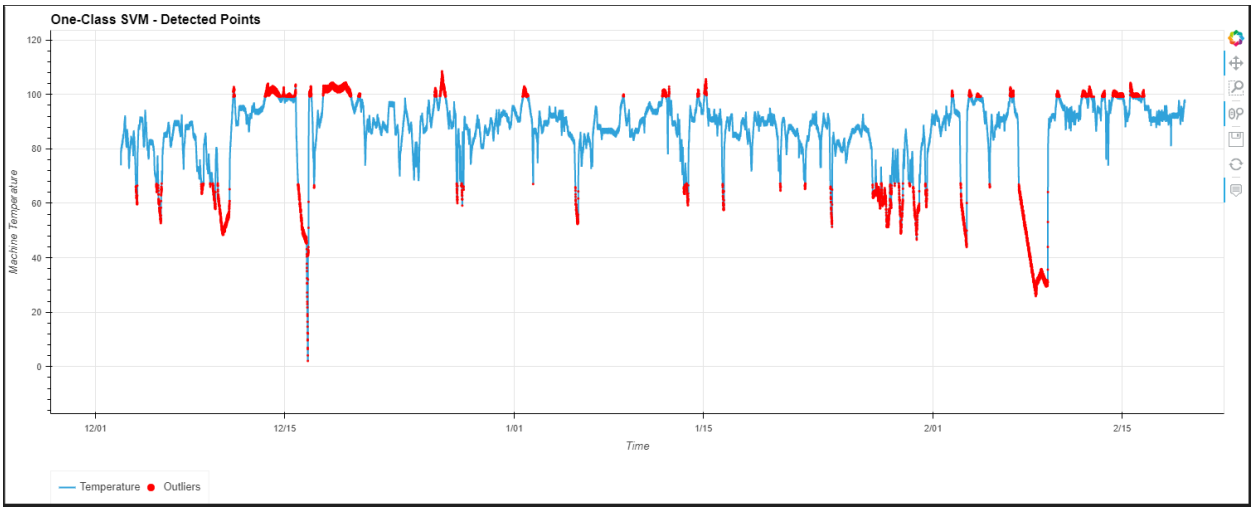
“nu” ελέγχει την αναλογία ακραίων τιμών και παίρνει τιμή από 0 έως 1. Η παράμετρος “gamma” καθορίζει το πλάτος του πυρήνα της συνάρτησης ακτινικής βάσης (RBF) που χρησιμοποιείται από τον αλγόριθμο OC-SVM και ελέγχει την επιρροή μεμονωμένων σημείων δεδομένων στο όριο απόφασης. Η παράμετρος kernel που ονομάζεται και παράμετρος πυρήνα καθορίζει τον τύπο του πυρήνα που χρησιμοποιείται από τον αλγόριθμο One Class SVM. Ο πυρήνας RBF ('rbf') είναι μια δημοφιλής επιλογή, καθώς μπορεί να συλλάβει αποτελεσματικά μη γραμμικές σχέσεις στα δεδομένα. Μετά από αρκετές δοκιμές ο συνδιασμός που μου έδωσε το μεγαλύτερο ποσοστό επιτυχίας ήταν για nu = 0.2 , gamma = 0.001 και kernel = rbf. Τέλος δημιουργούμε μία νέα στήλη στο ocsvm_df με το όνομα “prediction” στην οποία θα τοποθετήσουμε όλες τις τιμές τις οποίες ο αλγόριθμος θα προβλέψει ως ακραίες τιμές με την εντολή “fit_predict()”.

```
ocsvm_df = df.copy()
ocsvm_model =  OneClassSVM(nu=0.2, gamma=0.001, kernel='rbf')
ocsvm_df.loc[:,'prediction'] =
ocsvm_model.fit_predict(ocsvm_df[['value']].values)
```

Απεικονίζουμε γραφικά τα ακραία σημεία που εντόπισε ο αλγόριθμος One Class SVM για την στήλη “value” και δημιουργούμε μία λίστα με το όνομα “outliers” που θα εντοπίζει τα δείγματα της στήλης “prediction” του ocsvm_df με τιμή -1 και θα τα απεικονίζει με κόκκινο χρώμα μέσω ενός Holoviews curve object:

```
outliers = [[ind, value] for ind, value in zip(ocsvm_df[ocsvm_df['prediction']==-
1].index, ocsvm_df.loc[ocsvm_df['prediction']==-1,'value'])]
(hv.Curve(ocsvm_df['value'], label="Temperature") * hv.Points(outliers,
label="Outliers").opts(color='red', legend_position='bottom', size=2, title="One-
Class SVM - Detected Points"))\
    .opts(opts.Curve(xlabel="Time", ylabel="Machine Temperature", width=1500,
height=600,tools=['hover'],show_grid=True))
```

Τα ακραία σημεία που προέβλεψε ο αλγόριθμος One Class SVM :



Διάγραμμα 4

Παρατηρούμε ότι και σε αυτήν την περίπτωση τα αποτελέσματα που παίρνουμε αναδεικνύουν ως ανωμαλίες τις πολύ υψηλές και πολύ χαμηλές θερμοκρασίες όπως ήταν αναμενώμενο για ένα μηχάνημα βιομηχανικού τύπου.

Για τον υπολογισμό του f1_score συγκρίνουμε τα αποτελέσματα της στήλης “anomalies” του αρχικού μας df με τα αποτελέσματα της στήλης “prediction” του ocsvm_df που μόλις δημιουργήσαμε.

```
ocsvm_f1 = f1_score(df['anomalies'], ocsvm_df['prediction'])
print(f'One Class SVM F1 Score : {ocsvm_f1}')
```

Το f1_score του μοντέλου One Class SVM για την ανίχνευση ακραίων τιμών (outlier detection) στα θερμοκρασιακά δεδομένα αγγίζει το **89.80%**.

