ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Εντοπισμός Ανωμαλιών με Χρήση Δεδομένων Παρατηρησιμότητας Κατανεμημένων Εφαρμογών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεράσιμος Σ. Μουντάκης

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2024

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Εντοπισμός Ανωμαλιών με Χρήση Δεδομένων Παρατηρησιμότητας Κατανεμημένων Εφαρμογών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεράσιμος Σ. Μουντάκης

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16/10/2024.

.....	.....	.....
Συμεών Παπαβασιλείου	Ελένη Στάη	Γεώργιος Ματσόπουλος
Καθηγητής Ε.Μ.Π	Επ. Καθηγήτρια Ε.Μ.Π	Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2024

.....
Γεράσιμος Σ. Μουντάκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Στην εποχή της κατανεμημένης πληροφορικής, οι μικροϋπηρεσίες έχουν αναδειχθεί σε κυρίαρχο αρχιτεκτονικό στυλ για την ανάπτυξη και την κλιμάκωση εφαρμογών και υπηρεσιών. Ωστόσο, η διασφάλιση της αξιοπιστίας και της σταθερότητας τέτοιων συστημάτων παρουσιάζει σημαντικές προκλήσεις, ιδίως όσον αφορά τον εντοπισμό ανωμαλιών που θα μπορούσαν να οδηγήσουν σε υποβάθμιση της απόδοσης ή διακοπές λειτουργίας. Η παρούσα εργασία προτείνει μία νέα μέθοδο για την πολυτροπική ανίχνευση ανωμαλιών σε αρχιτεκτονικές βασισμένες σε μικροϋπηρεσίες, αξιοποιώντας αρχεία καταγραφής (logs), ίχνη εκτέλεσης (traces) και μετρικές (metrics). Η προσέγγιση που προτείνεται ενσωματώνει μεθόδους που βασίζονται σε πρότυπα, χρησιμοποιώντας τον αλγόριθμο Drain για την ανάλυση αρχείων καταγραφής, μαζί με τεχνικές Μηχανικής Μάθησης, όπως τους αλγορίθμους Birch, Local Outlier Factor και Isolation Forest για τον εντοπισμό ανωμαλιών σε πραγματικό χρόνο. Κάθε τύπος δεδομένων επεξεργάζεται με μεθόδους ειδικά ρυθμισμένες για τη μεγιστοποίηση των δυνατοτήτων ανίχνευσης ανωμαλιών στο συγκεκριμένο τομέα. Αυτές οι εξειδικευμένες μέθοδοι ενσωματώνονται στη συνέχεια σε ένα ενοποιημένο μοντέλο που συνδυάζει τα πλεονεκτήματά τους, παρέχοντας μια πιο ακριβή ανάλυση. Με την ταυτόχρονη ανάλυση διαφορετικών τύπων δεδομένων, η τεχνική ανίχνευσης αυτή παρέχει μια ολοκληρωμένη εικόνα της υγείας του συστήματος, βοηθώντας στον εντοπισμό προβλημάτων που μπορεί να μην εντοπιστούν από τις παραδοσιακές μεθόδους που εστιάζουν σε μεμονωμένες ροές δεδομένων. Η αποτελεσματικότητα των προτεινόμενων μεθόδων αξιολογείται και συγκρίνεται με εκτέλεση πειραμάτων σε μια ειδικά προσαρμοσμένη εφαρμογή που προσομοιώνει το κατανεμημένο περιβάλλον εκτέλεσης, παρουσιάζοντας βελτιωμένα ποσοστά ανίχνευσης σε σύγκριση με τις βασικές προσεγγίσεις. Η παρούσα εργασία συμβάλλει στον αναπτυσσόμενο τομέα της αυτοματοποιημένης ανίχνευσης ανωμαλιών σε κατανεμημένα συστήματα, με πιθανές εφαρμογές στην παρακολούθηση υποδομών νέφους και στη διαχείριση συστημάτων μεγάλης κλίμακας.

Λέξεις Κλειδιά:
μικροϋπηρεσίες, κατανεμημένα συστήματα, παρατηρησιμότητα, αρχεία καταγραφής, ανάλυση αρχείων καταγραφής, ίχνη, μετρικές, ανίχνευση ανωμαλιών, ανίχνευση ακραίων τιμών, μηχανική μάθηση

Abstract

In the era of distributed computing, microservices have emerged as the dominant architectural style for developing and scaling applications and services. However, ensuring the reliability and stability of such systems presents significant challenges, especially in terms of detecting anomalies that could lead to performance degradation or outages. This work proposes a novel method for multimodal anomaly detection in microservice-based architectures by exploiting logs, traces and metrics. The proposed approach integrates template-based methods, using the Drain algorithm for log analysis, along with Machine Learning techniques such as Birch, Local Outlier Factor and Isolation Forest algorithms for real-time anomaly detection. Each data type is processed with methods specifically tuned to maximize the anomaly detection capabilities in the specific domain. These specialized methods are then integrated into a unified model that combines their strengths, providing a more accurate analysis. By analyzing different types of data simultaneously, this detection technique provides a comprehensive picture of system health, helping to identify problems that may be missed by traditional methods that focus on individual data streams. The effectiveness of the proposed methods is evaluated and compared by running experiments in a customized application that simulates the distributed execution environment, showing improved detection rates compared to the baseline approaches. This work contributes to the growing field of automated anomaly detection in distributed systems, with potential applications in cloud infrastructure monitoring and large-scale system management.

Keywords:
microservices, distributed systems, observability, logs, log parsing, traces, metrics, anomaly detection, outlier detection, machine learning

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1 - Εισαγωγή

1.1 Κίνητρο Εργασίας

Στη σημερινή ψηφιακή εποχή, τα σύγχρονα συστήματα έχουν γίνει ολοένα και πιο πολύπλοκα, με τους οργανισμούς να βασίζονται σε μεγάλο βαθμό σε μεγάλης κλίμακας κατανεμημένες αρχιτεκτονικές για την υποστήριξη των εφαρμογών και των υπηρεσιών τους. Καθώς οι δομές αυτές κλιμακώνονται, παράγουν τεράστιες και συνεχείς ροές δεδομένων με τη μορφή αρχείων καταγραφής, ιχνών και μετρικών. Τα δεδομένα αυτά είναι ανεκτίμητα για την παρακολούθηση της απόδοσης του συστήματος, τον εντοπισμό αποτυχιών και τη διασφάλιση της αξιοπιστίας του συστήματος. Ωστόσο, ο μεγάλος όγκος, η ποικιλομορφία και η ταχύτητα παραγωγής αυτών των δεδομένων μπορεί να υπερκεράσει τις δυνατότητες των παραδοσιακών εργαλείων και τεχνικών παρακολούθησης, καθιστώντας την ανίχνευση ανωμαλιών ένα ολοένα και πιο σύνθετο και κρίσιμο έργο για τους διαχειριστές και τους μηχανικούς συστημάτων.

Οι ανωμαλίες (ασυνήθιστα μοτίβα ή συμπεριφορές) μπορεί να σηματοδοτήσουν πιθανές αποτυχίες, υποβάθμιση της ποιότητας των υπηρεσιών ή παραβιάσεις ασφαλείας. Ο εντοπισμός τέτοιων γεγονότων μέσα στο θόρυβο της κανονικής συμπεριφοράς του συστήματος δεν είναι πάντα απλός. Οι παραδοσιακές τεχνικές ανίχνευσης ανωμαλιών συχνά βασίζονται σε μεμονωμένες πηγές δεδομένων όπως αρχεία καταγραφής, ή μετρικές ή χρησιμοποιούν προκαθορισμένα συστήματα βασισμένα σε σύνολα κανόνων. Τεχνικές αυτής της μορφής έχουν καταστεί ανεπαρκείς για τη διαχείριση της πολυπλοκότητας που επιφέρει η αυξανόμενη χρήση κατανεμημένων συστημάτων, όπως αυτά που συναντώνται σε περιβάλλοντα νέφους (cloud). Για παράδειγμα, τα αρχεία καταγραφής από μόνα τους δεν μπορούν να καταγράψουν όλα τα σημάδια μιας επικείμενης αποτυχίας, και η ανίχνευση ανωμαλιών που βασίζεται σε μετρικές μπορεί να παραβλέψει κρίσιμες πληροφορίες πλαισίου που παρέχονται στα ίχνη. Αυτή η προσέγγιση απομόνωσης οδηγεί συχνά σε τυφλά σημεία στα συστήματα παρακολούθησης, καθιστώντας δύσκολη την απόκτηση μιας ολοκληρωμένης εικόνας της υγείας και της απόδοσης του συστήματος σε πραγματικό χρόνο.

Για την αντιμετώπιση αυτών των περιορισμών, η πολυτροπική ανίχνευση ανωμαλιών (multi-modal anomaly detection) έχει αναδειχθεί ως μία πολλά υποσχόμενη προσέγγιση. Με την ενσωμάτωση πολλαπλών τύπων δεδομένων (αρχεία καταγραφής, ίχνη, μετρικές) σε ένα ενοποιημένο πλαίσιο, η τεχνική αυτή παρέχει μια πιο ολοκληρωμένη κατανόηση της συμπεριφοράς ενός συστήματος. Κάθε τύπος δεδομένων προσφέρει τα δικά του πλεονεκτήματα στη διαδικασία εντοπισμού ανωμαλιών: τα αρχεία καταγραφής παρέχουν λεπτομερείς ακολουθίες συμβάντων, τα ίχνη αποκαλύπτουν τη ροή των αιτημάτων όπως αυτά διαδίδονται εντός ενός κατανεμημένου συστήματος και οι μετρικές καταγράφουν τις τάσεις της απόδοσης με την πάροδο του χρόνου. Όταν τα δεδομένα αυτά συνδυάζονται, επιτρέπουν έναν πλουσιότερο και ακριβέστερο εντοπισμό ανωμαλιών ακόμα και σε περιπτώσεις που αυτές δεν είναι εμφανείς στα μεμονωμένα δεδομένα. Η ενοποίηση αυτή μπορεί να οδηγήσει στον εντοπισμό κρυφών μοτίβων, συσχετίσεων και αλληλεξαρτήσεων που διαφορετικά θα περνούσαν απαρατήρητες, επιτρέποντας την έγκαιρη και ακριβέστερη ανίχνευση προβλημάτων, πριν αυτά κλιμακωθούν και γενικευθούν σε μεγαλύτερα ζητήματα.

Οι επιπτώσεις της αποτελεσματικής ανίχνευσης ανωμαλιών στον πραγματικό κόσμο είναι ιδιαίτερα σημαντικές. Στις υποδομές που βασίζονται σε τεχνολογίες νέφους, για παράδειγμα, οι μη εντοπισμένες ανωμαλίες μπορεί να οδηγήσουν σε απρογραμμάτιστες διακοπές λειτουργίας, σημαντική υποβάθμιση των επιδόσεων και πιθανά τρωτά σημεία στο σύστημα ασφαλείας. Κάθε μία από τις επιδράσεις αυτές μπορεί να μεταφραστεί σε οικονομική απώλεια και ζημιά στη φήμη. Κλάδοι όπως ο χρηματοπιστωτικός τομέας, ο τομέας υγείας και οι τηλεπικοινωνίες, όπου η υψηλή διαθεσιμότητα και αξιοπιστία είναι ζωτικής σημασίας, βασίζονται στην έγκαιρη ανίχνευση ανωμαλιών για την αποφυγή τέτοιων ανεπιθύμητων καταστάσεων. Η ικανότητα ταχείας ανίχνευσης και αντιμετώπισης των ανωμαλιών είναι το κλειδί για τη διατήρηση της συνέχειας των υπηρεσιών και τη βελτίωση της συνολικής ανθεκτικότητας του συστήματος.

Παρά τις δυνατότητες που προσφέρει, το πεδίο της πολυτροπικής ανίχνευσης ανωμαλιών δεν είναι απαλλαγμένο από προκλήσεις. Τα σύγχρονα εργαλεία συχνά δοκιμάζονται από την ανάγκη κλιμακωσιμότητας όταν εφαρμόζονται σε μεγάλης κλίμακας κατανεμημένα συστήματα με πολύπλοκες αρχιτεκτονικές. Επιπλέον, πολλά συστήματα στερούνται ερμηνευσιμότητας, καθιστώντας δύσκολο για τους χειριστές των συστημάτων να εντοπίσουν τις πηγές των ανωμαλιών που εντοπίζονται ή ακόμη και να εμπιστευτούν τα αυτοματοποιημένα αποτελέσματα χωρίς χειροκίνητη παρέμβαση. Επιπλέον, ο δυναμικός και εξελισσόμενος χαρακτήρας των σύγχρονων συστημάτων προσθέτει ένα επιπλέον επίπεδο πολυπλοκότητας, καθώς τα μοντέλα ανίχνευσης ανωμαλιών πρέπει να προσαρμόζονται στις αλλαγές των προτύπων κανονικής συμπεριφοράς, καθώς το σύστημα γύρω τους αλλάζει.

Η παρούσα εργασία στοχεύει να αντιμετωπίσει τα κενά αυτά, διερευνώντας νέες, πιο αποτελεσματικές προσεγγίσεις για τον πολυτροπικό εντοπισμό ανωμαλιών. Εστιάζοντας σε τεχνικές που ενσωματώνουν πολλαπλές πηγές δεδομένων διατηρώντας τη δυνατότητα κλιμακωσιμότητας, η έρευνα στοχεύει στην ανάπτυξη αξιόπιστων λύσεων που μπορούν να λειτουργήσουν σε περιβάλλοντα πραγματικού κόσμου με υψηλή ζήτηση. Η εργασία που παρουσιάζεται εδώ έχει στόχο την πρόοδο του τομέα της ανίχνευσης ανωμαλιών, παρέχοντας τόσο θεωρητικές γνώσεις όσο και πρακτικά εργαλεία για την καλύτερη παρακολούθηση και συντήρηση κατανεμημένων συστημάτων.

1.2 Συνεισφορές

Στην ενότητα αυτή παρουσιάζονται συνοπτικά οι κύριες συνεισφορές της εργασίας αυτής:

- Επισκόπηση και παρουσίαση βιβλιογραφίας πάνω στο πρόβλημα της ανίχνευσης ανωμαλιών χρησιμοποιώντας δεδομένα από αρχεία καταγραφής, ίχνη εκτέλεσης και μετρικές, είτε μεμονωμένα, είτε συνδυαστικά, δίνοντας έμφαση σε μεθόδους που βασίζονται σε τεχνικές Μηχανικής Μάθησης.
- Κατασκευή διαδικτυακής εφαρμογής που προσομοιώνει την αρχιτεκτονική και τη λειτουργία μιας κατανεμημένης εφαρμογής για την παραγωγή δεδομένων παρατηρησιμότητας.
- Κατασκευή ολοκληρωμένου συστήματος παρατηρησιμότητας για τη συλλογή, την αποθήκευση και την επεξεργασία των δεδομένων που παράγονται κατά την εκτέλεση της εφαρμογής

- Υλοποίηση επιμέρους μοντέλων εντοπισμού ανωμαλιών για τους διαφόρους τύπους δεδομένων.
- Συνδυασμός των μοντέλων σε ένα ενιαίο σύστημα για τον εντοπισμό ανωμαλιών σε πολυτροπικά δεδομένα.
- Πειραματική αξιολόγηση των προτεινόμενων μεθόδων πάνω σε δεδομένα που συλλέγονται από την προσομοιωμένη κατανεμημένη εφαρμογή.

1.3 Ορολογία

1.3.1 Αρχιτεκτονική Μικροϋπηρεσιών

Η αρχιτεκτονική μικροϋπηρεσιών (Microservice Architecture) [1] αποτελεί μία προσέγγιση σχεδιασμού λογισμικού που δομεί μια εφαρμογή ως ένα σύνολο από χαλαρά συνδεδεμένες και ανεξάρτητα αναπτύξιμες υπηρεσίες (services). Κάθε υπηρεσία εκτελεί μία συγκεκριμένη δυνατότητα, μπορεί να αναπτυχθεί και να κλιμακωθεί ανεξάρτητα και επικοινωνεί με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως το HTTP, ή συστήματα ανταλλαγής μηνυμάτων. Σε αντίθεση με τις παραδοσιακές μονολιθικές αρχιτεκτονικές, όπου όλη η λειτουργικότητα βρίσκεται συγκεντρωμένη μέσα σε ένα ενιαίο, στενά συνδεδεμένο σύστημα, η χρήση μικροϋπηρεσιών επιτρέπει τη δημιουργία πιο ευέλικτων και επεκτάσιμων εφαρμογών, καθώς κάθε υπηρεσία μπορεί να τροποποιηθεί, ενημερωθεί και να αναπτυχθεί ανεξάρτητα από την εφαρμογή της οποίας αποτελεί μέρος. Η αρχιτεκτονική αυτή είναι ιδανική για περιβάλλοντα cloud στα οποία η κλιμακωσιμότητα, ευελιξία και ανοχή σε σφάλματα είναι κρίσιμες.

1.3.2 Κατανεμημένη Αρχιτεκτονική

Η αυξανόμενη χρήση της αρχιτεκτονικής μικροϋπηρεσιών έχει οδηγήσει στην εμφάνιση ολοένα και περισσότερων κατανεμημένων εφαρμογών, στις οποίες κάθε υπηρεσία λειτουργεί ως ανεξάρτητη οντότητα και μπορεί να εκτελείται σε διαφορετικά συστήματα. Στις μονολιθικές εφαρμογές, όλες οι λειτουργίες εκτελούνται σε μία ενιαία πλατφόρμα, η ανάγκη αυτή όμως δεν υπάρχει για τις εφαρμογές βασισμένες σε μικροϋπηρεσίες, όπου κάθε διαφορετική υπηρεσία μπορεί να εκτελείται σε ξεχωριστό περιβάλλον, συχνά σε υποδομές νέφους (cloud), με τις υπηρεσίες αλλά και τα συστήματα που τις φιλοξενούν να επικοινωνούν μέσω δικτύου. Η προσέγγιση αυτή μπορεί να διασφαλίσει ότι οι εφαρμογές είναι πιο ευέλικτες και επεκτάσιμες, καθώς νέες δυνατότητες μπορούν να προστεθούν ή οι υπάρχουσες να κλιμακωθούν ανάλογα με τις ισχύουσες απαιτήσεις. Ωστόσο, αυτός ο κατανεμημένος χαρακτήρας προσθέτει πολυπλοκότητα στη διαχείριση, παρακολούθηση και ασφάλεια των αλληλεπιδράσεων μεταξύ των υπηρεσιών, κάνοντας την ανθεκτικότητα και τη διαλειτουργικότητα σημαντικά ζητήματα που πρέπει να αντιμετωπιστούν σε τέτοιες αρχιτεκτονικές.

1.3.3 Ανίχνευση Ανωμαλιών

Ο όρος ανίχνευση ανωμαλιών (anomaly detection) [2] ή ανίχνευση ακραίων τιμών (outlier detection) αναφέρεται στη διαδικασία σπάνιων αντικειμένων, γεγονότων ή παρατηρήσεων που αποκλίνουν σημαντικά από την πλειοψηφία των δεδομένων. Αυτές οι ανωμαλίες συχνά καταδεικνύουν κάποιο ασυνήθιστο γεγονός, κάποιο σφάλμα, ή γενικότερα

σπάνια αλλά σημαντικά φαινόμενα. Η ικανότητα ανίχνευσης ανωμαλιών είναι ζωτικής σημασίας σε διάφορους τομείς, όπως η ανίχνευση απάτης, η ασφάλεια στον κυβερνοχώρο, η ιατρικές διαγνώσεις, η παρακολούθηση δικτύων και η απόδοση βιομηχανικών συστημάτων. Οι ανωμαλίες χωρίζονται σε τρεις γενικές κατηγορίες:

- **Ανωμαλίες σημείου (point anomalies)**: ένα μοναδικό σημείο που αποκλίνει σημαντικά από τα υπόλοιπα.
- **Ανωμαλίες με βάση το πλαίσιο (context anomalies)**: Σημεία δεδομένων που θεωρούνται ανώμαλα μόνο σε ένα συγκεκριμένο πλαίσιο. Τα δεδομένα χρονοσειρών συχνά περιλαμβάνουν ανίχνευση ανωμαλιών σε συνάρτηση με το πλαίσιο.
- **Συλλογικές ανωμαλίες (collective anomalies)**: Ένα σύνολο σημείων δεδομένων που, συνολικά, είναι ανώμαλα, ακόμη και αν τα μεμονωμένα σημεία δεν είναι.

Κατά την ανίχνευση ανωμαλιών εμφανίζονται συχνά προκλήσεις. Συχνότερες αιτίες των προκλήσεων αυτών είναι οι παρακάτω:

- **Έλλειψη επισημασμένων δεδομένων**: Η απόκτηση μεγάλων, επισημασμένων συνόλων δεδομένων για την εκπαίδευση επιβλεπόμενων μοντέλων είναι συχνά δύσκολη. Αυτή η έλλειψη μπορεί να οδηγήσει σε εξάρτηση από τεχνικές μη επιβλεπόμενης μάθησης.
- **Ποικιλία ανωμαλιών**: Οι ανωμαλίες μπορούν να εκδηλωθούν σε διάφορες μορφές και πλαίσια και τα χαρακτηριστικά τους μπορεί να ποικίλλουν σε διάφορους τομείς. Αυτή η ετερογένεια καθιστά δύσκολη την ανάπτυξη γενικευμένων μοντέλων.
- **Ανισορροπία στα δεδομένα**: Συχνά τα σύνολα δεδομένων περιέχουν μια μεγάλη πλειοψηφία φυσιολογικών δεδομένων και ένα πολύ μικρό ποσοστό ανωμαλιών. Αυτή η ανισορροπία κλάσεων περιπλέκει τη διαδικασία μάθησης και την αξιολόγηση των μοντέλων.

Σε ένα κατανεμημένο περιβάλλον η ανίχνευση ανωμαλιών αποκτά ιδιαίτερη σημασία, αφού η αυτονομία των υπηρεσιών αυξάνει τον αριθμό των πιθανών σημείων αποτυχίας. Ακόμα και μικρές αποκλίσεις, όπως καθυστερήσεις ή σφάλματα επικοινωνίας μπορεί να κλιμακωθούν και να επηρεάσουν δυσανάλογα ολόκληρη την εφαρμογή. Η έγκαιρη ανίχνευση αυτών των ανωμαλιών επιτρέπει στους developers να εντοπίζουν και να επιλύουν προβλήματα πριν εξελιχθούν σε σοβαρές δυσλειτουργίες στο επίπεδο ολόκληρης της εφαρμογής. Επιπλέον, με τις εφαρμογές αυτές να εκτελούνται σε ένα δυναμικό κατανεμημένο περιβάλλον όπως αυτό του cloud, απαραίτητη κρίνεται η συνεχής και αυτοματοποιημένη παρακολούθηση για την εξασφάλιση συνεπών επιδόσεων και ομαλής λειτουργίας της εφαρμογής, συνεισφέροντας στη βέλτιστη εμπειρία για τον τελικό χρήστη.

1.3.4 Μηχανική Μάθηση

Στα κατανεμημένα συστήματα, η πολυπλοκότητα των πολλαπλών αλληλεπιδρώντων στοιχείων καθιστά δύσκολη την ανίχνευση μη φυσιολογικής συμπεριφοράς. Οι ανωμαλίες μπορεί να εκδηλωθούν ως αποτυχίες τμημάτων του συστήματος, υποβαθμισμένη απόδοση, λανθασμένες ρυθμίσεις ή θέματα ασφάλειας. Για την ανίχνευσή τους χρησιμοποιούνται διάφορες προσεγγίσεις, όπως συστήματα βασισμένα σε κανόνες, στατιστικές μέθοδοι και πιο πρόσφατα τεχνικές μηχανικής μάθησης.

Η Μηχανική Μάθηση (Machine Learning) [3] αποτελεί έναν κλάδο της Τεχνητής Νοημοσύνης (Artificial Intelligence) [4], ο οποίος εστιάζει στην ανάπτυξη αλγορίθμων και συστημάτων που έχουν την ικανότητα να μαθαίνουν και να βελτιώνονται αυτόνομα, χωρίς να απαιτείται ανθρώπινη παρέμβαση ή ρητός προγραμματισμός. Μέσω της ανάλυσης μεγάλων ποσοτήτων δεδομένων, τα μοντέλα μηχανικής μάθησης μπορούν να αναγνωρίζουν πρότυπα, να προβλέπουν αποτελέσματα και να λαμβάνουν αποφάσεις σε πραγματικό χρόνο. Η εφαρμογή της Μηχανικής Μάθησης καλύπτει ένα ευρύ φάσμα πεδίων, όπως η ανάλυση δεδομένων, η αναγνώριση εικόνας, η φυσική γλώσσα και η πρόβλεψη συμπεριφορών. Υπάρχουν τρεις βασικοί τύποι Μηχανικής Μάθησης: Επιβλεπόμενη, Μη Επιβλεπόμενη και Ημι-Επιβλεπόμενη Μάθηση.

- **Επιβλεπόμενη Μάθηση (Supervised Learning)**: Σε αυτόν τον τύπο, τα συστήματα εκπαιδεύονται με δεδομένα που είναι επισημασμένα (labeled), δηλαδή κάθε δείγμα δεδομένων έχει μια γνωστή έξοδο (ετικέτα). Τόσο οι φυσιολογικές, όσο και οι ανώμαλες συμπεριφορές είναι γνωστές εκ των προτέρων. Ο αλγόριθμος μαθαίνει να συσχετίζει τις εισόδους με τις σωστές εξόδους και μπορεί να χρησιμοποιηθεί για την πρόβλεψη νέων, άγνωστων δεδομένων. Χρησιμοποιείται ευρέως σε εφαρμογές όπως η ταξινόμηση (classification) και η παλινδρόμηση (regression). Αυτή η προσέγγιση λειτουργεί καλά όταν τα ιστορικά δεδομένα είναι άφθονα και σωστά επισημασμένα, αλλά μπορεί να είναι δύσκολο να εφαρμοστεί σε κατανεμημένα περιβάλλοντα όπου οι επισημασμένες ανωμαλίες είναι σπάνιες.
- **Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)**: Εδώ, τα δεδομένα που χρησιμοποιούνται δεν είναι επισημασμένα και ο αλγόριθμος πρέπει να βρει κρυφά μοτίβα ή δομές στα δεδομένα χωρίς να έχει συγκεκριμένες ετικέτες. Μοντέλα που ανήκουν σε αυτή την κατηγορία εντοπίζουν ανωμαλίες ως ακραίες τιμές ή αποκλίσεις από τη φυσιολογική συμπεριφορά του συστήματος. Χρησιμοποιείται κυρίως για την ομαδοποίηση (clustering) και τη μείωση διαστάσεων (dimensionality reduction), επιτρέποντας την ανακάλυψη κρυφών σχέσεων μεταξύ των δεδομένων. Η προσέγγιση αυτή είναι χρήσιμη όταν τα επισημασμένα δεδομένα είναι σπάνια, όπως συμβαίνει συχνά σε πολύπλοκα, κατανεμημένα συστήματα.
- **Ημι-Επιβλεπόμενη Μάθηση (Semi-Supervised Learning)**: Αυτός ο τύπος βρίσκεται ανάμεσα στην επιβλεπόμενη και την μη επιβλεπόμενη μάθηση. Χρησιμοποιεί ένα μικρό σύνολο επισημασμένων δεδομένων και ένα μεγαλύτερο σύνολο μη επισημασμένων δεδομένων για την εκπαίδευση του μοντέλου. Με τα μοντέλα αυτού του είδους να εκπαιδεύονται κυρίως σε κανονικά δεδομένα, η προσέγγιση αυτή αναγνωρίζει κάθε απόκλιση από την «κανονική» συμπεριφορά που έχει μάθει ως ανώμαλη. Είναι ιδιαίτερα χρήσιμη όταν η επισήμανση δεδομένων είναι δαπανηρή ή χρονοβόρα, αλλά υπάρχουν πολλά διαθέσιμα μη επισημασμένα δεδομένα. Η μέθοδος αυτή είναι ιδιαίτερα εφαρμόσιμη σε κατανεμημένα συστήματα όπου η κανονική συμπεριφορά είναι γνωστή, αλλά οι ανωμαλίες είναι σπάνιες.

Η καθεμία από αυτές τις προσεγγίσεις εφαρμόζεται σε διαφορετικά προβλήματα και έχει διαφορετικές χρήσεις ανάλογα με το είδος και τη διαθεσιμότητα των δεδομένων.

1.3.5 Παρατηρησιμότητα

Η ανίχνευση των ανωμαλιών συνδέεται άμεσα με την έννοια της παρατηρησιμότητας (observability) [5]. Η έννοια αυτή αναφέρεται στην ικανότητα κατανόησης της εσωτερικής κατάστασης ενός κατανεμημένου συστήματος με βάση τις εξόδους του. Αποτελεί βασική αρχή για τον εντοπισμό και τη διάγνωση ανωμαλιών σε κατανεμημένες αρχιτεκτονικές. Οι βασικές πηγές δεδομένων που χρησιμοποιούνται για την επίτευξη παρατηρησιμότητας είναι τρεις: τα αρχεία καταγραφής (logs), οι μετρικές (metrics) και τα ίχνη (traces).

Αρχεία Καταγραφής (Logs)

Τα αρχεία καταγραφής αποτελούνται από εγγραφές με χρονική σήμανση που καταγράφουν αναλυτικές πληροφορίες για διακριτά γεγονότα κατά την εκτέλεση ενός προγράμματος. Συχνά παράγονται από εφαρμογές, υπηρεσίες ή στοιχεία υποδομής (όπως βάσεις δεδομένων, διακομιστές ιστού ή λειτουργικά συστήματα) και περιέχουν σημαντικές λεπτομέρειες, όπως μηνύματα σφάλματος, προειδοποιήσεις ή πληροφοριακά συμβάντα. Τα αρχεία καταγραφής δημιουργούνται για να παρέχουν αναλυτική χρονολογική αναφορά της συμπεριφοράς ενός συστήματος. Σε ένα κατανεμημένο σύστημα, τα αρχεία καταγραφής από διάφορες μικροϋπηρεσίες, βάσεις δεδομένων και διακομιστές μπορούν να συγκεντρωθούν για την κατανόηση του τι συνέβη σε συγκεκριμένες χρονικές στιγμές. Τα αρχεία καταγραφής είναι συνήθως ημιδομημένα ή αδόμητα, αποτελούμενα από ένα μείγμα κειμένου, ζευγών κλειδιών-τιμών και χρονοσφραγίδων. Μπορεί να περιέχουν πληροφορίες σχετικά με την κατάσταση της εφαρμογής τη στιγμή του συμβάντος, όπως το αναγνωριστικό χρήστη, τον τύπο αίτησης ή τη χρήση πόρων. Τα αρχεία καταγραφής μπορούν να αναλυθούν και να μελετηθούν για τον εντοπισμό απροσδόκητων μοτίβων, σφαλμάτων ή ασυνήθιστων μηνυμάτων που υποδεικνύουν ζητήματα όπως υποβάθμιση της απόδοσης, αποτυχίες του συστήματος ή παραβιάσεις της ασφάλειας.

Ανάλυση Αρχείων Καταγραφής

Τα αρχεία καταγραφής αποτελούν ένα από τα βασικά δομικά συστατικά της παρατηρησιμότητας, παρέχοντας αρχεία κειμένου για τα συμβάντα του συστήματος, τα σφάλματα και τις αλληλεπιδράσεις μεταξύ υπηρεσιών. Ωστόσο, η ανάλυση των αρχείων καταγραφής σε μεγάλα κατανεμημένα συστήματα αποτελεί πρόκληση λόγω του όγκου και της αδόμητης φύσης των δεδομένων καταγραφής. Για να εξαχθούν σημαντικές πληροφορίες, τα αρχεία καταγραφής πρέπει να αναλυθούν σε δομημένα μορφή. Η ανάλυση των αρχείων καταγραφής είναι η διαδικασία κατά την οποία αδόμητες εγγραφές μετατρέπονται δομημένα δεδομένα αναγνωρίζοντας συχνά μοτίβα, πρότυπα και ζεύγη τιμών – κλειδιών στις εγγραφές. Αυτά τα δομημένα δεδομένα μπορούν να χρησιμοποιηθούν για ανίχνευση ανωμαλιών, παρακολούθηση επιδόσεων και αποσφαλμάτωση σε κατανεμημένα συστήματα.

Ίχνη (Traces)

Τα ίχνη καταγράφουν τον κύκλο ζωής μεμονωμένων αιτημάτων, καθώς αυτά διαδίδονται μέσω των τμημάτων ενός κατανεμημένου συστήματος, όπως οι μικροϋπηρεσίες ή οι προγραμματιστικές διεπαφές. Ένα ίχνος παρακολουθεί ένα αίτημα από τη στιγμή που εισέρχεται στο σύστημα μέχρι την ολοκλήρωσή του, καταγράφοντας τις αλληλεπιδράσεις και καθυστερήσεις μεταξύ των διαφόρων υπηρεσιών στην πορεία της διαδρομής του. Τα ίχνη παρέχουν ορατότητα στο πως διάφορα τμήματα διαχειρίζονται μεμονωμένα αιτήματα,

προσφέροντας πληροφορίες σχετικά με τα σημεία συμφόρησης, τις καθυστερήσεις και τις εξαρτήσεις μεταξύ υπηρεσιών σε κατανεμημένες αρχιτεκτονικές. Ένα ίχνος αποτελείται τυπικά από πολλαπλά spans, με κάθε span να αντιστοιχεί σε ένα τμήμα του ίχνους όπως μία κλήση συνάρτησης ή αλληλεπίδραση μεταξύ μικροϋπηρεσιών. Κάθε span περιέχει πληροφορία σχετικά με το χρόνο έναρξης και λήξης του, τις διεργασίες που εμπλέκονται και το χρόνο απόκρισης. Τα ίχνη χρησιμοποιούνται συχνά για τον εντοπισμό ανωμαλιών σε ροές αιτημάτων. Τέτοιες ανωμαλίες μπορεί να εμφανίζονται ως αυξημένοι χρόνοι απόκρισης, διαφορετικές ακολουθίες από spans ή μη αναμενόμενες εξαρτήσεις μεταξύ υπηρεσιών. Η ανίχνευση ανωμαλιών με βάση τα ίχνη μπορεί να βοηθήσει στον εντοπισμό προβλημάτων απόδοσης, διακοπών υπηρεσιών ή λανθασμένων ρυθμίσεων.

Μετρικές (Metrics)

Οι μετρικές αποτελούνται από αριθμητικές μετρήσεις που αποτυπώνουν την απόδοση και την αξιοποίηση πόρων μιας εφαρμογής ή ενός συστήματος με το πέρασμα του χρόνου. Συνήθως συλλέγονται ανά τακτά χρονικά διαστήματα και αντικατοπτρίζουν βασικούς δείκτες απόδοσης (Key Performance Indicators, KPIs) όπως όπως η χρήση CPU, η κατανάλωση μνήμης και οι χρόνοι εξυπηρέτησης αιτήσεων. Οι μετρικές παρέχουν μια ποσοτική εικόνα της υγείας και της απόδοσης ενός συστήματος τόσο σε επίπεδο υποδομής όσο και σε επίπεδο εφαρμογής. Σε κατανεμημένα συστήματα οι μετρικές συλλέγονται από πολλαπλές πηγές (π.χ. μικροϋπηρεσίες, βάσεις δεδομένων, εξυπηρετητές) για την παρακολούθηση της χρήσης πόρων και της λειτουργικής απόδοσης. Οι μετρικές συνήθως αναπαρίστανται ως δεδομένα χρονοσειρών, με κάθε μετρική να έχει μία χρονική σήμανση, μία αριθμητική τιμή και προαιρετικά κάποια ετικέτα που υποδεικνύει την πηγή της μετρικής. Οι ανωμαλίες στις μετρικές εντοπίζονται ως αποκλίσεις στις τιμές τους. Για παράδειγμα, εάν η χρήση της CPU μιας υπηρεσίας αυξάνεται ξαφνικά ή εάν ο χρόνος απόκρισης μιας αίτησης υπερβαίνει ένα προκαθορισμένο όριο, αυτό μπορεί να υποδεικνύει μια ανωμαλία απόδοσης. Η ανίχνευση ανωμαλιών με βάση τις μετρικές χρησιμοποιεί συνήθως στατιστικές μεθόδους ή μοντέλα μηχανικής μάθησης για τη διάκριση μεταξύ των κανονικών διακυμάνσεων και των πραγματικών ανωμαλιών.

1.3.6 Πολυτροπικότητα στην ανίχνευση ανωμαλιών

Η έννοια της πολυτροπικότητας (Multi-Modality) αναφέρεται στη χρήση πολλαπλών τύπων δεδομένων για τη βελτίωση της αποτελεσματικότητας ενός συστήματος, στην προκειμένη περίπτωση ενός συστήματος ανίχνευσης ανωμαλιών. Σε κατανεμημένες εφαρμογές, όπου η συνεχής παραγωγή διαφόρων μορφών πληροφοριών είναι συνηθισμένη (π.χ. αρχεία καταγραφής, μετρήσεις, ίχνη, δεδομένα αισθητήρων), η στήριξη σε έναν μόνο τύπο δεδομένων για την ανίχνευση ανωμαλιών μπορεί να μην αποτυπώνει την πλήρη εικόνα. Αντ' αυτού, μια πολυτροπική προσέγγιση αξιοποιεί πολλαπλές μορφές δεδομένων για να παρέχει έναν πιο ολοκληρωμένο και ακριβή μηχανισμό ανίχνευσης ανωμαλιών. Συνδυάζοντας τα δεδομένα αυτά, το σύστημα έχει την ικανότητα να εντοπίσει πιο πολύπλοκες ανωμαλίες που περνούν απαρατήρητες όταν το σύστημα εστιάζει σε μία μόνο πηγή δεδομένων. Για παράδειγμα, ένα μεμονωμένο σφάλμα στα αρχεία καταγραφής μπορεί να φαίνεται ακίνδυνο, όμως σε συνδυασμό με απότομη αύξηση στη χρήση της CPU και ανώμαλα ίχνη κατά την εξυπηρέτηση αιτήσεων, μπορεί να υποδεικνύει ένα σοβαρό πρόβλημα.

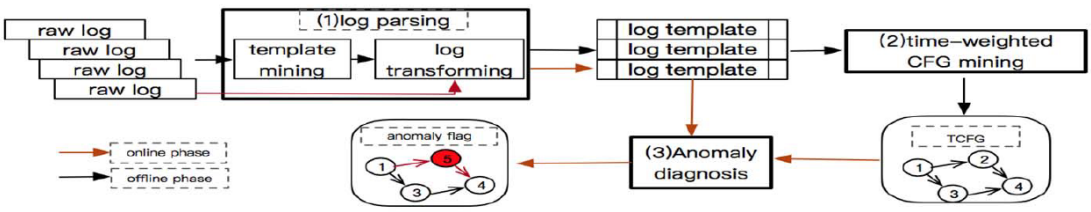
Σε ένα κατανεμημένο σύστημα, η ανίχνευση ανωμαλιών με χρήση μηχανικής μάθησης γίνεται πιο αποτελεσματική όταν αξιοποιούνται πολλαπλές μορφές δεδομένων (π.χ. αρχεία καταγραφής, ίχνη, μετρικές). Για παράδειγμα, ένα μοντέλο μπορεί να επεξεργάζεται ξεχωριστά κάθε τύπο δεδομένων και τα αποτελέσματά τους να συγχωνεύονται για να δημιουργηθεί ένα ακριβέστερο σύστημα ανίχνευσης.

Κεφάλαιο 2 - Βιβλιογραφική επισκόπηση

2.1 Εντοπισμός Ανωμαλιών με βάση αρχεία καταγραφής

Κάποιες εργασίες στην υπάρχουσα βιβλιογραφία χρησιμοποιούν αρχεία καταγραφής (logs) για τον εντοπισμό ανωμαλιών σε κατανεμημένες εφαρμογές. Εργασίες όπως οι [6], [7], [8] χρησιμοποιούν τα αρχεία καταγραφής για την κατασκευή Γράφων Ελέγχου Ροής (Control Flow Graphs, CFG) της εφαρμογής. Οι γράφοι αυτοί λειτουργούν ως αναπαράσταση της εφαρμογής, με τους κόμβους να αντιστοιχούν σε διαφορετικές κατηγορίες εγγραφών των αρχείων καταγραφής και τις ακμές να αντιστοιχούν σε μεταβάσεις μεταξύ των κατηγοριών. Οι κατηγορίες των εγγραφών προκύπτουν από την κατάλληλη ανάλυσή τους. Και στις τρεις εργασίες εφαρμόζεται ομαδοποίηση στις εγγραφές χωρίζοντάς τες σε ομάδες – κατηγορίες. Από τις κατηγορίες αυτές προκύπτουν οι κόμβοι του CFG. Για τη συνένωση των κόμβων που δημιουργούνται και οι τρεις εργασίες χρησιμοποιούν χρονικά κριτήρια. Διαδοχικές εγγραφές του αρχείου καταγραφής είναι πολύ πιθανό να προκύπτουν ως μέρος της ίδιας διεργασίας. Σε περιβάλλον κατανεμημένων εφαρμογών όμως, πολύ συχνές είναι οι παράλληλες διεργασίες. Σε τέτοια περίπτωση μπορεί να παρουσιάζονται διαδοχικές εγγραφές που προκύπτουν από διαφορετικές παράλληλες διεργασίες και άρα δεν πρέπει να συνδεθούν με ακμή. Για το λόγο αυτό στις εργασίες που αναφέρθηκαν εξετάζεται ένα σύνολο από τις πιο κοντινές χρονικά κατηγορίες και επιλέγεται η πιθανότερη με βάση τη συχνότητα εμφάνισης. Στις εργασίες [7] και [8] στις ακμές προστίθενται βάρη που αντιστοιχούν στον εκτιμώμενο χρόνο απόκρισης ανάμεσα στις συνδεόμενες κατηγορίες. Στην εργασία [7], εκτός από τον CFG δημιουργείται και ένας κατευθυνόμενος ακυκλικός γράφος για την αναπαράσταση της τοπολογίας της εφαρμογής. Ο γράφος αυτός προκύπτει με χρήση του αλγορίθμου Peter-Clark (PC), ο οποίος βασίζεται στον υπολογισμό της αιτιατότητας (causality) ανάμεσα στις διάφορες κατηγορίες εγγραφών για τον εντοπισμό εξαρτημένων στοιχείων.