4.3.3 LOCAL OUTLIER FACTOR MODEL (OUTLIER DETECTION)

Για την εφαρμογή του αλγορίθμου αυτού, δημιουργούμε ένα αντίγραφο του αρχικού DataFrame με το όνομα “lof_df”. Έπειτα δημιουργούμε έναν Local Outlier Factor Classifier όπου το “n_neighbors” υποδικνύει τον αριθμό των γειτόνων που λαμβάνονται υπόψη για τον υπολογισμό της τοπικής πυκνότητας κάθε σημείου δεδομένων. Πιο συγκεκριμένα καθορίζει το μέγεθος της «γειτονιάς» γύρω από κάθε σημείο εντός του οποίου εκτιμάται η τοπική πυκνότητα. Η παράμετρος “contamination” καθορίζει το αναμενώμενο ποσοστό ανωμαλιών στο σύνολο δεδομένων. Μετά από αρκετές δοκιμές ο συνδιασμός που μου έδωσε το μεγαλύτερο ποσοστό

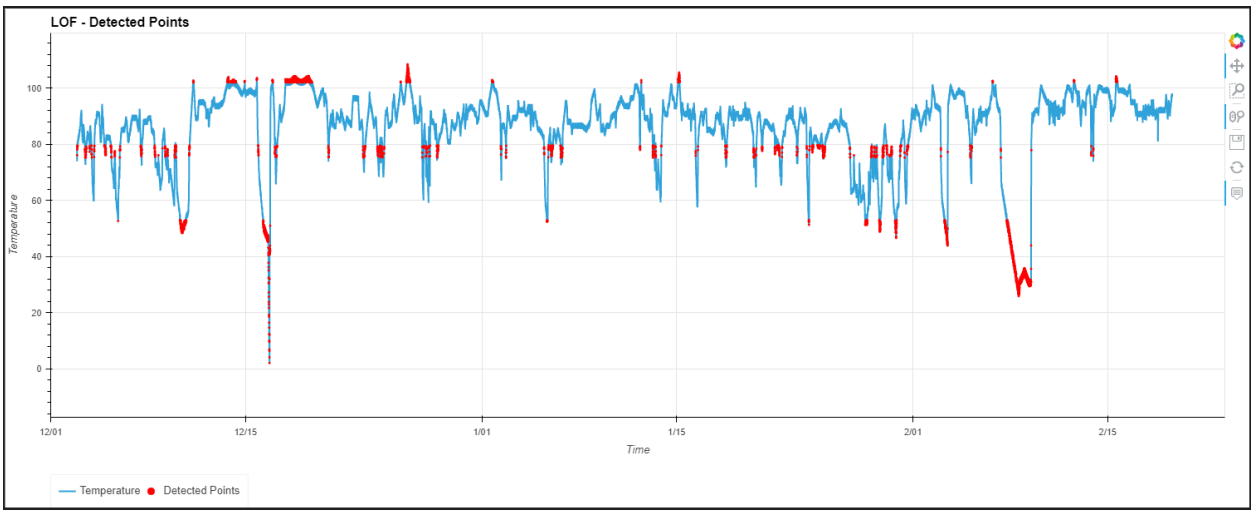
επιτυχίας ήταν για n_neighbors = 800 και contamination = 0.1. Τέλος δημιουργούμε μία νέα στήλη στο lof_df με το όνομα “prediction” στην οποία θα τοποθετήσουμε όλες τις τιμές τις οποίες ο αλγόριθμος θα προβλέψει ως ακραίες τιμές ή ανωμαλίες με την εντολή “fit_predict()”.

```
lof_df = df.copy()
lof_model = LocalOutlierFactor(n_neighbors=800, contamination=0.1)
lof_df.loc[:,'prediction'] = lof_model.fit_predict(lof_df[['value']].values)
```

Απεικονίζουμε γραφικά τα ακραία σημεία που εντόπισε ο αλγόριθμος Local Outlier Factor και δημιουργούμε μία λίστα με το όνομα “outliers” που θα εντοπίζει τα δείγματα της στήλης “prediction” του lof_df με τιμή -1 απεικονίζοντας τα με κόκκινο χρώμα μέσω ενός Holoviews curve object:

```
outliers = [[ind, value] for ind, value in zip(lof_df[lof_df['prediction']==-
1].index, lof_df.loc[lof_df['prediction']==-1,'value'])]
(hv.Curve(lof_df['value'], label="Temperature") * hv.Points(outliers,
label="Detected Points").opts(color='red', legend_position='bottom', size=2,
title="LOF - Detected Points"))\
    .opts(opts.Curve(xlabel="Time", ylabel="Temperature", width=1500,
height=600,tools=['hover'],show_grid=True))
```

Τα ακραία σημεία που προέβλεψε ο αλγόριθμος Local Outlier Factor:



Διάγραμμα 5

Παρατηρούμε ότι και σε αυτή την περίπτωση ο αλγόριθμος LOF καταφέρνει προσδιορίσει τις πολύ χαμηλές και πολύ υψηλές θερμοκρασίες ως ανωμαλίες. Ωστόσο για τις θερμοκρασίες απο 75 – 80 βαθμούς κελσίου παρατηρούμε ότι ο αλγόριθμος έχει προβλέψει και σε εκείνα τα σημεία ανωμαλίες παρά το γεγονός ότι στην πραγματικότητα αποτελούν φυσιολογικές θερμοκρασιακές τιμές. Το γεγονός αυτό οφείλεται στην πυκνότητα των δεδομένων σε αυτή την θερμοκρασιακή κλίμακα δεδομένου ότι ο αλγόριθμος Local Outlier Factor εντοπίζει τις ανωμαλίες με βάση την τοπική τους πυκνότητα.

Για τον υπολογισμό του f1_score συγκρίνουμε τα αποτελέσματα της στήλης “anomalies” του αρχικού μας df με τα αποτελέσματα της στήλης “prediction” του lof_df που μόλις δημιουργήσαμε.

```
lof_f1 = f1_score(df['anomalies'], lof_df['prediction'])
print(f'Local Outlier Factor F1 Score : {lof_f1}')
```

Το f1_score του Local Outlier Factor μοντέλου για ανίχνευση ακραίων τιμών (outlier detection) στα θερμοκρασιακά αυτά δεδομένα αγγίζει το **93.15%**

4.3.4 ELLIPTIC ENVELOPE MODEL (OUTLIER DETECTION)

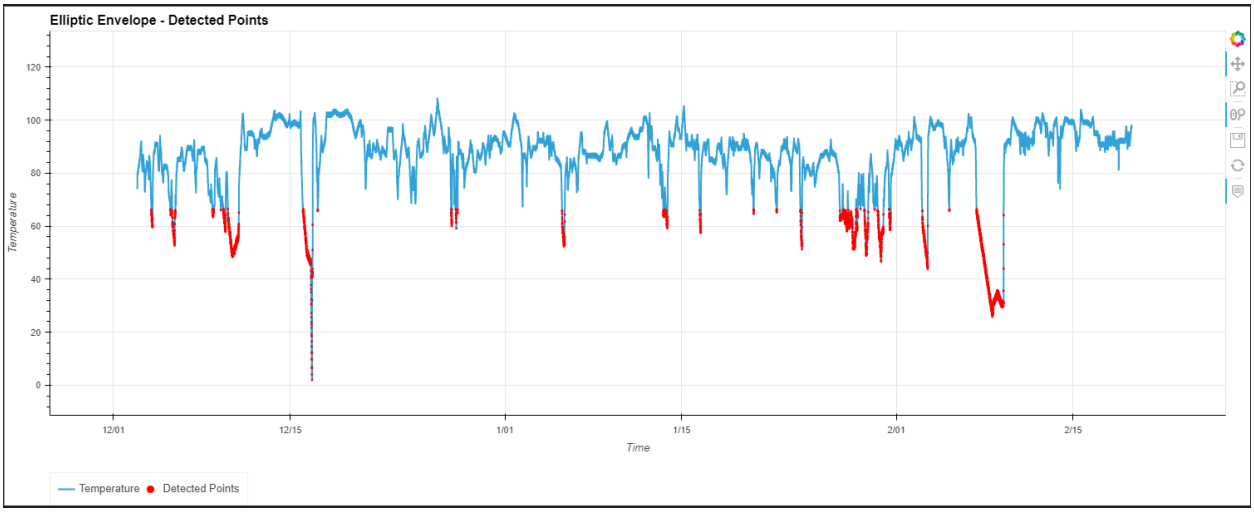
Για την εφαρμογή του αλγορίθμου αυτού, δημιουργούμε ένα αντίγραφο του αρχικού DataFrame με το όνομα “ee_df”. Έπειτα δημιουργούμε έναν Elliptic Envelope Classifier όπου η παράμετρος “contamination” καθορίζει το αναμενώμενο ποσοστό ανωμαλιών στο σύνολο δεδομένων. Μετά από αρκετές δοκιμές ο συνδιασμός που μου έδωσε το μεγαλύτερο ποσοστό ήταν για contamination = 0.1. Τέλος δημιουργούμε μία νέα στήλη στο ee_df με το όνομα “prediction” στην οποία θα τοποθετήσουμε όλες τις τιμές τις οποίες ο αλγόριθμος θα προβλέψει ως ακραίες τιμές ή ανωμαλίες με την εντολή “fit_predict()”.

```
ee_df = df.copy()
ee_model = EllipticEnvelope(contamination=0.1)
ee_df.loc[:,'prediction'] = ee_model.fit_predict(ee_df[['value']].values)
```

Απεικονίζουμε γραφικά τα ακραία σημεία που εντόπισε ο αλγόριθμος Elliptic Envelope και δημιουργούμε μία λίστα με το όνομα “outliers” που θα εντοπίζει τα δείγματα της στήλης “prediction” του lof_df με τιμή -1 απεικονίζοντας τα με κόκκινο χρώμα μέσω ενός Holoviews curve object:

```
outliers = [[ind, value] for ind, value in zip(ee_df[ee_df['prediction']==-
1].index, ee_df.loc[ee_df['prediction']==-1,'value'])]
(hv.Curve(ee_df['value'], label="Temperature") * hv.Points(outliers,
label="Detected Points").opts(color='red', legend_position='bottom', size=2,
title="Elliptic Envelope - Detected Points"))\
    .opts(opts.Curve(xlabel="Time", ylabel="Temperature", width=1500,
height=600,tools=['hover'],show_grid=True))
```

Τα ακραία σημεία που προέβλεψε ο αλγόριθμος Elliptic Envelope:

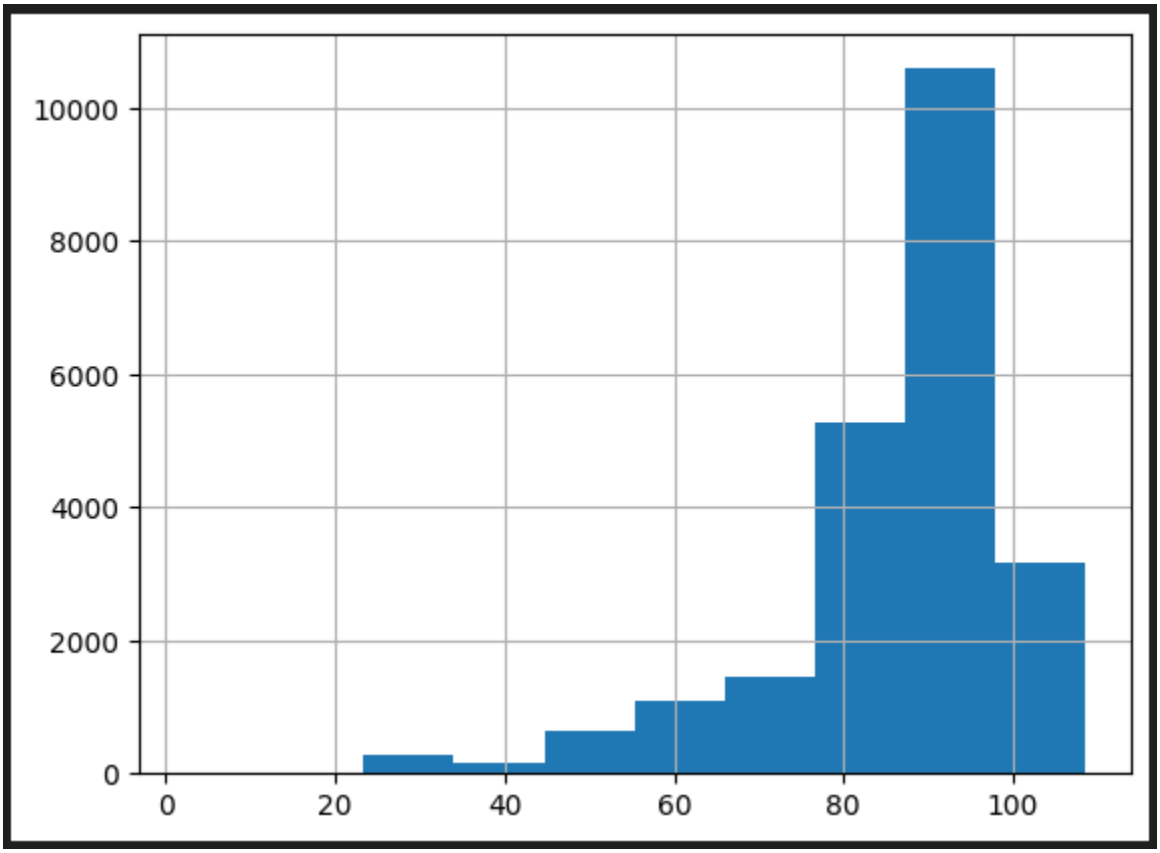


Διάγραμμα 6

Με μία πρώτη ματιά παρατηρούμε ότι ο αλγόριθμος, έχει εντοπίσει ανωμαλίες μόνο στις χαμηλότερες θερμοκρασίες κάτι που είναι εν μέρη σωστό αλλα όχι ιδανικό καθώς ο κίνδυνος βλάβης του μηχανήματος υπάρχει και στις υψηλότερες θερμοκρασίες. Ο λόγους που παρατηρείται αυτό το φαινόμενο είναι διότι ο αλγόριθμος Elliptic Envelope υποθέτει ότι τα δεδομένα ακολουθούν μια Γκαουσιανή κατανομή. Εάν τα δεδομένα έχουν χαμηλό επίπεδο ακραίων τιμών και είναι καλά μοντελοποιημένα από μια κατανομή Gauss (όπως στην συγκεκριμένη περίπτωση), ο αλγόριθμος μπορεί να εκτιμήσει με ακρίβεια τις παραμέτρους της κατανομής και να εντοπίσει ακραίες τιμές που βρίσκονται σημαντικά εκτός της εκτιμώμενης έλλειψης. Πιο αναλυτικά μπορούμε να παρατηρήσουμε με μία απλή γραμμή κώδικα την Gaussian κατανομή των δεδομένων μας δημιουργόντας ένα ιστόγραμμα των θερμοκρασιακών τιμών που διαθέτουμε:

```
df['value'].hist()
```

Το αποτέλεσμα της παραπάνω γραμμής κώδικα θα έχει ως εξής:



Διάγραμμα 7

Όπως ήταν αναμενώμενο τα δεδομένα μας φαίνεται να ακολουθούν με μεγάλη ακρίβεια μία gaussian κατανομή με αποτέλεσμα το μοντέλο καινοτομίας Elliptic Envelope να μπορεί να εντοπίζει ως ανωμαλίες, τις τιμές που εμφανίζονται σπανιότερα στο dataset δηλαδή τις τιμές από 50 βαθμούς και κάτω.

Για τον υπολογισμό του f1_score συγκρίνουμε τα αποτελέσματα της στήλης “anomalies” του αρχικού μας df με τα αποτελέσματα της στήλης “prediction” του ee_df που μόλις δημιουργήσαμε.

```
ee_f1 = f1_score(df['anomalies'], ee_df['prediction'])
print(f'Elliptic Envelope F1 Score : {ee_f1}')
```

Το f1_score του Elliptic Envelope μοντέλου για την ανίχνευση ακραίων τιμών (outlier detection) για αυτά τα θερμοκρασιακά δεδομένα αγγίζει το **95.38%**.

4.4 NOVELTY DETECTION (ΥΛΟΠΟΙΗΣΗ)

4.4.1 Προετοιμασία δεδομένων

Στην περίπτωση του outlier detection χρησιμοποιήθηκε το 100% του συνόλου δεδομένων για εκπαίδευση. Ωστόσο για το Novelty detection θα χωρίσουμε τα δεδομένα μας σε δεδομένα εισόδου X και δεδομένα εξόδου y. Στα δεδομένα εισόδου X έχουμε τις στήλες value, year, month, day, hour και minute του df ενώ τα δεδομένα εξόδου θα αποτελούνται από την στήλη anomalies η οποία περιέχει τις δωσμένες ανωμαλίες από το NAB.

```
X = df.drop(['anomalies'],axis=1)
y = df['anomalies']
```

Στην συνέχεια χωρίζουμε το σύνολο των δεδομένων σε δεδομένα εισόδου για εκπαίδευση (X_train), δεδομένα εισόδου για έλεγχο (X_test), και δεδομένα εξόδου για εκπαίδευση (y_train) και δεδομένα εξόδου για έλεγχο (y_test). Πιο αναλυτικά θα αξιοποιήσουμε το 75% των δεδομένων για εκπαίδευση και το 25% για έλεγχο με την εντολή:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
random_state=18)
print("Οι διαστάσεις του X_train είναι:",X_train.shape)
print("Οι διαστάσεις του X_test είναι:",X_test.shape)
print("Οι διαστάσεις του y_train είναι:",y_train.shape)
print("Οι διαστάσεις του y_test είναι:",y_test.shape)
```

της οποίας το αποτελέσμα θα έχει ως εξής:

```
Οι διαστάσεις του X_train είναι: (17021 6)
Οι διαστάσεις του X_test είναι: (5674 6)
Οι διαστάσεις του y_train είναι: (17021 )
Οι διαστάσεις του y_test είναι: (5674 )
```

Το X_train θα αποτελείται από 17.021 δείγματα και 6 συνόλικα στήλες, το X_test θα αποτελείται από 5.674 δείγματα και 6 στήλες, το y_train από 17.021 δείγματα και μία στήλη και το y_test από 5.674 δείγματα και μία στήλη.

4.4.2 ISOLATION FOREST (NOVELTY DETECTION)

Αρχικά δημιουργούμε έναν isolation forest classifier με παραμέτρους n_estimators = 300 και contamination = 0.05 (η παραμετροποίηση αυτή προέκυψε μετά από μία πληθώρα δοκιμών). Η παράμετρος contamination είναι λογικό να έχει μειωθεί εφόσον μειώθηκαν και τα δεδομένα εκπαίδευση μετά τον προηγούμενο διαχωρισμό. Έπειτα εκπαιδεύουμε το novelty μοντέλο Isolation forest στα δεδομένα X_train.

```
isfN_model = IsolationForest(n_estimators=300, contamination=0.05)
isfN_detector = isfN_model.fit(X_train)
```

Στην συνέχεια δημιουργούμε ένα αντίγραφο του υποσυνόλου X_test που το ονομάζουμε “isfN_df” έτσι ώστε να έχει τον ίδιο αριθμό δειγμάτων με το y_test το οποίο θα χρησιμοποιήσουμε μετέπειτα για την αξιολόγηση του μοντέλου. Τέλος χρησιμοποιούμε το μοντέλο για να προβλέψουμε ποια από τα δεδομένα εισόδου που είχαμε κρατήσει για έλεγχο είναι ανωμαλά και τοποθετούμε τα αποτελέσματα σε μία καινούργια στήλη με το όνομα “isfN_prediction”.