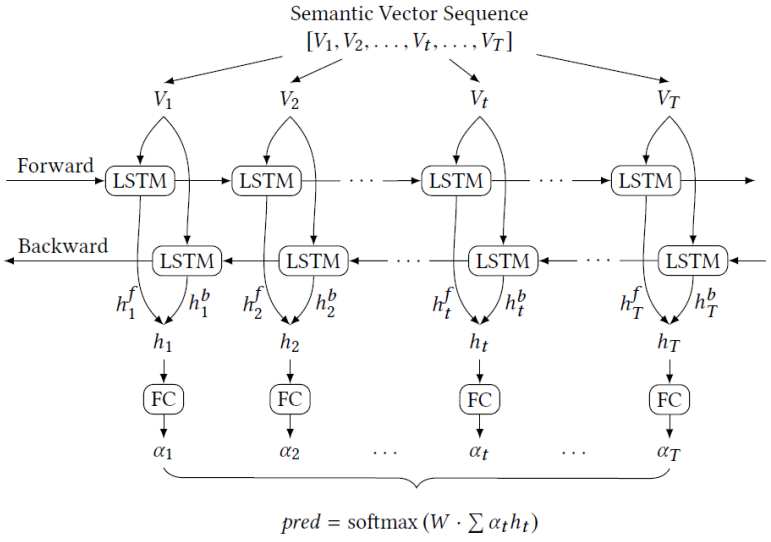
Οι γράφοι που σχηματίζονται χρησιμοποιούνται στην διαδικασία του εντοπισμού ανωμαλιών. Ο εντοπισμός ανωμαλιών και στις τρεις εργασίες γίνεται σε πραγματικό χρόνο. Οι εισερχόμενες εγγραφές των αρχείων καταγραφής αντιστοιχίζονται με την κατάλληλη κατηγορία και στη συνέχεια διακρίνουμε τις ακόλουθες περιπτώσεις. Στην εργασία [6] εντοπίζονται anomalies όταν παρατηρείται υπέρβαση χρονικού ορίου ανάμεσα σε διαδοχικές εγγραφές με βάση κάποιο προκαθορισμένο όριο ή όταν παρατηρείται αλλαγή στην κατανομή πιθανότητας στις μεταβάσεις ανάμεσα στους κόμβους του γράφου. Στις εργασίες [7], [8] ανωμαλίες εντοπίζονται όταν παρατηρείται υπέρβαση χρονικού ορίου ανάμεσα σε διαδοχικές εγγραφές με βάση το βάρος της αντίστοιχης ακμής του γράφου ή όταν εμφανίζεται μη γνωστή κατηγορία εγγραφής. Η διαδικασία επεξεργασίας των δεδομένων για το χτίσιμο των γράφων και τον εντοπισμό των ανωμαλιών που ακολουθείται στις τρεις εργασίες αναπαρίσταται και στο Σχήμα 2.1.



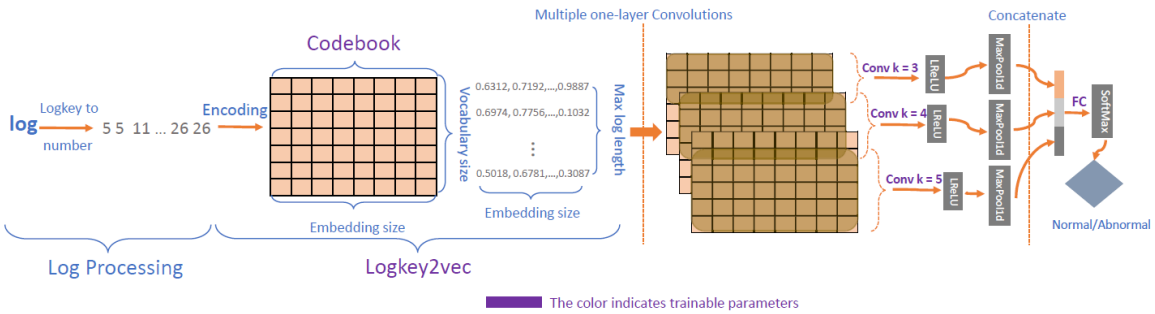
Σχήμα 2.1: Εντοπισμός Ανωμαλιών σε Logs με χρήση CFG

Και οι τρεις εργασίες χρησιμοποιούν τις μετρικές precision και recall για την αξιολόγηση των αποτελεσμάτων τους. Και οι τρεις πετυχαίνουν πολύ υψηλή αξιολόγηση. Στην [6] εξετάζονται διάφορα μοντέλα εκτέλεσης της εφαρμογής με διάφορους βαθμούς παραλληλίας. Στις περιπτώσεις που αντιστοιχούν σε real-world χρήση τα anomalies εντοπίζονται με πολύ υψηλό precision (περίπου 100%) και recall κοντά στο 90%. Στις εργασίες [7], [8] χρησιμοποιούνται κοινά δεδομένα και παρατηρούνται αντίστοιχα αποτελέσματα. Και στις δύο περιπτώσεις επιτυγχάνεται precision της τάξης του 90% και recall κοντά στο 80%.

Άλλες εργασίες χρησιμοποιούν τα αρχεία καταγραφής ως είσοδο για την εκπαίδευση μοντέλων μηχανικής μάθησης. Κάποιες από αυτές, όπως οι [9], [10], [11], [12] αξιοποιούν μη επιβλεπόμενα μοντέλα. Συγκεκριμένα, οι εργασίες [9], [10], [11] χρησιμοποιούν μοντέλα Μακράς Βραχύχρονης Μνήμης (Long Short Term Memory, LSTM) [13] (Σχήμα 2.2), ενώ η [12] χρησιμοποιεί Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Network, CNN) [14] (Σχήμα 2.3).

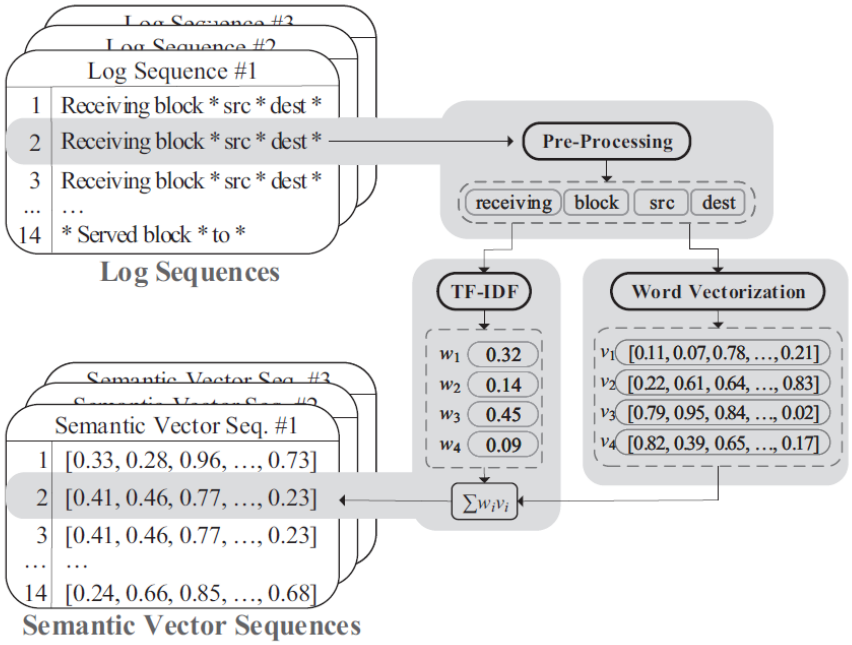


Σχήμα 2.2: Αρχιτεκτονική Μοντέλου Βασισμένο σε Δίκτυα LSTM



Σχήμα 2.3: Αρχιτεκτονική Μοντέλου Βασισμένο σε CNN

Οι εργασίες αυτές ακολουθούν παρόμοια δομή. Αρχικά αναλύουν κατάλληλα τις εγγραφές των αρχείων καταγραφής ώστε να καταλήξουν στις κατηγορίες, όπως αναφέρθηκε και προηγουμένως. Οι εργασίες [10], [11], [12] συμπεριλαμβάνουν ένα επιπλέον βήμα, αυτό της κωδικοποίησης των κατηγοριών που προκύπτουν σε σημασιολογικά διανύσματα. Στα [11], [12] χρησιμοποιείται ο αλγόριθμος word2Vec [15] για τη διανυσματική αναπαράσταση των λέξεων-κλειδιών που περιέχονται στις εγγραφές, ενώ στο [10] προτιμάται μια παραλλαγή του αλγορίθμου, ο αλγόριθμος template2Vec, πιο ταιριαστή για την ανάλυση των δεδομένων αρχείων καταγραφής. Επιπλέον, στο [11] χρησιμοποιείται η μετρική term frequency and inverse document frequency (TF-IDF) [16] ως βάρος για τις διανυσματικές αναπαραστάσεις των εγγραφών. Η μετρική αυτή υπολογίζει το βάρος κάθε λέξης λαμβάνοντας υπόψιν τη συχνότητα εμφάνισής της. Η διαδικασία μετασχηματισμού των εγγραφών σε διανύσματα παρουσιάζεται στην εργασία [11] χρησιμοποιώντας το Σχήμα 2.4.

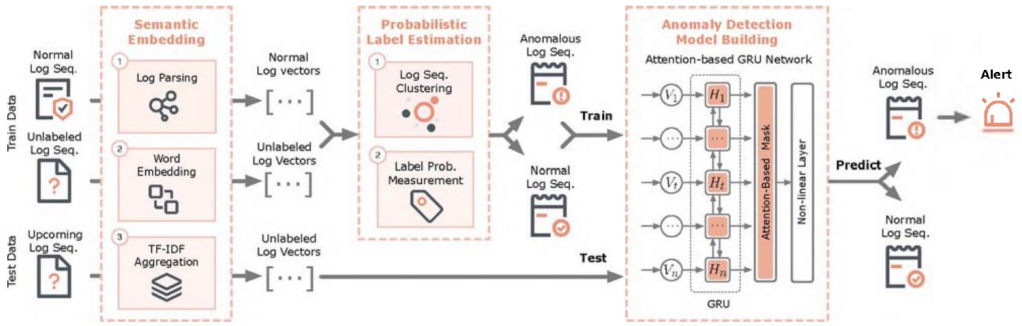


Σχήμα 2.4: Διαδικασία Μετασχηματισμού Εγγραφών σε Διανύσματα

Μετά την κατάλληλη προεπεξεργασία, τα δεδομένα χρησιμοποιούνται για την εκπαίδευση των μοντέλων. Οι εργασίες [9], [10] εκπαιδεύουν ένα απλό μοντέλο LSTM, το οποίο από ένα ιστορικό εγγραφών δεδομένου μήκους μαθαίνει και προβλέπει την πιθανότερη επόμενη εγγραφή. Η εργασία [11] χρησιμοποιεί ένα μοντέλο Bi-LSTM διπλής κατεύθυνσης, σε συνδυασμό με την τεχνική του attention. Το μοντέλο αυτό εντοπίζει περισσότερες συμφραζόμενες πληροφορίες εξετάζοντας και τις δύο κατευθύνσεις στην αλληλουχία των εγγραφών. Ο μηχανισμός του attention εφαρμόζεται για την κατάλληλη αντιστοίχιση βαρών στις εγγραφές ανάλογα τη σημασία τους στην ανάδειξη ανωμαλιών. Στην εργασία [12] χρησιμοποιείται ένα πιο απλό αλλά ευκολότερα εκπαιδεύσιμο δίκτυο CNN. Σε όλες τις εργασίες που αναφέρθηκαν, η έξοδος αποτελείται από μια λίστα με τις πιθανότερες επόμενες εγγραφές. Αν η εγγραφή που παρατηρείται δεν ανήκει σε αυτή τη λίστα δηλώνεται ως ανωμαλία.

Οι μετρικές που χρησιμοποιούνται για την αξιολόγηση των μοντέλων είναι οι precision, recall και f1 score. Και οι τέσσερεις εργασίες χρησιμοποιούν κοινά δεδομένα για την αξιολόγηση και συγκεκριμένα δεδομένα από το HDFS dataset [17], [18]. Τα καλύτερα αποτελέσματα φαίνεται να πετυχαίνει το [12] με precision 97.7%, recall 99.3% και f1 score 98.5%, ακολουθούμενο από τα [9], [10] με precision, recall και f1 score κοντά στο 95%. Για το [11] παρουσιάζονται αποτελέσματα αξιολόγησης για διάφορα ποσοστά anomalies στα δεδομένα. Για μικρά ποσοστά της τάξης του 5% παρουσιάζονται εξίσου καλά αποτελέσματα με τα παραπάνω, πετυχαίνοντας 100% precision, 91% recall και 95% f1 score. Καθώς αυξάνεται το ποσοστό των anomalies στα αρχικά δεδομένα, η επίδοση του μοντέλου μειώνεται σε μικρό βαθμό, παραμένοντας όμως σε υψηλά επίπεδα (f1 score 89% για anomalies σε ποσοστό 20%).

Εργασίες όπως η [19] συνδυάζουν τεχνικές μη επιβλεπόμενης και ημι-επιβλεπόμενης μηχανικής μάθησης στοχεύοντας στο συνδυασμό των θετικών των δύο μεθόδων. Πιο συγκεκριμένα, στο [19] χρησιμοποιείται η ημι-επιβλεπόμενη μέθοδος Πιθανοτικής Εκτίμησης Επισημάνσεων (Probabilistic Label Estimation, PLE) για την πιθανοτική εκτίμηση των ετικετών (normal ή anomaly) των εγγραφών μιας ακολουθίας αρχείων καταγραφής σε συνδυασμό με το επιβλεπόμενο Gated Recurrent Unit (GRU) [20] δίκτυο για τον εντοπισμό των ανωμαλιών (Σχήμα 2.5). Όπως και στις προηγούμενες περιπτώσεις οι εγγραφές αναλύονται κατάλληλα, ώστε να προκύψουν οι κατηγορίες στις οποίες ανήκουν. Οι κατηγορίες αυτές κωδικοποιούνται σε μορφή σημασιολογικών διανυσμάτων χρησιμοποιώντας ένα προ-εκπαιδευμένο μοντέλο word2Vec. Όπως και στο [11], έτσι κι εδώ τα διανύσματα εμπλουτίζονται με βάρη που προκύπτουν από τη χρήση της μετρικής TF-IDF. Αφού οι εγγραφές κωδικοποιηθούν κατάλληλα, ακολουθεί η πιθανοτική εκτίμηση των ετικετών. Η εκτίμηση γίνεται με βάση την τεχνική της θετικής μη Επισημασμένης Μάθησης (Positive Unlabeled (PU) learning), αφού η εκπαίδευση γίνεται μόνο σε θετικές (normal) εγγραφές. Οι κωδικοποιημένες εγγραφές χωρίζονται σε ομάδες με βάση τον ιεραρχικό αλγόριθμο HDBSCAN [21], ο οποίος παράλληλα αναθέτει σε κάθε εγγραφή και ένα βάρος που εκφράζει την σιγουριά της εκτίμησης. Με βάση το βάρος αυτό γίνεται και η πιθανοτική εκτίμηση της ετικέτας της εγγραφής. Τα δεδομένα που προκύπτουν δίνονται ως είσοδος σε ένα GRU δίκτυο, που αποτελεί παραλλαγή των τυπικών Επαναληπτικών Νευρωνικών Δικτύων (Recurrent Neural Networks, RNNs [22] ιδανικό για την εκμάθηση χρονικών σχέσεων σε ακολουθιακά δεδομένα. Το μοντέλο μπορεί να εντοπίσει ανωμαλίες σε επίπεδο ακολουθίας εγγραφών.

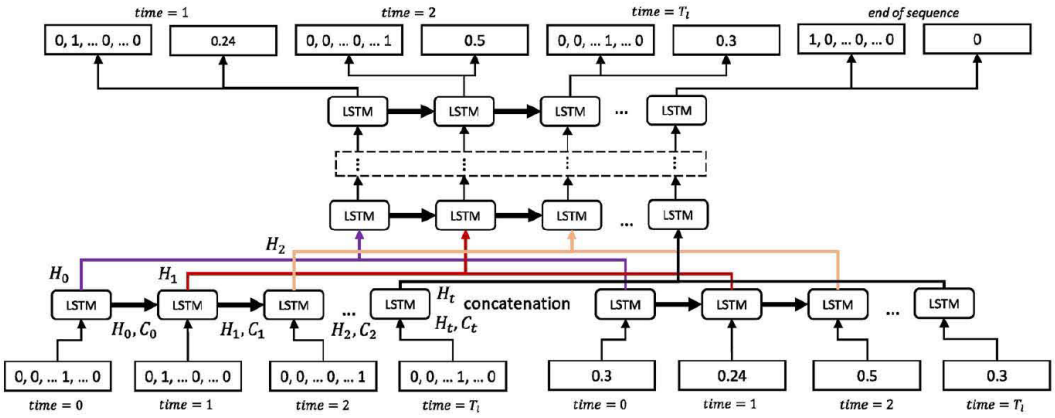


Σχήμα 2.5: Αρχιτεκτονική Συστήματος PLELog

Η αξιολόγηση του μοντέλου γίνεται σε κοινό σύνολο δεδομένων με τις προηγούμενες εργασίες (HDFS dataset) και χρησιμοποιούνται οι μετρικές precision, recall και f1 score. Το μοντέλο αυτό παρουσιάζει αντίστοιχη επίδοση με τις προηγούμενες εργασίες με 95% precision και 96% recall και f1 score.

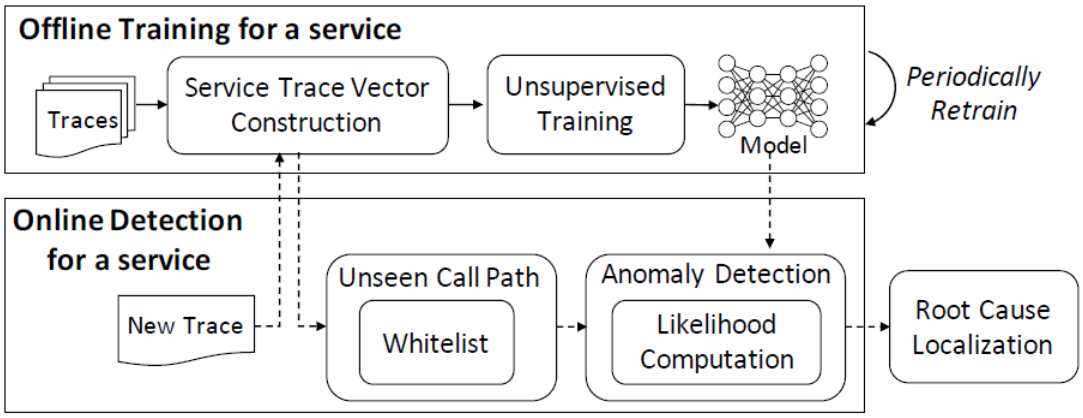
2.2 Εντοπισμός ανωμαλιών με βάση ίχνη εκτέλεσης

Άλλες εργασίες χρησιμοποιούν δεδομένα ιχνηλάτησης για την ανίχνευση ανωμαλιών στα πλαίσια εφαρμογών που βασίζονται σε μικροϋπηρεσίες. Κάποιες εξ αυτών όπως οι [23], [24] συνδυάζουν τα traces με μη επιβλεπόμενα μοντέλα μηχανικής μάθησης. Στην εργασία [23], όμοια με κάποιες από τις προσεγγίσεις που χρησιμοποιούσαν αρχεία καταγραφής, προτιμάται ένα πολυτροπικό μοντέλο LSTM (Σχήμα 2.6). Συγκεκριμένα, το μοντέλο εκπαιδεύεται πάνω σε διανυσματικές αναπαραστάσεις των ιχνών αλλά και σε χρονοσειρές αποτελούμενες από τους χρόνους απόκρισης που περιέχονται στα ίχνη. Για την προεπεξεργασία των δεδομένων αρχικά τα ίχνη κωδικοποιούνται ως διανύσματα με μια μέθοδο παρόμοια με το One Hot Encoding [25]. Σε κάθε ένα από τα μοναδικά spans που εμφανίζονται στο σύνολο των ιχνών ανατίθεται ένα αναγνωριστικό ID. Κάθε ίχνος επομένως μπορεί να αναπαρασταθεί ως ένα διάνυσμα σταθερού μήκους ίσο με τον αριθμό των μοναδικών spans με στοιχεία 0 και 1 ανάλογα με τα spans τα οποία περιέχει. Για τις τιμές του χρόνου απόκρισης εφαρμόζεται min-max scaling στο διάστημα [0,1]. Το μοντέλο που εκπαιδεύεται πάνω στις διανυσματικές αναπαραστάσεις των ιχνών δουλεύει αντίστοιχα με εργασίες που αναφέρθηκαν στον εντοπισμό ανωμαλιών με βάση αρχεία καταγραφής. Το LSTM μαθαίνει τις ακολουθιακές σχέσεις ανάμεσα σε διαδοχικά spans και εκφράζει, δοσμένου ενός χρονικού παραθύρου, την πιθανότητα κάθε ένα από τα γνωστά span να ακολουθεί τα δοσμένα. Αντίστοιχα λειτουργεί και το δεύτερο κομμάτι του μοντέλου, μόνο που αυτό εργάζεται με πραγματικούς αριθμούς ως δεδομένα εισόδου και ως έξοδο δίνει μια εκτίμηση για τον χρόνο απόκρισης. Το μοντέλο εντοπίζει δύο είδη ανωμαλιών, μία στην περίπτωση που το span που παρατηρείται δεν ανήκει στα πιθανότερα επόμενα σύμφωνα με την πρόβλεψη του μοντέλου και μία όταν ο χρόνος απόκρισης που καταγράφεται δεν βρίσκεται κοντά στον εκτιμώμενο.



Σχήμα 2.6: Αρχιτεκτονική Πολυτροπικού Μοντέλου LSTM

Η εργασία [24] ακολουθεί μια διαφορετική προσέγγιση, χρησιμοποιώντας Variational Autoencoder [26] για τον εντοπισμό ανωμαλιών (Σχήμα 2.7). Και πάλι, αρχικό βήμα είναι η κατάλληλη επεξεργασία και κωδικοποίηση των δεδομένων ιχνηλάτησης. Και σε αυτή την περίπτωση, τα δεδομένα κωδικοποιούνται σε μορφή διανύσματος. Συγκεκριμένα, δημιουργείται ένα διάνυσμα πολλαπλών διαστάσεων για κάθε ίχνος με κάθε διάσταση να αντιστοιχεί στα πιθανά «μονοπάτια» από spans που έχουν καταγραφεί στο ίχνος και την τιμή του διανύσματος στη διάσταση αυτή να αποτελείται από το χρόνο απόκρισης μεταξύ spans εντός του ίχνους. Το μοντέλο, με είσοδο τα διανύσματα που δημιουργούνται μαθαίνει την κατανομή των ιχνών. Το μοντέλο της εργασίας [24] προσφέρει επιπλέον τη δυνατότητα επανεκπαίδευσης ανά κάποιο χρονικό διάστημα, ώστε να εξασφαλίζει ότι η κατανομή των ιχνών που γνωρίζει συμβαδίζει με Κατά την ανίχνευση των ανωμαλιών σε πραγματικό χρόνο, αν ένα εισερχόμενο ίχνος δεν ακολουθεί την κατανομή που έχει μάθει το μοντέλο με μεγάλη πιθανότητα, δηλώνεται ως ανωμαλία.

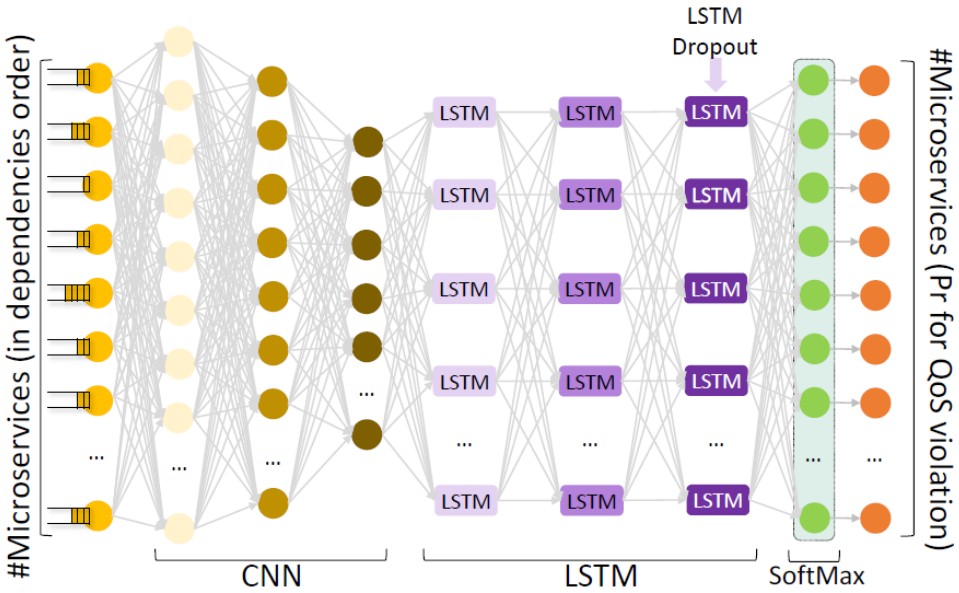


Σχήμα 2.7: Αρχιτεκτονική Συστήματος TraceAnomaly

Για την αξιολόγηση των μοντέλων, η εργασία [23] χρησιμοποιεί τις μετρικές precision και recall, ενώ η εργασία [24] μόνο τη μετρική accuracy. Και οι δύο εργασίες χρησιμοποιούν προσαρμοσμένα σύνολα δεδομένων, βασισμένα σε εφαρμογές με πολλαπλές μικροϋπηρεσίες. Οι δύο εργασίες φαίνεται να παρουσιάζουν εξίσου υψηλές μετρικές αξιολόγησης με την [23] να καταγράφει accuracy της τάξης του 97% στην καλύτερη περίπτωση και την [24] να αναφέρει 98% precision και 97% recall.

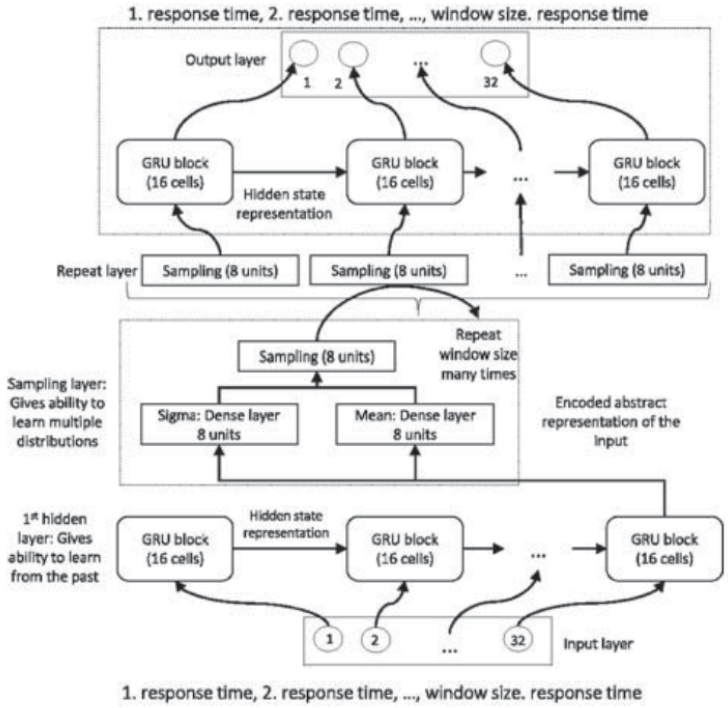
Άλλες εργασίες πάλι, όπως οι [27], [28], [29], [30], εφαρμόζουν τεχνικές επιβλεπόμενης μηχανικής μάθησης σε δεδομένα ιχνηλάτησης. Στις επιβλεπόμενες μεθόδους παρατηρείται μεγαλύτερη ποικιλία προσεγγίσεων. Σε αντίθεση με τις υπόλοιπες εργασίες που διαχειρίζονται δεδομένα από ίχνη, η [27] χρησιμοποιεί ως δεδομένα εισόδου μετρικές για το σύστημα που περιέχονται στα ίχνη, αντί για ολόκληρα τα ίχνη. Συγκεκριμένα, εξετάζονται διάφορες μετρικές και εν τέλει προτιμάται το μήκος των αιτημάτων σε αναμονή για κάθε μικροϋπηρεσία ως η μετρική με την μεγαλύτερη ακρίβεια στην ανάδειξη ανωμαλιών στη συμπεριφορά της εφαρμογής. Τα δεδομένα αυτά χρησιμοποιούνται για την εκπαίδευση ενός υβριδικού μοντέλου που συνδυάζει δίκτυα LSTM και CNN (Σχήμα 2.8). Το CNN χρησιμοποιείται για τη μείωση των διαστάσεων των δεδομένων και το φιλτράρισμα των μικροϋπηρεσιών που δεν επηρεάζουν την απόδοση της εφαρμογής, ενώ το LSTM, αντίστοιχα

με εργασίες που αναφέρονται παραπάνω, εξάγει την πιθανότητα για κάθε μικροϋπηρεσία η ανωμαλία του συστήματος να προέρχεται από αυτό. Το μοντέλο αυτό παρέχει και τη δυνατότητα επανεκπαίδευσης.



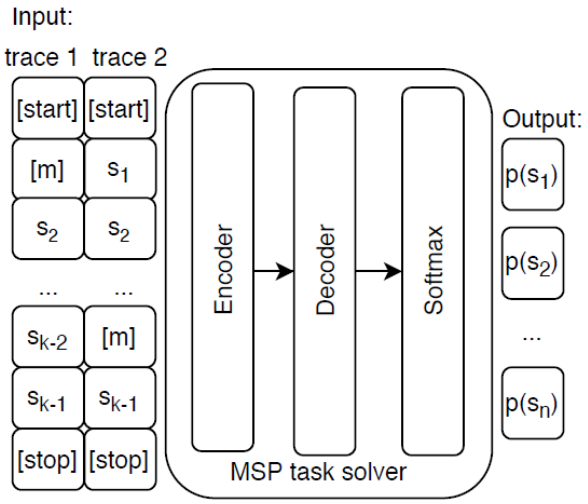
Σχήμα 2.8: Αρχιτεκτονική Υβριδικού Μοντέλου Seer

Στην εργασία [28] χρησιμοποιείται μία πιο σύνθετη ακολουθία αποτελούμενη από τρία στάδια, που περιλαμβάνουν έναν Variational Autoencoder (VAE), έναν post-processor και ένα CNN (Σχήμα 2.9). Η εκπαίδευση του μοντέλου γίνεται σε δεδομένα σε μορφή χρονοσειρών χωρίς ανωμαλίες, οπότε αρχικά εφαρμόζεται κατάλληλη επεξεργασία στα δεδομένα ώστε να αφαιρεθούν οι ακραίες τιμές, να κανονικοποιηθούν στο διάστημα [0,1] και να αφαιρεθεί τυχόν θόρυβος. Ο VAE εκπαιδεύεται πάνω στα δεδομένα αυτά και μαθαίνει να ανακατασκευάζει ίχνη χωρίς ανωμαλίες με μικρό σφάλμα. Όταν τα εισερχόμενα δεδομένα περιέχουν ανωμαλίες, το σφάλμα ανακατασκευής είναι μεγαλύτερο. Κατά την εκπαίδευση αποθηκεύεται η κατανομή των σφαλμάτων για την ανακατασκευή κανονικών ιχνών. Κατά τη φάση ανίχνευσης, ο post-processor αναλαμβάνει το φιλτράρισμα των ιχνών με μεγαλύτερο σφάλμα. Αν το σφάλμα ανακατασκευής δεν ακολουθεί την γνωστή κατανομή σφαλμάτων με μεγάλη πιθανότητα, το ίχνος δηλώνεται ως ανώμαλο και προωθείται στο δίκτυο CNN το οποίο καθορίζει το είδος ανωμαλίας που προέκυψε.



Σχήμα 2.9: Αρχιτεκτονική Νευρωνικού Δικτύου εργασίας

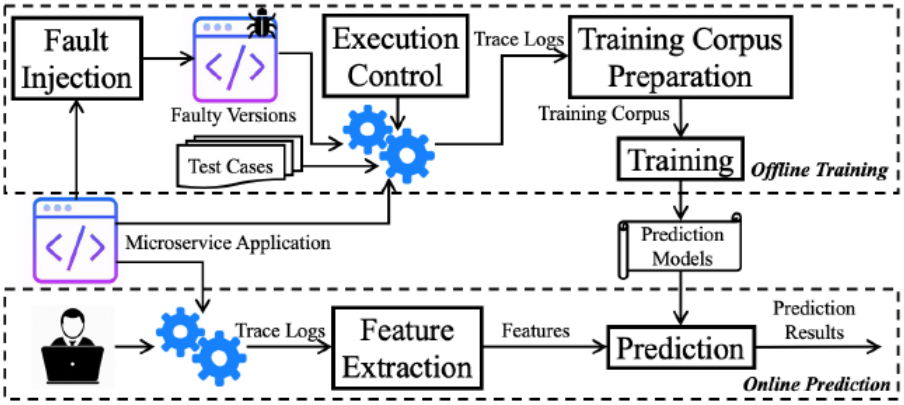
Η εργασία [29] χρησιμοποιώντας τεχνικές ανάλυσης αντίστοιχες με αυτές στα αρχεία καταγραφής χωρίζει τα spans των ιχνών σε διάφορες κατηγορίες. Στη συνέχεια, χρησιμοποιώντας έναν Autoencoder [31] σε συνδυασμό με ένα φίλτρο Softmax [32] κάνει προβλέψεις για το πιθανότερο πρότυπο span που μπορεί να αντιστοιχεί σε μία θέση ενός ίχνους. Κάθε κατηγορία span μπορεί να εμφανιστεί στο ίχνος μέσα σε ένα συγκεκριμένο πλαίσιο γειτονικών spans. Ο Autoencoder μαθαίνει το πλαίσιο αυτό, κρύβοντας ένα από τα spans σε ένα παράθυρο και προσπαθώντας να το μαντέψει από τα συμφραζόμενα. Η διαδικασία αυτή αποκαλείται Masked Span Prediction (MSP) και για την επίλυσή της χρησιμοποιείται το νευρωνικό δίκτυο που παρουσιάζεται στο Σχήμα 2.10.



Σχήμα 2.10: Νευρωνικό Δίκτυο για την επίλυση του MSP

Ως έξοδος του μοντέλου δίνεται μια λίστα από τα πιθανότερα spans που ταιριάζουν σε ένα συγκεκριμένο πλαίσιο. Αντίστοιχα με προηγούμενες προσεγγίσεις, αν το span που παρατηρείται στην πραγματικότητα δεν βρίσκεται μέσα στις πιο πιθανές προβλέψεις το ίχνος δηλώνεται ως ανώμαλο.

Και η εργασία [30] χρησιμοποιεί μετρικές που περιέχονται στα ίχνη σε συνδυασμό με τεχνικές ταξινόμησης (classification) για τον εντοπισμό πολλαπλών προκαθορισμένων τύπων ανωμαλιών (Σχήμα 2.11). Και σε αυτή την περίπτωση τα δεδομένα επεξεργάζονται κατάλληλα, ώστε να συμπληρωθούν κατάλληλα τα ελλιπή δεδομένα, να κωδικοποιηθούν τα μη αριθμητικά δεδομένα και να κανονικοποιηθεί το σύνολο των δεδομένων. Για τα μοντέλα ανίχνευσης των διαφόρων τύπων ανωμαλιών δοκιμάζονται αλγόριθμοι μηχανικής μάθησης όπως Random Forest (RF) [33], K-Nearest Neighbors (KNN) [34] και Multi-Layer Perceptron (MLP) [35]. Τα μοντέλα που παρουσιάζονται λειτουργούν είτε σε λογική δυαδικής ταξινόμησης έχοντας μόνο δύο πιθανές εξόδους (normal, anomaly) είτε σε λογική ταξινόμησης πολλαπλών τάξεων διαχωρίζοντας κατηγορίες ανωμαλιών.



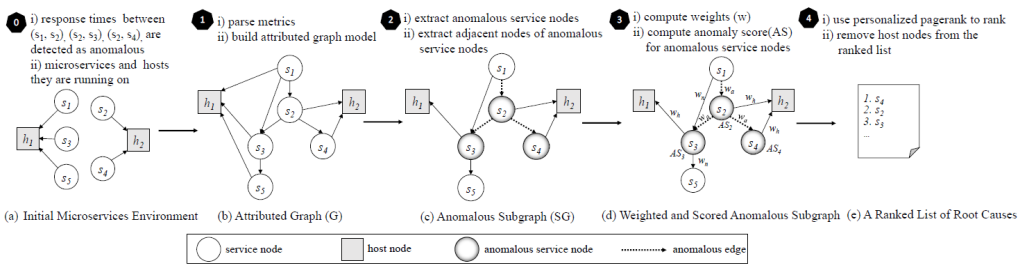
Σχήμα 2.11: Αρχιτεκτονική Συστήματος Εντοπισμού Ανωμαλιών σε Ίχνη με Ταξινόμηση

Για την αξιολόγηση των μοντέλων, όλες οι εργασίες χρησιμοποιούν δεδομένα από συγκεκριμένες περιπτώσεις εφαρμογών που είτε δημιούργησαν εξειδικευμένα για το σκοπό αυτό είτε τροποποιήθηκαν κατάλληλα. Οι εργασίες [27], [28] παρουσιάζουν τα αποτελέσματα με βάση τη μετρική accuracy. Στην [27] εξετάζονται διάφορα μεγέθη του συνόλου δεδομένων και χρονικά διαστήματα μεταξύ ιχνών, πετυχαίνοντας στην καλύτερη περίπτωση accuracy κοντά στο 95%. Από την άλλη, η εργασία [28] μελετά τα αποτελέσματα για διαφορετικά προφίλ ανωμαλιών (διαφορετικές αυξομειώσεις στις μετρικές) πετυχαίνοντας στη γενική περίπτωση accuracy της τάξης του 95% με την καλύτερη περίπτωση στο 97.9%. Οι εργασίες [29], [30] χρησιμοποιούν τις μετρικές precision, recall και f1 score. Η [29] εξετάζει τα αποτελέσματα για διάφορα μήκη ιχνών. Τα καλύτερα αποτελέσματα επιτυγχάνονται για ίχνη μικρού μήκους με τις τρεις μετρικές να έχουν πολύ υψηλή τιμή, κοντά στο 100%. Η εργασία [30] παρουσιάζει τα αποτελέσματα για τους διαφόρους αλγορίθμους μηχανικής μάθησης. Τα καλύτερα αποτελέσματα φαίνεται να επιτυγχάνονται κατά τη χρήση του μοντέλου MLP με όλες τις μετρικές να έχουν τιμή της τάξης του 98%.

2.3 Εντοπισμός ανωμαλιών με χρήση μετρικών

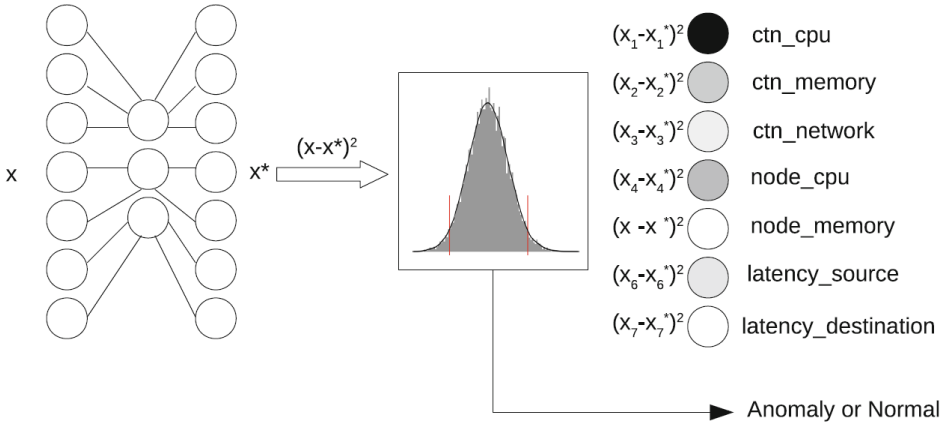
Κάποιες άλλες εργασίες χρησιμοποιούν μετρικές του συστήματος στο οποίο εκτελούνται οι εφαρμογές για να αξιολογήσουν την εμφάνιση ανωμαλιών. Οι μετρικές αυτές μπορεί να εκφράζουν την αξιοποίηση των πόρων του συστήματος, χρόνους απόκρισης κ.λπ.. Εργασίες όπως οι [36], [37], [38] χρησιμοποιούν μετρικές σε συνδυασμό με μεθόδους μη επιβλεπόμενης μηχανικής μάθησης. Οι εργασίες αυτές ακολουθούν παρόμοια προσέγγιση στο κομμάτι της ανίχνευσης ανωμαλιών και διαφοροποιούνται στην ανίχνευση της πηγής τους. Στις εργασίες αυτές χρησιμοποιούνται δεδομένα σε μορφή χρονοσειρών που περιέχουν πληροφορία για την αξιοποίηση των πόρων του συστήματος (CPU, RAM), πληροφορίες δικτυακής κίνησης αλλά και τους χρόνους απόκρισης μεταξύ των μικροϋπηρεσιών. Και οι τρεις εργασίες χρησιμοποιούν τον αλγόριθμο ομαδοποίησης BIRCH [39]. Ο αλγόριθμος αυτός αποτελεί παραλλαγή του KNN, ιδανικός για χρήση σε μεγάλα σύνολα δεδομένων, αφού εφαρμόζει ομαδοποίηση σε μία σύνοψη των δεδομένων, με την ελάχιστη δυνατή απώλεια πληροφορίας. Στο [36] τα δεδομένα από πολλαπλές μετρικές μοντελοποιούνται ως διανύσματα. Ένα αρχικό σύνολο δεδομένων χωρίς ανωμαλίες χρησιμοποιείται για την εκπαίδευση του μοντέλου και των υπολογισμό των ομάδων. Οι ανωμαλίες ανιχνεύονται σε πραγματικό χρόνο, εφόσον το εισερχόμενο διάνυσμα μετρικών δεν ανήκει σε κάποια από τις υπολογισμένες ομάδες. Η εργασία αυτή, σε αντίθεση με τις [37], [38], δεν περιλαμβάνει κάποιο στάδιο ανίχνευσης της πηγής των ανωμαλιών.