```
isfN_df = X_test.copy()
isfN_df.loc[:,'isfN_prediction'] = isfN_model.predict(X_test)
isfN_df.head()
```

το σύνολο δεδομένων isfN_df έχει πλεόν την εξής μορφή:

	value	year	month	day	hour	minute	isfN_prediction
timestamp							
2014-01-31 05:40:00	81.941526	2014	1	31	5	40	1
2013-12-17 16:35:00	90.749277	2013	12	17	16	35	1
2013-12-13 05:50:00	93.603815	2013	12	13	5	50	1
2013-12-03 04:25:00	90.302344	2013	12	3	4	25	1
2014-02-13 05:10:00	94.189268	2014	2	13	5	10	1

Συγκρίνουμε τα δείγματα της στήλης “isfN_prediction” με τις σίγουρες ανωμαλίες του συνόλου y_test για να υπολογίσουμε το f1_score και το accuracy του μοντέλου.

```
isfN_f1 = f1_score(isfN_df['isfN_prediction'], y_test)
print(f'Isolation Forest Novelty model F1 Score : {isfN_f1}')
isfN_acc = accuracy_score(isfN_df['isfN_prediction'], y_test)
print(f'Isolation Forest Novelty model accuracy Score : {isfN_acc}')
```

Το f1_score του μοντέλου isolation forest για το novelty detection είναι **94,88%** ενώ το accuracy **90.55%**

Κατα παρόμοιο τρόπο ενεργούμε και για τους υπόλοιπους αλγορίθμους που αναφέραμε.

4.4.3 ONE CLASS SVM (NOVELTY DETECTION)

Εκπαίδευση μοντέλου OCSVM

```
ocsvmN_model = OneClassSVM(nu=0.2, gamma=0.001, kernel='rbf')
ocsvmN_detector = ocsvmN_model.fit(X_train)
```

Πρόβλεψη στα δεδομένα ελέγχου X_test

```
ocsvmN_df = X_test.copy()
```

```
ocsvmN_df.loc[:,'ocsvmN_prediction'] = ocsvmN_model.predict(X_test)
```

Υπολογισμός f1_score και accuracy

```
ocsvmN_f1 = f1_score(ocsvmN_df['ocsvmN_prediction'], y_test)
print(f'One Class SVM Novelty model F1 Score : {ocsvmN_f1}')
ocsvmN_acc = accuracy_score(ocsvmN_df['ocsvmN_prediction'], y_test)
print(f'One Class SVM Novelty model accuracy Score : {ocsvmN_acc}')
```

Το f1_score του μοντέλου One Class SVM για το novelty detection είναι **88.24%** ενώ το accuracy **80.06%**

4.4.3 LOCAL OUTLIER FACTOR (NOVELTY DETECTION)

Εκπαίδευση μοντέλου Local Outlier Factor. Στον συγκεκριμένο αλγόριθμο είναι σημαντικό να θέσουμε την παράμετρο Novelty = True για να τον αξιοποιήσουμε για ανίχνευση καινοτομίας.

```
lofN_model = LocalOutlierFactor(n_neighbors=600, contamination=0.05,
novelty=True)
lofN_detector = lofN_model.fit(X_train)
```

Πρόβλεψη στα δεδομένα ελέγχου X_test

```
lofN_df = X_test.copy()
lofN_df.loc[:,'lofN_prediction'] = lofN_model.predict(X_test)
```

Υπολογισμός f1_score και accuracy

```
lofN_f1 = f1_score(lofN_df['lofN_prediction'], y_test)
print(f'Local Outlier Factor Novelty model F1 Score : {lofN_f1}')
lofN_acc = accuracy_score(lofN_df['lofN_prediction'], y_test)
print(f'Local Outlier Detection Novelty model accuracy Score : {lofN_acc}')
```

Το f1_score του μοντέλου Local Outlier Factor για το novelty detection είναι **93,53 %** ενώ το accuracy **88,10%**

4.4.4 ELLIPTIC ENVELOPE (NOVELTY DETECTION)

Εκπαίδευση μοντέλου Elliptic Envelope

```
eeN_model = EllipticEnvelope(contamination=0.05)
eeN_detector = eeN_model.fit(X_train)
```

Πρόβλεψη στα δεδομένα ελέγχου X_test

```
eeN_df = X_test.copy()
eeN_df.loc[:,'eeN_prediction'] = eeN_model.predict(X_test)
```

Υπολογισμός f1_score και accuracy

```
eeN_f1 = f1_score(eeN_df['eeN_prediction'], y_test)
print(f'Elliptic Envelope Novelty model F1 Score : {eeN_f1}')
eeN_acc = accuracy_score(eeN_df['eeN_prediction'], y_test)
print(f'Elliptic Envelope Novelty model accuracy Score : {eeN_acc}')
```

Το f1_score του μοντέλου Elliptic Envelope για το novelty detection είναι **94,02 %** ενώ το accuracy **88,94%**

4.4.5 ENSEMBLY MODEL (STACKING CLASSIFIER)

Για την κατασκευή του μοντέλου με την μέθοδο Stacking θα αξιοποιήσουμε τους αλγορίθμους που «ταιριάζουν» καλύτερα στα δεδομένα που διαθέτουμε. Από τα διαγράμματα 3 και 4 συμπαιρένουμε ότι οι αλγόριθμοι που έχουν την ικανότητα να προβλέψουν καλύτερα στα δεδομένα μας είναι ο αλγόριθμος Isolation Forest και ο One Class SVM. Για τον λόγο αυτό θα

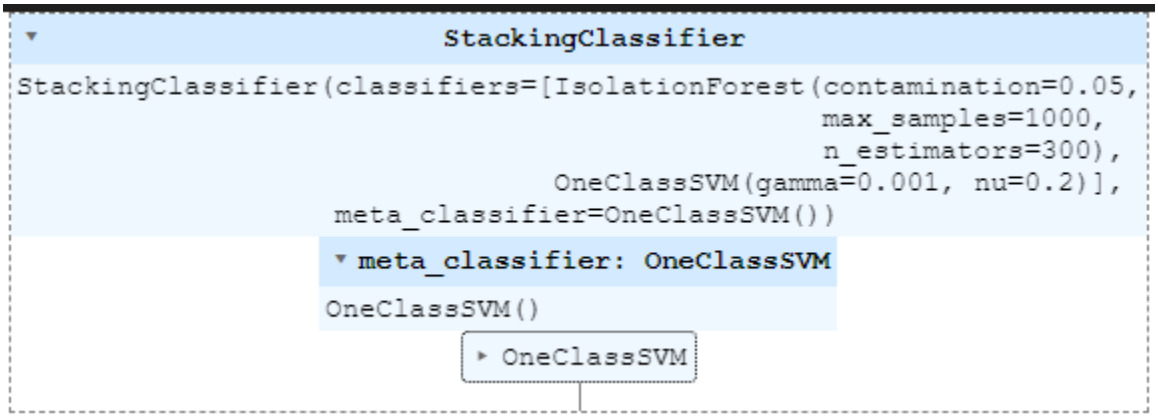
αξιοποιήσουμε τα novelty αυτά μοντέλα ως base models. Στην συνέχεια θα χρησιμοποιήσουμε τις προβλέψεις αυτών των μοντέλων ως δεδομένα εισόδου στο meta model classifier που θα κάνει χρήση του αλγορίθμου OCSVM. Η επιλογή του meta classifier προέκυψε από μία πληθώρα δοκιμών με διαφορετικούς τύπους ταξιμομητών (όπως Logistic Regression, knn classifier κ.α.) μεταξύ των οποίων ο OCSVM απέδωσε τα βέλτιστα.