Οι εργασίες [37], [38] χρησιμοποιούν ακριβώς την ίδια μέθοδο για την ανίχνευση των ανωμαλιών. Και στις δύο περιπτώσεις χρησιμοποιείται ο αλγόριθμος BIRCH πάνω σε χρονοσειρές αποτελούμενες από χρόνους απόκρισης μεταξύ διαφορετικών μικροϋπηρεσιών. Οι εργασίες διαφοροποιούνται όμως στον εντοπισμό της πηγής των ανωμαλιών. Στο [37], η εφαρμογή μοντελοποιείται ως ένας γράφος υπηρεσιών, με τους κόμβους να αποτελούνται από τις υπηρεσίες και τις ακμές να δηλώνουν επικοινωνία μεταξύ τους. Από το γράφο αυτό εξάγεται ένας υπογράφος ανωμαλιών, στον οποίο συμπεριλαμβάνονται μόνο κόμβοι και ακμές με μη φυσιολογικούς χρόνους απόκρισης. Στις ακμές του υπογράφου αυτού προστίθενται κατάλληλα βάρη, ανάλογα και με τα δεδομένα για την αξιοποίηση των πόρων του μηχανήματος στο οποίο εκτελούνται οι μικροϋπηρεσίες. Στον τελικό γράφο εκτελείται ο αλγόριθμος PageRank [40] για τον εντοπισμό της πιο κεντρικής μικροϋπηρεσίας, η οποία δηλώνεται και ως πιθανότερη πηγή της ανωμαλίας (Σχήμα 2.12).



Σχήμα 2.12: Διαδικασία Εντοπισμού Πηγής Ανωμαλιών στο Micro RCA

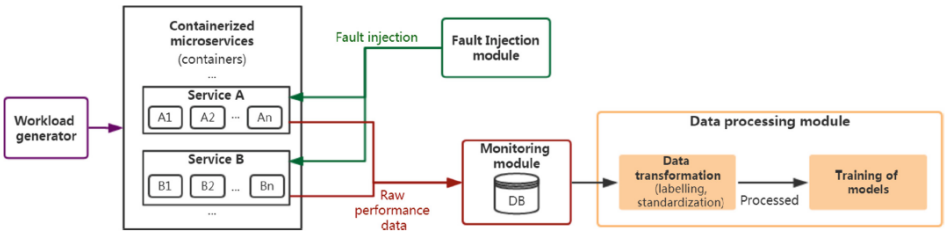
Η εργασία [38] χρησιμοποιεί το παραπάνω σύστημα εντοπισμού της μικροϋπηρεσίας που αποτελεί πηγή του προβλήματος, συνδυάζοντάς το όμως με ένα μοντέλο Autoencoder για την ανίχνευση της συγκεκριμένης υπαίτιας μετρικής, εντός της μικροϋπηρεσίας. Το μοντέλο του Autoencoder μαθαίνει τις σχέσεις μεταξύ μετρικών και ανακατασκευάζει παρόμοιους χρόνους απόκρισης με μικρό σφάλμα. Τα σφάλματα μοντελοποιούνται ως Γκαουσιανή κατανομή και σε περίπτωση που κάποιο σφάλμα ανακατασκευής δεν ακολουθεί την κατανομή με μεγάλη πιθανότητα, έχει εντοπιστεί η ζητούμενη μετρική. Η διαδικασία αυτή παρουσιάζεται στο Σχήμα 2.13.



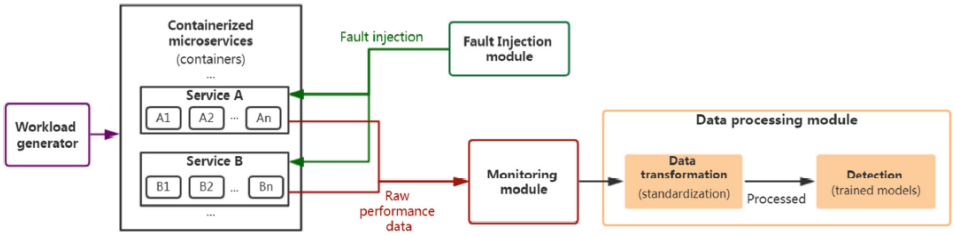
Σχήμα 2.13: Διαδικασία Εντοπισμού Πηγής Ανωμαλιών με Γκαουσιανή Κατανομή

Η εργασία [36] αξιολογεί την αποτελεσματικότητα του μοντέλου ανίχνευσης ανωμαλιών παρουσιάζοντας τα ποσοστά επιτυχών ανιχνεύσεων για διάφορα είδη ανωμαλιών και το χρόνο ανίχνευσης. Σε όλες τις περιπτώσεις οι ανωμαλίες ανιχνεύονται με υψηλά ποσοστά επιτυχίας σε πολύ μικρό χρονικό διάστημα. Στη χειρότερη περίπτωση οι ανωμαλίες ανιχνεύονται με ποσοστό επιτυχίας 83% και χρόνο ανίχνευσης μέχρι και ένα λεπτό. Οι [37], [38] χρησιμοποιούν πιο συμβατικές μετρικές αξιολόγησης παρουσιάζοντας precision at top k (πιθανότητα η σωστή απάντηση να βρίσκεται στις πρώτες k) και mean average precision για όλες τις μικροϋπηρεσίες. Από τις δύο εργασίες καλύτερα αποτελέσματα φαίνεται να πετυχαίνει η [37] με 89% precision και 97% mean average precision, έναντι 85.5% και 96.5% της [38].

Άλλες εργασίες, όπως οι [41], [42] συνδυάζουν δεδομένα μετρικών με μεθόδους επιβλεπόμενης μηχανικής μάθησης και συγκεκριμένα τεχνικές classification, όπως Support Vector Machine (SVM) [43], k-Nearest Neighbors (KNN), Naive Bayes (NB) [44], Random Forest (RF), Bayesian Network (BN), Best-First Decision Tree, (BFDT), Decision Table (DT), Logistic Model Tree (LMT) και Hidden NB. Η εργασία [41] δημιουργεί και συγκρίνει μοντέλα με βάση τους τέσσερεις πρώτους αλγορίθμους. Η εκπαίδευση γίνεται σε δεδομένα σε μορφή χρονοσειρών (Σχήμα 2.14). Με βάση την ταξινόμηση που προκύπτει εντοπίζονται οι μετρικές που περιέχουν ανωμαλίες (Σχήμα 2.15). Στην εργασία γίνεται η υπόθεση ότι κάθε μικροϋπηρεσία μπορεί να εκτελείται σε πολλαπλά κοντέινερ ταυτόχρονα και ότι μόνο ένα από αυτά μπορεί να παρουσιάζει προβληματική συμπεριφορά σε κάθε χρονική στιγμή. Με τις προϋποθέσεις αυτές, με τον εντοπισμό μιας προβληματικής χρονοσειράς μετρικών, χρησιμοποιείται ο αλγόριθμος Dynamic Time Warping (DTW) για τον υπολογισμό της ομοιότητας μεταξύ χρονοσειρών. Όλες οι χρονοσειρές της ανώμαλης μετρικής από κάθε κοντέινερ συγκρίνονται και η λιγότερο όμοια με τις υπόλοιπες και άρα και το κοντέινερ στο οποίο εκτελείται, επιλέγονται ως ρίζα του προβλήματος.



Σχήμα 2.14: Φάση Εκπαίδευσης

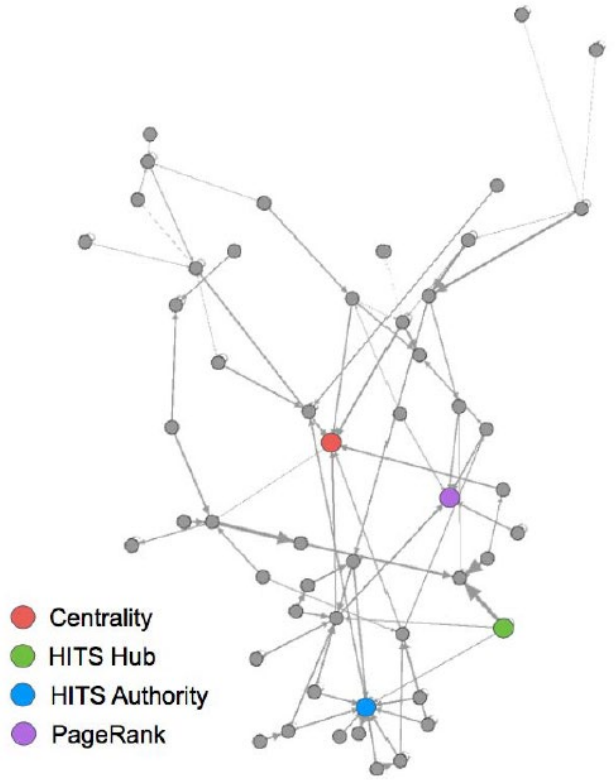


Σχήμα 2.15: Φάση Εντοπισμού

Η εργασία [42] συνδυάζει αλγορίθμους ταξινόμησης με εργαλεία Operation Analytics της IBM. Η ανίχνευση των ανωμαλιών γίνεται με χρήση των εργαλείων της IBM, υπολογίζοντας τη συσχέτιση μεταξύ των μετρικών που παρουσιάζονται σε μορφή χρονοσειρών. Συγκεκριμένα χρησιμοποιούνται πολλαπλά κριτήρια, όπως normal baseline variance, normal-to-flat variation, variance reduction, αιτιατότητα Granger, unexpected level κ.α.. Αφού εντοπιστούν οι προβληματικές μετρικές, ενεργοποιείται το μοντέλο εξαγωγής υπογραφών. Το μοντέλο αυτό χρησιμοποιεί έναν αλγόριθμο ταξινόμησης (στην εργασία υλοποιούνται και συγκρίνονται αυτοί που αναφέρθηκαν παραπάνω) για να διαχωρίσει το είδος ανωμαλίας το οποίο υποδηλώνει η προβληματική μετρική.

Η εργασία [41], για την αξιολόγηση των αποτελεσμάτων χρησιμοποιεί τις μετρικές precision, recall και f1 score. Συγκεκριμένα παρουσιάζονται τα αποτελέσματα που προκύπτουν για τρία διαφορετικά σύνολα δεδομένων με διαφορετικά μεγέθη. Στις περισσότερες περιπτώσεις τα μοντέλα παρουσιάζουν πολύ καλή απόδοση με όλες τις μετρικές να έχουν τιμές πάνω από 90%. Χαμηλότερη απόδοση σε κάποια από τα σύνολα δεδομένων φαίνεται να έχει ο αλγόριθμος SVM, ενώ την καλύτερη απόδοση φαίνεται να έχουν οι KNN και RF. Αντίστοιχες μετρικές αξιολόγησης χρησιμοποιούνται και από την εργασία [42] με την προσθήκη της μετρικής accuracy. Τα αποτελέσματα που παρουσιάζονται συγκρίνουν τους διαφορετικούς αλγορίθμους ταξινόμησης καθώς και διαφορετικά είδη προσομοιωμένων ανωμαλιών. Από τους αλγορίθμους που συγκρίνονται, καλύτερα αποτελέσματα επιτυγχάνουν οι LMT και SVM με όλες τις μετρικές αξιολόγησης να έχουν τιμές της τάξης του 98.8% και 98.6% αντίστοιχα. Όσον αφορά τα διαφορετικά είδη ανωμαλιών, σε όλες σχεδόν τις περιπτώσεις παρατηρούνται πολύ καλά αποτελέσματα με τιμές των μετρικών κοντά στο 100%. Εξαίρεση φαίνεται να αποτελούν κάποιες ανωμαλίες που εκδηλώνονται ως καθυστέρηση (latency), στις οποίες η μετρική precision φτάνει στην κατώτατη τιμή της (62%).

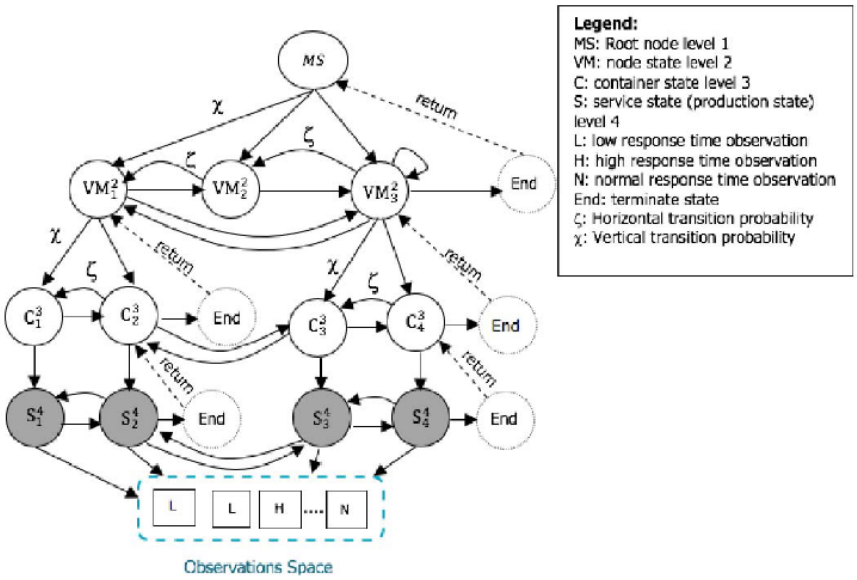
Οι μετρικές αποτελούνται τις περισσότερες φορές από αριθμητικά δεδομένα σε μορφή χρονοσειρών. Πολλές εργασίες, όπως οι [45], [46], [47], [48], [49], [50], [51] εκμεταλλεύονται το γεγονός αυτό και χρησιμοποιούν στατιστικές μεθόδους για την ανάλυση των μετρικών και την ανάδειξη των ανωμαλιών. Οι εργασίες αυτές ακολουθούν διάφορες προσεγγίσεις με τη βασική ιδέα όμως να αποτελείται από τον υπολογισμό της συσχέτισης μεταξύ των μετρικών. Στην [45] η ανίχνευση των ανωμαλιών γίνεται χρησιμοποιώντας τα έτοιμα εργαλεία Operation Analytics της IBM. Κατά την αρχική φάση εκπαίδευσης του μοντέλου, μαθαίνονται οι τυπικές τιμές των μετρικών και χτίζεται η αναπαράσταση της εφαρμογής μέσω ενός γράφου αιτιότητας των μετρικών. Οι κόμβοι του γράφου αντιστοιχούν στις διάφορες μετρικές και οι ακμές σε αιτιώδεις σχέσεις μεταξύ τους. Οι ακμές περιλαμβάνουν βάρη που υπολογίζονται ως ο συντελεστής Granger μεταξύ των χρονοσειρών και εκφράζουν τη συσχέτιση μεταξύ των δύο μετρικών που συνδέονται με ακμή. Αφού ανιχνευθεί κάποια προβληματική μετρική, ο γράφος χρησιμοποιείται για τον εντοπισμό της πιθανότερης πηγής του προβλήματος. Ο εντοπισμός αυτός γίνεται με βάση τη μετρική της κεντρικότητας των κόμβων, με τον κόμβο με τη μεγαλύτερη κεντρικότητα να αποτελεί την πιθανότερη αιτία του προβλήματος (Σχήμα 2.16). Στην εργασία εξετάζονται πολλαπλές μετρικές, όμως προτιμότερος κρίνεται ο αλγόριθμος PageRank.



Σχήμα 2.16: Σύγκριση Μεθόδων Υπολογισμού Κεντρικότητας

Η εργασία [46] βασίζει την υλοποίησή της σε ιεραρχικά μοντέλα Markov και συγκεκριμένα Hierarchical Hidden Markov Models (HHMM). Αρχικά, ο εντοπισμός των ανωμαλιών στα δεδομένα γίνεται παρατηρώντας τη μεταβολή των τιμών των μετρικών. Συγκεκριμένα, χρησιμοποιείται ο συντελεστής συσχέτισης του Spearman για την εκτίμηση της

επίδρασης του αριθμού εργασιών που εκτελούνται με τις τιμές των μετρικών που καταγράφονται. Αν ο συντελεστής που υπολογίζεται είναι αυξημένος, η μείωση της απόδοσης του συστήματος είναι άμεσα συσχετισμένη με τον αυξημένο αριθμό εργασιών και μπορεί να δηλωθεί ως ανωμαλία. Χρησιμοποιώντας το HHMM οι μετρικές μοντελοποιούνται ως ένα σύνολο καταστάσεων, με κάθε κατάσταση να αποτελεί ένα συνδυασμό τιμών των μετρικών (Σχήμα 2.17). Οι πιθανότητες μετάβασης μεταξύ των καταστάσεων αυτών υπολογίζονται από το ιστορικό του συνόλου δεδομένων, αντικατοπτρίζοντας την τυπική συμπεριφορά του συστήματος. Όταν εντοπιστεί κάποια ανωμαλία στις τιμές των μετρικών, η πιθανότερη αιτία του προβλήματος ανιχνεύεται προσδιορίζοντας τη μικροϋπηρεσία της οποίας η συμπεριφορά αποκλίνει περισσότερο από το μοντέλο Markov.

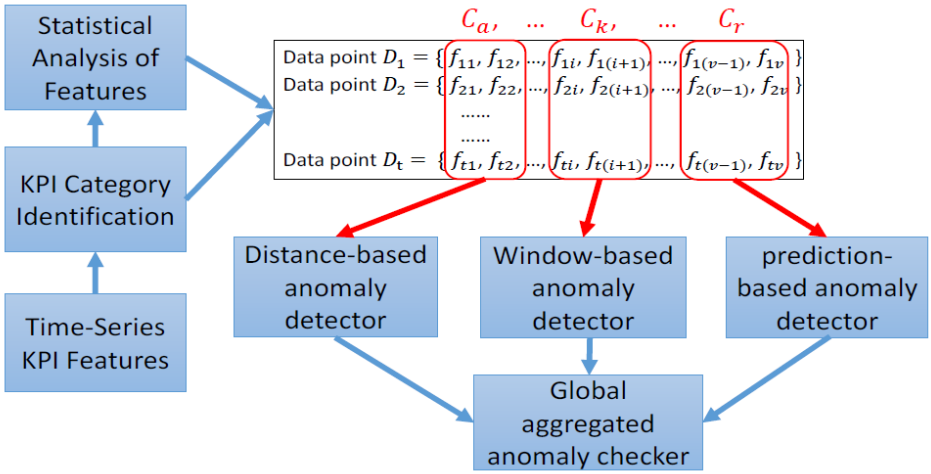


Σχήμα 2.17: Δομή Συστήματος με HHMM

Η εργασία [47] χρησιμοποιεί Granger Graphical Models (GGM) τα οποία υπολογίζουν τις χρονικές συσχετίσεις μεταξύ δεδομένων πολυμεταβλητών χρονοσειρών. Αφού το μοντέλο χτίσει τους γράφους χρονικής εξάρτησης της εφαρμογής, χρησιμοποιεί την απόσταση Kullback-Leibler (KL) για να υπολογίσει τη διαφορά των κατανομών των μετρικών. Αρχικά η απόσταση υπολογίζεται στα δεδομένα εκπαίδευσης που δεν περιέχουν ανωμαλίες, ώστε να υπολογιστεί το κατώφλι μέγιστης αποδεκτής φυσιολογικής απόστασης. Όταν κάποιο από τα εισερχόμενα δεδομένα υπερβεί το κατώφλι αυτό, δηλώνεται ως ανωμαλία.

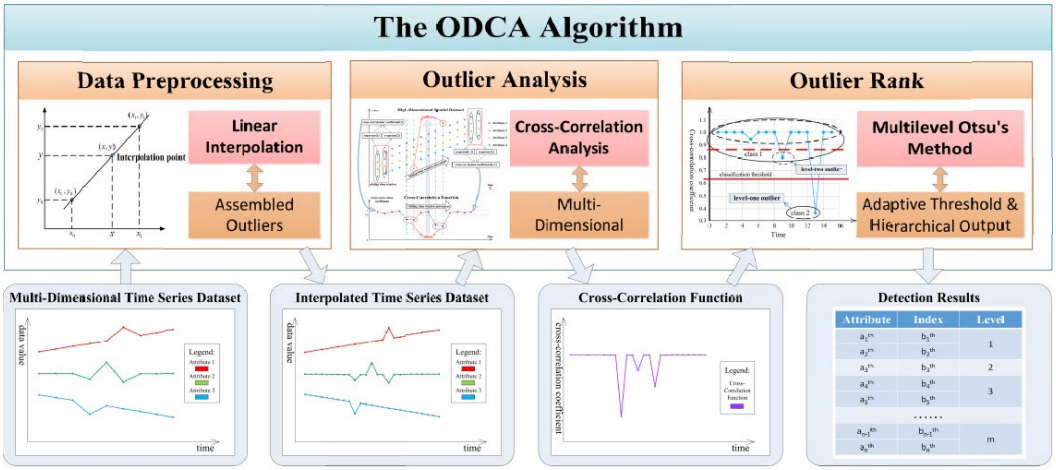
Η εργασία [48] ισχυρίζεται ότι η ανίχνευση ανωμαλιών σε διαφορετικές μετρικές επωφελείται από τη χρήση διαφορετικών μοντέλων. Στην εργασία αυτή οι διάφορες μετρικές χωρίζονται σε κατηγορίες με στατιστικά κριτήρια. Αρχικά εντοπίζονται γραμμικές εξαρτήσεις με χρήση του συντελεστή συσχέτισης Pearson. Ζεύγη μετρικών με υψηλή συσχέτιση συγχωνεύονται και χρησιμοποιείται η μετρική minimum-redundancy – maximum-relevance (MRMR) για την εξαγωγή των πιο σημαντικών χαρακτηριστικών των δεδομένων. Τα χαρακτηριστικά που επιλέγονται, ανάλογα με το είδος τους προωθούνται για ανίχνευση ανωμαλιών στο κατάλληλο μοντέλο (π.χ. μέτρηση απόστασης ή ταξινόμηση). Η ολοκληρωμένη διαδικασία κατηγοριοποίησης των δεδομένων μετρικών και η επιλογή του

αντίστοιχου μοντέλου για τον εντοπισμό ανωμαλιών παρουσιάζεται από την εργασία στο Σχήμα 2.18.



Σχήμα 2.18: Κατηγοριοποίηση Μετρικών και Επιλογή Μοντέλου

Στην εργασία [49] εφαρμόζεται μια διαδικασία τριών σταδίων για την ανίχνευση ανωμαλιών (Σχήμα 2.19). Αρχικά εφαρμόζεται γραμμική παρεμβολή, ώστε οι ακραίες τιμές που εμφανίζονται σε ομάδες μεγάλης πληθικότητας να διαχωριστούν και να είναι ευκολότερα παρατηρήσιμες. Στη συνέχεια υπολογίζεται η ετεροσυσχέτιση (cross-correlation) μεταξύ των διαφόρων μετρικών. Με τον τρόπο αυτό μειώνεται η διάσταση των δεδομένων εισόδου και οι ανωμαλίες μπορούν να εντοπιστούν ευκολότερα σε ένα «πηγάδι» της συνάρτησης cross-correlation. Για να διαχωριστεί η σοβαρότητα των ακραίων τιμών και να υπολογιστεί το κατώφλι πάνω από το οποίο αυτές θεωρούνται ανωμαλίες χρησιμοποιείται μία πολυεπίπεδη εκδοχή της μεθόδου του Otsu. Η μέθοδος αυτή συναντάται συχνά στην επεξεργασία εικόνας για το διαχωρισμό του φόντου από το κύριο αντικείμενο της εικόνας και χωρίζει τα δεδομένα με βάση τη μέγιστη διακύμανση μεταξύ διαφορετικών τάξεων και την ελάχιστη διακύμανση ανάμεσα σε δεδομένα της ίδιας τάξης. Παίρνοντας ως είσοδο τους συντελεστές συσχέτισης των μετρικών η μέθοδος υπολογίζει επαναλαμβανόμενα ένα υποσύνολο ανωμαλιών στα δεδομένα, μέχρι το πλήθος τους να φτάσει ένα προκαθορισμένο ποσοστό. Το κατώφλι που υπολογίζεται στο τελευταίο επίπεδο διαχωρισμού είναι το κατώφλι ανίχνευσης ανωμαλιών.



Σχήμα 2.19: Περιγραφή Συστήματος ODCA

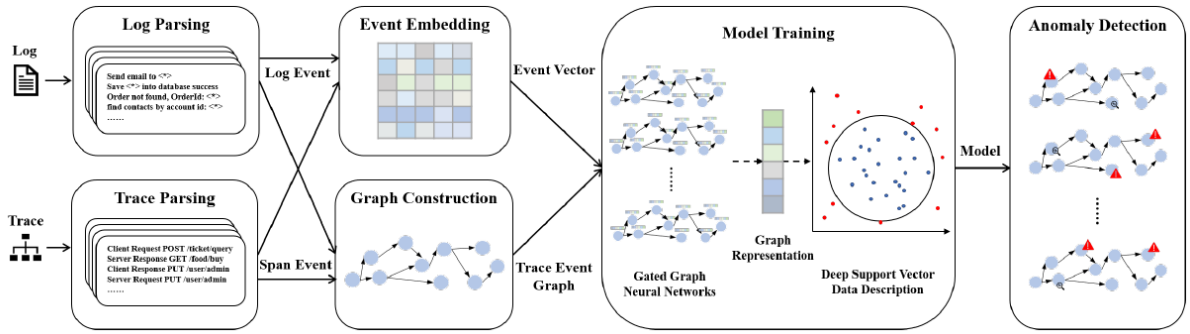
Οι εργασίες [50], [51] ακολουθούν παρόμοια προσέγγιση προσπαθώντας να προβλέψουν την εμφάνιση ανωμαλιών με βάση τα δεδομένα μετρικών. Το μοντέλο βασίζεται σε δύο αναπαραστάσεις της εφαρμογής με τη μορφή γράφων. Η μία από αυτές εκφράζει τη συσχέτιση των διαφορετικών τμημάτων λογισμικού που αποτελούν την εφαρμογή, τόσο μεταξύ τους, όσο και με το υλικό του υπολογιστή στον οποίο εκτελούνται. Ο γράφος αυτός μπορεί να δημιουργηθεί αυτόματα από ένα σύνολο μετρικών του συστήματος αλλά και δεδομένα ιχνηλάτησης από την εκτέλεση της εφαρμογής. Η δεύτερη αποτελεί μετασχηματισμό της πρώτης, έχει τη δομή ενός Bayesian network και αποτελεί μια μέθοδο αναπαράστασης της πληροφορίας για την διάδοση των ανωμαλιών στην εφαρμογή. Η αναπαράσταση αυτή περιέχει πληροφορία για όλες τις υπό προϋποθέσεις εξαρτήσεις μεταξύ μετρικών που εκφράζουν την πιθανότητα για κάθε μετρική-κόμβο να εμφανίζει ανωμαλίες δεδομένων των αντίστοιχων πιθανοτήτων των μετρικών-προγόνων της. Η πρόβλεψη των τιμών των χρονοσειρών γίνεται χρησιμοποιώντας την τεχνική Auto Regressive Moving Average (ARIMA). Με βάση τις τιμές που προκύπτουν, υπολογίζονται οι πιθανότητες αποτυχίας όλων των μικροϋπηρεσιών και ανανεώνονται οι αναπαραστάσεις της εφαρμογής.

Οι παραπάνω εργασίες κατά κύριο λόγο χρησιμοποιούν συμβατικές μετρικές όπως οι precision, recall, accuracy και f1 score για την αξιολόγηση των αποτελεσμάτων τους. Εξαίρεση αποτελεί η εργασία [46] που αξιολογεί τα αποτελέσματα με βάση μετρικές όπως οι Root Mean Square Error, Number of Correctly Detected Anomaly (CDA), Number of Correctly Identified Anomaly (CIA), Number of Incorrectly Detected Anomaly (IDA) και Number of Incorrectly Identified Anomaly (IIA). Το μοντέλο πετυχαίνει μικρό σφάλμα, της τάξης του 15% και πολύ υψηλά CDA και CIA (97.7% και 98.4% αντίστοιχα). Η εργασία [45] συγκρίνει αποτελέσματα για διάφορα είδη ανωμαλιών και αλγορίθμους κεντρικότητας παρουσιάζοντας διαγράμματα με τις μετρικές precision, recall και f1 score. Όπως αναφέρθηκε, την καλύτερη επίδοση φαίνεται να έχει ο αλγόριθμος PageRank σε όλα τα είδη ανωμαλιών, με τις υψηλότερες τιμές του f1 score να φτάνουν κοντά στο 100%. Η εργασία [47] παρουσιάζει τις τιμές των μετρικών precision, recall και f1 score (89%, 99% και 94% αντίστοιχα). Στην εργασία [48], η υβριδική μέθοδος κατηγοριοποίησης και ανίχνευσης ανωμαλιών πετυχαίνει 95% accuracy και 92% precision. Η εργασία [49] παρουσιάζει τους αναλυτικούς αριθμούς TP/TN, FP/FN και αντιστοιχίζοντας σε μετρικές όμοια με τις υπόλοιπες εργασίες πετυχαίνει στην καλύτερη

περίπτωση 99.9% precision, 100% recall και 99.5% f1 score. Οι εργασίες [50], [51] παρουσιάζουν αντίστοιχα αποτελέσματα με 43% precision, 87.7% recall και 89% accuracy.

2.4 Εντοπισμός ανωμαλιών με χρήση πολυτροπικών δεδομένων

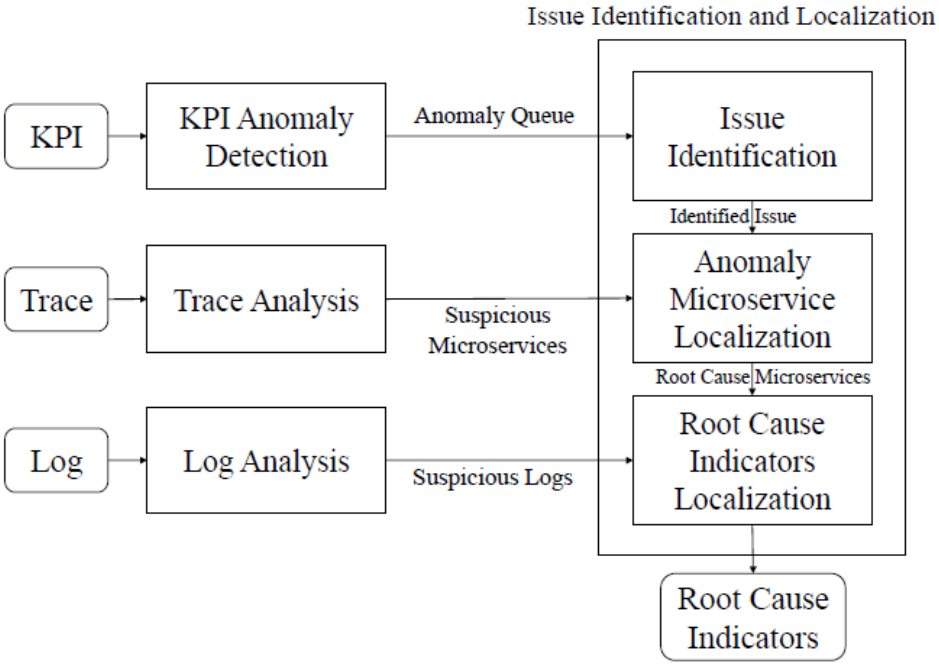
Κάποιες εργασίες, προσπαθώντας να βελτιώσουν την ικανότητα ανίχνευσης ανωμαλιών, χρησιμοποιούν δεδομένα από πολλαπλές πηγές. Για παράδειγμα, στην εργασία [52] συνδυάζονται δεδομένα από ίχνη και αρχεία καταγραφής, ενώ εργασίες όπως οι [53], [54] χρησιμοποιούν δεδομένα από αρχεία καταγραφής, ίχνη και μετρικές. Η εργασία [52] επεξεργάζεται κατάλληλα τα δεδομένα εισόδου, ώστε να προκύψουν οι κατηγορίες των εγγραφών των αρχείων καταγραφής και τα είδη spans που περιέχονται στα ίχνη. Τα πρότυπα αυτά στη συνέχεια κωδικοποιούνται με τη μορφή διανυσμάτων. Οι λέξεις κωδικοποιούνται με βάση ένα προ-εκπαιδευμένο μοντέλο GloVe. Στα διανύσματα που προκύπτουν ανατίθεται το κατάλληλο βάρος λαμβάνοντας υπόψιν τη σημασία των λέξεων με την τεχνική TF-IDF. Τα τελικά διανύσματα αξιοποιούνται για την κατασκευή ενός γράφου, με κόμβους που αντιστοιχούν σε αρχεία καταγραφής και πρότυπα span και ακμές που εκφράζουν τις χρονικές σχέσεις μεταξύ τους. Το πρόβλημα της ανίχνευσης ανωμαλιών αντιμετωπίζεται ως ένα πρόβλημα ταξινόμησης μιας τάξης (one-class classification). Το μοντέλο που χρησιμοποιείται στην εργασία είναι ένα Deep Support Vector Data Description (Deep SVDD) υλοποιημένο με Gated Graph Neural Networks (GGNNs). Στο μοντέλο αυτό οι κόμβοι του γράφου αναπαρίστανται ως μονάδες το νευρωνικού δικτύου και συνδέονται σύμφωνα με τον πίνακα γειτνίασης του γράφου. Το μοντέλο μαθαίνει μία λανθάνουσα αναπαράσταση κάθε γράφου, γύρω από την οποία δημιουργεί μια υπερσφαίρα. Για κάθε εισερχόμενο ίχνος εντοπίζονται τα κατάλληλα αρχεία καταγραφής με βάση τα αναγνωριστικά τους και σχηματίζεται η διανυσματική αναπαράστασή τους καθώς και ο γράφος. Ο γράφος δίνεται ως είσοδος στο μοντέλο, το οποίο υπολογίζει την λανθάνουσα αναπαράστασή του. Σε κάθε ίχνος ανατίθεται ένα σκορ ανωμαλίας που αντιστοιχεί στην ελάχιστη απόσταση της λανθάνουσας αναπαράστασης από την υπερσφαίρα που έχει μάθει το μοντέλο. Αν το σκορ υπερβεί ένα προκαθορισμένο κατώφλι, τα δεδομένα εισόδου χαρακτηρίζονται ως ανωμαλία. Η συνολική αρχιτεκτονική του συστήματος, από την επεξεργασία των δεδομένων εισόδου, την εκπαίδευση των μοντέλων, μέχρι και τον εντοπισμό ανωμαλιών παρουσιάζεται στο Σχήμα 2.20.



Σχήμα 2.20: Αρχιτεκτονική Συστήματος DeepTraLog

Η εργασία [53] αξιοποιεί δεδομένα από αρχεία καταγραφής, ίχνη και μετρικές. Οι χρονοσειρές των μετρικών χρησιμοποιούνται για την αρχική ανίχνευση των ανωμαλιών και τα

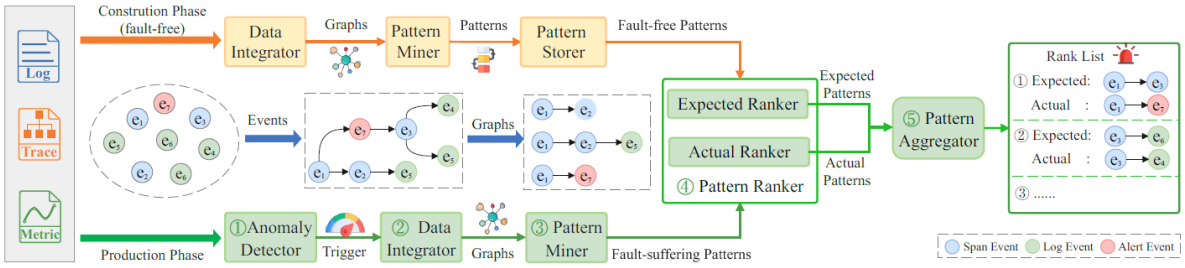
δεδομένα από τα αρχεία καταγραφής και τα ίχνη για την επαλήθευση του αποτελέσματος. Για την ανάλυση των χρονοσειρών χρησιμοποιούνται τεχνικές όπως Kernel Density Estimation (KDE) και Weighted Moving Average (WMA). Η μέθοδος KDE χρησιμοποιείται για τον υπολογισμό μιας εκτίμησης της συνάρτησης πυκνότητας πιθανότητας μιας τυχαίας μεταβλητής με βάση προηγούμενες τιμές της μεταβλητής με μη επιβλεπόμενο τρόπο. Η μέθοδος WMA από την άλλη χρησιμοποιεί το μέσο όρο ενός κυλιόμενου παραθύρου για την πρόβλεψη μελλοντικών τιμών της χρονοσειράς. Τα αρχεία καταγραφής αναλύονται ελέγχοντας για την ύπαρξη λέξεων κλειδιών στις εγγραφές και τα ίχνη αναλύονται με βάση το χρόνο απόκρισης κάθε span. Με βάση τις ανωμαλίες που εντοπίζονται στις μετρικές σχηματίζεται μια λίστα πιθανών ανωμαλιών, η οποία επικυρώνεται με βάση την ανάλυση των αντίστοιχων αρχείων καταγραφής και ιχνών. Για τον εντοπισμό της μικροϋπηρεσίας που αποτέλεσε την πιθανότερη αιτία του προβλήματος χρησιμοποιείται ένα σύστημα ψηφοφορίας με κάθε μικροϋπηρεσία να παίρνει ψήφους για κάθε μετρική που της αντιστοιχεί στη λίστα των μετρικών με ανωμαλίες. Μία γενική αρχιτεκτονική του συστήματος παρουσιάζεται από την εργασία στο Σχήμα 2.21.



Σχήμα 2.21: Αρχιτεκτονική του Συστήματος PDiagnose

Αντίστοιχη προσέγγιση ακολουθεί και η εργασία [54], χρησιμοποιώντας τις χρονοσειρές των μετρικών για την αρχική ανίχνευση των ανωμαλιών και τα δεδομένα από τα αρχεία καταγραφής και τα ίχνη για την αναπαράσταση της εφαρμογής μέσω ενός γράφου και τον εντοπισμό της πιθανής αιτίας του προβλήματος. Η αρχική ανίχνευση της ανωμαλίας γίνεται με βάση τον κανόνα k-σ, σύμφωνα με τον οποίο στοιχεία της χρονοσειράς με απόκλιση από τη μέση τιμή μεγαλύτερη από k φορές τη διασπορά θεωρούνται ανωμαλίες. Αφού εντοπιστεί ανωμαλία, τα δεδομένα από τις μετρικές, τα αρχεία καταγραφής και τα ίχνη υπόκεινται κατάλληλη επεξεργασία για να σχηματίσουν ένα γράφο γεγονότων. Από τις χρονοσειρές σημειώνονται μόνο οι χρονικές στιγμές που η τιμή μιας μετρικής υπερβαίνει τα προκαθορισμένα όρια, ενώ τα αρχεία καταγραφής και τα ίχνη αναλύονται όπως προηγουμένως

για να προκύψουν τα πρότυπα. Σε κάθε γράφο που δημιουργείται διασχίζονται όλα τα δυνατά μονοπάτια και εντοπίζονται αυτά που περιέχουν ακραίες τιμές μετρικών. Οι γράφοι ταξινομούνται ανάλογα με το ποσοστό μονοπατιών με ανωμαλίες που περιέχουν και εντοπίζεται η πιθανότερη αιτία του προβλήματος. Η αναλυτική αρχιτεκτονική του συστήματος παρουσιάζεται σε μορφή διαγράμματος από την εργασία και παρατίθεται στο Σχήμα 2.22.



Σχήμα 2.22: Αρχιτεκτονική Συστήματος Nezha

Οι εργασίες [52] και [54] αξιολογούν τα μοντέλα τους χρησιμοποιώντας ένα κοινό σύνολο δεδομένων. Στην [27] καταγράφεται 93% precision, 97.8% recall και 95.4% f1 score. Η εργασία [54] από την άλλη παρουσιάζει μόνο τη μετρική accuracy at top k πετυχαίνοντας accuracy 86.7%, 97.8% και 97.8% για τις τιμές του k=1, 3, 5 αντίστοιχα. Η εργασία [53] ελέγχει την επίδοση του μοντέλου χρησιμοποιώντας μόνο τη μετρική recall. Κατά μέσο όρο, στα σύνολα δεδομένων που εξετάζονται πετυχαίνει 84.8% recall.

2.5 Μοντέλα μη Επιβλεπόμενης Μηχανικής Μάθησης

2.5.1 BIRCH

Ο αλγόριθμος BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [39] είναι ένας αποδοτικός αλγόριθμος ομαδοποίησης (clustering) που έχει σχεδιαστεί ειδικά για μεγάλες ποσότητες δεδομένων. Βασική έννοια στη λειτουργία του αλγορίθμου BIRCH είναι η λειτουργία ομαδοποίησης Clustering Feature (CF). Κάθε CF αποτελεί μία συμπαγή αναπαράσταση ενός υποσυνόλου δεδομένων η οποία περιέχει πληροφορίες που επιτρέπουν στον αλγόριθμο να υπολογίζει χαρακτηριστικά όπως το κέντρο και τη διάμετρο των ομάδων. Διατηρώντας αυτή την πληροφορία για κάθε σημείο του συνόλου δεδομένων, ο αλγόριθμος συμπιέζει το συνολικό τους μέγεθος και μπορεί εύκολα να διαχειριστεί μεγάλα σύνολα δεδομένων, χωρίς αυξημένες απαιτήσεις μνήμης. Ο αλγόριθμος διατηρεί τις πληροφορίες της ομαδοποίησης σε μια δενδρική δομή που ονομάζεται δέντρο CF (CFTree). Τα φύλλα του δέντρου αντιστοιχούν σε ομάδες που αποθηκεύονται με τη μορφή CFs και οι ενδιάμεσοι κόμβοι βοηθούν στην ιεραρχική οργάνωση. Η δομή του δέντρου CF ανανεώνεται διαρκώς, καθώς προστίθενται νέα δεδομένα. Κατά την εισαγωγή δεδομένων, ο αλγόριθμος προσπαθεί να τα τοποθετήσει στο δέντρο. Αν το νέο σημείο βρίσκεται κοντά σε μία υπάρχουσα ομάδα (δηλαδή το σημείο βρίσκεται μέσα στη διάμετρο της ομάδας που υπάρχει αποθηκευμένη στο CF), τότε μπορεί να προστεθεί στην ομάδα, τα στοιχεία της οποίας ανανεώνονται. Αν δεν υπάρχει καμία κοντινή ομάδα, δημιουργείται μια νέα. Παράλληλα με τη διαδικασία αυτή στο δέντρο εφαρμόζεται μια μέθοδος εξισορρόπησης για τη διατήρηση των επιδόσεων του αλγορίθμου ανεξάρτητα του όγκου δεδομένων. Σαν προαιρετικό τελικό βήμα του αλγορίθμου

περιλαμβάνεται μια πιο λεπτομερής διαδικασία ομαδοποίησης (όπως για παράδειγμα ο αλγόριθμος k-means) για την αύξηση της ακρίβειας των αποτελεσμάτων.