```
classifiers = [isfN_model, ocsvmN_model]
stacking_model = StackingClassifier(classifiers=[clf.fit(X_train, y_train) for
clf in classifiers], meta_classifier= OneClassSVM())
```

Στην συνέχεια εκπαιδεύουμε το Stacking μοντέλο στα δεδομένα εκπαίδευσης X_train

```
stacking_model.fit(X_train, y_train)
```

Στο παρακάτω σχήμα (διάγραμμα 8) μπορούμε να δούμε την αρχιτεκτονική του μοντέλου Stacking ή οποία αποτελείται από 2 επίπεδα. Το Stacking Classifier που αποτελείται από τον συνδιασμό των “isfN_model” και του “ocsvmN_model” και το meta_classifier ή αλλιώς meta model που αποτελείται από έναν OCSVM classifier.



Διάγραμμα 8

Στην συνέχεια χρησιμοποιούμε το stacking μοντέλο για να προβλέψουμε τα δεδομένα του X_test που είχαμε κρατήσει για έλεγχο.

```
y_pred = stacking_model.predict(X_test)
```

Και υπολογίζουμε το f1 score και το accuracy του stacking μοντέλου συγκρίνοντας τα αποτελέσματα των προβλέψεων που πήραμε από το y_pred με τις σίγουρες ανωμαλίες y_test

```
stacking_f1 = f1_score(y_pred, y_test)
print(f'Stacking model F1 Score : {stacking_f1}')
stacking_acc = accuracy_score(y_pred, y_test)
print(f'Stacking model accuracy Score : {stacking_acc}')
```

Το f1_score του stacking μοντέλου για το novelty detection είναι **88,21 %** ενώ το accuracy **80,01 %**.

4.6 ΔΟΚΙΜΗ ΜΟΝΤΕΛΩΝ

Για την δοκιμή των Novelty μοντέλων δημιουργούμε 3 νέα δείγματα τα οποία θα περιλαμβάνουν μία θερμοκρασιακή τιμή και μία ημερομηνία χωρισμένη στις στήλες year, month, day, hour και minute. Έπειτα δημιουργούμε μία λίστα που θα περιέχει αυτές τις τιμές και την ονομάζουμε “new_data”.

```
nd1 = np.asarray([['110.749277','2015','12','17','10','35']], dtype= float)
nd2 = np.asarray([['55.181315','2013','3','20','16','00']], dtype= float)
nd3 = np.asarray([['88.123548','2014','6','8','18','20']], dtype= float)
new_data = [nd1, nd2, nd3]
```

Όπως παρατηρούμε η πρώτη τιμή nd1 περιέχει μία υψηλή τιμή θερμοκρασίας και το ίδιο και η δεύτερη τιμή nd2 περιέχει μία χαμηλή τιμή θερμοκρασίας. Οι τιμές αυτές αποτελούν εσκεμένες ανωμαλίες που δεν απέχουν ωστόσο εξωφρενικά από τις υπόλοιπες φυσιολογικές τιμές και αυτό γιατί επιθυμούμε να δοκιμάσουμε το μοντέλο σε αμφιλεγώμενες ανωμαλίες που είναι δύσκολο να εντοπιστούν. Τέλος η τιμή nd3 αποτελεί μία τιμή που αναμένουμε να εμπίπτει στα φυσιολογικά πλάισια λειτουργίας του βιομηχανικού μηχανήματος.

Έπειτα δημιουργούμε μία for loop η οποία θα χρησιμοποιεί τις παραπάνω τιμές της λίστα new_data για πρόβλεψη σε κάθε ένα από τα Novelty μοντέλα που δημιουργήσαμε όπως φαίνεται παρακάτω:

Δοκιμή του isolation forest novelty μοντέλου:

```
for i in new_data:
    is_anomalous = isfN_model.predict(i)
    if is_anomalous == -1:
        print("The new data point is anomalous!")
    else:
        print("The new data point is not anomalous.")
```

Δοκιμή του One Class SVM novelty μοντέλου:

```
for i in new_data:
    is_anomalous = ocsvmN_model.predict(i)
    if is_anomalous == -1:
        print("The new data point is anomalous!")
    else:
        print("The new data point is not anomalous.")
```

Δοκιμή του Local Outlier Factor novelty μοντέλου:

```
for i in new_data:
    is_anomalous = lofN_model.predict(i)
    if is_anomalous == -1:
        print("The new data point is anomalous!")
    else:
        print("The new data point is not anomalous.")
```

Δοκιμή του Elliptic Envelope novelty μοντέλου:

```
for i in new_data:
    is_anomalous = eeN_model.predict(i)
    if is_anomalous == -1:
        print("The new data point is anomalous!")
    else:
        print("The new data point is not anomalous.")
```

Δοκιμή του Stacking μοντέλου για Novelty detection:

```
for i in new_data:
    is_anomalous = stacking_model.predict(i)
    if is_anomalous == -1:
        print("The new data point is anomalous!")
    else:
        print("The new data point is not anomalous.")
```

Τα αποτελέσματα των προβλέψεων όλων των παραπάνω 5 μοντέλων καινοτομίας, φαίνονται συγκεντρωμένα στον παρακάτω πίνακα:

Μοντέλα	nd1 (anomaly)	nd2 (anomaly)	nd3(not anomalous)
Isolation Forest	1	-1	1
One Class SVM	-1	-1	1
Local Outlier Factor	-1	1	1
Elliptic Envelope	1	1	1
Stacking model	-1	-1	1

Πίνακας Αποτελεσμάτων για Novelty Detection

Στον πίνακα αποτελεσμάτων των μοντέλων καινοτομίας με τον αριθμό 1 συμβολίζουμε τις προβλέψεις που δεν αποτελούν ανωμαλία ενώ με τον αριθμό -1 συμβολίζουμε τις ανωμαλίες που εντοπίστηκαν. Με πράσινο χρώμα αναγράφουμε τις προβλέψεις που αναμένουμε να είναι αληθείς (σύμφωνα με τα δεδομένα μας) ενώ στην περίπτωση που είναι λανθασμένες τις απεικονίζουμε με κόκκινο χρώμα.

4.6.1 Παρατηρήσεις

Σύμφωνα με τον πίνακα αποτελεσμάτων, συμπεραίνουμε ότι από τα μεμωνομένα μοντέλα που χρησιμοποιήσαμε, το μοντέλο Elliptic Envelope έχει την ασθενέστερη απόδοση από τα υπόλοιπα. Στον αντίποδα το μοντέλο One Class SVM φαίνεται να αποδίδει καλύτερα τα αποτελέσματα καταφέρνοντας να προβλέψει επιτυχώς και τα 3 νέα δείγματα που καλύπτουν όλο το θερμοκρασιακό εύρος του συνόλου δεδομένων. Παρατηρούμε επίσης ότι τα μοντέλα Isolation Forest και Local Outlier Factor κατάφεραν να πραγματοποιήσουν επιτυχώς 2 σωστές προβλέψεις το καθένα. Τέλος όπως ήταν αναμενόμενο το Stacking μοντέλο που συνδιάζει τα μοντέλα One Class SVM και Isolation Forest, μας αποδίδει και τα 3 νέα δείγματα με τις σωστές προβλέψεις που αναμέναμε.

Άξιο παρατήρησης είναι επίσης το γεγονός ότι το μεμωνομένο μοντέλο One Class SVM προβλέπει με μεγαλύτερη επιτυχία τις νέες τιμές (nd1, nd2, nd3) που χρησιμοποιήσαμε για πρόβλεψη σε σχέση με το Isolation Forest μοντέλο παρά το γεγονός ότι έχει μικρότερο f1 score. Το γεγονός αυτό οφείλεται στο ότι το SVM μιας κατηγορίας έχει ένα συγκεκριμένο πλεονέκτημα σε ορισμένα σενάρια, όπως για παράδειγμα όταν οι καινοτομίες μοιάζουν περισσότερο με την πλειοψηφική κατηγορία όπως στην συγκεκριμένη περίπτωση. Το One Class SVM έχει σχεδιαστεί για να δημιουργεί ένα όριο απόφασης γύρω από την πλειονότητα των σημείων δεδομένων και στόχος του είναι να εντοπίσει περιπτώσεις που αποκλίνουν σημαντικά από αυτό το όριο. Σε τέτοιες περιπτώσεις, το SVM μιας κατηγορίας μπορεί να είναι πιο αποτελεσματικό στον εντοπισμό των λεπτών διαφορών στα δεδομένα. Από την άλλη πλευρά, ο αλγόριθμος Isolation Forest κατασκευάζει τυχαία δυαδικά δέντρα για να απομονώσει ανωμαλίες. Εάν οι καινοτομίες στα νέα δεδομένα βρίσκονται σε περιοχές με αραιά κατανεμημένα δεδομένα ή παρουσιάζουν σημαντικά διαφορετικά χαρακτηριστικά από την πλειοψηφική τάξη, το Isolation Forest μπορεί να είναι πιο αποτελεσματικό στην καταγραφή αυτών των ανωμαλιών.

4.7 Συμπεράσματα

Ανάμεσα στο Stacking μοντέλο και στο μεμονωμένο μοντέλο καινοτομίας One Class SVM θα επιλέξουμε ως καλύτερό μοντέλο εκείνο που δημιουργήθηκε με την μέθοδο του Stacking. Ο λόγος για τον οποίο οδηγούμαστε σε αυτή την επιλογή έγκειται στο ότι με τον συνδυασμό πολλαπλών μοντέλων με την μέθοδο Stacking, μας παρέχεται η δυνατότητα να αξιοποιήσουμε τα μεμονωμένα δυνατά σημεία των αλγορίθμων και να αντισταθμίσουμε έτσι τις αδυναμίες τους. Το μοντέλο καινοτομίας Isolation Forest και One-Class SVM έχουν διαφορετικούς βασικούς αλγόριθμους και υποθέσεις και ο συνδυασμός τους μπορεί να οδηγήσει σε καλύτερα αποτελέσματα. Επιπλέον τα stacking μοντέλα μπορούν να μειώσουν τον κίνδυνο του overfitting καθώς κάθε μοντέλο στη στοίβα εκπαιδεύεται σε διαφορετικές πτυχές δεδομένων, γεγονός που βοηθά στον ευκολότερο εντοπισμό διαφορετικών μοτίβων δεδομένων. Κατα αυτόν τον τρόπο οδηγούμαστε σε ένα πιο αξιόπιστο και γενικεύσιμο μοντέλο. Επιπρόσθετα ο συνδυασμός του Isolation Forest με το μοντέλο καινοτομίας One-Class SVM, μπορεί να αξιοποιήσει τα δυνατά σημεία και των δύο μοντέλων για να εντοπίσει και να προβλέψει αποτελεσματικά τις ανωμαλίες. Τέλος τα μοντέλα στοίβαξης παρουσιάζουν μεγαλύτερη προσαρμοστικότητα στην αλλαγή των μοτίβων δεδομένων σε σύγκριση με τα μεμονωμένα μοντέλα. Πιο συγκεκριμένα εάν τα χαρακτηριστικά του συνόλου δεδομένων που διαθέτουμε εξελίσσονται με την πάροδο του χρόνου ή εφανίζονται νέοι τύποι ανωμαλιών, το μοντέλο στοίβαξης μπορεί να είναι πιο αποτελεσματικό στην καταγραφή αυτών των αλλαγών προσαρμόζοντας το σε νέες προκλήσεις στα δεδομένα.

ΤΕΛΟΣ

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

ΕΙΚΟΝΕΣ

1. https://www.oreilly.com/api/v2/epubs/9781492078180/files/assets/aiml_2001.png
2. https://www.frontiersin.org/files/Articles/1130229/fonc-13-1130229-HTML/image_m/fonc-13-1130229-g001.jpg
3. https://www.linkedin.com/pulse/what-overfitting-underfitting-he-hao/
4. https://static.javatpoint.com/tutorial/machine-learning/images/stacking-in-machine-learning2.png
5. https://cdn.analyticsvidhya.com/wp-content/uploads/2020/11/outlier.png
6. https://miro.medium.com/v2/resize:fit:720/format:webp/1*cY9PvTtR_B2NdJEo0WOUvA.png
7. https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F260738754%2Ffigure%2Ffig2%2FAS%3A667618331598856%401536184022668%2FA-simple-example-of-contextual-anomalies-on-f-T-Here-anomalies-are-defined-as-the.png&tbnid=Zqbqb75hLmIdyM&vet=12ahUKEwjBxdeBs-r9AhVNiP0HHak_B4cQMygBegUIARClAQ..i&imgrefurl=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FA-simple-example-of-contextual-anomalies-on-f-T-Here-anomalies-are-defined-as-the_fig2_260738754&docid=OcWDR_XUgoTM0M&w=617&h=343&q=contextual%20anomalies&ved=2ahUKEwjBxdeBs-r9AhVNiP0HHak_B4cQMygBegUIARClAQ