2.5.2 Local Outlier Factor

Ο αλγόριθμος Local Outlier Factor (LOF) [55] είναι σχεδιασμένος για μη επιβλεπόμενο εντοπισμό ακραίων τιμών (outlier detection). Ο αλγόριθμος εντοπίζει σημεία που αποκλίνουν του κανονικού υπολογίζοντας και συγκρίνοντας την τοπική πυκνότητα κάθε σημείου με αυτή των γειτόνων του. Βασική έννοια στη λειτουργία του αλγορίθμου αποτελεί η τοπική πυκνότητα, η οποία υπολογίζεται με βάση τις αποστάσεις ενός σημείου από τους γείτονές του (σημεία σε πυκνές περιοχές έχουν περισσότερους γείτονες σε μικρή απόσταση, ενώ σημεία σε περιοχές χαμηλής πυκνότητας που αποτελούν πιθανές ανωμαλίες, εμφανίζουν λιγότερους γείτονες σε μεγαλύτερη απόσταση). Μία άλλη σημαντική έννοια είναι η απόσταση προσπελασιμότητας (reachability distance, RD) ενός σημείου p από ένα άλλο o. Η απόσταση αυτή υπολογίζεται ως η μέγιστη εκ των: αποστάσεων του o από τον κοντινότερό του γείτονα και της πραγματικής απόστασης μεταξύ p και o. Αντίστοιχα υπολογίζεται και η τοπική απόσταση προσπελασιμότητας (LRD), ως ο αντίστροφος του μέσου όρου όλων των RD ενός σημείου από τους γείτονές του. Κατά την εκτέλεση του αλγορίθμου LOF αρχικά υπολογίζονται οι k κοντινότεροι γείτονες για κάθε σημείο του συνόλου δεδομένων. Ο αριθμός k είναι παράμετρος του μοντέλου και μπορεί να τροποποιηθεί ανάλογα τα χαρακτηριστικά των δεδομένων που χρησιμοποιούνται. Στη συνέχεια υπολογίζεται η RD κάθε σημείου από τους γείτονες αυτούς, αλλά και οι αντίστοιχες LRD. Τέλος υπολογίζεται το σκορ LOF κάθε σημείου, ως ο λόγος της LRD του σημείου προς τη μέση LRD των γειτόνων του. Όσο μεγαλύτερο το σκορ, τόσο πιθανότερο για το σημείο να αποτελεί ανωμαλία. Το όριο στη διάκριση ανωμαλιών καθορίζεται από το ποσοστό θετικών (ανώμαλων) δειγμάτων στο σύνολο.

2.5.3 Isolation Forest

Το μοντέλο Isolation Forest [56] αποτελείται από έναν αλγόριθμο μη επιβλεπόμενης μάθησης ειδικά σχεδιασμένο για ανίχνευση ανωμαλιών (anomaly detection) ή ακραίων τιμών (outlier detection). Σε αντίθεση με προσεγγίσεις όπως ο LOF ο αλγόριθμος αυτός λειτουργεί απομονώνοντας σημεία. Βασίζεται στην παραδοχή ότι οι ανωμαλίες ή οι ακραίες τιμές είναι αραιές και διαφορετικές από την πλειονότητα των δεδομένων, γεγονός που καθιστά ευκολότερη την απομόνωσή τους. Σύμφωνα με την παραδοχή αυτή, οι ακραίες τιμές είναι λίγες και συχνά διαφορετικές ως προς ορισμένα χαρακτηριστικά και άρα μία τυχαία κατάτμηση των δεδομένων θα τις διαχωρίσει γενικά ταχύτερα από ότι για τα κανονικά σημεία. Όσο λιγότεροι διαχωρισμοί απαιτούνται για την απομόνωση ενός σημείου δεδομένων, τόσο πιο πιθανό είναι να πρόκειται για ακραίο σημείο. Η διαδικασία που ακολουθείται για την απομόνωση των σημείων βασίζεται στην αναδρομική κατάτμηση. Ο αλγόριθμος επιλέγει τυχαία ένα από τα χαρακτηριστικά και καθορίζει πάλι τυχαία μία τιμή του χαρακτηριστικού αυτού με βάση την οποία τα δεδομένα χωρίζονται στα δύο. Η διαδικασία αυτή επαναλαμβάνεται αναδρομικά μέχρι τα σημεία να απομονωθούν. Η ακολουθία κατατμήσεων αναπαρίσταται από τον αλγόριθμο με μία δενδρική δομή που αποκαλείται δέντρο απομόνωσης (Isolation Tree, iTree). Το βάθος του δέντρου αντιστοιχεί στον αριθμό κατατμήσεων που απαιτούνται για την απομόνωση ενός σημείου. Κατά την εκτέλεση του αλγορίθμου, αρχικά κατασκευάζονται

πολλαπλά δέντρα με τυχαίες επιλογές χαρακτηριστικών και τιμών κατάτμησης για κάθε κόμβο του δέντρου. Το σύνολο των δέντρων σχηματίζει ένα «δάσος» όπως και στον αλγόριθμο Τυχαίου Δάσους (Random Forest). Για κάθε σημείο του συνόλου δεδομένων, υπολογίζεται ο αριθμός κατατμήσεων μέχρι την απομόνωσή του με βάση τα δέντρα που έχουν σχηματιστεί. Το τελικό σκορ κάθε σημείου προκύπτει από το μέσο αριθμό κατατμήσεών του. Σημεία με μικρά μέσα μονοπάτια δηλώνονται ως ανωμαλίες.

2.6 Ρύθμιση Παραμέτρων και Υπερπαραμέτρων

Η διαδικασία εύρεσης βέλτιστου συνδυασμού παραμέτρων είναι ζωτικής σημασίας για την επίτευξη της καλύτερης δυνατής απόδοσης ενός μοντέλου μηχανικής μάθησης. Βασική μέθοδος για το σκοπό αυτό είναι η μέθοδος Grid Search με διασταυρούμενη επικύρωση (Cross-Validation, CV). Η μέθοδος αυτή δοκιμάζει συστηματικά συνδυασμούς υπερπαραμέτρων και χρησιμοποιεί διασταυρούμενη επικύρωση για να αξιολογήσει την απόδοση κάθε συνδυασμού, διασφαλίζοντας ότι το μοντέλο γενικεύει καλά σε άγνωστα δεδομένα. Οι υπερπαράμετροι αποτελούν παραμέτρους που καθορίζονται πριν από την έναρξη της διαδικασίας μάθησης, σε αντίθεση με τις παραμέτρους του μοντέλου που μαθαίνονται κατά τη διάρκεια της εκπαίδευσης. Παραδείγματα υπερπαραμέτρων περιλαμβάνουν την ισχύ κανονικοποίησης στη λογιστική παλινδρόμηση (logistic regression), τον αριθμό των δέντρων σε ένα τυχαίο δάσος (random forest) ή τον ρυθμό μάθησης στην ενίσχυση κλίσης (gradient boost). Κατά την εκτέλεση του Grid Search, ελέγχονται εξαντλητικά όλοι οι συνδυασμοί υπερπαραμέτρων που παρέχονται. Η εφαρμογή της διασταυρούμενης επικύρωσης εξασφαλίζει ότι το μοντέλο γενικεύει επαρκώς σε άγνωστα δεδομένα. Αυτό εξασφαλίζεται χωρίζοντας τα δεδομένα εκπαίδευσης σε k τμήματα. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας k-1 από αυτά τα τμήματα, χρησιμοποιώντας το άλλο για την επαλήθευση. Η διαδικασία αυτή επαναλαμβάνεται k φορές, με διαφορετικό τμήμα να χρησιμοποιείται για την επαλήθευση κάθε φορά. Ως τελικό αποτέλεσμα θεωρείται ο μέσος όρος των αποτελεσμάτων για τις k επαναλήψεις.

2.7 Μέθοδος Ανάλυσης Αρχείων Καταγραφής Drain

Ο Drain [57] είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος ανάλυσης αρχείων καταγραφής που επικεντρώνεται στην αποτελεσματική εξαγωγή δομημένων προτύπων από αδόμητα αρχεία καταγραφής. Είναι ιδιαίτερα κατάλληλος για κατανεμημένα συστήματα μεγάλης κλίμακας, όπου τα αρχεία καταγραφής μπορεί να διαφέρουν σημαντικά ως προς τη δομή και τη μορφή τους σε διαφορετικές υπηρεσίες. Η λειτουργία του αλγορίθμου βασίζεται στην έννοια της ιεραρχικής ομαδοποίησης. Τα αρχεία καταγραφής οργανώνονται σε μία δενδροειδή δομή όπου κάθε κόμβος αναπαριστά μία απόφαση βασισμένη σε ένα σύμβολο (π.χ. μια λέξη ή ένα σύμβολο στην εγγραφή ενός αρχείου καταγραφής). Διατρέχοντας το δέντρο, παρόμοιες καταχωρήσεις εγγραφών ομαδοποιούνται και προκύπτει ένα δομημένο πρότυπο για κάθε ομάδα. Το δέντρο αυτό έχει σταθερό βάθος, επιτρέποντας στον αλγόριθμο να κλιμακώνει καλά για μεγάλο όγκο δεδομένων εισόδου, καθώς περιορίζει τον αριθμό αποφάσεων που απαιτούνται για την κατηγοριοποίηση μιας εγγραφής. Κατά την εκτέλεση του αλγορίθμου, κάθε εγγραφή χωρίζεται σε σύμβολα αποτελούμενα από μεμονωμένες λέξεις ή στοιχεία της εγγραφής. Στη συνέχεια διασχίζεται το δέντρο απόφασης που έχει δημιουργηθεί για τον

εντοπισμό των πιο όμοιων εγγραφών με βάση τα κοινά τους σύμβολα. Μετά το πέρας της ομαδοποίησης των εγγραφών, ο αλγόριθμος δημιουργεί πρότυπα αντικαθιστώντας τα δυναμικά στοιχεία κάθε εγγραφής (στοιχεία όπως χρονικές σημάνσεις, αναγνωριστικά αιτημάτων ή διευθύνσεις IP) με σύμβολα μπαλαντέρ, γενικεύοντας ουσιαστικά τη δομή των εγγραφών του αρχείου καταγραφής. Ο αλγόριθμος προσφέρει δυνατότητα λειτουργίας σε πραγματικό χρόνο, με τις νέες εγγραφές που καταφθάνουν να χρησιμοποιούνται για την ανανέωση της δομής του δέντρου αλλά και των προτύπων.

Τα δομημένα πρότυπα που προκύπτουν μέσω της μεθόδου Drain μπορούν να χρησιμοποιηθούν για τον εντοπισμό ανωμαλιών, επισημαίνοντας τις καταχωρήσεις που δεν ταιριάζουν με κανένα υπάρχον πρότυπο. Αυτές οι καταχωρήσεις που δεν ταιριάζουν μπορεί να αντιπροσωπεύουν προηγουμένως αθέατα συμβάντα ή σφάλματα, καθιστώντας τες υποψήφιες για περαιτέρω διερεύνηση.

Κεφάλαιο 3 - Μεθοδολογία

Στο πλαίσιο της εργασίας, δημιουργείται μία εφαρμογή που προσομοιώνει το περιβάλλον μιας κατανεμημένης εφαρμογής. Η εφαρμογή αυτή αξιοποιείται για την παραγωγή και τη συλλογή δεδομένων παρατηρησιμότητας. Τα δεδομένα αυτά, έπειτα από κατάλληλη επεξεργασία χρησιμοποιούνται για τον εντοπισμό ανωμαλιών σε ένα πολυτροπικό σύστημα εντοπισμού ανωμαλιών. Παρακάτω παρουσιάζονται τα επιμέρους τμήματα της εφαρμογής, της διαδικασίας παραγωγής, συλλογής και επεξεργασίας των δεδομένων παρατηρησιμότητας καθώς και του εντοπισμού ανωμαλιών.

3.1 Μοντέλο Κατανεμημένης Εφαρμογής

Η ανίχνευση ανωμαλιών με τη χρήση συνδυασμού δεδομένων παρατηρησιμότητας βασίζεται στη λειτουργία μιας κατανεμημένης εφαρμογής. Στα πλαίσια της παρούσας εργασίας, μία τέτοια εφαρμογή αποτελείται από μικροϋπηρεσίες που αλληλεπιδρούν για την εκτέλεση ενός συνόλου λειτουργιών. Κάθε υπηρεσία εκτελείται απομονωμένα, συχνά σε διαφορετικά συστήματα, και επικοινωνεί με τις υπόλοιπες μέσω δικτύου. Το ακόλουθο μοντέλο απεικονίζει τα βασικά στοιχεία της εφαρμογής:

- **Μικροϋπηρεσίες**: Οι μικροϋπηρεσίες είναι τα θεμελιώδη δομικά στοιχεία της κατανεμημένης εφαρμογής. Κάθε μικροϋπηρεσία είναι μία ανεξάρτητη μονάδα που εκτελεί συγκεκριμένες εργασίες. Οι μικροϋπηρεσίες χειρίζονται αιτήματα είτε από εξωτερικούς χρήστες, είτε από άλλες υπηρεσίες του συστήματος. Κάθε μικροϋπηρεσία διατηρεί τη δική της κατάσταση για να διασφαλίζει την απομόνωσή της και την επεκτασιμότητα. Αυτή η αυτονομία επιτρέπει στις επιμέρους υπηρεσίες να κλιμακώνονται ανεξάρτητα και μειώνει τον κίνδυνο αλυσιδωτών αποτυχιών.
- **Σύστημα Επικοινωνίας**: Για τη διαχείριση της επικοινωνίας μεταξύ υπηρεσιών, μια κατανεμημένη εφαρμογή πρέπει να ενσωματώνει ένα σύστημα επικοινωνίας. Η αλληλεπίδραση μεταξύ μικροϋπηρεσιών μπορεί να αναπαρασταθεί από έναν Ακυκλικό Κατευθυνόμενο Γράφο (Directed Acyclic Graph, DAG), με κάθε κόμβο του γράφου να αναπαριστά μία υπηρεσία και τις ακμές να αντιστοιχούν στη ροή της επικοινωνίας μεταξύ τους. Η επικοινωνία μεταξύ υπηρεσιών μπορεί να γίνεται με σύγχρονο ή και με ασύγχρονο τρόπο. Στη σύγχρονη επικοινωνία (όπως σε κλήσεις HTTP) η υπηρεσία που καλεί περιμένει την απάντηση πριν συνεχίσει, σχηματίζοντας πιο γραμμικές διαδρομές στο DAG. Στην ασύγχρονη επικοινωνία (συχνά μέσω συστημάτων ανταλλαγής μηνυμάτων), τα μηνύματα αποστέλλονται μεταξύ των υπηρεσιών χωρίς να περιμένουν απάντηση, επιτρέποντας στις υπηρεσίες να συνεχίσουν ανεξάρτητα την εργασία τους. Αυτό μπορεί να δημιουργήσει πιο σύνθετες, παράλληλες διαδρομές μέσα στο DAG, επιτρέποντας την επεκτασιμότητα και την ανοχή σε σφάλματα.
- **Βάσεις Δεδομένων - Κρυφές Μνήμες**: Οι περισσότερες μικροϋπηρεσίες αλληλεπιδρούν με κάποια μορφή μόνιμης αποθήκευσης, όπως οι βάσεις δεδομένων ή προσωρινής αποθήκευσης όπως οι κρυφές μνήμες για την αποθήκευση και την ανάκτηση δεδομένων. Κάθε υπηρεσία μπορεί να έχει τη δική της αποκλειστική βάση δεδομένων για τη διατήρηση της απομόνωσης και την αποφυγή των διασταυρούμενων εξαρτήσεων μεταξύ των υπηρεσιών. Το επίπεδο προσωρινής αποθήκευσης, χρησιμοποιείται για τη μείωση της καθυστέρησης και την αποφόρτιση της κίνησης από

τις βάσεις δεδομένων, αποθηκεύοντας δεδομένα στα οποία γίνεται συχνή πρόσβαση στη μνήμη. Οι αλληλεπιδράσεις τόσο με τις βάσεις δεδομένων όσο και με τις κρυφές μνήμες αποτελούν βασικά στοιχεία του προφίλ απόδοσης του συστήματος. Αυτές οι αλληλεπιδράσεις καταγράφονται παρέχοντας πληροφορίες σχετικά με τους χρόνους ανάκτησης δεδομένων, τα ποσοστά επιτυχίας/αποτυχίας της κρυφής μνήμης και άλλες σχετικές μετρικές, οι οποίες συμβάλλουν στα δεδομένα παρατηρησιμότητας.

3.2 Instrumentation Εφαρμογής

3.2.1 Υποδομή παρακολούθησης και καταγραφής

Η κατανεμημένη εφαρμογή αυτή πρέπει να συνοδεύεται και από ένα σύστημα ολοκληρωμένης παρατηρησιμότητας, με δυνατότητα συλλογής, αποθήκευσης και εμφάνισης αρχείων καταγραφής, ιχνών και μετρικών. Η διαδικασία του Instrumentation είναι μία κρίσιμη πτυχή στην ανάπτυξη της παρατηρησιμότητας μιας εφαρμογής. Μέσω αυτής εξασφαλίζεται ότι τα απαραίτητα δεδομένα συλλέγονται από κάθε στοιχείο του συστήματος, επιτρέποντας την αποτελεσματική παρακολούθηση και την ανίχνευση ανωμαλιών. Αυτή η υποδομή περιλαμβάνει τρεις βασικούς πυλώνες της παρατηρησιμότητας: ίχνη, αρχεία καταγραφής και μετρικές.

Αρχεία Καταγραφής

Τα αρχεία καταγραφής καταγράφουν λεπτομερείς πληροφορίες σχετικά με τα συμβάντα που λαμβάνουν χώρα σε κάθε υπηρεσία, όπως μηνύματα σφάλματος, ενημερώσεις κατάστασης και λειτουργικές πληροφορίες. Κάθε μικροϋπηρεσία είναι εξοπλισμένη με συστήματα καταγραφής (π.χ. Fluentd, Fluent-Bit) για την παραγωγή δομημένων ή ημιδομημένων αρχείων καταγραφής σε μορφή εγγράφων JSON ή άλλες μορφές αναγνώσιμες από μηχανήματα. Αυτά τα αρχεία καταγραφής περιλαμβάνουν συνήθως χρονικές σημάνσεις, επίπεδα σοβαρότητας καταγραφής (info, warn, error) και δεδομένα πλαισίου (π.χ. αναγνωριστικό αίτησης, αναγνωριστικό χρήστη). Με την υιοθέτηση δομημένης καταγραφής, οι προγραμματιστές διασφαλίζουν ότι τα αρχεία καταγραφής περιέχουν συνεπή ζεύγη κλειδιών-τιμών, διευκολύνοντας το φιλτράρισμα και την αναζήτηση στα αρχεία καταγραφής κατά τη διάρκεια της αποσφαλμάτωσης ή όταν τα συστήματα ανίχνευσης ανωμαλιών αναλύουν αυτά τα αρχεία καταγραφής για μοτίβα. Ένα σύστημα συνάθροισης των εγγραφών αρχείων καταγραφής (Log Aggregator) συλλέγει και αποθηκεύει αρχεία καταγραφής από όλες τις μικροϋπηρεσίες. Μέσω των προγραμμάτων αυτών, μπορούν να δημιουργηθούν δομές ευρετηρίων με βάση τα αρχεία καταγραφής, επιτρέποντας τη συσχέτιση μεταξύ συμβάντων σε όλες τις υπηρεσίες και την ταχύτερη ανίχνευση ασυνήθιστων μοτίβων ή σφαλμάτων.

Ίχνη

Η κατανεμημένη ιχνηλάτηση είναι μια βασική τεχνική στις σύγχρονες αρχιτεκτονικές μικροϋπηρεσιών για την παρακολούθηση της ροής των αιτημάτων σε διάφορες υπηρεσίες. Τα δεδομένα που αποτελούν τα ίχνη εκτέλεσης της εφαρμογής συλλέγονται κατά την εξυπηρέτηση αιτημάτων χρηστών. Κάθε μικροϋπηρεσία της κατανεμημένης εφαρμογής είναι εξοπλισμένη για την καταγραφή spans, δηλαδή χρονομετρημένες λειτουργίες που αντιπροσωπεύουν μονάδες εργασίας, τα οποία θα δομήσουν το ίχνος της εκάστοτε εκτέλεσης. Αυτά τα spans επισημαίνονται με μεταδεδομένα όπως το όνομα της υπηρεσίας, η διεύθυνση

της υπηρεσίας που το παρήγαγε, ο κωδικός κατάστασης και οποιαδήποτε άλλα προσαρμοσμένα δεδομένα που σχετίζονται με το αίτημα. Τα spans συνενώνονται σε μία δενδρική δομή που αποτελεί το ενιαίο ίχνος εκτέλεσης με βάση αναγνωριστικά που περιλαμβάνουν. Τα ίχνη που σχηματίζονται συλλέγονται και αποθηκεύονται από κατάλληλα προγράμματα (π.χ. Zipkin, Jaeger) που εκτελούν τη συνάθροιση (Trace Aggregators). Τα προγράμματα αυτά συχνά παρέχουν μια οπτικοποίηση του τρόπου με τον οποίο τα αιτήματα διατρέχουν τις μικροϋπηρεσίες. Αυτό το οπτικό χρονοδιάγραμμα βοηθά στον εντοπισμό σημείων συμφόρησης καθυστέρησης και πιθανών αστοχιών σε όλες τις υπηρεσίες.

Μετρικές

Οι μετρικές παρέχουν ποσοτικά δεδομένα σχετικά με την υγεία και την απόδοση των υπηρεσιών, όπως η χρήση της CPU, η μνήμη, η καθυστέρηση αιτήσεων και τα ποσοστά σφαλμάτων. Κάθε υπηρεσία είναι εξοπλισμένη με βιβλιοθήκες που συλλέγουν αυτόματα βασικές μετρήσεις σε επίπεδο συστήματος και εφαρμογής. Οι μετρήσεις σε επίπεδο συστήματος περιλαμβάνουν CPU, μνήμη και I/O δίσκου, ή στατιστικά του δικτύου ενώ οι μετρήσεις σε επίπεδο εφαρμογής επικεντρώνονται σε ποσοστά αιτήσεων, ποσοστά επιτυχίας/σφαλμάτων και χρόνους επεξεργασίας. Οι προγραμματιστές μπορούν επίσης να καταγράφουν προσαρμοσμένες μετρήσεις, όπως ο αριθμός των στοιχείων που επεξεργάζονται ανά εξυπηρέτηση αιτήματος ή ο χρόνος που δαπανάται σε κρίσιμα τμήματα του κώδικα. Αυτές οι μετρήσεις παρέχουν βαθύτερη εικόνα για συγκεκριμένες συμπεριφορές της εφαρμογής. Οι μετρήσεις συλλέγονται μέσω βιβλιοθηκών (όπως ο Prometheus), που συλλέγουν μετρήσεις από υπηρεσίες σε τακτά χρονικά διαστήματα. Τα δεδομένα των μετρήσεων αποθηκεύονται στη συνέχεια σε μια βάση δεδομένων χρονοσειρών, όπου μπορούν να αναζητηθούν και να απεικονιστούν με τη χρήση εργαλείων παρατηρησιμότητας (όπως το Grafana).

3.2.2 Παραδείγματα και Προσεγγίσεις Instrumentation

Η διαδικασία του Instrumentation μπορεί να εφαρμοστεί με δύο κύριους τρόπους: χειροκίνητα και αυτόματα. Και οι δύο προσεγγίσεις είναι απαραίτητες για τη συλλογή δεδομένων παρατηρησιμότητας, αλλά διαφέρουν ως προς την πολυπλοκότητα, τον έλεγχο και το είδος των δεδομένων που παράγουν. Παρακάτω, περιγράφονται και οι δύο προσεγγίσεις με παραδείγματα.

Χειροκίνητο Instrumentation

Το χειροκίνητο Instrumentation περιλαμβάνει τη ρητή προσθήκη κώδικα από τους προγραμματιστές για τη συλλογή δεδομένων παρατηρησιμότητας. Αυτό δίνει πλήρη έλεγχο σε ό,τι μετράται και καταγράφεται, αλλά απαιτεί μεγαλύτερη προσπάθεια και προσεκτικό σχεδιασμό. Ακολουθούν παραδείγματα που αποτυπώνουν τη διαδικασία της χειροκίνητης συλλογής δεδομένων. Ως πρώτο παράδειγμα, θεωρείται μια πλατφόρμα ηλεκτρονικού εμπορίου, όπου μια υπηρεσία επεξεργασίας παραγγελιών χειρίζεται τις παραγγελίες των πελατών. Σε ένα σύστημα που πραγματοποιείται χειροκίνητη συλλογή δεδομένων ιχνηλάτησης, ο προγραμματιστής πρέπει να προσθέσει ειδικά προσαρμοσμένα spans για κάθε βασικό βήμα της διαδικασίας (π.χ. επικύρωση της παραγγελίας, χρέωση της πληρωμής και αποστολή email επιβεβαίωσης). Κάθε span περιέχει επιπλέον πληροφορία πλαισίου, όπως το αναγνωριστικό της παραγγελίας ή τη μέθοδο πληρωμής, παρέχοντας μια λεπτομερή εικόνα της διαδρομής του αιτήματος στο σύστημα. Ως ένα δεύτερο παράδειγμα παρουσιάζεται μία

μικροϋπηρεσία που διαχειρίζεται εισιτήρια τεχνικής υποστήριξης (support tickets). Η χειροκίνητη καταγραφή (logging) προστίθεται σε βασικά βήματα της διαδικασίας, όπως όταν ένα εισιτήριο ανοίγει, ανατίθεται σε έναν εκπρόσωπο και επιλύεται.

Αυτόματο Instrumentation

Η αυτόματη διαδικασία συλλογής δεδομένων απλοποιεί τη διαδικασία χρησιμοποιώντας χειριστές ή βιβλιοθήκες που καταγράφουν δεδομένα της εφαρμογής χωρίς να τροποποιούν τη βασική επιχειρησιακή λογική. Για παράδειγμα, ένα σύστημα πληρωμών βασισμένο σε υποδομές νέφους έχει ρυθμιστεί ώστε να παρακολουθεί αυτόματα όλα τα αιτήματα HTTP και τα ερωτήματα στη βάση δεδομένων χρησιμοποιώντας μια βιβλιοθήκη παρακολούθησης. Χωρίς την ανάγκη χειροκίνητης δημιουργίας span, δημιουργούνται ίχνη για όλα τα αιτήματα, συμπεριλαμβανομένων των συναλλαγών πελατών και των επαληθεύσεων πληρωμών, παρέχοντας ορατότητα από άκρο σε άκρο σε όλο το σύστημα. Παρουσιάζοντας ένα δεύτερο παράδειγμα για την αυτόματη συλλογή μετρικών, μελετάται μια εφαρμογή που χρησιμοποιεί ένα εργαλείο παρακολούθησης μετρήσεων για την αυτόματη συλλογή μετρήσεων σχετικά με τη χρήση CPU, την κατανάλωση μνήμης και την καθυστέρηση αιτήσεων από όλες τις μικροϋπηρεσίες. Τα δεδομένα αυτά εξάγονται σε ένα σύστημα παρακολούθησης όπου ορίζονται προκαθορισμένες ειδοποιήσεις για υψηλή καθυστέρηση ή υπερβολική χρήση πόρων, επιτρέποντας την παρακολούθηση της υγείας του συστήματος σε πραγματικό χρόνο χωρίς πρόσθετη κωδικοποίηση.

Συμπερασματικά, η προσέγγιση χειροκίνητου Instrumentation προσφέρει στους προγραμματιστές πλήρη έλεγχο των δεδομένων παρατηρησιμότητας που συλλέγονται, επιτρέποντας ειδικά προσαρμοσμένες μετρικές, αρχεία καταγραφής και ίχνη που αφορούν την επιχειρησιακή λογική της εφαρμογής υπό παρακολούθηση. Αυτή η λεπτομέρεια καθιστά δυνατή την παρακολούθηση κρίσιμων διεργασιών με λεπτομέρεια, η οποία μπορεί να είναι απαραίτητη για τον εντοπισμό λεπτών ζητημάτων απόδοσης ή ανωμαλιών. Ωστόσο, το βασικό μειονέκτημα της χειροκίνητης αυτής προσέγγισης είναι η πρόσθετη πολυπλοκότητα και η απαιτούμενη προσπάθεια. Οι προγραμματιστές πρέπει να γράφουν και να διατηρούν πρόσθετο κώδικα, ο οποίος μπορεί να γίνει δυσκίνητος, ιδίως σε μεγάλα ή εξελισσόμενα συστήματα. Η ασυνέπεια στην υλοποίηση μεταξύ των υπηρεσιών μπορεί επίσης να οδηγήσει σε κενά στην παρατηρησιμότητα. Από την άλλη, η αυτόματη προσέγγιση παρουσιάζεται ως μια βελτιωμένη λύση που εξασφαλίζει ομοιόμορφη συλλογή δεδομένων με ελάχιστη προσπάθεια του προγραμματιστή. Τα εργαλεία και οι βιβλιοθήκες που παρέχονται μπορούν να καταγράφουν αυτόματα βασικές μετρικές και ίχνη (όπως καθυστερήσεις εξυπηρέτησης αιτημάτων και χρήση πόρων), προσφέροντας καλύτερη επίγνωση στην απόδοση του συστήματος. Η προσέγγιση αυτή είναι ιδανική σε περιπτώσεις που το ζητούμενο είναι γρήγορη εγκατάσταση και παρακολούθηση σε μια κατανεμημένη αρχιτεκτονική. Ωστόσο, οι αυτόματες προσεγγίσεις ενδέχεται να παραλείπουν ορισμένες λεπτομέρειες που αφορούν συγκεκριμένες υπηρεσίες ή να εισάγουν επιβάρυνση. Ενώ παρέχουν καλή γενική κάλυψη, ενδέχεται να μην έχουν το βάθος και την ευελιξία που απαιτούνται για πιο σύνθετες ή προσαρμοσμένες περιπτώσεις χρήσης, γεγονός που καθιστά αναγκαίο ένα μείγμα και των δύο προσεγγίσεων.

3.3 Επεξεργασία Δεδομένων Παρατηρησιμότητας

Πριν τη διαδικασία της ανάλυσης, τα ακατέργαστα δεδομένα παρατηρησιμότητας που έχουν συλλεγεί και αποθηκευτεί από τα επιμέρους εργαλεία πρέπει να υποστούν προεπεξεργασία για τον καθαρισμό, το φιλτράρισμα και τη μορφοποίηση των δεδομένων. Ενδεικτικά στάδια της διαδικασίας αυτής είναι τα:

- **Αφαίρεση Διπλότυπων Δεδομένων**: Οι διπλές εγγραφές, ειδικά στα αρχεία καταγραφής και τα ίχνη, αφαιρούνται για να αποφευχθεί η στρεβλή ανάλυση.
- **Κανονικοποίηση**: Δεδομένα παρατηρησιμότητας από διαφορετικές υπηρεσίες μπορεί να έχουν διάφορες μορφές, ετικέτες, ή συμβάσεις ονοματοδοσίας. Η κανονικοποίηση των δεδομένων εξασφαλίζει συνεπή ονομασία και δομή σε όλα τα αρχεία καταγραφής, τα ίχνη και τις μετρικές, διευκολύνοντας την ανάλυση και τη συσχέτιση μεταξύ των υπηρεσιών.
- **Σύνοψη Δεδομένων**: Τα ακατέργαστα δεδομένα μπορούν να συνενωθούν για να μειωθούν οι απαιτήσεις αποθήκευσης και να βελτιωθούν οι επιδόσεις κατά την ανάλυση. Για παράδειγμα, τα δεδομένα μετρικών μπορούν να συναθροιστούν ανά λεπτό ή ανά ώρα, ενώ τα ίχνη μπορούν να συνοψιστούν υπολογίζοντας τους μέσους χρόνους απόκρισης.

3.3.1 Δημιουργία Προτύπων

Μια σημαντική πτυχή της επεξεργασίας δεδομένων παρατηρησιμότητας είναι η δημιουργία προτύπων για τα αρχεία καταγραφής και τα ίχνη. Αυτά τα πρότυπα παρέχουν δομημένες μορφές που διευκολύνουν την ευκολότερη ανάλυση, την ανίχνευση ανωμαλιών και τη συσχέτιση μεταξύ διαφορετικών υπηρεσιών. Παρακάτω παρουσιάζεται αναλυτικά η διαδικασία δημιουργίας αυτών των προτύπων και οι μέθοδοι μέσω των οποίων μπορούν να δημιουργηθούν. Υπάρχουν διάφοροι τρόποι για τη δημιουργία προτύπων αρχείων καταγραφής και ιχνών, που κυμαίνονται από χειροκίνητες μεθόδους έως πλήρως αυτοματοποιημένους αλγορίθμους. Στη χειροκίνητη προσέγγιση, οι προγραμματιστές αναλύουν χειροκίνητα τα αρχεία καταγραφής και τα ίχνη, εντοπίζουν μοτίβα και καθορίζουν πρότυπα με βάση τις γνώσεις τους για το σύστημα. Ενώ η χειροκίνητη δημιουργία προτύπων προσφέρει υψηλή ακρίβεια και προσαρμογή, είναι χρονοβόρα και επιρρεπής σε ανθρώπινα λάθη, ιδίως σε μεγάλα, δυναμικά περιβάλλοντα. Η αυτοματοποιημένη δημιουργία προτύπων χρησιμοποιεί αλγόριθμους μηχανικής μάθησης ή αντιστοίχισης προτύπων για τον εντοπισμό κοινών δομών σε αρχεία καταγραφής και ίχνη. Ορισμένες από τις δημοφιλείς τεχνικές περιλαμβάνουν αλγορίθμους εξόρυξης προτύπων (template mining) όπως τους αλγορίθμους Drain και Spell ή ακόμα και μεθόδους μηχανικής μάθησης όπως η ομαδοποίηση.

Πρότυπα Αρχείων Καταγραφής

Τα αρχεία καταγραφής που παράγονται από κατανεμημένα συστήματα έχουν συχνά διάφορες μορφές, γεγονός που μπορεί να καταστήσει την ανάλυση δύσκολη. Με τη δημιουργία τυποποιημένων προτύπων, τα αρχεία καταγραφής ομαλοποιούνται, διευκολύνοντας τη συνεπή εξαγωγή χρήσιμων πληροφοριών. Η διαδικασία ακολουθεί συνήθως τα εξής βήματα:

1. **Συλλογή Αρχείων Καταγραφής**: Ακατέργαστα αρχεία καταγραφής συλλέγονται από όλες τις μικροϋπηρεσίες ή τα στοιχεία του συστήματος. Οι εγγραφές που συλλέγονται μπορεί να διαφέρουν σημαντικά ως προς τη μορφή τους ιδίως σε κατανεμημένα

συστήματα, όπου διαφορετικές υπηρεσίες μπορεί να χρησιμοποιούν διαφορετικές συμβάσεις κατά την καταγραφή δεδομένων.

2. **Ανάλυση Αρχείων Καταγραφής**: Χρησιμοποιώντας ένα εργαλείο ανάλυσης αρχείων καταγραφής, οι εγγραφές αναλύονται σε δομημένα πεδία. Από κάθε εγγραφή εξάγονται πληροφορίες όπως η χρονική σήμανση, το επίπεδο σοβαρότητας, το όνομα της υπηρεσίας από την οποία προέρχονται, το περιεχόμενο του μηνύματος και οποιαδήποτε συμπεριλαμβανόμενη πληροφορία πλαισίου.
3. **Εντοπισμός Μοτίβων**: Μέσω χειροκίνητης εξέτασης ή αυτοματοποιημένων εργαλείων, εντοπίζονται επαναλαμβανόμενα μοτίβα στα αρχεία καταγραφής. Για παράδειγμα, πολλές εγγραφές καταγραφής μπορεί να ακολουθούν μια κοινή μορφή όπως: [TIMESTAMP] - [LEVEL] - [SERVICE] - Message. Αυτά τα επαναλαμβανόμενα μοτίβα αποτελούν τη βάση του προτύπου.
4. **Δημιουργία Προτύπων**: Μόλις εντοπιστούν κοινά μοτίβα, δημιουργούνται πρότυπα που ταιριάζουν με αυτά τα μοτίβα. Αυτά τα πρότυπα καθορίζουν την αναμενόμενη δομή των αρχείων καταγραφής, επιτρέποντας την αυτοματοποιημένη εξαγωγή των σχετικών πεδίων.
5. **Εφαρμογή Προτύπων**: Τα πρότυπα που προκύπτουν κατά την ανάλυση εφαρμόζονται σε νέες εισερχόμενες εγγραφές για το μετασχηματισμό τους σε δομημένη μορφή.
6. **Ανανέωση Προτύπων (προαιρετικά)**: Τα πρότυπα θα πρέπει να ενημερώνονται ή να βελτιώνονται περιοδικά καθώς εμφανίζονται νέες μορφές ή πρότυπα καταγραφής. Αυτό το βήμα διασφαλίζει ότι το σύστημα παραμένει ευέλικτο στις εξελισσόμενες μορφές καταγραφής.

Πρότυπα Ιχνών Εκτέλεσης

Παρόμοια με τα πρότυπα καταγραφής, τα πρότυπα ιχνών είναι χρήσιμα για τη δημιουργία μιας τυποποιημένης δομής για τα κατανεμημένα δεδομένα ιχνηλάτησης. Ένα ίχνος αποτελείται συνήθως από μια σειρά από spans, καθένα από τα οποία αντιπροσωπεύει μια λειτουργία σε μια κατανεμημένη εξυπηρέτηση αιτήματος. Τα πρότυπα επιτρέπουν τον ευκολότερο εντοπισμό ανωμαλιών κανονικοποιώντας τη μορφή των ιχνών σε όλες τις υπηρεσίες. Η διαδικασία κατασκευής των προτύπων αυτών τυπικά αποτελείται από τα ακόλουθα βήματα:

1. **Συλλογή Δεδομένων Ιχνηλάτησης**: Δεδομένα ιχνηλάτησης συλλέγονται από μια κατανεμημένη εφαρμογή χρησιμοποιώντας ειδικά εργαλεία ιχνηλάτησης. Αυτά τα ίχνη συνήθως καταγράφουν χρόνους εξυπηρέτησης αιτημάτων αλλά και τη ροή των αιτημάτων αυτών μεταξύ των υπηρεσιών.
2. **Κανονικοποίηση Spans**: Τα δεδομένα spans που συλλέγονται από τις διάφορες υπηρεσίες πρέπει να κανονικοποιηθούν ορίζοντας τα βασικά πεδία που πρέπει να περιέχει κάθε span. Τα κοινά πεδία περιλαμβάνουν το όνομα της υπηρεσίας, το όνομα της λειτουργίας, την ώρα έναρξης, τη διάρκεια και τυχόν δείκτες σφάλματος. Η κανονικοποίηση των πεδίων επιτρέπει την ευκολότερη σύγκριση μεταξύ των spans σε διαφορετικές υπηρεσίες.
3. **Αναγνώριση Μοτίβων**: Κοινά μοτίβα εντοπίζονται στη δομή των spans ή και των ιχνών. Για παράδειγμα, ένα ίχνος για ένα αίτημα HTTP μπορεί να περιλαμβάνει πάντα τα spans σχετικά με λειτουργίες όπως «authentication», «database query» και

«response». Η αναγνώριση αυτών των επαναλαμβανόμενων μοτίβων βοηθά στη δημιουργία προτύπων.

4. **Δημιουργία Προτύπων**: Με βάση τα κοινά μοτίβα, δημιουργούνται πρότυπα των ιχνών που καθορίζουν τη ροή ενός τυπικού αιτήματος σε πολλαπλές υπηρεσίες. Το πρότυπο μπορεί να περιλαμβάνει πληροφορίες σχετικά με την αναμενόμενη σειρά των spans, τα κρίσιμα διαστήματα για τη μέτρηση των επιδόσεων και τα κατώτατα όρια για τις αποδεκτές καθυστερήσεις.
5. **Εφαρμογή Προτύπων**: Μόλις δημιουργηθούν τα πρότυπα, εφαρμόζονται σε νέα ίχνη για το μετασχηματισμό τους σε δομημένες εγγραφές.
6. **Ανανέωση Προτύπων (προαιρετικά)**: Καθώς προστίθενται νέες υπηρεσίες ή εξελίσσεται η συμπεριφορά του συστήματος, τα πρότυπα ιχνηλάτησης πρέπει να ενημερώνονται ώστε να αντικατοπτρίζουν την μεταβαλλόμενη αρχιτεκτονική και τα πρότυπα αιτήσεων.

3.4 Προσομοίωση Ανωμαλιών

Για την αποτελεσματική αξιολόγηση των συστημάτων ανίχνευσης ανωμαλιών, είναι ζωτικής σημασίας η προσομοίωση ρεαλιστικών ανώμαλων συνθηκών στα δεδομένα παρατηρησιμότητας. Αυτές οι ανωμαλίες μπορούν να εισαχθούν σε αρχεία καταγραφής, ίχνη ή μετρικές, ανάλογα με τη φύση του υπό δοκιμή συστήματος και την επιθυμητή εστίαση της δοκιμής. Η προσθήκη των προσομοιωμένων ανωμαλιών μπορεί να γίνει με διάφορες μεθόδους. Χρησιμοποιώντας τη χειροκίνητη προσέγγιση, οι προγραμματιστές ή οι χειριστές εισάγουν χειροκίνητα ανωμαλίες κατά τη διάρκεια των πειραματικών δοκιμών. Αυτό μπορεί να περιλαμβάνει την επεξεργασία εγγραφών αρχείων καταγραφής, την τροποποίηση ιχνών ή την προσαρμογή μετρικών κατά τη διάρκεια ενός ελεγχόμενου πειράματος. Ενώ αυτή η προσέγγιση προσφέρει υψηλή ακρίβεια, απαιτεί σημαντική χειρωνακτική προσπάθεια και δεν κλιμακώνει καλά για συστήματα μεγάλης κλίμακας. Για το λόγο αυτό, η χρήση απλών προγραμμάτων (scripts) για την αυτοματοποίηση της προσομοίωσης ανωμαλιών είναι μια κοινή πρακτική. Με τη μέθοδο αυτή παρέχεται περισσότερος έλεγχος και επαναληψιμότητα, ενώ παράλληλα μειώνεται η ανάγκη για χειροκίνητη παρέμβαση. Η διαδικασία προσομοίωσης ανωμαλιών συχνά αναλαμβάνεται εξ ολοκλήρου και από εξειδικευμένα εργαλεία. Παρακάτω παρουσιάζονται ορισμένες από τις πιο συνηθισμένες προσεγγίσεις για την προσομοίωση ανωμαλιών.

3.4.1 Προσομοίωση Ανωμαλιών σε Αρχεία Καταγραφής

Τα δεδομένα αρχείων καταγραφής είναι μια κρίσιμη πηγή για τον εντοπισμό προβλημάτων σε κατανεμημένα συστήματα. Η εισαγωγή ανωμαλιών στα αρχεία καταγραφής μπορεί να προσομοιώσει διάφορα σενάρια αποτυχίας που μιμούνται προβλήματα του πραγματικού κόσμου. Ορισμένες μέθοδοι περιλαμβάνουν:

- **Εισαγωγή Μηνυμάτων Σφάλματος**: Ένας απλός τρόπος για την εισαγωγή ανωμαλιών στα αρχεία καταγραφής είναι η εισαγωγή συνθετικών μηνυμάτων σφάλματος σε καίρια σημεία της ροής εγγραφών. Αυτά τα μηνύματα σφάλματος θα πρέπει να είναι δομημένα παρόμοια με τις πραγματικές εγγραφές καταγραφής, ώστε να συνδυάζονται άψογα με την κανονική ροή δεδομένων. Οι εγγραφές που περιέχουν τα

μηνύματα αυτά συνοδεύονται και από αυξημένο επίπεδο σοβαρότητας (όπως WARN, ERROR).

- **Αναπάντεχη Εμφάνιση νέων Μοτίβων**: Με την εισαγωγή απροσδόκητων ή σπάνιων προτύπων εγγραφών, όπως μηνύματα που υποδεικνύουν ξαφνικές αλλαγές διαμόρφωσης ή εξαιρέσεις κατά την εκτέλεση της εφαρμογής, μπορεί να δοκιμαστεί η ικανότητα του συστήματος να ανιχνεύει ανωμαλίες.

3.4.2 Προσομοίωση Ανωμαλιών σε Ίχνη

Στα κατανεμημένα συστήματα, τα ίχνη παρέχουν πληροφορίες για τον κύκλο ζωής των αιτημάτων, που καλύπτουν πολλαπλές υπηρεσίες. Η εισαγωγή ανωμαλιών στα δεδομένα ιχνών βοηθά στην προσομοίωση συμφορήσεων που μειώνουν την απόδοση, σφαλμάτων που οφείλονται στις εξαρτήσεις υπηρεσιών και προβλημάτων καθυστέρησης μεταξύ υπηρεσιών. Ορισμένες μέθοδοι προσομοίωσης ανωμαλιών σε δεδομένα ιχνηλάτησης παρατίθενται:

- **Τεχνητή Αύξηση Χρόνου Απόκρισης**: Ίσως η πιο συνηθισμένη προσομοιωμένη ανωμαλία στα ίχνη είναι η αύξηση του χρόνου απόκρισης, με τεχνητές καθυστερήσεις να εισάγονται σε συγκεκριμένα spans ή υπηρεσίες. Για παράδειγμα, σε ένα κατανεμημένο σύστημα μικροϋπηρεσιών, η προσθήκη επιπλέον καθυστέρησης σε ένα span ερωτήματος στη βάση δεδομένων (π.χ. αύξηση του χρόνου ερωτήματος από 50ms σε 500ms) προσομοιώνει την αργή απόδοση της βάσης δεδομένων. Αυτό βοηθά στον έλεγχο του πόσο καλά το σύστημα ανιχνεύει υπολειτουργικά στοιχεία ή υπηρεσίες.
- **Εισαγωγή Σφαλμάτων στα Spans**: Για την προσομοίωση αποτυχιών σε ένα κατανεμημένο αίτημα, μπορούν να εισαχθούν κωδικοί σφάλματος σε συγκεκριμένα spans. Για παράδειγμα, η εισαγωγή ενός span με κατάσταση HTTP 500 (εσωτερικό σφάλμα διακομιστή) ή αποτυχία σύνδεσης με βάση δεδομένων μιμείται αποτυχίες υπηρεσιών. Το σύστημα ανίχνευσης ανωμαλιών θα πρέπει να είναι σε θέση να αναγνωρίζει αυτά τα ίχνη με τα προσομοιωμένα σφάλματα ως ασυνήθιστη συμπεριφορά.
- **Μη Εμφάνιση Spans στην Δομή Ιχνών**: Μια άλλη μορφή ανωμαλίας είναι η παράλειψη ορισμένων spans που κανονικά αναμένονται στη δομή ενός ίχνους. Για παράδειγμα, παίρνοντας ένα ίχνος που ακολουθεί τη δομή request - query - reply και παραλείποντας το span που αφορά το βήμα του ερωτήματος (query) προσομοιώνεται μια διακοπή της αντίστοιχης υπηρεσίας.

3.4.3 Προσομοίωση Ανωμαλιών στις Μετρικές

Οι μετρικές παρέχουν ποσοτικά δεδομένα σχετικά με την απόδοση του συστήματος, όπως η χρήση πόρων, η καθυστέρηση εξυπηρέτησης αιτήσεων και τα ποσοστά σφαλμάτων. Η εισαγωγή ανωμαλιών στα δεδομένα μετρικών βοηθά στην προσομοίωση της εξάντλησης των πόρων, της επιβράδυνσης του συστήματος και της υποβάθμισης των επιδόσεων. Μερικές μέθοδοι που χρησιμοποιούνται για την προσομοίωση ανωμαλιών σε δεδομένα μετρικών είναι οι ακόλουθες:

- **Υψηλή Χρήση CPU και Μνήμης**: Οι ανωμαλίες στις μετρικές μπορούν να εισαχθούν με τεχνητή αύξηση της χρήσης της CPU ή της μνήμης για σύντομο χρονικό διάστημα. Για παράδειγμα, η προσομοίωση μιας αιχμής χρήσης CPU μπορεί να μιμηθεί ένα

σύστημα υπό μεγάλο φορτίο, ενώ μια ξαφνική αύξηση της κατανάλωσης μνήμης μπορεί να προσομοιώσει μια διαρροή μνήμης (memory leak).

- **Τεχνητή Στέρηση Πόρων**: Η έλλειψη πόρων μπορεί να προσομοιωθεί με την τεχνητή μείωση της διαθεσιμότητας κρίσιμων πόρων του συστήματος (π.χ. εύρος ζώνης δικτύου, είσοδος/έξοδος δίσκου ή διαθέσιμες συνδέσεις στη βάση δεδομένων).

3.5 Εντοπισμός Ανωμαλιών

Λόγω της διαφορετικής φύσης των δεδομένων παρατηρησιμότητας που εξάγονται από μια κατανεμημένη εφαρμογή, η ανίχνευση ανωμαλιών σε αυτά διεξάγεται με βέλτιστο τρόπο χρησιμοποιώντας διάφορα μέσα. Ανεξάρτητα του τρόπου διάγνωσης των ανωμαλιών όμως, οι μέθοδοι που εφαρμόζονται ακολουθούν κοινή δομή. Η γενική διαδικασία εντοπισμού ανωμαλιών αποτελείται από δύο στάδια. Αρχικά παρατηρείται ένα στάδιο εκπαίδευσης. Το στάδιο αυτό χαρακτηρίζεται από την offline φύση του. Για την εκπαίδευση απαιτείται η συγκέντρωση ενός όγκου δεδομένων που θα χρησιμοποιηθεί ώστε τα εκάστοτε μοντέλα να μάθουν την κανονική λειτουργία της εφαρμογής (ή την κατανομή των δεδομένων). Η συλλογή αυτών των δεδομένων γίνεται σε ένα ενδεχομένως εκτεταμένο χρονικό διάστημα και σε καμία περίπτωση σε πραγματικό χρόνο. Αντίθετα, το στάδιο εντοπισμού που ακολουθεί, βασιζόμενο στη γνώση που έχει συγκεντρωθεί στο στάδιο εκπαίδευσης, έχει τη δυνατότητα λειτουργίας σε πραγματικό χρόνο, καθώς τα δεδομένα που συλλέγονται από την εφαρμογή προωθούνται στο σύστημα εντοπισμού.

3.5.1 Αρχεία Καταγραφής & Ίχνη

Όσον αφορά τα αρχεία καταγραφής και τα ίχνη, έχοντας δομήσει τα δεδομένα χρησιμοποιώντας πρότυπα, όπως αναφέρθηκε, ο εντοπισμός ανωμαλιών μπορεί να επιτευχθεί με άμεση σύγκριση των προτύπων που έχουν προκύψει κατά την επεξεργασία. Χρησιμοποιώντας ως σύνολο εκπαίδευσης δεδομένα από εκτελέσεις της εφαρμογής που δεν περιέχουν ανωμαλίες, υπολογίζονται όλα τα πρότυπα εγγραφών αρχείων καταγραφής και ιχνών που μπορεί να εντοπιστούν κατά την κανονική λειτουργία. Συγκρίνοντας τα πρότυπα που προκύπτουν από δεδομένα ελέγχου με προσθήκη σφαλμάτων με αυτά που δημιουργήθηκαν κατά τη διαδικασία εκπαίδευσης, τα μόνα πρότυπα που θα διαφέρουν είναι αυτά που αντιστοιχούν στις ανωμαλίες. Με τη μέθοδο αυτή μπορούν να υπολογιστούν ετικέτες που σημειώνουν τα δεδομένα ως κανονικά (normal) ή εσφαλμένα (anomalous). Η ανίχνευση ανωμαλιών γίνεται σε επίπεδο εγγραφής στην περίπτωση των αρχείων καταγραφής και σε επίπεδο ίχνους για τα δεδομένα ιχνών εκτέλεσης.

3.5.2 Μετρικές

Οι μετρικές από την άλλη, αποτελούμενες από αριθμητικές τιμές στη μορφή χρονοσειρών, ευνοούν τη χρήση μοντέλων μηχανικής μάθησης. Στο πλαίσιο της εργασίας συγκρίνεται η απόδοση τριών μεθόδων στον εντοπισμό ακραίων τιμών (outlier detection) και συγκεκριμένα των αλγορίθμων BIRCH, Local Outlier Factor και Isolation Forest. Τα τρία αυτά μοντέλα, ανήκουν στην κατηγορία αλγορίθμων μη επιβλεπόμενης μηχανικής μάθησης, επομένως δεν απαιτούν την ύπαρξη ετικετών για τα δεδομένα εκπαίδευσης, γεγονός ιδιαίτερα θετικό όπως έχει αναφερθεί. Στην περίπτωση των μοντέλων BIRCH και Isolation Forest, η διαδικασία αποτελείται από ένα στάδιο εκπαίδευσης χρησιμοποιώντας δεδομένα από την

κανονική λειτουργία και εμπλουτισμένα με ανωμαλίες αντίστοιχα, ακολουθούμενο από το στάδιο εντοπισμού ανωμαλιών. Στην περίπτωση του LOF, το αρχικό στάδιο εκπαίδευσης παραλείπεται, λόγω του τρόπου λειτουργίας του αλγορίθμου στον οποίο βασίζεται το μοντέλο. Ο εντοπισμός ανωμαλιών γίνεται σε επίπεδο υπηρεσιών, με τις τιμές των μετρικών να χαρακτηρίζονται ως κανονικές ή ανώμαλες.

BIRCH

Για την ανίχνευση ανωμαλιών χρησιμοποιώντας τον αλγόριθμο BRICH, ακολουθείται η παρακάτω διαδικασία. Αρχικά το μοντέλο εκπαιδεύεται πάνω σε δεδομένα από εκτέλεση της εφαρμογής χωρίς την προσθήκη ανωμαλιών. Στα δεδομένα εφαρμόζεται κλιμάκωση για την εξισορρόπηση της επίδρασης όλων των χαρακτηριστικών και την βελτίωση της επίδοσης του αλγορίθμου. Κατά την εκπαίδευση, το μοντέλο υπολογίζει το όριο για τον εντοπισμό ανωμαλιών με βάση τις αποστάσεις των σημείων από τα κέντρα των ομάδων που σχηματίζονται. Σημεία με απόσταση μεγαλύτερη από το όριο κρίνονται ανώμαλα. Στη συνέχεια ακολουθεί κλιμάκωση των δεδομένων και εκτέλεση προβλέψεων, σε δεδομένα με προσθήκη ανωμαλιών. Το μοντέλο επιστρέφει τις ετικέτες που παράγει για τα δεδομένα ελέγχου.

Local Outlier Factor

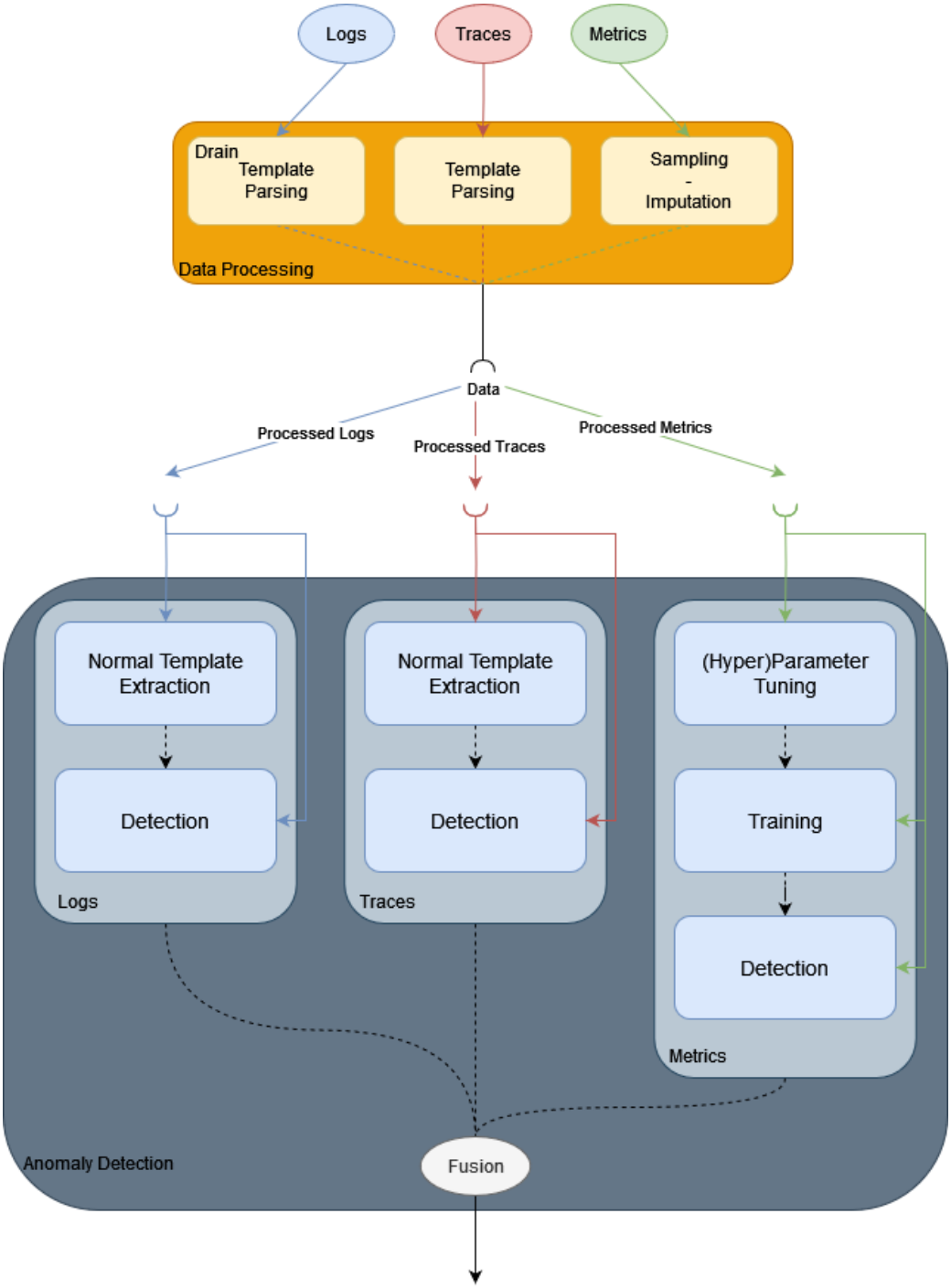
Σε αντίθεση με τον αλγόριθμο BIRCH, στην περίπτωση του LOF δεν απαιτείται εκπαίδευση σε δεδομένα καθαρά από ανωμαλίες. Αφού ο αλγόριθμος βασίζεται στην πυκνότητα των σημείων στο χώρο για τον καθορισμό των ανωμαλιών, η εκπαίδευση και ο έλεγχος γίνεται πάνω στο ίδιο σύνολο δεδομένων. Και σε αυτό το μοντέλο δίνεται επιλογή κλιμάκωσης των δεδομένων. Το μοντέλο επιστρέφει τις προβλέψεις που παρήγαγε σε μορφή ετικετών.

Isolation Forest

Στο μοντέλο Isolation Forest τόσο η εκπαίδευση όσο και η πρόβλεψη γίνονται σε δεδομένα με προσθήκη ανωμαλιών. Αντίθετα με το μοντέλο LOF όμως, η εκπαίδευση και ο έλεγχος δεν εκτελούνται στο ίδιο σύνολο δεδομένων. Αντίστοιχα με πριν, δίνεται η επιλογή για κλιμάκωση των δεδομένων. Το μοντέλο επιστρέφει τις ετικέτες που παράγονται κατά την πρόβλεψη.

3.5.3 Αρχιτεκτονική Συστήματος Εντοπισμού Ανωμαλιών

Η συνολική αρχιτεκτονική του συστήματος ανίχνευσης ανωμαλιών φαίνεται αναλυτικά στο Σχήμα 3.1.

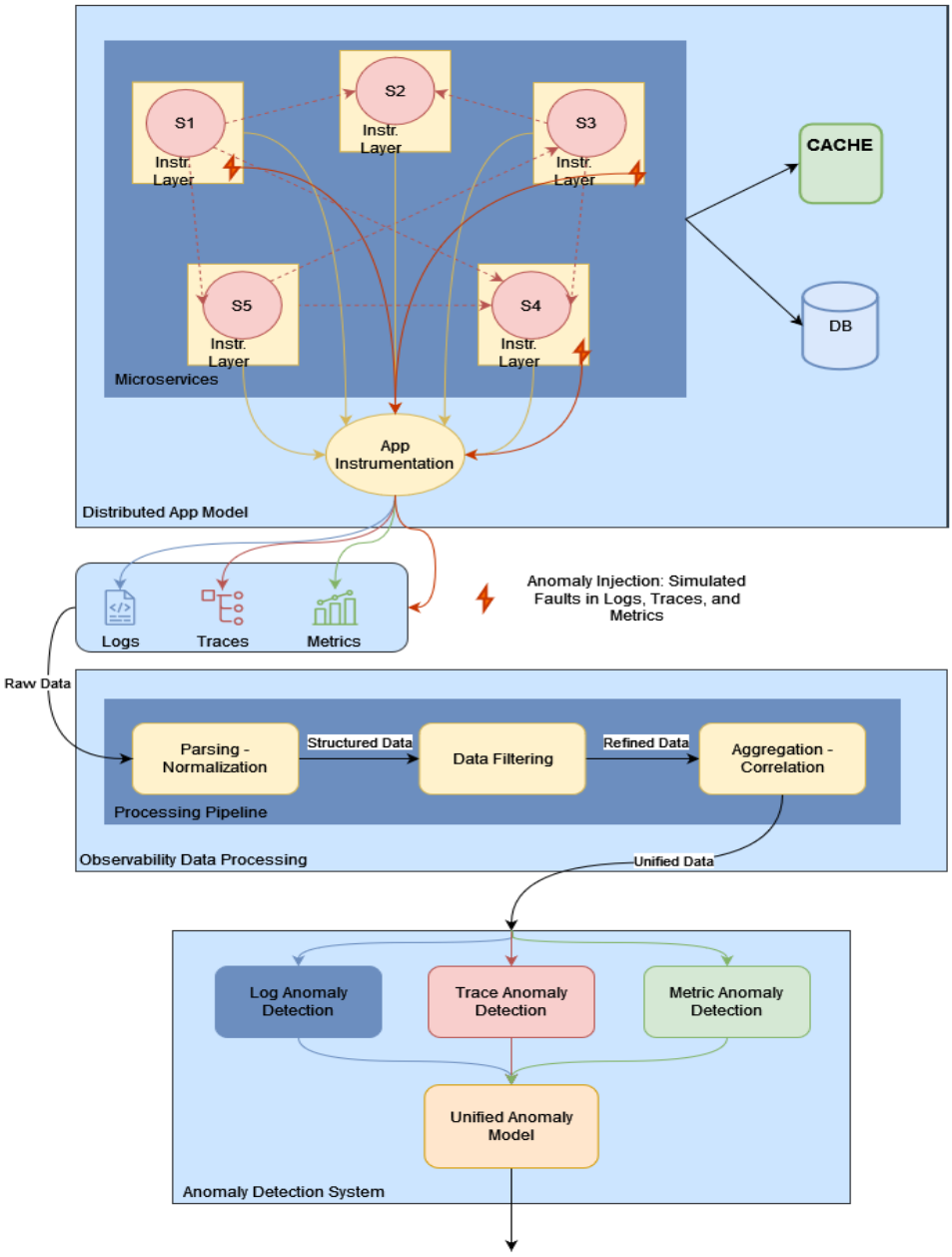


Σχήμα 3.1: Αρχιτεκτονική Συστήματος Ανίχνευσης Ανωμαλιών

3.6 Συγκεντρωτική Παρουσίαση Μεθοδολογίας

Η μεθοδολογία που παρουσιάζεται στην παρούσα εργασία ακολουθεί μία συστηματική προσέγγιση στην ανίχνευση ανωμαλιών σε ένα κατανεμημένο περιβάλλον μικροϋπηρεσιών. Η διαδικασία ξεκινά με μία κατάλληλα διαμορφωμένη κατανεμημένη εφαρμογή, από την οποία συλλέγονται δεδομένα παρατηρησιμότητας (αρχεία καταγραφής, ίχνη και μετρικές) από κάθε υπηρεσία. Τα δεδομένα αυτά υφίστανται κατάλληλη επεξεργασία, μέσω μιας ακολουθίας

σταδίων κανονικοποίησης, φιλτραρίσματος και συνάθροισης, ώστε να προετοιμαστούν για περαιτέρω ανάλυση. Για να αξιολογηθεί η ανθεκτικότητα του συστήματος ανίχνευσης ανωμαλιών, εισάγονται στο σύστημα διάφοροι τύποι συνθετικών ανωμαλιών, που μιμούνται σφάλματα του πραγματικού κόσμου, όπως αιχμές καθυστέρησης, σφάλματα αρχείων καταγραφής και υποβαθμίσεις επιδόσεων. Στα δεδομένα που προκύπτουν μετά την επεξεργασία, εφαρμόζονται εξειδικευμένες τεχνικές ανίχνευσης ανωμαλιών σε κάθε τύπο δεδομένων παρατηρησιμότητας. Τα αποτελέσματα από αυτές τις μεθόδους ανίχνευσης συνδυάζονται σε ένα ενοποιημένο μοντέλο, παρέχοντας μια ολοκληρωμένη εικόνα των τυχόν ανωμαλιών που έχουν εντοπιστεί. Το Σχήμα 3.2 παρέχει μια λεπτομερή επισκόπηση ολόκληρης αυτής της μεθοδολογίας, από τη συλλογή δεδομένων έως την ανίχνευση ανωμαλιών.



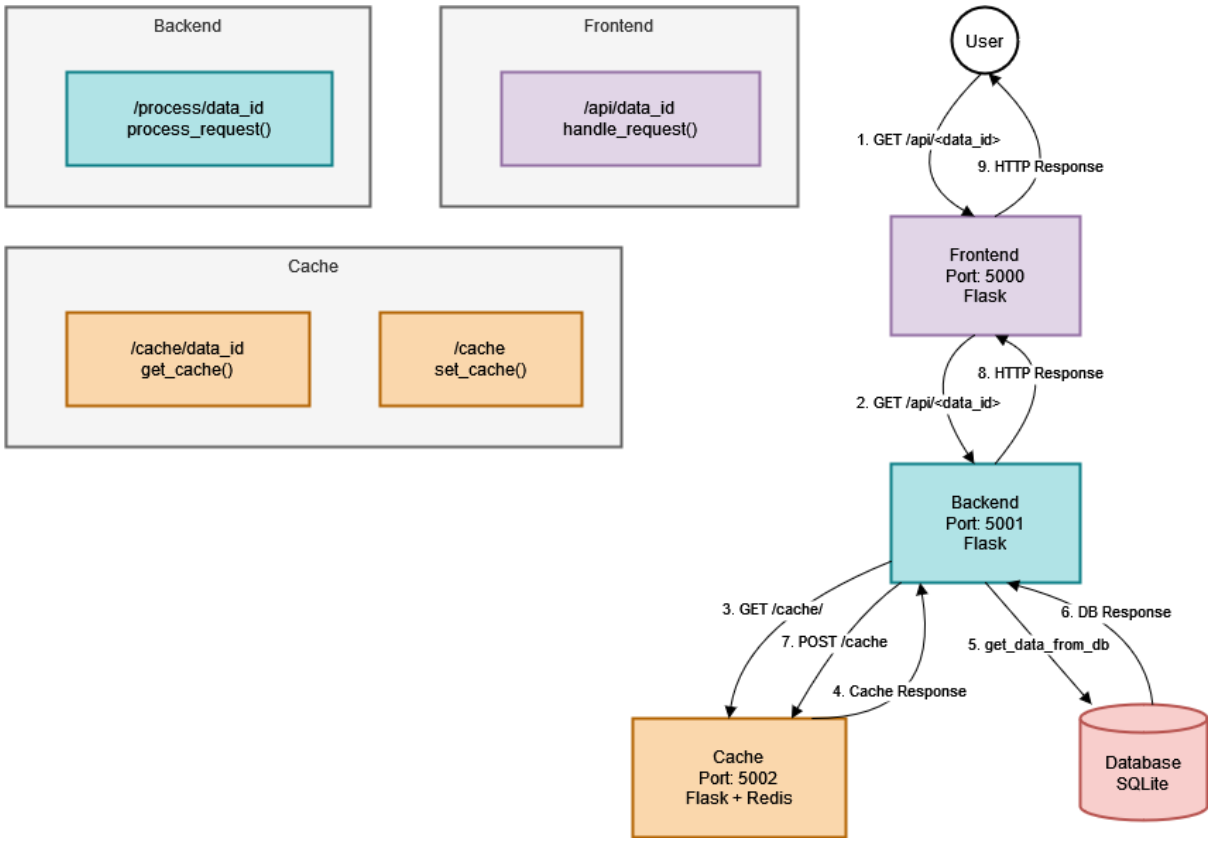
Σχήμα 3.2: Συγκεντρωτική Παρουσίαση Μεθοδολογίας

Κεφάλαιο 4 - Υλοποίηση και Πειραματική Αξιολόγηση

4.1 Υλοποίηση

4.1.1 Αρχιτεκτονική Εφαρμογής

Για την παραγωγή δεδομένων που θα χρησιμοποιηθούν στην εκπαίδευση των μοντέλων, αρχικά δημιουργείται μία διαδικτυακή εφαρμογή που προσομοιώνει μία κατανεμημένη εφαρμογή βασισμένη σε αρχιτεκτονική μικροϋπηρεσιών. Η εφαρμογή ακολουθεί την δομή που φαίνεται στο Σχήμα 4.1.



Σχήμα 4.1: Αναλυτική Τοπολογία Εφαρμογής

Πιο συγκεκριμένα, η εφαρμογή αποτελείται από τις ακόλουθες υπηρεσίες. Η υπηρεσία Frontend συνήθως αποτελεί τον κύριο τρόπο διεπαφής του χρήστη με το σύστημα. Είναι υπεύθυνη για την εμφάνιση των δεδομένων που λαμβάνονται από το backend και την υλοποίηση της λογικής που αφορά την εμπειρία του χρήστη. Στη συγκεκριμένη εφαρμογή, η υπηρεσία αυτή έχει πιο περιορισμένο ρόλο, προωθώντας τα αιτήματα των χρηστών στο backend και παρουσιάζοντας τα αποτελέσματα που αυτό επιστρέφει. Η υπηρεσία Backend αποτελεί το θεμέλιο για την επικοινωνία και την επεξεργασία δεδομένων μεταξύ των διαφόρων τμημάτων της εφαρμογής. Ειδικότερα, αποτελεί μεσάζοντα ανάμεσα στην υπηρεσία Frontend, τη βάση δεδομένων (Database) και την υπηρεσία κρυφής μνήμης (Cache). Η υπηρεσία της βάσης δεδομένων αποτελεί το βασικό σύστημα διαχείρισης και αποθήκευσης δεδομένων της εφαρμογής, αποθηκεύοντας δομημένες πληροφορίες που αξιοποιούνται από το Backend για

την επεξεργασία και την ανάκτηση δεδομένων. Τέλος, η υπηρεσία κρυφής μνήμης, χρησιμοποιείται για τη βελτίωση της απόδοσης της εφαρμογής αποθηκεύοντας συχνά χρησιμοποιούμενα δεδομένα σε προσωρινή μνήμη (συνήθως στη μνήμη RAM), ώστε να είναι γρηγορότερα διαθέσιμα όταν ζητούνται.

Η εφαρμογή που παρουσιάζεται υλοποιείται χρησιμοποιώντας προγράμματα βασισμένα στη γλώσσα προγραμματισμού Python [58]. Οι υπηρεσίες Frontend και Backend της εφαρμογής χρησιμοποιούν το web framework Flask [59]. Το Flask είναι ένα δημοφιλές micro web framework για την Python, σχεδιασμένο να είναι ελαφρύ, ευέλικτο και εύκολο στη χρήση για την ανάπτυξη web εφαρμογών και APIs. Ακολουθεί μια μινιμαλιστική προσέγγιση, δίνοντας στους προγραμματιστές τον έλεγχο για το πώς να δομήσουν τις εφαρμογές τους, παρέχοντας ταυτόχρονα τα απαραίτητα εργαλεία για δρομολόγηση, διαχείριση αιτημάτων και άλλα. Το Flask περιγράφεται συχνά ως "micro" framework επειδή περιλαμβάνει μόνο τα βασικά χαρακτηριστικά που χρειάζονται για να ξεκινήσει μια web εφαρμογή (δρομολόγηση, διαχείριση αιτημάτων κ.λπ.), χωρίς να επιβάλλει συγκεκριμένη δομή έργου ή να περιλαμβάνει πολλές προεγκατεστημένες βιβλιοθήκες.

Παρόλο που το Flask είναι ελαφρύ, μπορεί να επεκταθεί με τη χρήση διαφόρων extensions για τη διαχείριση βάσεων δεδομένων, την πιστοποίηση χρηστών, τις φόρμες κ.λπ.. Στην περίπτωση αυτής της εφαρμογής ενσωματώνεται η επέκταση Flask-SQLAlchemy [60] για ενσωμάτωση με την υπηρεσία βάσης δεδομένων. Για την υπηρεσία βάσης δεδομένων προτιμάται το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων SQLite [61]. Το SQLite είναι μια ελαφριά, χωρίς διακομιστή (serverless) και αυτοτελής μηχανή βάσης δεδομένων SQL, η οποία χρησιμοποιείται ευρέως σε εφαρμογές όπου η απλότητα και η αποδοτικότητα είναι κρίσιμες απαιτήσεις. Η βάση δεδομένων αποθηκεύεται σε ένα αρχείο, και η μηχανή λειτουργεί άμεσα μέσα από την εφαρμογή. Αυτό εξαλείφει την ανάγκη για επικοινωνία πελάτη-διακομιστή, καθιστώντας τη βάση πολύ γρήγορη για πολλές λειτουργίες. Η βάση δεδομένων που δημιουργείται αποτελεί μια ενσωματωμένη βάση δεδομένων, υπό την έννοια ότι είναι ενσωματωμένη και στενά συνδεδεμένη με την ίδια την εφαρμογή αντί να αποτελεί μία ξεχωριστή αυτόνομη διεργασία, όμως στο πλαίσιο της εφαρμογής αυτής μπορεί να θεωρηθεί ως μια συμβατική βάση με τη δική της διεργασία που επικοινωνεί με τις υπόλοιπες.

Η διεργασία που αναλαμβάνει την υλοποίηση της κρυφής μνήμης βασίζεται στο σύστημα αποθήκευσης Redis Cache [62]. Το Redis είναι ένα ταχύτατο σύστημα διαχείρισης βάσης δεδομένων που λειτουργεί στη μνήμη (in-memory), το οποίο χρησιμοποιείται κυρίως ως κρυφή μνήμη (cache) και ως σύστημα αποθήκευσης δεδομένων κλειδιού-τιμής (key-value store). Είναι γνωστό για την εξαιρετική του απόδοση και την υποστήριξη σύνθετων δομών δεδομένων, όπως λίστες, σύνολα, και πίνακες κατακερματισμού (hashes). Τα δεδομένα που αποθηκεύονται σε αυτή την cache δομούνται σε μορφή λεξικών με ζεύγη κλειδιών – τιμών και αποθηκεύονται στη μνήμη RAM για γρηγορότερη πρόσβαση.

4.1.2 Εργαλεία Παρατηρησιμότητας

Το σύστημα παρατηρησιμότητας που υλοποιείται παράλληλα με την εφαρμογή βασίζεται σε μία συλλογή διαφορετικών εργαλείων για τη συλλογή, την επεξεργασία και την παρουσίαση των δεδομένων που παράγονται μέσω της εφαρμογής. Για τα αρχεία καταγραφής αξιοποιείται το σύστημα επεξεργασίας και προώθησης Fluent-Bit [63]. Το σύστημα αυτό παρέχει δυνατότητες συλλογής δεδομένων αρχείων καταγραφής από πολλαπλές πηγές,

επεξεργασίας των δεδομένων αυτών με την εφαρμογή φίλτρων και προώθησής τους σε διάφορους προορισμούς. Το σύστημα αυτό είναι σχεδιασμένο για παρατηρησιμότητα μεγάλης κλίμακας σε δυναμικά και κατανεμημένα περιβάλλοντα και προτιμάται συχνά για χρήση σε περιβάλλοντα νέφους (cloud). Στην εφαρμογή που δημιουργείται, το προκαθορισμένο σύστημα διαχείρισης αρχείων καταγραφής που προσφέρει η Python αντικαθίσταται από έναν χειριστή, ειδικά προσαρμοσμένο για την προώθηση των εγγραφών αρχείων καταγραφής στο Fluent-Bit μέσω μηνυμάτων HTTP, καθώς αυτές προκύπτουν. Το Fluent Bit με τη σειρά του, συλλέγει τις εγγραφές και χωρίς να εφαρμόσει κάποια επιπλέον επεξεργασία, τις προωθεί στο επόμενο στάδιο που αποτελείται από το σύστημα συνάθροισης αρχείων καταγραφής Loki [64]. Το Loki είναι μια ανοιχτού κώδικα πλατφόρμα καταγραφής και παρακολούθησης που έχει σχεδιαστεί για να αποθηκεύει, να επεξεργάζεται και να αναζητά αρχεία καταγραφής (logs) από διάφορες πηγές. Είναι ιδιαίτερα προσανατολισμένο προς τη διαχείριση καταγραφών σε περιβάλλοντα νέφους για χρήση με κοντέινερ, προσφέροντας μια εύκολη και αποδοτική λύση για τη συλλογή και την ανάλυση των δεδομένων αρχείων καταγραφής. Το σύστημα αυτό είναι υπεύθυνο για την αποθήκευση, την επεξεργασία και την εκτέλεση ερωτημάτων (queries) πάνω στα αρχεία καταγραφής. Το Loki δεν παρέχει τη δυνατότητα εμφάνισης των εγγραφών που έχει αποθηκεύσει ούτε και την απευθείας εκτέλεση ερωτημάτων. Το ρόλο αυτό αναλαμβάνει η υπηρεσία Grafana [65], μια πλατφόρμα που προσφέρει πολλαπλές λειτουργίες παρατηρησιμότητας. Μέσω της πλατφόρμας αυτής οι χρήστες μπορούν να εκτελούν ερωτήματα, να οπτικοποιούν, να οργανώνουν και να μελετούν δεδομένα όπως μετρικές και αρχεία καταγραφής που συλλέγονται από πολλαπλές πηγές. Συνδυάζοντας τις υπηρεσίες αυτές δημιουργείται ένα ολοκληρωμένο σύστημα διαχείρισης αρχείων καταγραφής.

Η συλλογή των δεδομένων που αποτελούν τα ίχνη της εκτέλεσης της εφαρμογής πραγματοποιείται από το ανοιχτού κώδικα framework παρατηρησιμότητας Open Telemetry [66]. Με μια μεγάλη συλλογή διεπαφών προγραμματισμού εφαρμογών (API) και εργαλείων ανάπτυξης λογισμικού (SDK) παρέχει τη δυνατότητα δημιουργίας ενός συστήματος παρατηρησιμότητας ανεξαρτήτως των εργαλείων και της γλώσσας προγραμματισμού που χρησιμοποιούνται. Τα δεδομένα που συλλέγονται, αποτελούμενα από spans κατανεμημένα σε ίχνη ανάλογα με το αίτημα που εξυπηρετούν, προωθούνται στην κατάλληλη διεύθυνση για τη συλλογή τους από το κατανεμημένο σύστημα ιχνηλάτησης Zipkin [67]. Το Zipkin είναι ένα σύστημα κατανεμημένης ιχνηλάτησης ανοιχτού κώδικα που έχει σχεδιαστεί για να βοηθά στη συλλογή και ανάλυση δεδομένων χρονισμού από εφαρμογές που βασίζονται σε μικροϋπηρεσίες. Βοηθά τους προγραμματιστές να κατανοήσουν τις επιδόσεις των εφαρμογών τους, παρέχοντας πληροφορίες σχετικά με την καθυστέρηση και τη ροή των αιτημάτων καθώς αυτά διαδίδονται μέσω διαφόρων υπηρεσιών. Αυτό είναι ιδιαίτερα χρήσιμο σε πολύπλοκες αρχιτεκτονικές, όπου η παρακολούθηση της απόδοσης μεμονωμένων στοιχείων μπορεί να είναι δύσκολη. Το σύστημα αυτό δίνει στους χρήστες τη δυνατότητα να εκτελούν ερωτήματα, να εμφανίζουν και να μελετούν τα αποθηκευμένα δεδομένα ιχνηλάτησης.

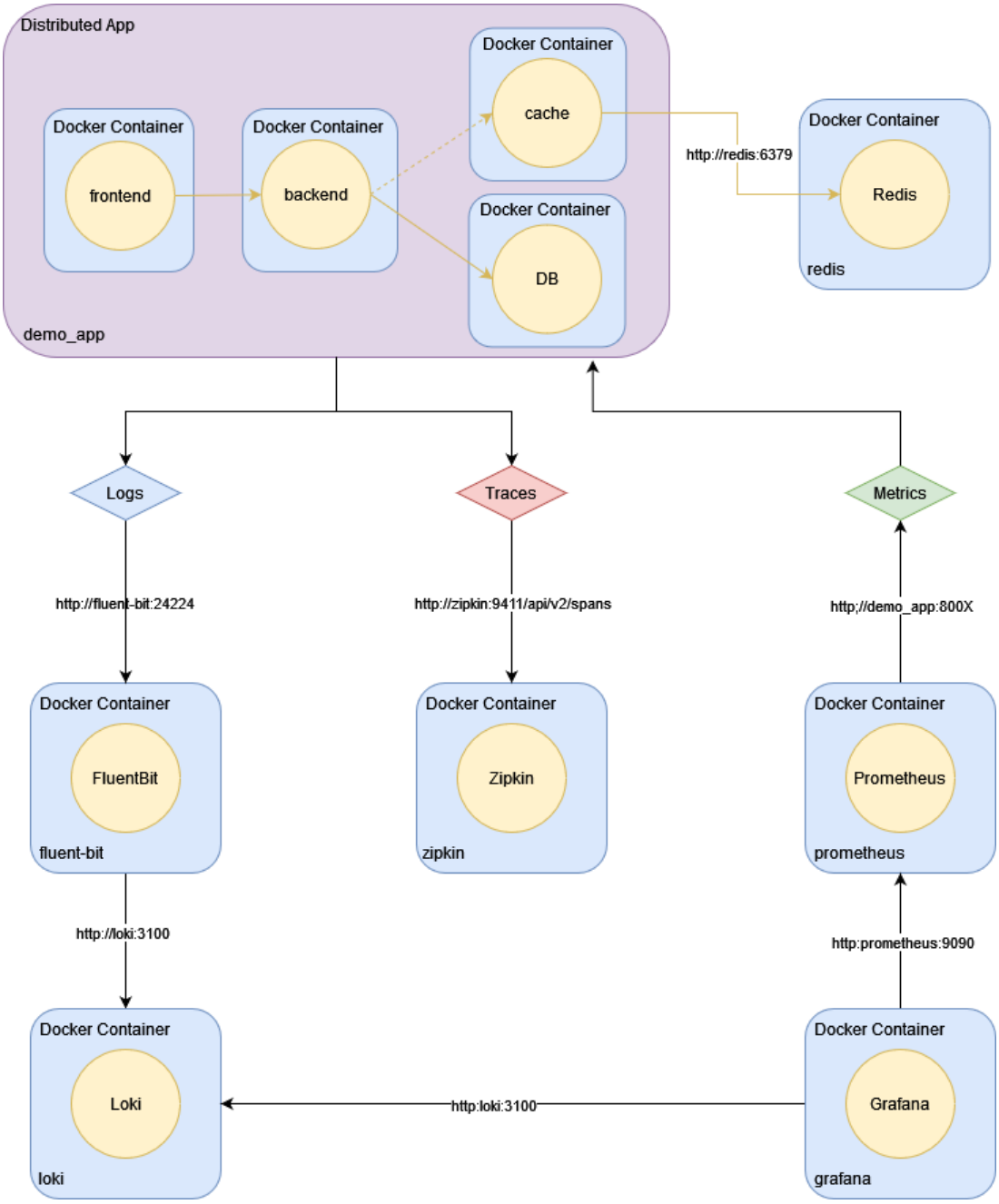
Τη συλλογή, αποθήκευση και εμφάνιση των μετρικών που εξάγονται κατά την εκτέλεση της εφαρμογής αναλαμβάνει εξ ολοκλήρου η υπηρεσία Prometheus [68]. Το Prometheus αποτελεί μία συλλογή εργαλείων ανοιχτού κώδικα για την παρακολούθηση μετρικών που χρησιμοποιείται ευρέως για την καταγραφή και επεξεργασία μετρικών σε εφαρμογές cloud-native και αρχιτεκτονικές μικροϋπηρεσιών. Το Prometheus έχει γίνει θεμελιώδες συστατικό του Cloud Native Computing Foundation (CNCF) και είναι γνωστό για

τις ισχυρές δυνατότητες ερωτημάτων και την ευέλικτη αρχιτεκτονική του. Το σύστημα αυτό μπορεί να συνδεθεί άμεσα με την πλατφόρμα Grafana για το συνδυασμό και την καλύτερη παρακολούθηση δεδομένων παρατηρησιμότητας.