8. https://www.researchgate.net/profile/Vipin-Kumar-54/publication/220565847/figure/fig4/AS:668202891763715@1536323392099/Collective-anomaly-corresponding-to-an-Atrial-Premature-Contraction-in-an-human.png
9. https://www.tuv.com/content-media-files/germany/bs-industrial-service/landingpages/functional-safety-training-cyber-security/distribution-pages/tuv-rheinland-fs-engineer-functional-safety-of-machinery-sk-587205803.jpg
10. https://axa.biopapyrus.jp/media/one-class-svm-result-01.png
11. https://www.baeldung.com/wp-content/uploads/sites/4/2021/03/svm-all.png

12. https://www.researchgate.net/publication/352017898/figure/fig1/AS:1029757483372550@1622524724599/Isolation-Forest-learned-iForest-construction-for-toy-dataset.png
13. https://scikit-learn.org/stable/_images/sphx_glr_plot_lof_outlier_detection_001.png
14. https://i.stack.imgur.com/Nw3dx.png

ΠΗΓΕΣ

1. Lavin, A., & Ahmad, S. (2015). *Evaluating Real-Time Anomaly Detection Algorithms -- The Numenta Anomaly Benchmark. 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA).* doi:10.1109/icmla.2015.141
2. Singh, N., & Olinsky, C. (2017). Demystifying Numenta anomaly benchmark. 2017 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2017.7966038
3. Mao, W., Cao, X., zhou, Q., Yan, T., & Zhang, Y. (2018). *Anomaly Detection for Power Consumption Data based on Isolated Forest. 2018 International Conference on Power System Technology (POWERCON).* doi:10.1109/powercon.2018.8602251
4. Braei, M. and Wagner, S. (2020) *Anomaly detection in Univariate Time-Series: A Survey on the state-of-the-art, arXiv.org*. Available at: https://arxiv.org/abs/2004.00433 (Accessed: 15 June 2023).
5. *A survey of network anomaly detection techniques*. (2015, December 11). A Survey of Network Anomaly Detection Techniques - ScienceDirect. https://doi.org/10.1016/j.jnca.2015.11.016
6. https://avinetworks.com/glossary/anomaly-detection/
7. https://www.datrics.ai/anomaly-detection-definition-best-practices-and-use-cases
8. https://towardsdatascience.com/a-note-about-finding-anomalies-f9cedee38f0b
9. https://millimetric.ai/blog/5-key-benefits-of-anomaly-detection/
10. https://www.qlik.com/us/kpi
11. https://www.ibm.com/topics/machine-learning
12. https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps
13. https://aws.amazon.com/what-is/overfitting/
14. https://elitedatascience.com/overfitting-in-machine-learning
15. https://www.ibm.com/topics/bagging
16. https://aws.amazon.com/what-is/boosting/
17. https://vitalflux.com/stacking-classifier-sklearn-python-example/
18. https://www.analyticsvidhya.com/blog/2021/03/advanced-ensemble-learning-technique-stacking-and-its-variants/
19. https://www.seldon.io/supervised-vs-unsupervised-learning-explained
20. https://www.datarobot.com/blog/semi-supervised-learning/
21. https://www.seldon.io/outlier-detection-and-analysis-methods
22. https://www.techopedia.com/definition/30345/novelty-detection
23. https://www.kaggle.com/datasets/boltzmannbrain/nab

24. https://arxiv.org/abs/1510.03336
25. https://jupyter-notebook.readthedocs.io/en/stable/notebook.html
26. https://medium.com/analytics-vidhya/anaconda-navigator-an-overview-4e5d27ca8047
27. https://pypi.org/project/pandas/
28. https://numpy.org/doc/stable/user/whatisnumpy.html
29. https://scikit-learn.org/stable/datasets/toy_dataset.html
30. https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/
31. https://holoviews.org/
32. https://notebook.community/vascotenner/holoviews/doc/Tutorials/Introduction
33. https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/
34. https://www.scaler.com/topics/matplotlib/matplotlib-in-python/
35. https://holoviews.org/user_guide/Plotting_with_Bokeh.html
36. https://www.analyticsvidhya.com/blog/2022/06/one-class-classification-using-support-vector-machines/
37. https://www.baeldung.com/cs/svm-feature-scaling
38. https://blog.paperspace.com/anomaly-detection-isolation-forest/
39. https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/
40. https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html
41. https://towardsdatascience.com/local-outlier-factor-lof-algorithm-for-outlier-identification-8efb887d9843
42. https://www.geeksforgeeks.org/local-outlier-factor/
43. https://medium.com/@pramodch/understanding-lof-local-outlier-factor-for-implementation-1f6d4ff13ab9
44. https://aiblog.co.za/technology/elliptic-envelope-anomaly-detection
45. https://www.geeksforgeeks.org/what-is-reinforcement-learning/
46. https://online.york.ac.uk/what-is-reinforcement-learning/
47. https://github.com/numenta/NAB
48. https://towardsdatascience.com/6-ways-to-test-for-a-normal-distribution-which-one-to-use-9dcf47d8fa93
49. https://medium.com/datasciencearth/local-outlier-factor-7821b5651bc5
50. https://impoff.com/importance-of-industry/
51. https://iap.unido.org/articles/why-industrial-development-matters-now-more-ever
52. https://towardsdatascience.com/anomaly-detection-in-manufacturing-part-1-an-introduction-8c29f70fc68b
53. https://towardsdatascience.com/4-machine-learning-techniques-for-outlier-detection-in-python-21e9cfacb81d