4.1.3 Εκτέλεση (Deployment) Εφαρμογής & Εργαλείων Παρατηρησιμότητας

Η εφαρμογή αλλά και τα εργαλεία που αναφέρθηκαν εκτελούνται σε containerized περιβάλλον χρησιμοποιώντας την πλατφόρμα Docker [69]. Το Docker είναι μια πλατφόρμα ανοικτού κώδικα που έχει σχεδιαστεί για την αυτοματοποίηση της ανάπτυξης, της κλιμάκωσης και της διαχείρισης εφαρμογών με τη χρήση κοντέινερ (containers). Επιτρέπει στους προγραμματιστές να συμπιέζουν τις εφαρμογές και τις εξαρτήσεις τους σε φορητά πακέτα, εξασφαλίζοντας τη συνοχή σε διάφορα περιβάλλοντα, από την ανάπτυξη έως την παραγωγή. Τα κοντέινερ είναι ελαφριά σε απαιτήσεις, μοιράζονται τον πυρήνα (kernel) του κεντρικού λειτουργικού συστήματος και είναι απομονωμένα μεταξύ τους, επιτρέποντας την αποτελεσματική χρήση των πόρων. Το Docker έχει γίνει μια θεμελιώδης τεχνολογία στις αρχιτεκτονικές cloud-native και μικροϋπηρεσιών, παρέχοντας μια βελτιωμένη προσέγγιση για την ανάπτυξη και την ανάπτυξη εφαρμογών. Η πλατφόρμα αυτή είναι κατάλληλη για αρχιτεκτονικές μικροϋπηρεσιών, όπου οι εφαρμογές αναλύονται σε μικρότερες, χαλαρά συνδεδεμένες υπηρεσίες. Κάθε υπηρεσία μπορεί να συσκευαστεί και να αναπτυχθεί ανεξάρτητα στο δικό της κοντέινερ, διευκολύνοντας την επεκτασιμότητα και τη συντηρησιμότητα. Η χρήση της πλατφόρμας Docker εξυπηρετεί, πέρα από το σκοπό της προσομοίωσης της λειτουργίας μιας κατανεμημένης εφαρμογής και την πιθανή εγκατάσταση και εκτέλεση της εφαρμογής σε πραγματικό κατανεμημένο περιβάλλον, ενδεχομένως με τη χρήση Kubernetes. Όλες οι υπηρεσίες της εφαρμογής συμπεριλαμβάνονται σε ένα ενιαίο κοντέινερ, κάτι το οποίο μπορεί εύκολα να αλλάξει καθώς η πολυπλοκότητα και οι λειτουργίες που επιτελεί κάθε μία από αυτές αυξάνεται. Οι υπηρεσίες που εξυπηρετούν την παρατηρησιμότητα της εφαρμογής καταλαμβάνουν ένα κοντέινερ η κάθε μία. Για λόγους απλότητας, όλα τα κοντέινερ βρίσκονται σε κοινό δίκτυο.

Η συνολική αρχιτεκτονική της εφαρμογής σε συνδυασμό με τα εργαλεία παρατηρησιμότητας που αξιοποιούνται, καθώς και οι αλληλεπιδράσεις μεταξύ τους χρησιμοποιώντας τα κατάλληλα endpoints φαίνεται αναλυτικά στο Σχήμα 4.2.



Σχήμα 4.2: Αρχιτεκτονική Συστήματος Παρατηρησιμότητας

Όπως φαίνεται, οι εγγραφές αρχείων καταγραφής που προκύπτουν από τις υπηρεσίες της εφαρμογής συλλέγονται από κατάλληλους χειριστές και προωθούνται στο Fluent-Bit και στη συνέχεια στο Loki για αποθήκευση. Αντίστοιχα για τα ίχνη, τα δεδομένα σε μορφή spans προωθούνται από χειριστές που συμπεριλαμβάνονται στον κώδικα της εφαρμογής, συλλέγονται και αποθηκεύονται στο Zipkin. Η συλλογή των δεδομένων μετρικών πραγματοποιείται από το σύστημα Prometheus, το οποίο αναζητά τα κατάλληλα δεδομένα σε προκαθορισμένες διευθύνσεις για κάθε υπηρεσία της εφαρμογής. Τα δεδομένα από τα συστήματα Loki και Prometheus συλλέγονται στην πλατφόρμα Grafana για μία συνολική εποπτεία.

4.1.4 Προσομοίωση Ανωμαλιών

Για την παραγωγή και την εξαγωγή δεδομένων από την εφαρμογή προσομοιώνεται η κίνηση πολλαπλών χρηστών, χρησιμοποιώντας τη συνάρτηση ThreadPoolExecutor της βιβλιοθήκης concurrent της Python. Ο ThreadPoolExecutor είναι μια διεπαφή υψηλού επιπέδου που χρησιμοποιείται για την ασύγχρονη εκτέλεση εργασιών σε πολλαπλά νήματα, διευκολύνοντας την ταυτόχρονη εκτέλεση εργασιών που δεσμεύονται από I/O ή CPU, χωρίς να χρειάζεται να διαχειρίζεστε χειροκίνητα τα νήματα. Στο πλαίσιο της κατανεμημένης εφαρμογής που σχεδιάζεται, ο ThreadPoolExecutor αξιοποιείται για την αποστολή ταυτόχρονων αιτημάτων προς την εφαρμογή. Κατά την προσομοίωση της κίνησης χρηστών μέσω της διεπαφής αυτής καθορίζεται ο αριθμός των συνολικών επιθυμητών αιτημάτων προς την εφαρμογή, προσδιορίζεται η κατάλληλη διεύθυνση για την προγραμματιστική διεπαφή και στη συνέχεια τα αιτήματα εκτελούνται με προσαρμόσιμο βαθμό παραλληλίας σε περιόδους χαμηλής και υψηλής χρήσης. Κατά την εκτέλεση της εφαρμογής μέσω του προγράμματος προσομοίωσης δεν παρατηρούνται σφάλματα ή αποκλίσεις από την κανονική λειτουργία. Για τον εμπλουτισμό των σφαλμάτων και την προσομοίωση ενός πιο ρεαλιστικού σεναρίου, προστίθενται εσκεμμένα ανωμαλίες στην εκτέλεση του κώδικα της εφαρμογής (anomaly injection). Συγκεκριμένα, έχει προστεθεί η δυνατότητα για καθορισμό των παραγόμενων ανωμαλιών μέσω παραμέτρων στη διεύθυνση URL κατά την αρχική αποστολή των αιτήσεων στην εφαρμογή. Οι ανωμαλίες που μπορούν να παραχθούν είναι οι ακόλουθες: παύσεις του προγράμματος που αυξάνουν το συνολικό χρόνο εξυπηρέτησης του αιτήματος και προσομοιώνουν αυξημένο υπολογιστικό φόρτο, εξάντληση του αριθμού διαθέσιμων συνδέσεων στη βάση δεδομένων και εξαιρέσεις κατά την εκτέλεση του κώδικα. Οι ανωμαλίες αυτές εμφανίζονται κατά την εκτέλεση της εφαρμογής, με την προϋπόθεση ότι η απαραίτητη παράμετρος έχει προσδιοριστεί, με συχνότητα που μπορεί να παραμετροποιηθεί.

4.1.5 Επεξεργασία Δεδομένων Παρατηρησιμότητας

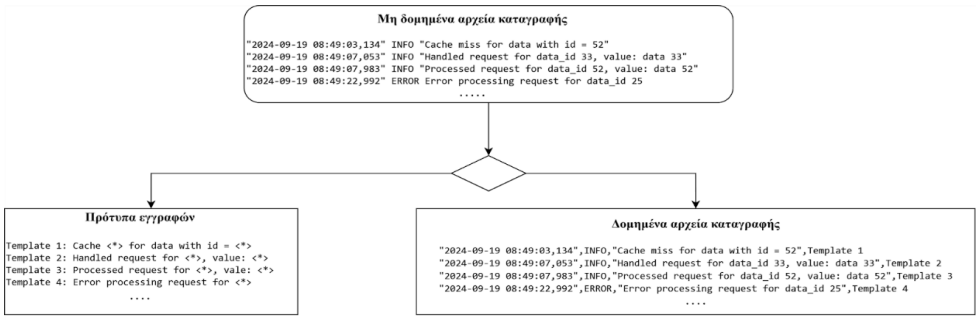
Συλλογή Δεδομένων

Έχοντας εκτελέσει την εφαρμογή και παράγει δεδομένα, απαραίτητη είναι μία μέθοδος εξαγωγής των δεδομένων αυτών από τα εργαλεία παρατηρησιμότητας που τα έχουν συλλέξει. Όλα τα εργαλεία παρέχουν προγραμματιστικές διεπαφές για το σκοπό αυτό, επιστρέφοντας τα δεδομένα ως απαντήσεις σε ερωτήματα που θέτει ο χρήστης. Σε κάθε ερώτημα μπορεί να προσδιοριστεί ένα πλήθος παραμέτρων όπως το χρονικό διάστημα κατά το οποίο καταγράφησαν τα δεδομένα, το πλήθος των εγγραφών προς αποστολή αλλά και τη συχνότητα μεταξύ διαδοχικών εγγραφών. Η απάντηση στο χρήστη έρχεται σε μορφή εγγράφου JSON, το οποίο πρέπει να υποστεί κατάλληλη επεξεργασία για την σωστή δόμηση των δεδομένων που περιέχει. Η πληροφορία για τα ίχνη εκτέλεσης της εφαρμογής είναι δομημένη σε spans, με επιπλέον λεπτομέρειες για το χρόνο καταγραφής του κάθε span, το ίχνος στο οποίο ανήκει, τον πρόγονο του στη σειρά κλήσεων, το όνομά του, το συνολικό χρόνο εκτέλεσής του αλλά και την υπηρεσία από την οποία προέκυψε. Από τα ίδια τα spans εξάγεται και η μετρική του χρόνου απόκρισης της κάθε υπηρεσίας στο πλαίσιο ενός ίχνους. Από τα δεδομένα μετρικών που συλλέγονται εξάγονται μετρικές, όπως για παράδειγμα ο αριθμός συνδέσεων της βάσης δεδομένων και ο αριθμός hits και misses στην κρυφή μνήμη. Τέλος, συλλέγονται τα δεδομένα που αφορούν τα αρχεία καταγραφής. Κάθε εγγραφή που επιστρέφεται περιέχει πληροφορία

για το χρόνο καταγραφής της, την υπηρεσία από την οποία προήλθε, το επίπεδο σοβαρότητάς της αλλά και το μήνυμα που τη συνοδεύει, σε μορφή κειμένου. Για την συσχέτιση των εγγραφών αυτών με τα ίχνη που παράγονται παράλληλα, κάθε εγγραφή περιέχει και ένα αναγνωριστικό για το ίχνος και το span στα οποία ανήκει. Όλα τα δεδομένα που συλλέγονται, αφού υποστούν την κατάλληλη επεξεργασία αποθηκεύονται σε μορφή αρχείων CSV για την ευκολότερη αξιοποίησή τους.

Αρχεία Καταγραφής

Τα δεδομένα από τα αρχεία καταγραφής και τα ίχνη είναι κατά κύριο λόγο αδόμητα και πρέπει να υποστούν κατάλληλη επεξεργασία για να μπορέσουν να αξιοποιηθούν κατάλληλα στην ανίχνευση ανωμαλιών. Η μετατροπή των αδόμητων δεδομένων σε δομημένα είναι απαραίτητη, αφού καθιστά ευκολότερη τη σύγκριση ακολουθιών και την ανίχνευση αποκλίσεων. Ο αλγόριθμος ανάλυσης των αρχείων καταγραφής επεξεργάζεται τις εγγραφές που παρέχονται ως είσοδος και επιστρέφει με τη μορφή αρχείων CSV τα δομημένα αρχεία καταγραφής αλλά και τα πρότυπα που προκύπτουν κατά την επεξεργασία (Σχήμα 4.3).



Σχήμα 4.3: Διαδικασία Ανάλυσης Αρχείων Καταγραφής

Ίχνη

Τα ίχνη που παράγονται κατά την εκτέλεση της εφαρμογής, αν και πιο δομημένα εκ φύσεως από τα αρχεία καταγραφής, απαιτούν μία επιπλέον επεξεργασία για την κατασκευή προτύπων. Κάθε διαφορετική εκτέλεση της εφαρμογής μπορεί να προκαλέσει τη δημιουργία ιχνών που ακολουθούν διαφορετικά μονοπάτια κλήσης των υπηρεσιών που συμμετέχουν στην εξυπηρέτηση ενός αιτήματος. Μία συνεπής μέθοδος αναπαράστασης των διαφορετικών αλληλουχιών υπηρεσιών που εμφανίζονται στα spans ενός ίχνους είναι απαραίτητη για την ευκολότερη ανίχνευση ανωμαλιών, όπως αναφέρθηκε και παραπάνω. Η αναπαράσταση αυτή βασίζεται στη δημιουργία δενδρικών δομών με βάση τα ίχνη. Πιο συγκεκριμένα, τα spans ενός ίχνους οργανώνονται σε δέντρα, με κάθε μοναδικό δέντρο να αποθηκεύεται ως πρότυπο.

Μετρικές

Οι μετρικές που συλλέγονται απαιτούν τη λιγότερη επεξεργασία, καθώς βρίσκονται ήδη σε δομημένη μορφή πολύ κοντά στην επιθυμητή. Όπως αναφέρθηκε και προηγουμένως, τα δεδομένα χρόνου απόκρισης για τις υπηρεσίες της εφαρμογής εξάγονται από τα spans που περιλαμβάνονται στα ίχνη εκτέλεσης. Κάθε span όμως μπορεί να διαφέρει στις υπηρεσίες που

περιλαμβάνει και άρα δεν περιέχει όλους τους χρόνους απόκρισης εκείνης της χρονικής στιγμής. Το πρόβλημα αυτό διορθώνεται συμπληρώνοντας με μηδενικά τους χρόνους για τις υπηρεσίες που δεν εμφανίζονται σε κάθε εγγραφή. Με τον τρόπο αυτό όλες οι υπηρεσίες αποκτούν τον ίδιο αριθμό εγγραφών και τα δεδομένα αναπαρίστανται με μεγαλύτερη ακρίβεια. Όσον αφορά τη μετρική που αναπαριστά τον αριθμό συνδέσεων στη βάση δεδομένων, ένα πρόβλημα αποτελεί η συχνότητα με την οποία συλλέγεται από τις υπηρεσίες παρατηρησιμότητας. Με τις τιμές του μετρητή να ανανεώνονται σε σταθερά χρονικά διαστήματα, τα οποία μπορούν να προσαρμοστούν στην υπηρεσία συλλογής, σπάνια οι χρονικές σημάνσεις τους ταυτίζονται με αυτές των εγγραφών για το χρόνο απόκρισης. Για την επίλυση του προβλήματος αυτού απαιτείται ένα είδος αναδειγματοληψίας (upsampling), αφού οι τιμές του μετρητή ενεργών συνδέσεων στη βάση δεδομένων είναι λιγότερες σε πλήθος. Η δειγματοληψία αυτή γίνεται χρησιμοποιώντας μια τεχνική κυλιόμενου παραθύρου. Το παράθυρο αυτό έχει μεταβλητό μήκος με έκταση μερικών δευτερολέπτων και τοποθετείται με το κέντρο του πάνω στη χρονική σήμανση κάθε εγγραφής χρόνου απόκρισης. Χρησιμοποιώντας τη μέγιστη τιμή του αριθμού ενεργών συνδέσεων μέσα στο παράθυρο αυτό συμπληρώνονται οι τιμές για όλες τις χρονικές σημάνσεις που λείπουν.

4.1.6 Αλγόριθμοι Εντοπισμού Ανωμαλιών

Εντοπισμός Ανωμαλιών σε Αρχεία Καταγραφής

Όπως αναφέρθηκε προηγουμένως, για τον εντοπισμό ανωμαλιών σε εγγραφές αρχείων καταγραφής αξιοποιούνται τα δεδομένα που συλλέγονται από την εφαρμογή σε δομημένη όμως μορφή. Η δομημένη αυτή μορφή προκύπτει από εφαρμογή του αλγορίθμου Drain στα δεδομένα. Ο αλγόριθμος Drain αρχικοποιείται με τις κατάλληλες παραμέτρους που προσδιορίζουν την αναμενόμενη μορφή των εγγραφών (log_format), το συνολικό βάθος των δέντρων που θα δημιουργηθούν (depth) και το όριο ομοιότητας για την κατηγοριοποίηση των εγγραφών σε πρότυπα (st). Για τα αρχεία καταγραφής που παράγονται από την εφαρμογή οι ιδανικές τιμές των παραμέτρων αυτών παρουσιάζονται στον Πίνακα 4.1.

Πίνακας 4.1: Παράμετροι Αλγορίθμου Drain

Παράμετρος	**Τιμή**
log_format	'<Timestamp> - <Service> - <Level> - <Content> - <TraceId> - <SpanId>'
st	0.5
depth	4

Ο αλγόριθμος στη συνέχεια τροποποιείται κατάλληλα, ώστε να χρησιμοποιεί ως είσοδο ένα προσωρινό αρχείο με εγγραφές της αναμενόμενης μορφής, να εξάγει τα αποτελέσματα και να τα αποθηκεύει με τη μορφή αρχείων CSV στη ζητούμενη τοποθεσία. Ο αλγόριθμος επιστρέφει τα αρχεία καταγραφής σε δομημένη μορφή αλλά και τα πρότυπα που προκύπτουν κατά την επεξεργασία με τη μορφή Pandas DataFrames. Τα DataFrames είναι μια ισχυρή δομή

δεδομένων στην Python, η οποία παρέχεται από τη βιβλιοθήκη Pandas και έχει σχεδιαστεί για τον αποτελεσματικό χειρισμό και την ανάλυση δομημένων δεδομένων. Ένα DataFrame είναι μια δισδιάστατη, πινακοποιημένη δομή δεδομένων με επισημασμένους άξονες (γραμμές και στήλες), παρόμοια με ένα λογιστικό φύλλο ή έναν πίνακα SQL. Επιτρέπει τον εύκολο χειρισμό δεδομένων, όπως φιλτράρισμα, μετασχηματισμό, συνάθροιση και ανάλυση μεγάλων συνόλων δεδομένων.

Το σύνολο δεδομένων εκπαίδευσης αποτελείται από αρχεία καταγραφής που προέκυψαν από εκτελέσεις της εφαρμογής σε κανονική λειτουργία, χωρίς την προσθήκη ανωμαλιών. Κατά την εκπαίδευση λοιπόν δημιουργείται ένα σύνολο προτύπων που αντικατοπτρίζει την ορθή λειτουργία της εφαρμογής. Στο στάδιο του εντοπισμού ανωμαλιών σε κανονικό χρόνο, οι εισερχόμενες εγγραφές υφίστανται την κατάλληλη επεξεργασία για την εξαγωγή του προτύπου στο οποίο ανήκουν και εν συνεχεία συγκρίνονται με τα γνωστά πρότυπα ορθής λειτουργίας. Σε περίπτωση που το πρότυπο μιας εγγραφής δεν ταιριάζει με κανένα από τα γνωστά πρότυπα, το σύστημα το επισημαίνει ως ανώμαλο.

Εντοπισμός Ανωμαλιών σε Ίχνη Εκτέλεσης

Αντίστοιχη διαδικασία ακολουθείται και για τα ίχνη εκτέλεσης, με διαφορετικό όμως στάδιο αρχικής επεξεργασίας. Είναι γνωστό ότι τα ίχνη σχηματίζουν δενδρικές δομές αποτελούμενες από τα spans που τους αντιστοιχούν. Ο αλγόριθμος εξαγωγής προτύπων λοιπόν ξεκινά αναπαριστώντας το κάθε span ως μία ξεχωριστή δομή που θα αποτελέσει έναν κόμβο στο δέντρο του ίχνους στο οποίο ανήκει. Η δομή αυτή περιέχει όλες τις πληροφορίες που υπάρχουν με τη μορφή κειμένου στα δεδομένα παρατηρησιμότητας που έχουν εξαχθεί από την εφαρμογή. Πιο συγκεκριμένα, περιλαμβάνει πληροφορία για τη χρονική σήμανση του span, το αναγνωριστικό του ίχνους στο οποίο ανήκει, ένα αναγνωριστικό για τον άμεσο πρόγονό του στη δενδρική δομή, ένα αναγνωριστικό για τον εαυτό του, το όνομά του, τη χρονική διάρκεια εκτέλεσής του, την υπηρεσία της εφαρμογής από την οποία προέκυψε αλλά και μία λίστα των απογόνων του στη δενδρική δομή. Όλα τα ίχνη που περιέχονται στα δεδομένα οργανώνονται σε δενδρικές δομές, με τα spans χωρίς έγκυρο αναγνωριστικό προγόνου να τίθενται ως τη ρίζα του εκάστοτε δέντρου. Με τα δέντρα να έχουν σχηματιστεί σωστά, τα πρότυπα των ιχνών προκύπτουν μέσω μιας αναδρομικής preorder διάσχισης του δέντρου, με τον κάθε κόμβο να επεξεργάζεται πλήρως πριν από την αντίστοιχη επεξεργασία των απογόνων του. Με τον τρόπο αυτό, κάθε δέντρο μετασχηματίζεται σε μία ακολουθία εμφανίσεων από spans, με τις ακολουθίες αυτές να χρησιμοποιούνται ως πρότυπα-κλειδιά. Ακολουθώντας τη διαδικασία αυτή, κατά τη διάρκεια της φάσης εκπαίδευσης, χρησιμοποιώντας δεδομένα από εκτελέσεις της εφαρμογής χωρίς την προσθήκη ανωμαλιών θεσπίζεται το σύνολο προτύπων ορθής λειτουργίας για τα ίχνη εκτέλεσης. Το σύνολο αυτό θα χρησιμοποιηθεί κατά την ανίχνευση ανωμαλιών σε πραγματικό χρόνο. Τα εισερχόμενα δεδομένα μετασχηματίζονται σε ακολουθίες spans όπως παραπάνω και τα πρότυπα που εμφανίζονται συγκρίνονται με το σύνολο προτύπων ορθής λειτουργίας. Σε περίπτωση εμφάνισης άγνωστου προτύπου, το αντίστοιχο ίχνος δηλώνεται ως ανώμαλο.

Εντοπισμός Ανωμαλιών σε Μετρικές

Τα δεδομένα μετρικών που εξάγονται από την εφαρμογή σχηματίζουν χρονοσειρές. Οι χρονοσειρές αυτές δίνονται ως είσοδοι για την εκπαίδευση τριών μοντέλων μηχανικής μάθησης. Τα μοντέλα αυτά βασίζονται στους αλγορίθμους BIRCH, LOF και iForest και

συγκεκριμένα στην υλοποίηση που παρέχεται από τη βιβλιοθήκη Scikit-Learn [70] της Python. Το Scikit-learn είναι μια δημοφιλής βιβλιοθήκη μηχανικής μάθησης ανοικτού κώδικα σε Python. Παρέχει απλά και αποτελεσματικά εργαλεία για την εξόρυξη δεδομένων, την ανάλυση δεδομένων και τη μηχανική μάθηση. Η βιβλιοθήκη Scikit-learn είναι βασισμένη πάνω σε άλλες βιβλιοθήκες όπως οι NumPy [71], SciPy [72] και Matplotlib [73] και είναι γνωστή για το συνεπές API, την ευκολία χρήσης και το ισχυρό σύνολο αλγορίθμων που παρέχει.

Το μοντέλο birch βασίζεται στην υλοποίηση Birch [74] που παρέχει η βιβλιοθήκη αλγορίθμων ομαδοποίησης του sklearn, γύρω από την οποία σχηματίζεται ένας ειδικά τροποποιημένος εκτιμητής με την ονομασία BinaryBirch. Ο εκτιμητής αυτός παρέχει το ίδιο σύνολο υπερπαραμέτρων με τον Birch(), προσφέροντας λειτουργίες εκπαίδευσης και πρόγνωσης όπως συνήθως, εμπλουτίζεται όμως με μία επιπλέον λειτουργία αυτόματου καθορισμού του ορίου ανίχνευσης ανωμαλιών με βάση την παράμετρο anomaly_percentile που παρέχεται κατά την αρχικοποίησή του. Κατά τη διαδικασία εκπαίδευσης του μοντέλου χρησιμοποιούνται δεδομένα από εκτελέσεις της εφαρμογής που αντικατοπτρίζουν την κανονική λειτουργία. Τα σημεία που αντιστοιχούν στα δεδομένα αυτά χωρίζονται σε ομάδες στο χώρο. Με βάση τις ομάδες αυτές αλλά και την παράμετρο anomaly_percentile καθορίζεται αυτόματα το όριο για την ανίχνευση ανωμαλιών στο επόμενο στάδιο λειτουργίας του μοντέλου. Πιο συγκεκριμένα, το όριο αυτό ορίζεται ως η απόσταση ενός σημείου από το κέντρο της κοντινότερης ομάδας. Η επιλογή του σημείου γίνεται με βάση την ποσοστιαία θέση του στο σύνολο δεδομένων. Με άλλα λόγια, οι τιμές της παραμέτρου anomaly_percentile εκφράζουν ποσοστιαία θέση της απόστασης που θα επιλεγεί ως όριο. Το στάδιο ανίχνευσης ανωμαλιών εφαρμόζεται σε δεδομένα με προσθήκη ανωμαλιών, για τα οποία και επιστρέφονται οι ετικέτες που παράγονται.

Το μοντέλο local_outlier_factor βασίζεται στον αλγόριθμο LocalOutlierFactor [75] της βιβλιοθήκης κοντινότερων γειτόνων του sklearn. Σε αντίθεση με το μοντέλο birch, στην περίπτωση του lof δεν απαιτείται εκπαίδευση σε δεδομένα καθαρά από ανωμαλίες. Αφού ο αλγόριθμος βασίζεται στην πυκνότητα των σημείων στο χώρο για τον καθορισμό των ανωμαλιών, η εκπαίδευση και ο έλεγχος γίνεται πάνω στο ίδιο σύνολο δεδομένων. Το μοντέλο επιστρέφει τις προβλέψεις που παρήγαγε σε μορφή ετικετών.

Το μοντέλο isolation_forest βασίζεται στην υλοποίηση IsolationForest [76] που παρέχει το sklearn ως έναν από τους αλγορίθμους ανίχνευσης ανωμαλιών με βάση τα σύνολα (ensemble-based). Στο μοντέλο isolation_forest τόσο η εκπαίδευση όσο και η πρόβλεψη γίνονται σε δεδομένα με προσθήκη ανωμαλιών. Αντίθετα με το μοντέλο lof όμως, η εκπαίδευση και ο έλεγχος δεν εκτελούνται στο ίδιο σύνολο δεδομένων. Κατά την εκπαίδευσή του, το μοντέλο εκπαιδεύει το σύνολο δέντρων του με δεδομένα που περιέχουν ανωμαλίες, μαθαίνοντας να τις διαχωρίζει από τα φυσιολογικά δεδομένα. Επομένως η ανίχνευση πρέπει να γίνει σε ξεχωριστό σύνολο δεδομένων με δυνατότητα ανίχνευσης σε πραγματικό χρόνο. Το μοντέλο επιστρέφει τις ετικέτες που παράγονται κατά την πρόβλεψη.

Παράμετροι και Υπερπαράμετροι των Μοντέλων

Αναζήτηση Βέλτιστων Παραμέτρων

Κάθε ένα από αυτά τα παραπάνω μοντέλα παρέχει ένα πλήθος παραμέτρων και υπερπαραμέτρων που καθορίζουν τον τρόπο λειτουργίας του. Η σωστή επιλογή των

παραμέτρων αυτών δεν αποτελεί απλή διαδικασία λόγω των πολλών δυνατών συνδυασμών τους. Την αναζήτηση του κατάλληλου συνδυασμού κάνει ευκολότερη η χρήση της μεθόδου Grid Search. Με τη μέθοδο αυτή, επιλέγονται οι παράμετροι προς έλεγχο και ελέγχονται όλοι οι πιθανοί συνδυασμοί τους, επιλέγοντας τον καλύτερο με βάση κάποια μετρική αξιολόγησης. Στην περίπτωση αυτή ως μετρική αξιολόγησης επιλέγεται η μετρική F1 score. Η διαδικασία εύρεσης των βέλτιστων παραμέτρων γίνεται χρησιμοποιώντας το εκάστοτε υποσύνολο δεδομένων εκπαίδευσης. Από την προσέγγιση αυτή συγκεντρώνονται τα αποτελέσματα για κάθε υπηρεσία της εφαρμογής ξεχωριστά και οι βέλτιστες παράμετροι επιλέγονται για την τελική εκπαίδευση των μοντέλων.

Για τον αλγόριθμο BRICH, δεν μπορούμε να εφαρμόσουμε συμβατικό Grid Search, εφόσον η εκπαίδευση γίνεται σε δεδομένα χωρίς ανωμαλίες και στη συνέχεια το μοντέλο πραγματοποιεί προβλέψεις σε σύνολο δεδομένων με την προσθήκη ανωμαλιών. Για το λόγο αυτό ακολουθείται η τυπική διαδικασία με μόνη διαφοροποίηση το γεγονός ότι η εκπαίδευση και η αξιολόγηση γίνονται από τυχαία τμήματα διαφορετικών συνόλων δεδομένων. Οι παράμετροι που εξετάζονται είναι οι threshold, branching_factor και η ειδική παράμετρος anomaly_percentile. Η παράμετρος threshold καθορίζει τη μέγιστη ακτίνα μίας υποομάδας που δημιουργείται, με μικρότερες τιμές να ωθούν στη δημιουργία μικρότερων και πυκνότερων ομάδων και μεγαλύτερες τιμές να διευκολύνουν την ομαδοποίηση και να δημιουργούν λιγότερες και μεγαλύτερες σε μέγεθος ομάδες. Για την παράμετρο αυτή ελέγχονται τιμές από το 0.1 μέχρι και το 1.0 με βήμα 0.1. Η παράμετρος branching_factor από την άλλη, επηρεάζει το βαθμό διακλάδωσης του δέντρου που κατασκευάζει ο αλγόριθμος, ελέγχοντας το μέγιστο επιτρεπόμενο αριθμό κόμβων που μπορούν να δημιουργηθούν σε κάθε επίπεδο του δέντρου. Η τιμή της παραμέτρου αλλάζει το βάθος του δέντρου επηρεάζοντας την υπολογιστική πολυπλοκότητα του αλγορίθμου αλλά και την αποτελεσματικότητά του, αφού ένα δέντρο με μεγαλύτερο βάθος μπορεί να είναι πιο απαιτητικό υπολογιστικά αλλά διαχειρίζεται καλύτερα την ομαδοποίηση. Για την παράμετρο αυτή εξετάζονται οι τιμές 20, 30, 50, 70, 100. Όπως αναφέρθηκε και προηγουμένως, η ειδική παράμετρος anomaly_percentile αξιοποιείται στην παραλλαγή του αλγορίθμου που εξετάζεται στα πλαίσια αυτής της εργασίας. Η παράμετρος αυτή ουσιαστικά καθορίζει το όριο για την ανίχνευση ανωμαλιών. Εξετάζονται τιμές της από το 95 έως το 99 με μοναδιαίο βήμα. Συνολικά εξετάζονται 225 πιθανοί συνδυασμοί παραμέτρων για 5 τμήματα του συνόλου δεδομένων καταλήγοντας σε 1125 εκπαιδεύσεις του μοντέλου.

Για τα μοντέλα Local Outlier Factor χρησιμοποιείται η έτοιμη υλοποίηση της διαδικασίας Grid Search με διασταυρούμενη επικύρωση που παρέχεται από τη βιβλιοθήκη Scikit-Learn [77]. Η συνάρτηση αυτή δέχεται ως παραμέτρους το μοντέλο που θα χρησιμοποιηθεί, το σύνολο παραμέτρων προς αναζήτηση, τη μετρική αξιολόγησης, τον αριθμό διασταυρούμενων επικυρώσεων κ.α.. Για τον αλγόριθμο Local Outlier Factor όμως, η αναζήτηση βέλτιστων παραμέτρων εμφανίζει ένα σημαντικό πρόβλημα. Στις προεπιλεγμένες ρυθμίσεις του μοντέλου για εντοπισμό ανωμαλιών, δεν παρέχεται η συνάρτηση predict(), αφού τόσο η εκπαίδευση όσο και η πρόγνωση γίνεται σε κοινό σύνολο δεδομένων. Για το λόγο αυτό κατασκευάζεται ένα περιτύλιγμα (wrapper) για το βασικό μοντέλο που αντικαθιστά τη συνάρτηση predict(), με την fit_predict(), η οποία εκτελεί διαδοχικά τις διαδικασίες εκπαίδευσης και πρόγνωσης. Εφαρμόζοντας αυτά, ελέγχονται οι παράμετροι n_neighbors, metric, leaf_size, algorithm και contamination. Η παράμετρος n_neighbors καθορίζει τον

αριθμό των γειτόνων που χρησιμοποιείται για τον υπολογισμό της τοπικής πυκνότητας. Μικρότερες τιμές της παραμέτρου κάνουν τον αλγόριθμο πιο ευαίσθητο σε μικρές αποκλίσεις, διευκολύνοντας τον εντοπισμό ανωμαλιών σε πυκνές περιοχές. Αντίθετα, μεγαλύτερες τιμές της παραμέτρου μειώνουν την ευαισθησία του αλγορίθμου και μαζί της τον αριθμό των ψευδών θετικών αποτελεσμάτων. Για την παράμετρο αυτή επιλέγονται οι τιμές 20, 50, 100, 150, 200. Η παράμετρος metric από την άλλη επηρεάζει τον τρόπο υπολογισμού της απόστασης μεταξύ σημείων και άρα τον τελικό υπολογισμό της τοπικής πυκνότητας. Παρέχονται επιλογές όπως η Ευκλείδεια απόσταση, απόσταση Manhattan κ.α. με την κάθε μία να εκφράζει καλύτερα την απόσταση για διαφορετικά είδη δεδομένων. Για τη διαδικασία της αναζήτησης επιλέγονται οι τιμές 'euclidean', 'manhattan' και 'chebyshev'. Η παράμετρος algorithm αναφέρεται στον αλγόριθμο που χρησιμοποιείται για τον εντοπισμό των κοντινότερων γειτόνων κάθε σημείου. η επιλογή αλγορίθμου πρέπει να γίνει παρέχοντας την κατάλληλη αντιστάθμιση ανάμεσα στην επίδοση και τις απαιτήσεις σε υπολογιστικούς πόρους. Οι τιμές της παραμέτρου που ελέγχονται είναι οι 'auto', 'ball_tree', 'kd_tree', 'brute'. Η παράμετρος leaf_size αφορά αλγορίθμους που βασίζονται σε δέντρα (όπως οι KDTree, BallTree), επηρεάζοντας την αποδοτικότητά τους. Μικρότερος αριθμός φύλλων αυξάνει την ταχύτητα εκτέλεσης του αλγορίθμου αυξάνοντας όμως ταυτόχρονα την χρήση μνήμης. Για την παράμετρο αυτή εξετάζονται οι τιμές 20, 30, 40, 50. Τέλος, η παράμετρος contamination προσδιορίζει το ποσοστό του συνόλου δεδομένων που αποτελείται από θετικά (ανώμαλα) δείγματα. Η τιμή της παραμέτρου βοηθά το μοντέλο να καθορίσει τα όρια για την ανίχνευση ανωμαλιών. Οι τιμές της παραμέτρου που συμπεριλαμβάνονται στην αναζήτηση είναι 'auto' και η υπολογιζόμενη τιμή για το σύνολο δεδομένων που χρησιμοποιείται. Συνολικά εξετάζονται 480 συνδυασμοί παραμέτρων για 5 τμήματα του συνόλου δεδομένων οδηγώντας σε 2400 εκπαιδεύσεις του μοντέλου.

Για τον αλγόριθμο Isolation Forest, η αναζήτηση βέλτιστων παραμέτρων γίνεται χρησιμοποιώντας τις παραμέτρους n_estimators, max_smaples, bootstrap και contamination. Η παράμετρος n_estimators καθορίζει τον αριθμό δέντρων απομόνωσης που δημιουργούνται στο δάσος. Ο αριθμός των δέντρων επηρεάζει την αποτελεσματικότητα του μοντέλου, με μεγαλύτερο πλήθος δέντρων να αυξάνει την αξιοπιστία του αποτελέσματος αυξάνοντας ταυτόχρονα και τις υπολογιστικές απαιτήσεις. Για την παράμετρο αυτή εξετάζονται τιμές στο διάστημα 50, 500 με βήμα 50. Η παράμετρος max_samples εκφράζει τον αριθμό ή το ποσοστό των δεδομένων που θα χρησιμοποιηθούν για την εκπαίδευση του κάθε δέντρου. Μικρότερα ποσοστά επιταχύνουν την εκτέλεση του αλγορίθμου και αυξάνουν την ποικιλία μεταξύ των δέντρων, ενδεχομένως μειώνοντας την ικανότητα του μοντέλου να εντοπίσει μοτίβα μεγαλύτερης κλίμακας. Μεγαλύτερα ποσοστά συνήθως αυξάνουν τις υπολογιστικές απαιτήσεις αλλά και την ακρίβεια του μοντέλου, όμως μειώνουν την ποικιλία στα δέντρα που δημιουργούνται. Για την παράμετρο αυτή δοκιμάζονται οι τιμές 'auto', 50, 100, 200, 400. Τέλος η παράμετρος bootstrap χρησιμοποιείται για την προσθήκη τυχαιότητας και ποικιλίας στο δάσος, συμβάλλοντας στην αποφυγή της υπερπροσαρμογής. Οι τιμές που εξετάζονται είναι True και False, με την παράμετρο True να επιτρέπει την επαναχρησιμοποίηση των ίδιων σημείων για την εκπαίδευση των δέντρων. Όπως και πριν για την παράμετρο contamination ελέγχονται οι τιμές 'auto' και η υπολογιζόμενη για το υποσύνολο εκπαίδευσης. Συνολικά δημιουργούνται 180 συνδυασμοί παραμέτρων που χρησιμοποιούνται για την εκπαίδευση του μοντέλου σε 5 τμήματα του συνόλου δεδομένων, καταλήγοντας σε 900 εκπαιδεύσεις.

Επιπλέον Παράμετροι

Μετά το πέρας της αναζήτησης βέλτιστων παραμέτρων για τα τρία μοντέλα, έχουν συγκεντρωθεί οι πλειάδες τιμών με την καλύτερη απόδοση στο υποσύνολο δεδομένων εκπαίδευσης για τα δεδομένα που αφορούν κάθε υπηρεσία της εφαρμογής. Από τις παραμέτρους αυτές προκύπτουν οι τελικές παράμετροι για την εκπαίδευση των μοντέλων. Πέρα από αυτές, παρέχονται και κάποιες επιπλέον παράμετροι, των οποίων οι τιμές ήταν γνωστές εκ των προτέρων. Στον αλγόριθμο BIRCH προστίθεται η παράμετρος n_clusters με τιμή None. Η παράμετρος αυτή καθορίζει το πλήθος των ομάδων στις οποίες ο αλγόριθμος θα προσπαθήσει να χωρίσει τα δεδομένα. Στην περίπτωση αυτή δεν είναι επιθυμητό να δοθεί μία συγκεκριμένη τιμή, για να μην δημιουργηθεί κάποιου είδους μεροληψία στο μοντέλο. Δίνοντας την τιμή None εξασφαλίζεται ακριβώς αυτό, με το τελευταίο βήμα του αλγορίθμου, αυτό της ομαδοποίησης των υποομάδων, να μην εκτελείται και να επιστρέφονται οι υποομάδες που έχουν δημιουργηθεί. Στο μοντέλο βασισμένο στον αλγόριθμο Local Outlier Factor δεν χρειάζεται να προστεθεί κάποια επιπλέον παράμετρος. Στο μοντέλο Isolation Forest πάλι, προστίθενται οι παράμετροι n_jobs και random state. Οι παράμετροι αυτοί καθορίζουν τον αριθμό εργασιών που εκτελούνται παράλληλα κατά την εκπαίδευση του μοντέλου αλλά και τις προβλέψεις και ελέγχουν την ψευδοτυχαιότητα κατά την εκτέλεση του αλγορίθμου αντίστοιχα. Στις περιπτώσεις των αλγορίθμων Local Outlier Factor και Isolation Forest η τιμή της παραμέτρου contamination τίθεται ίση με τη μέση τιμή των αντίστοιχων για τα υποσύνολα εκπαίδευσης και ελέγχου.

4.2 Πειραματικά Αποτελέσματα

4.2.1 Μετρικές Αξιολόγησης

Για την αξιολόγηση της αποδοτικότητας των μοντέλων στις διαφορετικές διαμορφώσεις που ακολουθούν απαιτείται η χρήση μετρικών αξιολόγησης. Οι μετρικές αυτές αντανακλούν το πόσο καλά το μοντέλο μπορεί να εντοπίσει τις ανωμαλίες (ακραίες τιμές) και να τις διαχωρίσει από τις κανονικές τιμές. Οι μετρικές precision, recall και το F1-score είναι τρεις ευρέως χρησιμοποιούμενες μετρικές για την αξιολόγηση μοντέλων ταξινόμησης, ιδίως όταν τα δεδομένα είναι ανισόρροπα (όπως συμβαίνει συχνά στην ανίχνευση ανωμαλιών). Ακολουθεί μια λεπτομερής εξήγηση αυτών των μετρικών και της σημασίας τους για την ανίχνευση ανωμαλιών:

1. **Precision**: η μετρική precision είναι το ποσοστό των αληθώς θετικών προβλέψεων (σωστά εντοπισμένες ανωμαλίες) επί του συνόλου των περιπτώσεων που προβλέφθηκαν ως ανωμαλίες (αληθώς θετικές + ψευδώς θετικές). Με άλλα λόγια, δηλώνει πόσες από τις περιπτώσεις που το μοντέλο χαρακτήρισε ως ανωμαλίες είναι πράγματι ανωμαλίες. Για τον υπολογισμό της χρησιμοποιείται ο μαθηματικός τύπος:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

Υψηλή τιμή της μετρικής precision σημαίνει ότι όταν το μοντέλο χαρακτηρίζει μια περίπτωση ως ανωμαλία, είναι πολύ πιθανό να είναι σωστή. Αυτό είναι σημαντικό σε περιπτώσεις όπου τα ψευδώς θετικά αποτελέσματα (κανονικές περιπτώσεις που

κατατάσσονται εσφαλμένα ως ανωμαλίες) κοστίζουν ακριβά ή οδηγούν σε περιττές έρευνες.

2. **Recall**: η μετρική recall είναι το ποσοστό των αληθώς θετικών προβλέψεων (σωστά εντοπισμένες ανωμαλίες) επί του συνόλου των πραγματικών ανωμαλιών (αληθώς θετικές + ψευδώς αρνητικές). Μετράει πόσο καλά το μοντέλο μπορεί να συλλάβει τις πραγματικές ανωμαλίες. Για τον υπολογισμό της χρησιμοποιείται ο τύπος:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}$$

Υψηλές τιμές της μετρικής αυτής δηλώνουν ότι το μοντέλο είναι καλό στον εντοπισμό των περισσότερων πραγματικών ανωμαλιών. Η επίτευξη υψηλών τιμών της μετρικής recall είναι ιδιαίτερα σημαντική όταν η απουσία μιας ανωμαλίας (ψευδώς αρνητικά αποτελέσματα) είναι ιδιαίτερα επικίνδυνη. Για παράδειγμα, στα συστήματα ασφαλείας, η παράλειψη μιας απειλής ή εισβολής μπορεί να έχει σοβαρές συνέπειες, οπότε η μεγιστοποίηση του recall είναι ζωτικής σημασίας.

3. **F1-Score**: Το F1-score είναι ο αρμονικός μέσος όρος των precision και recall, παρέχοντας μια ενιαία μετρική που εξισορροπεί και τα δύο. Είναι ιδιαίτερα χρήσιμο όταν πρέπει να βρεθεί ένας συμβιβασμός μεταξύ των δύο μετρικών, κάτι που είναι σύνηθες σε προβλήματα ανίχνευσης ανωμαλιών. Για τον υπολογισμό του F1-Score χρησιμοποιείται ο μαθηματικός τύπος:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Το F1-score είναι η ιδανική μετρική όταν υπάρχει ανάγκη εξισορρόπησης του κόστους των ψευδώς θετικών (λανθασμένη επισήμανση φυσιολογικών περιπτώσεων) και των ψευδώς αρνητικών (απουσία ανωμαλιών). Ένα υψηλό F1-score υποδεικνύει ότι το μοντέλο έχει καλή ισορροπία μεταξύ ακρίβειας και ανάκλησης, εντοπίζοντας αποτελεσματικά τις ανωμαλίες και ελαχιστοποιώντας τις εσφαλμένες προβλέψεις.

4.2.2 Επισήμανση Δεδομένων

Για τη σωστή αξιολόγηση των αποτελεσμάτων με χρήση των παραπάνω μετρικών, απαραίτητη είναι η ύπαρξη επισημασμένων δεδομένων. Για το σύνολο δεδομένων που χρησιμοποιείται στα πλαίσια της εργασίας αυτής η επισήμανση πρέπει να γίνει με χειροκίνητο τρόπο, γεγονός εφικτό, αφού υπάρχει πλήρης γνώση πάνω στην εμφύτευση των ανωμαλιών στα δεδομένα. Όπως έχει αναφερθεί, τα δεδομένα μπορεί να περιέχουν τρία είδη ανωμαλιών που εκφράζονται με διαφορετικό τρόπο στα δεδομένα.

- **Τυχαίες Αυξήσεις** στο χρόνο απόκρισης (**Random Slowdowns**): οι ανωμαλίες αυτής της μορφής παρουσιάζονται στις χρονοσειρές της μετρικής χρόνου απόκρισης ως ξαφνικές αυξήσεις του χρόνου απόκρισης με τιμή σημαντικά μεγαλύτερη του μέσου όρου. Η επισήμανση τέτοιων ανωμαλιών εύκολα επιτυγχάνεται εφαρμόζοντας ένα φίλτρο στη χρονοσειρά της αντίστοιχης μετρικής δηλώνοντας ως ανώμαλες όλες τις

τιμές πάνω από ένα συγκεκριμένο όριο. Για το σύνολο δεδομένων που χρησιμοποιείται, το όριο αυτό προκύπτει για χρόνους απόκρισης μεγαλύτερους των 8 δευτερολέπτων.

- **Αυξήσεις** στο χρόνο απόκρισης της **βάσης δεδομένων** που οφείλονται σε εξάντληση του αριθμού ταυτόχρονων συνδέσεων (**DB Connection Limit**): οι ανωμαλίες αυτού του είδους εκδηλώνονται στις χρονοσειρές των μετρικών χρόνου απόκρισης και αριθμού ενεργών συνδέσεων στη βάση δεδομένων. Αντίστοιχα με πριν, η επισήμανσή τους βασίζεται στην εφαρμογή φίλτρων στις χρονοσειρές των δύο αυτών μετρικών. Για τη μετρική χρόνου απόκρισης εφαρμόζεται το ίδιο όριο των 8 δευτερολέπτων, ενώ για τη μετρική αριθμού ενεργών συνδέσεων το όριο τίθεται στην ακραία τιμή των 20 συνδέσεων. Στη βάση δεδομένων έχει εφαρμοστεί τεχνητός περιορισμός των ενεργών συνδέσεων μέσω του κώδικα της εφαρμογής. Η τιμή αυτή επιλέχθηκε για το σκοπό της προσομοίωσης της λειτουργικότητας της εφαρμογής και μπορεί να τροποποιηθεί μέσω του κώδικα. Τα τελικά αποτελέσματα της επισήμανσης προκύπτουν εφαρμόζοντας την πράξη τομής (logical OR) στις επιμέρους επισημάνσεις.
- **Τυχαία Σφάλματα** κατά την εκτέλεση του κώδικα (**Random Exceptions**): οι ανωμαλίες αυτής της μορφής εμφανίζονται στα άμεσα στα δεδομένα των αρχείων καταγραφής και έμμεσα στα ίχνη εκτέλεσης. Κατά την εμφάνιση του σφάλματος του κώδικα δημιουργείται κατάλληλη εγγραφή στο αρχείο καταγραφής με το επίπεδο σοβαρότητας ERROR. Αντίστοιχα στα δεδομένα που αφορούν το ίχνος της εκτέλεσης της εφαρμογής, μία τέτοια ανωμαλία μπορεί να εκφραστεί ως ένα μη ολοκληρωμένο μονοπάτι εκτέλεσης (μικρότερο μήκος ή διαφορετική ακολουθία από spans). Η επισήμανση των δεδομένων που περιέχουν ανωμαλίες τέτοιου τύπου γίνεται προτιμώντας τη χρήση των δεδομένων αρχείων καταγραφής και συγκεκριμένα εφαρμόζοντας κατάλληλο φίλτρο για εντοπισμό εγγραφών με επίπεδο σοβαρότητας ERROR.

4.2.3 Πείραμα 1ο - Εντοπισμός ανωμαλιών που ανήκουν σε μία κατηγορία

Στο πείραμα αυτό στόχος είναι η αξιολόγηση της ικανότητας των μοντέλων να εντοπίζουν επιμέρους ανωμαλίες. Για κάθε ένα από τα τρία είδη ανωμαλιών που αναφέρθηκαν χρησιμοποιείται το είδος δεδομένων στο οποίο εκφράζονται καλύτερα. Με άλλα λόγια, εξετάζεται η ικανότητα των μοντέλων να εντοπίζουν ανωμαλίες τύπου Random Slowdowns σε δεδομένα χρόνου απόκρισης, ανωμαλίες τύπου DB Connection Limit σε συνδυασμό δεδομένων χρόνου απόκρισης και αριθμού συνδέσεων στη βάση, και ανωμαλίες τύπου Random Exceptions σε δεδομένα από αρχεία καταγραφής και ίχνη εκτέλεσης.

Για την ανίχνευση ανωμαλιών στα δεδομένα αρχείων καταγραφής ακολουθείται η διαδικασία που αναλύθηκε προηγουμένως στην ενότητα 4.1.4. Εγγραφές από εκτέλεση της εφαρμογής κατά την κανονική λειτουργία αξιοποιούνται για την εξαγωγή των προτύπων ορθής λειτουργίας. Τα πρότυπα αυτά συγκεντρώνονται και χρησιμοποιούνται για αντιπαραβολή με τα πρότυπα των δεδομένων ελέγχου. Το σύνολο δεδομένων εκπαίδευσης αποτελείται από 2926 εγγραφές. Ο έλεγχος στην περίπτωση αυτή γίνεται χρησιμοποιώντας αρχεία καταγραφής από εκτέλεση της εφαρμογής στην οποία έχει γίνει εμφύτευση ανωμαλιών τύπου Random Exceptions. Τα δεδομένα ελέγχου αποτελούνται από 2947 εγγραφές εκ των οποίων 106 (ποσοστό 3.6%) είναι γνωστές ανωμαλίες. Το μοντέλο πετυχαίνει άριστες επιδόσεις με όλες τις μετρικές αξιολόγησης να έχουν τη μέγιστη τιμή (100%).

Αντίστοιχα, για τα δεδομένα από τα ίχνη εκτέλεσης ακολουθείται η γνωστή διαδικασία. Για την εκπαίδευση του μοντέλου και την εξαγωγή των προτύπων ορθής λειτουργίας χρησιμοποιούνται τα spans που παρήχθησαν κατά την ίδια κανονική εκτέλεση της εφαρμογής, όπως και τα αρχεία καταγραφής. Τα δεδομένα εκπαίδευσης αποτελούνται από 5240 spans, τα οποία ανήκουν σε 1000 διακριτά ίχνη. Από την επεξεργασία προκύπτουν 3 πρότυπα ορθής λειτουργίας. Για τον εντοπισμό ανωμαλιών χρησιμοποιούνται πάλι ίχνη στα οποία περιλαμβάνονται ανωμαλίες τύπου Random Exceptions. Συγκεκριμένα το σύνολο δεδομένων περιέχει 5416 spans που προέρχονται από 1000 διακριτά ίχνη εκτέλεσης. Και σε αυτή την περίπτωση επιτυγχάνονται άριστα αποτελέσματα, με τις τρεις μετρικές αξιολόγησης να παίρνουν τη μέγιστη τιμή 100%.

Για την ανίχνευση ανωμαλιών τύπου Random Slowdowns παρουσιάζονται και συγκρίνονται οι επιδόσεις των τριών μοντέλων μηχανικής μάθησης που μελετώνται. Η αναζήτηση βέλτιστων παραμέτρων γίνεται χρησιμοποιώντας ένα υποσύνολο δεδομένων με προσθήκη ανωμαλιών, αποτελούμενο από 960 τιμές για το χρόνο απόκρισης κάθε υπηρεσίας της εφαρμογής. Το μέσο ποσοστό ανωμαλιών στο σύνολο των δεδομένων ανέρχεται στο 2.08%. Οι βέλτιστοι συνδυασμοί παραμέτρων για τα μοντέλα παρατίθενται (Πίνακας 4.2, Πίνακας 4.3, Πίνακας 4.4).

Πίνακας 4.2: Πείραμα 1, Ανωμαλίες Random Slowdown, Παράμετροι Μοντέλου Birch

Παράμετροι Μοντέλου BIRCH	
n_clusters	None
threshold	0.6
branching_factor	20
anomaly_percentile	99

Πίνακας 4.3: Πείραμα 1, Ανωμαλίες Random Slowdown, Παράμετροι Μοντέλου LOF

Παράμετροι Μοντέλου LOF	
algorithm	auto
metric	manhattan
leaf_size	20
n_neighbors	50
contamination	0.02

Πίνακας 4.4: Πείραμα 1, Ανωμαλίες Random Slowdown, Παράμετροι Μοντέλου iForest

Παράμετροι Μοντέλου iForest	
bootstrap	True
max_samples	400
n_estimators	150
contamination	0.02
n_jobs	-1
random_state	42

Για την εκπαίδευση του μοντέλου Birch χρησιμοποιούνται οι χρόνοι απόκρισης των υπηρεσιών της εφαρμογής κατά την ίδια εκτέλεση αντιπροσωπευτική της κανονικής λειτουργίας που χρησιμοποιήθηκε και στις προηγούμενες περιπτώσεις. Οι χρονοσειρές που περιέχει το σύνολο δεδομένων παρουσιάζουν 980 εγγραφές για το χρόνο απόκρισης κάθε υπηρεσίας. Όπως έχει αναφερθεί, το μοντέλο LOF δεν απαιτεί διακριτό σύνολο δεδομένων εκπαίδευσης. Για το μοντέλο iForest πάλι, χρησιμοποιείται το σύνολο δεδομένων που αξιοποιήθηκε κατά την αναζήτηση βέλτιστων παραμέτρων. Για τη διαδικασία εντοπισμού ανωμαλιών και τα τρία μοντέλα χρησιμοποιούν κοινό σύνολο δεδομένων με προσθήκη ανωμαλιών. Το σύνολο δεδομένων αποτελείται από 1000 εγγραφές χρόνου απόκρισης για κάθε υπηρεσία. Επομένως η αναλογία μεγέθους των συνόλων εκπαίδευσης και ελέγχου είναι πολύ κοντά στο 1-1.

Και στα τρία μοντέλα, όλα τα δεδομένα που χρησιμοποιούνται υφίστανται κατάλληλη κλιμάκωση χρησιμοποιώντας το εργαλείο κλιμάκωσης RobustScaler [78] που παρέχει η βιβλιοθήκη Scikit-Learn. Το RobustScaler είναι ένα εργαλείο προεπεξεργασίας δεδομένων που χρησιμοποιείται για την κλιμάκωση των χαρακτηριστικών σε ένα συγκεκριμένο εύρος, του οποίου βασικό χαρακτηριστικό είναι η μικρότερη ευαισθησία στις ακραίες τιμές σε σύγκριση με τις τυπικές μεθόδους κλιμάκωσης όπως το StandardScaler [79] ή το MinMaxScaler [80]. Λειτουργεί με τη χρήση εύρωστων στατιστικών στοιχείων (όπως η διάμεσος (median) και το ενδοτεταρτημοριακό εύρος (IQR)) αντί της μέσης τιμής (mean) και της τυπικής απόκλισης (standard deviation), καθιστώντας το ιδανικό για σύνολα δεδομένων με σημαντικές ακραίες τιμές. Αντί για την αφαίρεση του μέσου όρου (όπως στο StandardScaler), το RobustScaler κεντράρει τα δεδομένα αφαιρώντας τη διάμεσο κάθε χαρακτηριστικού. Η διάμεσος είναι μια πιο εύρωστη στατιστική, καθώς δεν επηρεάζεται από ακραίες τιμές. Επιπλέον, αντί να κλιμακώνει τα δεδομένα με βάση την τυπική απόκλιση (όπως στο StandardScaler) ή με βάση το εύρος (όπως στο MinMaxScaler), το RobustScaler κλιμακώνει τα χαρακτηριστικά χρησιμοποιώντας το ενδοτεταρτημοριακό εύρος (IQR). Το IQR είναι το εύρος μεταξύ του 25ου εκατοστημορίου (Q1) και του 75ου εκατοστημορίου (Q3) των δεδομένων. Αυτό ελαχιστοποιεί αποτελεσματικά τον αντίκτυπο των ακραίων τιμών στη διαδικασία κλιμάκωσης.

Μετά την εκτέλεση του πειράματος, υπολογίζονται και συγκεντρώνονται οι τιμές των μετρικών αξιολόγησης για τα αποτελέσματα κάθε υπηρεσίας και υπολογίζεται ένας μέσος όρος για κάθε μετρική. Εφόσον η εμφάνιση ανωμαλιών περιορίζεται σε ένα υποσύνολο των υπηρεσιών της εφαρμογής, μόνο οι υπηρεσίες αυτές λαμβάνονται υπόψιν κατά τον υπολογισμό του μέσου όρου των μετρικών. Από την αξιολόγηση προκύπτει το συμπέρασμα ότι και τα τρία μοντέλα εντοπίζουν τις ανωμαλίες με τον ιδανικό τρόπο, πετυχαίνοντας άριστα αποτελέσματα και για τις τρεις μετρικές (Precision, Recall, F1-Score = 100%).

Για την ανίχνευση ανωμαλιών τύπου DB Connection Limit παρουσιάζονται και συγκρίνονται οι επιδόσεις και των τριών μοντέλων μηχανικής μάθησης που μελετώνται. Για την περίπτωση αυτή υλοποιείται ένας συνδυασμός μετρικών και συγκεκριμένα των μετρικών χρόνου απόκρισης και του αριθμού ενεργών συνδέσεων στη βάση δεδομένων. Ο συνδυασμός των δεδομένων γίνεται με τη μέθοδο που παρουσιάστηκε στο κεφάλαιο 3.4.4, με τις τιμές της μετρικής αριθμού ενεργών συνδέσεων να αναδειγματολειπτούνται ώστε να παρέχεται μία τιμή για κάθε μία αντίστοιχη της χρονοσειράς χρόνων απόκρισης. Η αναζήτηση βέλτιστων παραμέτρων για τα μοντέλα γίνεται χρησιμοποιώντας ένα υποσύνολο δεδομένων με προσθήκη ανωμαλιών, αποτελούμενο από 1000 τιμές για το χρόνο απόκρισης κάθε υπηρεσίας της εφαρμογής και 601 τιμές για τον αριθμό συνδέσεων στη βάση που αναδειγματολειπτούνται στις 1000. Το μέσο ποσοστό ανωμαλιών στο σύνολο των δεδομένων ανέρχεται στο 4.4%. Οι παράμετροι που προκύπτουν και εν τέλει χρησιμοποιούνται για την αρχικοποίηση των τριών μοντέλων παρουσιάζονται παρακάτω (Πίνακας 4.5, Πίνακας 4.6, Πίνακας 4.7).

Πίνακας 4.5: Πείραμα 1, Ανωμαλίες DB Connection Limit, Παράμετροι Μοντέλου Birch

Παράμετροι Μοντέλου BIRCH	
n_clusters	None
threshold	0.8
branching_factor	20
anomaly_percentile	99

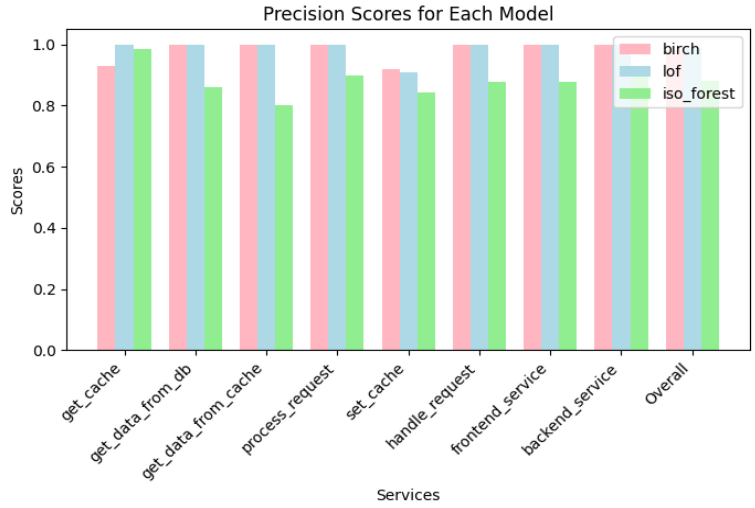
Πίνακας 4.6: Πείραμα 1, Ανωμαλίες DB Connection Limit, Παράμετροι Μοντέλου LOF

Παράμετροι Μοντέλου LOF	
algorithm	auto
metric	manhattan
leaf_size	20
n_neighbors	200
contamination	0.09

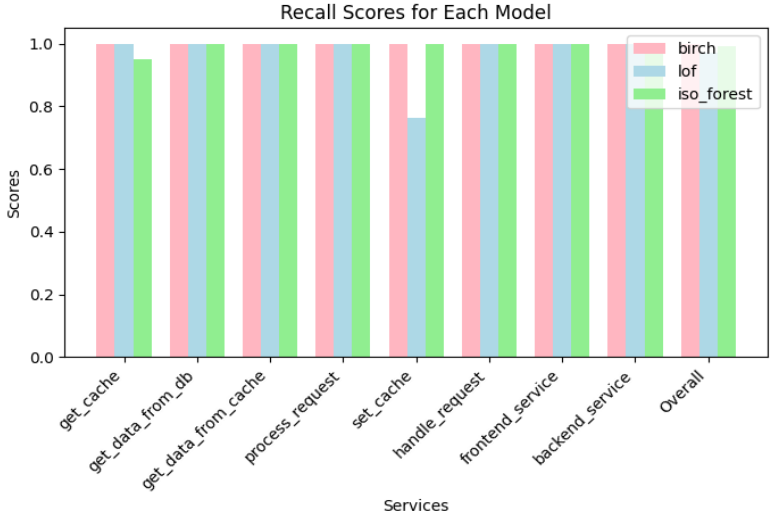
Πίνακας 4.7: Πείραμα 1, Ανωμαλίες DB Connection Limit, Παράμετροι Μοντέλου iForest

Παράμετροι Μοντέλου iForest	
bootstrap	True
max_samples	100
n_estimators	50
contamination	0.06
n_jobs	-1
random_state	42

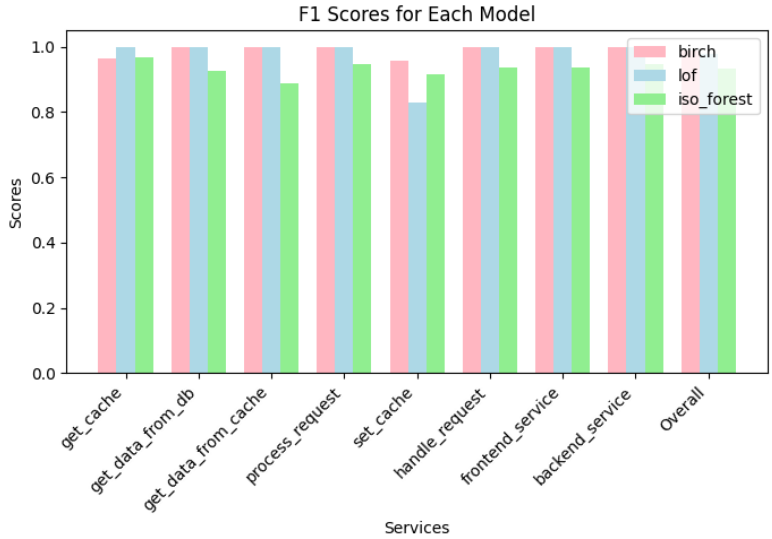
Για την εκπαίδευση του μοντέλου Birch χρησιμοποιούνται οι χρόνοι απόκρισης των υπηρεσιών της εφαρμογής κατά την ίδια εκτέλεση αντιπροσωπευτική της κανονικής λειτουργίας που χρησιμοποιήθηκε και στις προηγούμενες περιπτώσεις. Οι χρονοσειρές που περιέχει το σύνολο δεδομένων παρουσιάζουν 980 εγγραφές για κάθε μετρική μετά τη διαδικασία της δειγματοληψίας. Όπως έχει αναφερθεί, το μοντέλο LOF δεν απαιτεί διακριτό σύνολο δεδομένων εκπαίδευσης. Για το μοντέλο iForest πάλι, χρησιμοποιείται το σύνολο δεδομένων που αξιοποιήθηκε κατά την αναζήτηση βέλτιστων παραμέτρων. Για τη διαδικασία εντοπισμού ανωμαλιών και τα τρία μοντέλα χρησιμοποιούν κοινό σύνολο δεδομένων με προσθήκη ανωμαλιών. Το σύνολο δεδομένων αποτελείται από 940 εγγραφές για κάθε υπηρεσία. Επομένως η αναλογία μεγέθους των συνόλων εκπαίδευσης και ελέγχου είναι πολύ κοντά στο 1-1. Το ποσοστό των γνωστών ανωμαλιών στο σύνολο δεδομένων ελέγχου ανέρχεται στο 8.5%. Όπως και πριν, όλα τα δεδομένα κλιμακώνονται χρησιμοποιώντας το RobustScaler. Παρουσιάζονται τα αποτελέσματα της επίδοσης των τριών μοντέλων με βάση τις μετρικές αξιολόγησης Precision (Σχήμα 4.4), Recall (Σχήμα 4.5) και F1-Score (Σχήμα 4.6).



Σχήμα 4.4: Πείραμα 1, Ανωμαλίες DB Connection Limit, Αποτελέσματα Μετρικής Precision



Σχήμα 4.5: Πείραμα 1, Ανωμαλίες DB Connection Limit, Αποτελέσματα Μετρικής Recall



Σχήμα 4.6: Πείραμα 1, Ανωμαλίες DB Connection Limit, Αποτελέσματα Μετρικής F1-Score

Στην περίπτωση αυτή ανωμαλίες εντοπίζονται στα δεδομένα όλων των υπηρεσιών, επομένως όλες οι τιμές λαμβάνονται υπόψιν στον υπολογισμό του μέσου όρου. Το μοντέλο Birch έχει επιδόσεις αντίστοιχες με πριν, με την ανίχνευση στις περισσότερες υπηρεσίες να πετυχαίνει τη μέγιστη τιμή για όλες τις μετρικές (100%), με μόνη εξαίρεση τις υπηρεσίες get_cache και set_cache στις οποίες παρατηρείται μικρή πτώση στις μετρικές Precision και F1-Score με τιμές Precision = 93% και 91.95% και F1-Score = 96.39% και 95.81% αντίστοιχα. Τα συνολικά αποτελέσματα για το μοντέλο είναι Precision = 98.12%, Recall = 100%, F1-Score = 99.02%. Σχεδόν εξίσου καλά αποτελέσματα πετυχαίνει και το μοντέλο LOF, με άριστες τιμές για όλες τις μετρικές και υπηρεσίες εκτός της set_cache (Pr = 91.04%, Rec = 76.25%, F1 = 82.99%). Τα συνολικά αποτελέσματα για το μοντέλο είναι Precision = 98.88%, Recall = 97.03%, F1-Score = 97.87%. Μεγαλύτερη πτώση στις επιδόσεις παρατηρείται στο μοντέλο iForest με όλες τις υπηρεσίες να διατηρούν άριστο Recall, αλλά μειωμένο Precision με ελάχιστη τιμή το 80% για την υπηρεσία get_data_from_cache. Σε όλες τις υπόλοιπες υπηρεσίες το μοντέλο διατηρεί τιμές του F1-Score πάνω από 90%. Τα συνολικά αποτελέσματα για το μοντέλο είναι Precision = 88.07%, Recall = 99.38%, F1-Score = 93.26%.

Σχολιασμός - Παρατηρήσεις

Τα παραπάνω αποτελέσματα αποτυπώνουν σαφώς τη διαφορά στην πληροφορία που περιέχεται στους διαφορετικούς τύπους δεδομένων. Τα αρχεία καταγραφής και τα ίχνη, έχουν δομές που τους επιτρέπουν να αποτυπώνουν περισσότερες και πιο ουσιαστικές πληροφορίες για την κατάσταση του συστήματος απ' ότι τα δεδομένα μετρικών. Η διαφορά αυτή αποτυπώνεται και από τη μικρή αλλά σε ορισμένες περιπτώσεις ουσιαστική διαφορά στην απόδοση των αντίστοιχων μοντέλων.

Στην περίπτωση των αρχείων καταγραφής και των ιχνών, για τον εντοπισμό των ανωμαλιών αξιοποιούνται συστήματα που βασίζονται σε ένα είδος κανόνων, τα οποία χτίζουν ένα πλαίσιο προτύπων κανονικής συμπεριφοράς και απορρίπτουν όσα δεδομένα αποκλίνουν από τα πρότυπα αυτά. Τα αρχεία καταγραφής και τα ίχνη συχνά περιέχουν δεδομένα με χαρακτηριστική πληροφορία πλαισίου, όπως η ακριβής ακολουθία κλήσεων συναρτήσεων, η κατάσταση των μικροϋπηρεσιών ή λεπτομερή μηνύματα σφάλματος. Αυτές οι δομημένες πληροφορίες επιτρέπουν την ανίχνευση ανωμαλιών πλαισίου, όπου οι αποκλίσεις αφορούν απροσδόκητα πρότυπα σε όλες τις υπηρεσίες, που συλλαμβάνονται έγκαιρα. Τα δομημένα δεδομένα που χρησιμοποιούνται κατά τον εντοπισμό ανωμαλιών προσφέρουν χρήσιμη πληροφορία όπως για παράδειγμα κωδικούς σφαλμάτων, ή αναφορές της κατάστασης του συστήματος.

Για τις μετρικές πάλι, αξιοποιούνται τα πολλά υποσχόμενα μοντέλα μηχανικής μάθησης, τα οποία όμως περιορίζονται από την έλλειψη πληροφορίας πλαισίου στις μονοσήμαντες μετρικές. Δεδομένου ότι μετρικές όπως ο αριθμός συνδέσεων στη βάση δεδομένων ή η καθυστέρηση εξυπηρέτησης αιτήσεων συλλέγονται μεμονωμένα, συχνά αποτυγχάνουν να καταγράψουν αλληλεπιδράσεις μεταξύ υπηρεσιών ή συσχετισμένες συμπεριφορές που συχνά σηματοδοτούν σύνθετες ανωμαλίες. Σε περιπτώσεις εύκολα διαχωρίσιμων ανωμαλιών, τα μοντέλα επιτυγχάνουν εξαιρετική απόδοση. Ωστόσο, όταν η πηγή της ανωμαλίας είναι πιο σύνθετη - όπως αλληλεπιδράσεις πολλαπλών υπηρεσιών ή σταδιακή υποβάθμιση του συστήματος - τα δεδομένα μετρικών από μόνα τους μπορεί να μην παρέχουν επαρκείς πληροφορίες για ακριβή ανίχνευση.

Τα συστήματα που βασίζονται σε κανόνες υπερέχουν στην ανίχνευση δομικών ανωμαλιών ή ανωμαλιών ακολουθίας (π.χ. απροσδόκητα μοτίβα καταγραφής), ενώ τα μοντέλα μηχανικής μάθησης για μετρικές είναι καλύτερα στον εντοπισμό σταδιακών ή ανεπαίσθητων μεταβολών στη συμπεριφορά του συστήματος. Οι δυνατότητες τέτοιων μοντέλων είναι πολλές, ιδίως όταν συνδυάζονται με δεδομένα πλούσια σε περιεχόμενο από αρχεία καταγραφής και ίχνη, προσφέροντας το καλύτερο δυνατό αποτέλεσμα τόσο για τη λεπτομέρεια στην ανίχνευση όσο και για τη διορατικότητα για τη βελτίωση της ανίχνευσης ανωμαλιών σε πολύπλοκα, κατανεμημένα περιβάλλοντα.

4.2.4 Πείραμα 2ο - Εντοπισμός μεικτών ανωμαλιών

Ως επόμενο λογικό βήμα, στο πείραμα αυτό εξετάζεται η ικανότητα των μοντέλων να εντοπίζουν ανωμαλίες σε σύνολα δεδομένων που περιέχουν μίξη ανωμαλιών πολλαπλών τύπων αλλά και η επίδραση της χρήσης πολυτροπικών δεδομένων για το σκοπό αυτό. Συγκεκριμένα, μελετάται η απόδοση των τριών μοντέλων μηχανικής μάθησης σε κοινό σύνολο δεδομένων ελέγχου για δύο περιπτώσεις. Στην πρώτη περίπτωση η εκπαίδευση των μοντέλων γίνεται μόνο με δεδομένα χρόνου απόκρισης των υπηρεσιών (δεδομένα μετρικών), ενώ στη δεύτερη χρησιμοποιείται ένα εμπλουτισμένο σύνολο δεδομένων στο οποίο αξιοποιούνται αρχεία καταγραφής, ίχνη και μετρικές. Το κοινό σύνολο πάνω στο οποίο εκτελείται ο εντοπισμός ανωμαλιών περιέχει ανωμαλίες και από τα τρία είδη που παρουσιάστηκαν (Random Slowdowns, DB connection Limit, Random Exceptions).

Η διαδικασία ανίχνευσης ανωμαλιών στην περίπτωση που χρησιμοποιείται μόνο η μετρική χρόνου απόκρισης είναι αντίστοιχη αυτής που περιγράφηκε στο προηγούμενο πείραμα για τις ανωμαλίες τύπου Random Slowdowns. Όπως και στις προηγούμενες περιπτώσεις, πραγματοποιείται αναζήτηση βέλτιστων παραμέτρων χρησιμοποιώντας κατάλληλα υποσύνολα δεδομένων για το κάθε μοντέλο. Η αξιολόγηση των συνδυασμών παραμέτρων γίνεται πάνω σε σύνολο δεδομένων αποτελούμενο από 920 εγγραφές στις οποίες περιέχονται ανώμαλα δείγματα σε ποσοστό 13.5%. Οι παράμετροι που προκύπτουν και εν τέλει χρησιμοποιούνται για την αρχικοποίηση των τριών μοντέλων παρουσιάζονται (Πίνακας 4.8, Πίνακας 4.9, Πίνακας 4.10).

Πίνακας 4.8: Πείραμα 2, Δεδομένα Latency, Παράμετροι Μοντέλου Birch

Παράμετροι Μοντέλου BIRCH	
n_clusters	None
threshold	0.6
branching_factor	20
anomaly_percentile	96

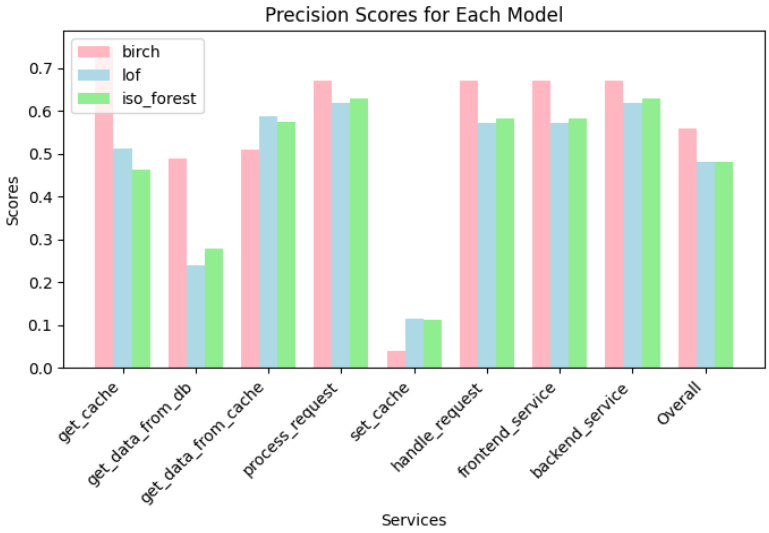
Πίνακας 4.9: Πείραμα 2, Δεδομένα Latency, Παράμετροι Μοντέλου LOF

Παράμετροι Μοντέλου LOF	
algorithm	auto
metric	manhattan
leaf_size	20
n_neighbors	100
contamination	0.13

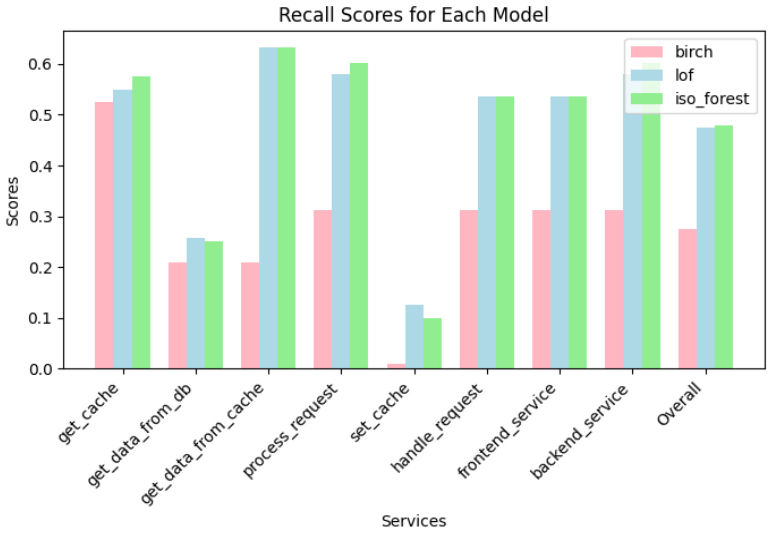
Πίνακας 4.10: Πείραμα 2, Δεδομένα Latency, Παράμετροι Μοντέλου iForest

Παράμετροι Μοντέλου iForest	
bootstrap	True
max_samples	50
n_estimators	100
contamination	0.13
n_jobs	-1
random_state	42

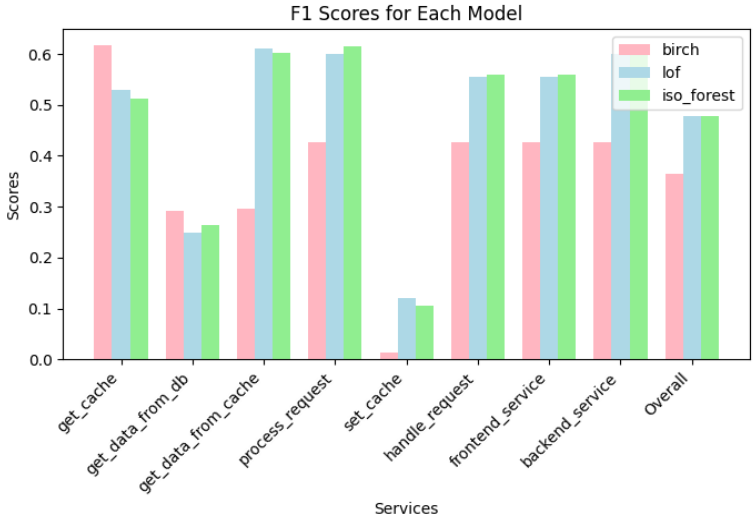
Για την εκπαίδευση του μοντέλου Birch, όπως και στις προηγούμενες περιπτώσεις χρησιμοποιούνται δεδομένα χωρίς προσθήκη ανωμαλιών. Τα δεδομένα αυτά αποτελούνται μόνο από τις χρονοσειρές της μετρικής χρόνου απόκρισης για τις υπηρεσίες της εφαρμογής. Το μοντέλο Isolation Forest εκπαιδεύεται χρησιμοποιώντας δεδομένα με προσθήκη ανωμαλιών. Το σύνολο δεδομένων εκπαίδευσης είναι αντίστοιχο του συνόλου ελέγχου, με παρουσία και των τριών ειδών ανωμαλιών οι οποίες καταλαμβάνουν το 13.5% των 920 εγγραφών. Τέλος, το μοντέλο LOF εκπαιδεύεται και εντοπίζει τις ανωμαλίες πάνω στο κοινό σύνολο δεδομένων ελέγχου. Τα δεδομένα ελέγχου αποτελούνται μόνο από χρονοσειρές της μετρικής χρόνου απόκρισης τα οποία όμως περιέχουν μίξη ανωμαλιών από τις τρεις κατηγορίες με δυνατότητα εμφύτευσης. Συγκεκριμένα κάθε μοντέλο πραγματοποιεί τις προβλέψεις του σε σύνολο 980 εγγραφών (13.2% εκ των οποίων αντιστοιχούν σε ανωμαλίες). Παρατίθενται τα αποτελέσματα της επίδοσης των τριών μοντέλων με βάση τις μετρικές αξιολόγησης Precision (Σχήμα 4.7), Recall (Σχήμα 4.8) και F1-Score (Σχήμα 4.9).



Σχήμα 4.7: Πείραμα 2, Δεδομένα Latency, Αποτελέσματα Μετρικής Precision



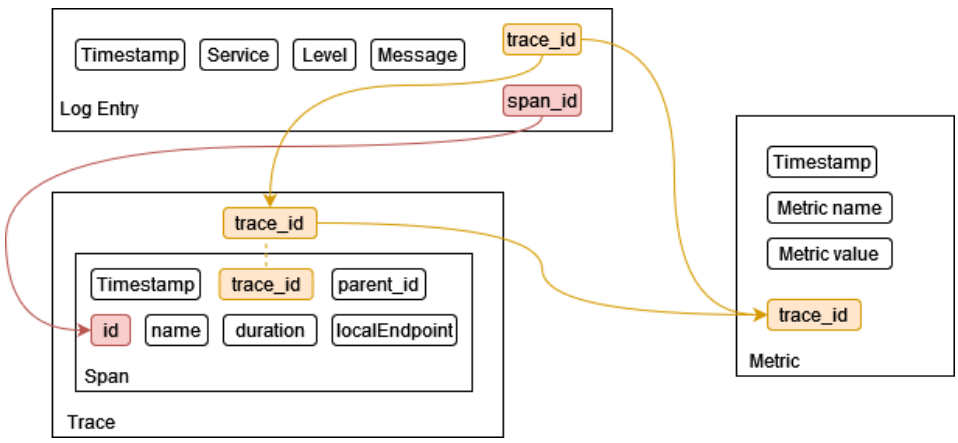
Σχήμα 4.8: Πείραμα 2, Δεδομένα Latency, Αποτελέσματα Μετρικής Recall



Σχήμα 4.9: Πείραμα 2, Δεδομένα Latency, Αποτελέσματα Μετρικής F1-Score

Η απόδοση των μοντέλων στα δεδομένα, όπως αυτή παρατηρείται, δεν θα μπορούσε να χαρακτηριστεί ικανοποιητική. Όσον αφορά τη μετρική Precision, κανένα μοντέλο δεν πέρασε την τιμή 75% με τη μεγάλη πλειοψηφία των αποτελεσμάτων να παρατηρείται κάτω και από το 65%. Από τα τρία μοντέλα οριακά καλύτερο εμφανίζεται το Birch με μέσο Precision = 55.97%. Αντίστοιχα αποτελέσματα παρατηρούνται και για τη μετρική Recall, με κανένα μοντέλο να φτάνει επίδοση μεγαλύτερη του 63%. Στην περίπτωση της μετρικής αυτής το μοντέλο Birch έχει σαφώς κατώτερη απόδοση από τα LOF και iForest, πετυχαίνοντας μόλις 27.45%. Συνολικά, τα άλλα δύο μοντέλα διατηρούν επιδόσεις αντίστοιχες με τη μετρική Precision με τιμές κοντά στο 48%. Κρίνοντας από τη μετρική F1-Score, οριακά καλύτερο προκύπτει το μοντέλο iForest με F1 = 47.9% έναντι F1 = 47.77% του LOF.

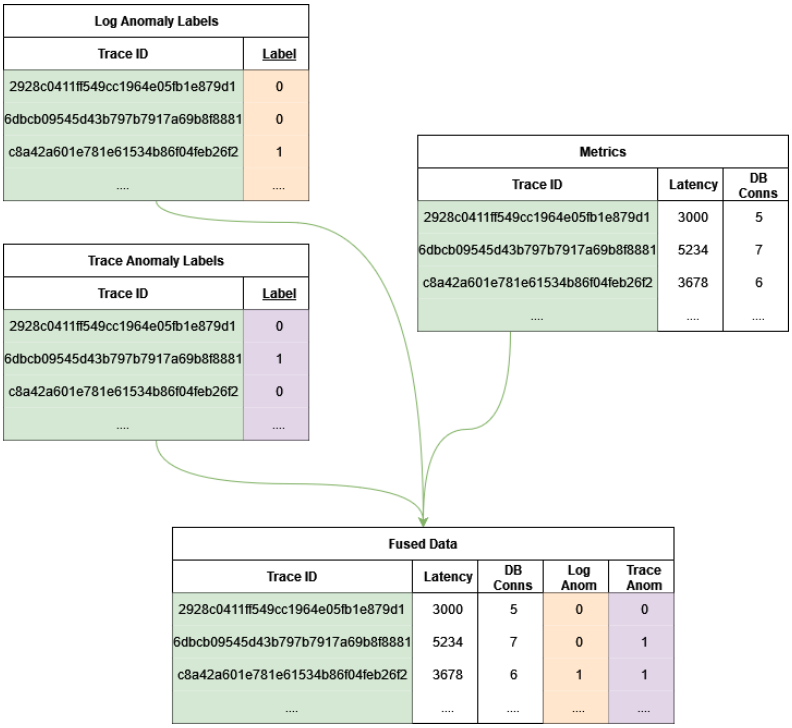
Στο δεύτερο μέρος του πειράματος επαναλαμβάνεται η παραπάνω διαδικασία, χρησιμοποιώντας όμως πολυτροπικά δεδομένα κατά την εκπαίδευση και τον εντοπισμό ανωμαλιών. Με τον τρόπο αυτό αξιοποιείται η πληροφορία που προέρχεται από τις πολλαπλές πηγές παρατηρησιμότητας που παρέχονται από την εφαρμογή. Η διαδικασία εμπλουτισμού έχει ως εξής: αρχικά εντοπίζονται οι ανωμαλίες στα δεδομένα αρχείων καταγραφής και ιχνών εκτέλεσης ακολουθώντας τη διαδικασία που περιγράφεται στο κεφάλαιο 4.1.4 (logs, traces). Για τη διαδικασία της εκπαίδευσης χρησιμοποιούνται δεδομένα συγκεντρωμένα κατά τη διάρκεια εκτέλεσης της εφαρμογής σε κανονική λειτουργία και για τον εντοπισμό ανωμαλιών χρησιμοποιούνται αρχεία καταγραφής και ίχνη από εκτέλεση στην οποία πραγματοποιήθηκε εμφύτευση πολλαπλών ειδών ανωμαλιών. Ενδεικτικά, το ποσοστό ανωμαλιών που είναι γνωστό ότι περιέχεται στα δεδομένα αυτά ανέρχεται στο 5%. Τα αποτελέσματα συνδέονται με τις εγγραφές των χρονοσειρών των μετρικών (χρόνος απόκρισης, αριθμός συνδέσεων στη βάση δεδομένων) με βάση αναγνωριστικά που περιλαμβάνονται σε όλες τις δομές δεδομένων (Σχήμα 4.10).



Σχήμα 4.10: Συσχέτιση Δεδομένων Παρατηρησιμότητας

Βασική μονάδα στη διαδικασία συσχέτισης των εγγραφών είναι το αναγνωριστικό ίχνους. Η πληροφορία αυτή περιλαμβάνεται στα δεδομένα κάθε είδους συνδέοντάς τα μεταξύ τους. Οι ετικέτες πρόβλεψης για τα δεδομένα αρχείων καταγραφής και τα ίχνη εκτέλεσης αποθηκεύονται σε συνδυασμό με το αναγνωριστικό ίχνους που τους αντιστοιχεί. Τα δεδομένα μετρικών ενοποιούνται σε μία ενιαία χρονοσειρά ακολουθώντας τη διαδικασία

δειγματοληψίας όπως αυτή περιγράφεται. Στα ενοποιημένα δεδομένα μετρικών προστίθενται οι επισημάνσεις ανωμαλιών που έχουν προκύψει, σύμφωνα με το αναγνωριστικό ίχνους της μετρικής και το αντίστοιχο της επισήμανσης (Σχήμα 4.11).



Σχήμα 4.11: Διαδικασία Ενοποίησης Δεδομένων

Όπως και πριν, εκτελείται αναζήτηση βέλτιστων υπερπαραμέτρων για τα τρία μοντέλα, χρησιμοποιώντας όμως τα εμπλουτισμένα δεδομένα. Οι βέλτιστοι συνδυασμοί που προκύπτουν και χρησιμοποιούνται για την αρχικοποίηση των μοντέλων παρουσιάζονται παρακάτω (Πίνακας 4.11, Πίνακας 4.12, Πίνακας 4.13).

Πίνακας 4.11: Πείραμα 2, Μεικτά Δεδομένα, Παράμετροι Μοντέλου Birch

Παράμετροι Μοντέλου BIRCH	
n_clusters	None
threshold	0.6
branching_factor	20
anomaly_percentile	99

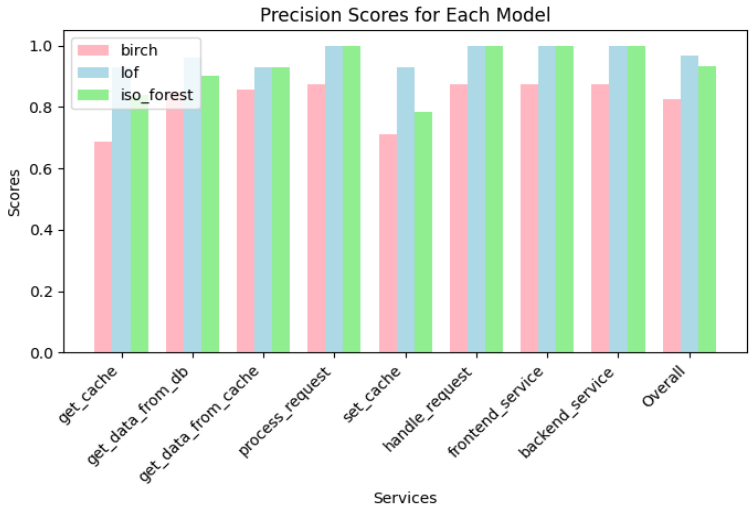
Πίνακας 4.12: Πείραμα 2, Μεικτά Δεδομένα, Παράμετροι Μοντέλου LOF

Παράμετροι Μοντέλου LOF	
algorithm	auto
metric	manhattan
leaf_size	20
n_neighbors	200
contamination	0.13

Πίνακας 4.13: Πείραμα 2, Μεικτά Δεδομένα, Παράμετροι Μοντέλου iForest

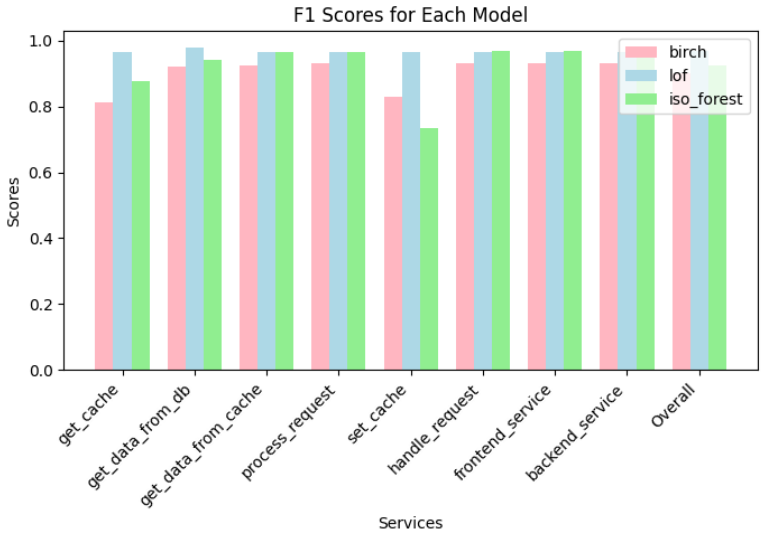
Παράμετροι Μοντέλου iForest	
bootstrap	True
max_samples	100
n_estimators	300
contamination	0.13
n_jobs	-1
random_state	42

Τα τρία μοντέλα ανιχνεύουν ανωμαλίες στο ίδιο σύνολο δεδομένων με το πρώτο μέρος του πειράματος. Ακολουθούν τα αποτελέσματα της ανίχνευσης με βάση τις μετρικές αξιολόγησης που χρησιμοποιούνται (Σχήμα 4.12, Σχήμα 4.13, Σχήμα 4.14).



Σχήμα 4.12: Πείραμα 2, Μεικτά Δεδομένα, Αποτελέσματα Μετρικής Precision

Σχήμα 4.13: Πείραμα 2, Μεικτά Δεδομένα, Αποτελέσματα Μετρικής Recall



Σχήμα 4.14: Πείραμα 2, Μεικτά Δεδομένα, Αποτελέσματα Μετρικής F1-Score

Όπως φαίνεται από τα παραπάνω διαγράμματα, τα μοντέλα μηχανικής μάθησης αποδίδουν σαφώς καλύτερα όταν χρησιμοποιούνται τα εμπλουτισμένα δεδομένα. Ξεκινώντας από τη μετρική Precision, όλα τα μοντέλα παρουσιάζουν τιμές μεγαλύτερες του 68%, με τα LOF και iForest να μην έχουν καμία τιμή κάτω του 93% και 78% αντίστοιχα. Τα καλύτερα αποτελέσματα κατά μέσο όρο πετυχαίνει το μοντέλο LOF με Precision = 96.88%, ακολουθούμενο από τα iForest και Birch με Precision = 93.17% και 82.47% αντίστοιχα. Στη μετρική Recall, τα αποτελέσματα είναι εξίσου καλά αν όχι καλύτερα, με όλα τα μοντέλα να πετυχαίνουν μέσο Recall πάνω από 90%. Το καλύτερο αποτέλεσμα έχει το μοντέλο Birch με Recall = 100%, ακολουθούμενο από τα LOF και iForest με Recall = 96.74% και 91.82% αντίστοιχα. Μελετώντας τα αποτελέσματα για τη μετρική F1 διαπιστώνεται ότι και τα τρία μοντέλα πετυχαίνουν πολύ καλή ισορροπία μεταξύ των μετρικών Precision και Recall, με όλα τους να έχουν F1 > 90%, την καλύτερη απόδοση από τα τρία όμως εμφανίζει το μοντέλο LOF, το οποίο παρουσιάζει μέσο F1-Score = 96.7% (για το Birch παρατηρείται F1 = 90.2% και για το iForest F1 = 92.35%).

Σχολιασμός - Παρατηρήσεις

Όπως φαίνεται από τα παραπάνω αποτελέσματα, η ανίχνευση ανωμαλιών χρησιμοποιώντας μόνο δεδομένα χρόνου απόκρισης των υπηρεσιών περιορίζεται σαφώς στην ικανότητα καταγραφής του πλήρους φάσματος των ανώμαλων συμπεριφορών στο σύστημα. Ο χρόνος απόκρισης ως μετρική, μπορεί να δείξει το αποτέλεσμα μιας ανωμαλίας, σπάνια όμως εντοπίζει τη βασική αιτία πίσω από αυτήν ή παρέχει το απαραίτητο πλαίσιο για να κατανοηθεί η εμφάνιση της ανωμαλίας. Η προσέγγιση αυτή εμφανίζει σημαντικές διακυμάνσεις στην απόδοσή της από υπηρεσία σε υπηρεσία. Σε ορισμένες υπηρεσίες, τα μοντέλα έχουν σχετικά καλύτερη απόδοση, υποδηλώνοντας ότι οι ανωμαλίες εκδηλώνονται με σαφή τρόπο στα δεδομένα των υπηρεσιών αυτών. Σε άλλες υπηρεσίες όμως, με βασικό παράδειγμα την υπηρεσία set_cache τα μοντέλα φαίνεται να δυσκολεύονται, γεγονός που δείχνει ότι η καθυστέρηση από μόνη της μπορεί να κρύβει πιο σύνθετες ανωμαλίες. Αυτές θα μπορούσαν να συνδέονται με συμφόρηση του συστήματος ή με αλληλεπιδράσεις πολλαπλών υπηρεσιών, τις οποίες η καθυστέρηση από μόνη της δεν μπορεί να συλλάβει. Στη συγκεκριμένη περίπτωση, η ύπαρξη ανωμαλιών που σχετίζονται με σφάλματα του κώδικα της εφαρμογής που εκδηλώνονται σε άλλες υπηρεσίες, δεν εκφράζεται στο χρόνο απόκρισης της υπηρεσίας set_cache και ο εντοπισμός τέτοιων ανωμαλιών δεν είναι εφικτός μόνο με χρήση της μετρικής χρόνου απόκρισης. Ενώ οι μετρικές όπως ο χρόνος απόκρισης είναι πολύτιμες για τον εντοπισμό της υποβάθμισης των επιδόσεων, συχνά δεν έχουν βάθος πληροφορίας πλαισίου. Οι αιχμές στο χρόνο απόκρισης μπορεί να οφείλονται σε διάφορους υποκείμενους λόγους, όπως αυξημένος αριθμός αιτημάτων, σφάλματα σε άλλες υπηρεσίες ή εξωτερικές εξαρτήσεις, αλλά η ίδια η μετρική δεν μεταφέρει αυτές τις πληροφορίες. Ως εκ τούτου, τα μοντέλα μηχανικής μάθησης που προσπαθούν να ανιχνεύσουν ανωμαλίες καθαρά από δεδομένα μετρικής χρόνου απόκρισης δυσκολεύονται να συλλάβουν σύνθετες, πλούσιες σε περιεχόμενο ανωμαλίες.

Στο δεύτερο μέρος του πειράματος παρουσιάζονται σημαντικά βελτιωμένα αποτελέσματα σε όλους τους τομείς με τον συνδυασμό δεδομένων αρχείων καταγραφής, ιχνών και μετρικών για την ανίχνευση ανωμαλιών. Αυτή η βελτίωση αντικατοπτρίζει την αξία της πολυτροπικής προσέγγισης, όπου τα πλεονεκτήματα κάθε τύπου δεδομένων συμπληρώνουν

τις αδυναμίες των άλλων. Ο συνδυασμός των δεδομένων οδηγεί σε σταθερά υψηλότερες βαθμολογίες μετρικών σε όλες τις υπηρεσίες, υποδεικνύοντας ότι τα μοντέλα έχουν πλέον πρόσβαση σε πιο ολοκληρωμένες πληροφορίες. Τα αρχεία καταγραφής και τα ίχνη προσθέτουν λεπτομέρειες σε επίπεδο συμβάντων, ενώ οι μετρικές παρέχουν ποσοτικά δεδομένα, επιτρέποντας από κοινού την ανίχνευση πιο διαφοροποιημένων ανωμαλιών. Για παράδειγμα, υπηρεσίες στις οποίες παρατηρούνταν κακές επιδόσεις όταν η ανίχνευση ανωμαλιών βασιζόταν μόνο σε δεδομένα χρόνου απόκρισης, όπως η υπηρεσία set_cache, εμφανίζουν τώρα σημαντικά καλύτερες επιδόσεις. Οι ανωμαλίες τύπου Random Exceptions που περιλαμβάνονταν στην υπηρεσία αυτή δεν σχετίζονταν καθαρά με το χρόνο απόκρισης, αλλά αφορούσαν μοτίβα συμπεριφοράς που μπορούσαν να παρατηρηθούν μόνο στα αρχεία καταγραφής ή τα ίχνη. Παρέχοντας την πληροφορία αυτή στα μοντέλα, είχαν τη δυνατότητα επίτευξης βελτιωμένων επιδόσεων.

Η σύγκριση των αποτελεσμάτων μεταξύ των πηγών δεδομένων καταδεικνύει τη σημασία της ποικιλομορφίας των δεδομένων στην ανίχνευση ανωμαλιών. Ενώ οι μετρικές είναι χρήσιμες για τον εντοπισμό γενικών προβλημάτων απόδοσης, δεν διαθέτουν το βάθος που απαιτείται για την κατανόηση σύνθετων ανωμαλιών του συστήματος. Όταν τα αρχεία καταγραφής, τα ίχνη και οι μετρικές χρησιμοποιούνται μαζί, το σύστημα ανίχνευσης γίνεται πολύ πιο ολοκληρωμένο και ακριβές, αξιοποιώντας τα πλεονεκτήματα κάθε τύπου δεδομένων. Αυτή η συνδυαστική προσέγγιση όχι μόνο βελτιώνει τις συνολικές επιδόσεις, αλλά καθιστά επίσης το σύστημα πιο ανθεκτικό σε πολύπλοκες, πολυδιάστατες ανωμαλίες, οι οποίες είναι όλο και πιο συχνές στις σύγχρονες κατανεμημένες αρχιτεκτονικές. Τα αποτελέσματα του πειράματος δείχνουν ότι μια ολιστική, πολυτροπική στρατηγική είναι το κλειδί για την επίτευξη υψηλών επιδόσεων ανίχνευσης ανωμαλιών σε διάφορες υπηρεσίες και τύπους ανωμαλιών.

Κεφάλαιο 5 - Συμπεράσματα και Μελλοντική Επέκταση

5.1 Επίλογος

Σε αυτή τη διπλωματική εργασία μελετάται η αποδοτικότητα διαφόρων μεθόδων στην πολυτροπική ανίχνευση ανωμαλιών σε περιβάλλοντα κατανεμημένων εφαρμογών, χρησιμοποιώντας δεδομένα από αρχεία καταγραφής, ίχνη εκτέλεσης και μετρικές. Αρχικά παρουσιάζεται μια σύντομη επεξήγηση των βασικών όρων και εννοιών του πεδίου αυτού, καλύπτοντας τεχνολογίες που εφαρμόζονται στην προσπάθεια επίλυσης του προβλήματος που μελετάται. Στη συνέχεια ακολουθεί εκτενής ανασκόπηση της σχετικής βιβλιογραφίας, στην οποία παρουσιάζονται και συγκρίνονται μέθοδοι εντοπισμού ανωμαλιών σε περιβάλλοντα κατανεμημένων εφαρμογών χρησιμοποιώντας ένα ή περισσότερα από τα είδη δεδομένων που αναφέρθηκαν. Η έμφαση δόθηκε σε προσεγγίσεις που αξιοποιούν μεθόδους Μηχανικής Μάθησης και στατιστικές μεθόδους, οι οποίες έχουν διερευνηθεί ευρέως, αλλά συχνά με αποσπασματικό τρόπο, βασιζόμενες σε έναν μόνο τύπο δεδομένων παρατηρησιμότητας.

Παρατηρώντας ότι η χρήση πολυτροπικών δεδομένων, με άλλα λόγια ο συνδυασμός πολλαπλών πηγών παρατηρησιμότητας, παραμένει ένας τομέας του πεδίου με σημαντικά περιθώρια περαιτέρω διερεύνησης, στην εργασία αυτή προτείνεται ένα σύστημα ανίχνευσης ανωμαλιών που ενσωματώνει δεδομένα αρχείων καταγραφής, ιχνών εκτέλεσης και μετρικών απόδοσης. Η προσέγγιση αυτή αντιμετωπίζει το κενό που παρατηρήθηκε κατά τη διάρκεια της βιβλιογραφικής ανασκόπησης: την έλλειψη συνεπών τυποποιημένων συνόλων δεδομένων προσαρμοσμένων για την ανίχνευση πολυτροπικών ανωμαλιών. Για την κάλυψη αυτού του κενού, παρουσιάζεται η υλοποίηση μιας ολοκληρωμένης διαδικτυακής εφαρμογής που προσομοιώνει εφαρμογές που εκτελούνται σε κατανεμημένα περιβάλλοντα. Η εφαρμογή αυτή, σε συνδυασμό με ένα σύνολο εργαλείων παρατηρησιμότητας παρέχει όλα τα εφόδια για την παραγωγή, τη συλλογή και την αποθήκευση των απαραίτητων δεδομένων. Τα δεδομένα αυτά, αφού υποβληθούν σε κατάλληλη επεξεργασία, θα αποτελέσουν τη βάση του συνόλου δεδομένων που χρησιμοποιείται για την ανίχνευση ανωμαλιών στα πειράματα που ακολουθούν.

Στο πλαίσιο της εργασίας προτείνεται η χρήση διαφορετικών τεχνικών για την ανίχνευση ανωμαλιών για κάθε τύπο δεδομένων και ο μετέπειτα συνδυασμός τους σε ένα ενιαίο μοντέλο για τον εντοπισμό ανωμαλιών. Οι τεχνικές που παρουσιάζονται περιλαμβάνουν συστήματα βασισμένα σε σύνολα κανόνων για τα δεδομένα από τα αρχεία καταγραφής και τα ίχνη και τα μοντέλα μηχανικής μάθησης Birch, Local Outlier Factor και Isolation Forest για τα δεδομένα μετρικών. Η επιλογή αυτών των μοντέλων ήταν σκόπιμη, καθώς το καθένα αντιπροσωπεύει μια διαφορετική προσέγγιση για τον εντοπισμό ανωμαλιών, επιτρέποντας έτσι μια συγκριτική ανάλυση με στόχο τον προσδιορισμό της πιο αποτελεσματικής μεθόδου. Κάθε μοντέλο υποβλήθηκε σε βελτιστοποίηση παραμέτρων, εξασφαλίζοντας μια δίκαιη αξιολόγηση των επιδόσεών τους σε ένα κοινό σύνολο δεδομένων. Από τα πειράματα αξιολόγησης των προτεινόμενων μεθόδων προκύπτει το συμπέρασμα ότι αν και ο εντοπισμός ανωμαλιών στα επιμέρους δεδομένα εύκολα πετυχαίνει άριστα αποτελέσματα, αυτή δεν είναι πάντα η περίπτωση για πιο σύνθετες και λεπτές ανωμαλίες. Ωστόσο, ο συνδυασμός δεδομένων

παρατηρησιμότητας από πολλαπλές πηγές, οδήγησε σε σημαντική βελτίωση της απόδοσης ανίχνευσης, προσφέροντας μια πιο ολοκληρωμένη εικόνα της υγείας του συστήματος. Η σύγκριση μεταξύ των μοντέλων αποκάλυψε ελάχιστες διαφορές στις επιδόσεις όταν χρησιμοποιήθηκαν πολυτροπικά δεδομένα, αναδεικνύοντας την ευρωστία της ολοκληρωμένης προσέγγισης. Αυτό αποδεικνύει την αξία της πολυτροπικής ανίχνευσης ανωμαλιών, επιβεβαιώνοντας ότι η αξιοποίηση πολλαπλών τύπων δεδομένων παρέχει μια πιο ολιστική κατανόηση της συμπεριφοράς του συστήματος και ενισχύει την ικανότητα ανίχνευσης σύνθετων ανωμαλιών.

5.2 Μελλοντικές Επεκτάσεις

Η παρούσα διπλωματική εργασία παρουσίασε μια πολλά υποσχόμενη προσέγγιση για την πολυτροπική ανίχνευση ανωμαλιών με τη χρήση αρχείων καταγραφής, ιχνών και μετρικών σε κατανεμημένα συστήματα, παρέχοντας ένα σύστημα για την παραγωγή, την αποθήκευση και την επεξεργασία των δεδομένων αυτών, δεν παύει όμως να αποτελεί μια proof-of-concept προσέγγιση της ροής εργασίας στην οποία αναφέρεται, αφήνοντας αρκετούς τομείς ώριμους για περαιτέρω διερεύνηση και βελτίωση.

Μία από τις κύριες προκλήσεις κατά τη διάρκεια της εργασίας ήταν η έλλειψη τυποποιημένων συνόλων δεδομένων που ενσωματώνουν πολλαπλές πηγές παρατηρησιμότητας για την ανίχνευση ανωμαλιών. Το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διατριβή δημιουργήθηκε μέσω μιας προσομοιωμένης διαδικτυακής εφαρμογής, αλλά ένα ευρύτερο, τυποποιημένο σύνολο δεδομένων που αντικατοπτρίζει τις συμπεριφορές συστημάτων του πραγματικού κόσμου σε διάφορους κλάδους θα ήταν ανεκτίμητο για τη μελλοντική έρευνα. Η δημιουργία και η επιμέλεια τέτοιων συνόλων δεδομένων όχι μόνο θα βελτίωνε τη συνοχή των αξιολογήσεων μεταξύ των μελετών, αλλά θα επέτρεπε επίσης στους ερευνητές να συγκρίνουν τα αποτελέσματά τους με πιο διαφορετικά, ρεαλιστικά περιβάλλοντα. Οι μελλοντικές εργασίες θα μπορούσαν να επικεντρωθούν στην ανάπτυξη συνόλων δεδομένων ανοικτού κώδικα που αντικατοπτρίζουν ένα ευρύ φάσμα συμπεριφορών κατανεμημένων εφαρμογών, συμπεριλαμβανομένων σεναρίων έγχυσης σφαλμάτων, γεγονός που θα μπορούσε να προωθήσει σημαντικά την έρευνα στον τομέα αυτό.

Η ικανότητα κλιμακωσιμότητας των μεθόδων εντοπισμού ανωμαλιών, ώστε αυτά να μπορούν να διαχειριστούν το συνεχώς αυξανόμενο μέγεθος και την πολυπλοκότητα των σύγχρονων κατανεμημένων συστημάτων είναι ένας άλλος κρίσιμος τομέας για μελλοντική διερεύνηση. Καθώς οι κατανεμημένες αρχιτεκτονικές επεκτείνονται, ο όγκος των παραγόμενων δεδομένων παρατηρησιμότητας αυξάνεται εκθετικά, δημιουργώντας προκλήσεις όσον αφορά το χρόνο επεξεργασίας και την έγκαιρη ανίχνευση ανωμαλιών σε πραγματικό χρόνο. Η βελτιστοποίηση των υπολογιστικών απαιτήσεων των μεθόδων πολυτροπικού εντοπισμού ανωμαλιών διατηρώντας ακριβή αποτελέσματα αποτελεί μία λεπτή ισορροπία. Οι μελλοντικές εργασίες θα μπορούσαν να διερευνήσουν τη χρήση τεχνικών κατανεμημένου υπολογισμού όπως η παράλληλη επεξεργασία, σε συνδυασμό με την αξιοποίηση υποδομών νέφους, για τη δημιουργία συστημάτων ικανών να κλιμακώσουν τις μεθόδους ανίχνευσης ανωμαλιών ώστε να ανταποκρίνονται στις απαιτήσεις μεγαλύτερων συνόλων δεδομένων και πιο περίπλοκων περιβαλλόντων.

Τα κατανεμημένα συστήματα είναι ιδιαίτερα δυναμικά, με τα συστατικά στοιχεία να αλλάζουν συχνά και τις συμπεριφορές να εξελίσσονται με την πάροδο του χρόνου. Τα τρέχοντα μοντέλα ανίχνευσης ανωμαλιών συχνά δυσκολεύονται να προσαρμοστούν σε αυτές τις αλλαγές, οδηγώντας σε προβλήματα όπως ψευδώς θετικά αποτελέσματα ή ανωμαλίες που χάνονται όταν το μοντέλο έχει καταστεί απαρχαιωμένο. Οι μελλοντικές εργασίες θα πρέπει να επικεντρωθούν στην ανάπτυξη προσαρμοστικών μοντέλων που μπορούν να ενημερώνονται και να εξελίσσονται αυτόματα με βάση τις αλλαγές στη συμπεριφορά του συστήματος. Τεχνικές όπως η συνεχής μάθηση (continual learning) [81], η διαδικτυακή μάθηση (online learning) [82] και η ενισχυτική μάθηση (reinforcement learning) [83] θα μπορούσαν να διερευνηθούν για τη δημιουργία συστημάτων που προσαρμόζουν τις στρατηγικές ανίχνευσης σε πραγματικό χρόνο καθώς αναδύονται νέα πρότυπα, χωρίς την ανάγκη συνεχούς επανεκπαίδευσης.

Καθώς η τεχνολογία εξελίσσεται, νέες μορφές δεδομένων παρατηρησιμότητας καθίστανται διαθέσιμες. Μελλοντικές εργασίες θα μπορούσαν να διερευνήσουν την ενσωμάτωση επιπλέον τύπων δεδομένων -όπως δεδομένα κίνησης δικτύου, ανάλυση συμπεριφοράς χρήστη ή δεδομένα συμβάντων ασφαλείας- σε συστήματα πολυτροπικής ανίχνευσης ανωμαλιών. Η ενσωμάτωση πρόσθετων πηγών παρατηρησιμότητας θα μπορούσε να παρέχει μια ακόμη πιο ολοκληρωμένη εικόνα της υγείας του συστήματος και να προσφέρει νέες ευκαιρίες για τον εντοπισμό προηγουμένως μη ανιχνεύσιμων ανωμαλιών. Η έρευνα σχετικά με τον καλύτερο τρόπο συγχώνευσης αυτών των νέων μορφών με τις υπάρχουσες πηγές δεδομένων, διατηρώντας παράλληλα την αποτελεσματικότητα και την ερμηνευσιμότητα, αποτελεί μια πολλά υποσχόμενη κατεύθυνση για περαιτέρω διερεύνηση.

Τέλος, ενώ ο εντοπισμός των ανωμαλιών είναι ζωτικής σημασίας, η ικανότητα εκτέλεσης Ανάλυσης Βασικής Αιτίας (Root Cause Analysis, RCA) [84] για τη διάγνωση των υποκείμενων λόγων μιας ανωμαλίας είναι εξίσου σημαντική για την αξιοπιστία του συστήματος. Η παρούσα διπλωματική εργασία, επικεντρώθηκε στον εντοπισμό ανωμαλιών σε πολλαπλές μορφές δεδομένων, αλλά η μελλοντική έρευνα θα μπορούσε να ενισχύσει αυτό το σύστημα με την ενσωμάτωση δυνατοτήτων RCA. Μόλις εντοπιστεί μια ανωμαλία, ο αυτόματος εντοπισμός της βαθύτερης αιτίας -είτε πρόκειται για μια συγκεκριμένη υπηρεσία, ένα πρόβλημα δικτύου ή μια συμφόρηση λόγω έλλειψης πόρων- μπορεί να μειώσει σημαντικά τον μέσο χρόνο αποκατάστασης (mean time to recovery, MTTR) σε κατανεμημένα συστήματα. Οι μελλοντικές εργασίες θα μπορούσαν να διερευνήσουν την αξιοποίηση τεχνικών εξαγωγής σχέσεων αιτιατότητας (causality), χαρτογράφησης εξαρτήσεων και ανάλυσης συσχέτισης για τον αποτελεσματικότερο εντοπισμό της προέλευσης των ανωμαλιών. Με την ενσωμάτωση της RCA στο σύστημα ανίχνευσης ανωμαλιών, οι μηχανικοί όχι μόνο θα ειδοποιούνται για την παρουσία ενός προβλήματος, αλλά θα τους παρέχονται επίσης πληροφορίες σχετικά με την πηγή του προβλήματος, επιτρέποντας ταχύτερη και αποτελεσματικότερη αποκατάσταση.

Συμπερασματικά, ο τομέας της πολυτροπικής ανίχνευσης ανωμαλιών παραμένει πλούσιος σε δυνατότητες καινοτομίας. Αν και η παρούσα εργασία έθεσε τις βάσεις αποδεικνύοντας την αποτελεσματικότητα της ενσωμάτωσης των αρχείων καταγραφής, των ιχνών και των μετρικών, η περαιτέρω πρόοδος στην τυποποίηση δεδομένων, την κλιμακωσιμότητα, την ανάλυση αιτιών πίσω από τις ανωμαλίες και τη συγχώνευση

περισσότερων τύπων δεδομένων θα είναι απαραίτητη για την πλήρη αξιοποίηση των δυνατοτήτων αυτών των τεχνικών σε πραγματικές εφαρμογές.

Βιβλιογραφία

[1] “What are microservices?,” microservices.io. Accessed: Oct. 02, 2024. [Online]. Available: http://microservices.io/index.html

[2] “What Is Anomaly Detection? | IBM.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.ibm.com/topics/anomaly-detection

[3] “What Is Machine Learning (ML)? | IBM.” Accessed: Sep. 26, 2024. [Online]. Available: https://www.ibm.com/topics/machine-learning

[4] “What Is Artificial Intelligence (AI)? | IBM.” Accessed: Sep. 26, 2024. [Online]. Available: https://www.ibm.com/topics/artificial-intelligence

[5] “What is Observability? | IBM.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.ibm.com/topics/observability

[6] A. Nandi, A. Mandal, S. Atreja, G. B. Dasgupta, and S. Bhattacharya, “Anomaly Detection Using Program Control Flow Graph Mining From Execution Logs,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco California USA: ACM, Aug. 2016, pp. 215–224. doi: 10.1145/2939672.2939712.

[7] T. Jia, P. Chen, L. Yang, Y. Li, F. Meng, and J. Xu, “An Approach for Anomaly Diagnosis Based on Hybrid Graph Model with Logs for Distributed Services,” in 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA: IEEE, Jun. 2017, pp. 25–32. doi: 10.1109/ICWS.2017.12.

[8] T. Jia, L. Yang, P. Chen, Y. Li, F. Meng, and J. Xu, “LogSed: Anomaly Diagnosis through Mining Time-Weighted Control Flow Graph in Logs,” in 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honolulu, CA, USA: IEEE, Jun. 2017, pp. 447–455. doi: 10.1109/CLOUD.2017.64.

[9] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning,” in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas Texas USA: ACM, Oct. 2017, pp. 1285–1298. doi: 10.1145/3133956.3134015.

[10] W. Meng et al., “LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs,” in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 4739–4745. doi: 10.24963/ijcai.2019/658.

[11] X. Zhang et al., “Robust log-based anomaly detection on unstable log data,” in Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn Estonia: ACM, Aug. 2019, pp. 807–817. doi: 10.1145/3338906.3338931.

[12] S. Lu, X. Wei, Y. Li, and L. Wang, “Detecting Anomaly in Big Data System Logs Using Convolutional Neural Network,” in 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Athens: IEEE, Aug. 2018, pp. 151–158. doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00037.

[13] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” 2013, arXiv. doi: 10.48550/ARXIV.1301.3781.
[16] C. Sammut and G. I. Webb, Eds., “TF–IDF,” in Encyclopedia of Machine Learning, Boston, MA: Springer US, 2011, pp. 986–987. doi: 10.1007/978-0-387-30164-8_832.
[17] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, “Detecting large-scale system problems by mining console logs,” in Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, Big Sky Montana USA: ACM, Oct. 2009, pp. 117–132. doi: 10.1145/1629575.1629587.
[18] J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, “Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics,” Sep. 12, 2023, arXiv: arXiv:2008.06448. Accessed: Oct. 04, 2024. [Online]. Available: http://arxiv.org/abs/2008.06448
[19] L. Yang et al., “PLELog: Semi-Supervised Log-Based Anomaly Detection via Probabilistic Label Estimation,” in 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Madrid, ES: IEEE, May 2021, pp. 230–231. doi: 10.1109/ICSE-Companion52605.2021.00106.
[20] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” Sep. 02, 2014, arXiv: arXiv:1406.1078. Accessed: Oct. 04, 2024. [Online]. Available: http://arxiv.org/abs/1406.1078
[21] L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical density based clustering,” J. Open Source Softw., vol. 2, no. 11, p. 205, Mar. 2017, doi: 10.21105/joss.00205.
[22] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network,” Phys. Nonlinear Phenom., vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.
[23] S. Nedelkoski, J. Cardoso, and O. Kao, “Anomaly Detection from System Tracing Data Using Multimodal Deep Learning,” in 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy: IEEE, Jul. 2019, pp. 179–186. doi: 10.1109/CLOUD.2019.00038.
[24] P. Liu et al., “Unsupervised Detection of Microservice Trace Anomalies through Service-Level Deep Bayesian Networks,” in 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), Coimbra, Portugal: IEEE, Oct. 2020, pp. 48–58. doi: 10.1109/ISSRE5003.2020.00014.
[25] Jamell Ivor Samuels, “One-Hot Encoding and Two-Hot Encoding: An Introduction,” 2024, doi: 10.13140/RG.2.2.21459.76327.
[26] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” Dec. 10, 2022, arXiv: arXiv:1312.6114. Accessed: Oct. 04, 2024. [Online]. Available: http://arxiv.org/abs/1312.6114
[27] Y. Gan et al., “Seer: Leveraging Big Data to Navigate the Complexity of Performance Debugging in Cloud Microservices,” in Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence RI USA: ACM, Apr. 2019, pp. 19–33. doi:

10.1145/3297858.3304004.
[28] S. Nedelkoski, J. Cardoso, and O. Kao, “Anomaly Detection and Classification using Distributed Tracing and Deep Learning,” in 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Larnaca, Cyprus: IEEE, May 2019, pp. 241–250. doi: 10.1109/CCGRID.2019.00038.
[29] J. Bogatinovski, S. Nedelkoski, J. Cardoso, and O. Kao, “Self-Supervised Anomaly Detection from Distributed Traces,” in 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), Leicester, UK: IEEE, Dec. 2020, pp. 342–347. doi: 10.1109/UCC48980.2020.00054.
[30] X. Zhou et al., “Latent error prediction and fault localization for microservice applications by learning from system trace logs,” in Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn Estonia: ACM, Aug. 2019, pp. 683–694. doi: 10.1145/3338906.3338961.
[31] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” AIChE J., vol. 37, no. 2, pp. 233–243, Feb. 1991, doi: 10.1002/aic.690370209.
[32] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. in Adaptive computation and machine learning. Cambridge, Mass: The MIT press, 2016.
[33] L. Breiman, “Random Forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[34] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN Model-Based Approach in Classification,” in On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, vol. 2888, R. Meersman, Z. Tari, and D. C. Schmidt, Eds., in Lecture Notes in Computer Science, vol. 2888. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 986–996. doi: 10.1007/978-3-540-39964-3_62.
[35] S. S. Haykin, Neural networks: a comprehensive foundation, 2. ed., [Nachdr.]. Upper Saddle River, NJ: Prentice Hall, 1999.
[36] A. Gulenko, F. Schmidt, A. Acker, M. Wallschlager, O. Kao, and F. Liu, “Detecting Anomalous Behavior of Black-Box Services Modeled with Distance-Based Online Clustering,” in 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA: IEEE, Jul. 2018, pp. 912–915. doi: 10.1109/CLOUD.2018.00134.
[37] L. Wu, J. Tordsson, E. Elmroth, and O. Kao, “MicroRCA: Root Cause Localization of Performance Issues in Microservices,” in NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary: IEEE, Apr. 2020, pp. 1–9. doi: 10.1109/NOMS47738.2020.9110353.
[38] L. Wu, J. Bogatinovski, S. Nedelkoski, J. Tordsson, and O. Kao, “Performance Diagnosis in Cloud Microservices Using Deep Learning,” in Service-Oriented Computing – ICSOC 2020 Workshops, vol. 12632, H. Hacid, F. Outay, H. Paik, A. Alloum, M. Petrocchi, M. R. Bouadjenek, A. Beheshti, X. Liu, and A. Maaradji, Eds., in Lecture Notes in Computer Science, vol. 12632. , Cham: Springer International Publishing, 2021, pp. 85–96. doi: 10.1007/978-3-030-76352-7_13.
[39] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method

for very large databases," ACM SIGMOD Rec., vol. 25, no. 2, pp. 103–114, Jun. 1996, doi: 10.1145/235968.233324.
[40] L.Page, S.Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," 1998, [Online]. Available: https://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/pagerank.pdf
[41] Q. Du, T. Xie, and Y. He, "Anomaly Detection and Diagnosis for Container-Based Microservices with Performance Monitoring," in Algorithms and Architectures for Parallel Processing, vol. 11337, J. Vaidya and J. Li, Eds., in Lecture Notes in Computer Science, vol. 11337. , Cham: Springer International Publishing, 2018, pp. 560–572. doi: 10.1007/978-3-030-05063-4_42.
[42] L. Mariani, M. Pezzè, O. Riganelli, and R. Xin, "Predicting failures in multi-tier distributed systems," J. Syst. Softw., vol. 161, p. 110464, Mar. 2020, doi: 10.1016/j.jss.2019.110464.
[43] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," IEEE Intell. Syst. Their Appl., vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.
[44] Vikramkumar, V. B, and Trilochan, "Bayes and Naive Bayes Classifier," Apr. 03, 2014, arXiv: arXiv:1404.0933. Accessed: Oct. 04, 2024. [Online]. Available: http://arxiv.org/abs/1404.0933
[45] L. Mariani, C. Monni, M. Pezze, O. Riganelli, and R. Xin, "Localizing Faults in Cloud Systems," in 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Vasteras: IEEE, Apr. 2018, pp. 262–273. doi: 10.1109/ICST.2018.00034.
[46] A. Samir and C. Pahl, "DLA: Detecting and Localizing Anomalies in Containerized Microservice Architectures Using Markov Models," in 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey: IEEE, Aug. 2019, pp. 205–213. doi: 10.1109/FiCloud.2019.00036.
[47] H. Qiu, Y. Liu, N. A. Subrahmanya, and W. Li, "Granger Causality for Time-Series Anomaly Detection," in 2012 IEEE 12th International Conference on Data Mining, Brussels, Belgium: IEEE, Dec. 2012, pp. 1074–1079. doi: 10.1109/ICDM.2012.73.
[48] S. Jin, Z. Zhang, K. Chakrabarty, and X. Gu, "Accurate anomaly detection using correlation-based time-series analysis in a core router system," in 2016 IEEE International Test Conference (ITC), Fort Worth, TX, USA: IEEE, Nov. 2016, pp. 1–10. doi: 10.1109/TEST.2016.7805836.
[49] H. Lu, Y. Liu, Z. Fei, and C. Guan, "An Outlier Detection Algorithm Based on Cross-Correlation Analysis for Time Series Dataset," IEEE Access, vol. 6, pp. 53593–53610, 2018, doi: 10.1109/ACCESS.2018.2870151.
[50] T. Pitakrat, D. Okanovic, A. Van Hoorn, and L. Grunske, "An Architecture-Aware Approach to Hierarchical Online Failure Prediction," in 2016 12th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA), Venice: IEEE, Apr. 2016, pp. 60–69. doi: 10.1109/QoSA.2016.16.
[51] T. Pitakrat, D. Okanović, A. Van Hoorn, and L. Grunske, "Hora: Architecture-aware online failure prediction," J. Syst. Softw., vol. 137, pp. 669–685, Mar. 2018, doi: 10.1016/j.jss.2017.02.041.

[52] C. Zhang et al., “DeepTraLog: trace-log combined microservice anomaly detection through graph-based deep learning,” in Proceedings of the 44th International Conference on Software Engineering, Pittsburgh Pennsylvania: ACM, May 2022, pp. 623–634. doi: 10.1145/3510003.3510180.
[53] C. Hou, T. Jia, Y. Wu, Y. Li, and J. Han, “Diagnosing Performance Issues in Microservices with Heterogeneous Data Source,” in 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), New York City, NY, USA: IEEE, Sep. 2021, pp. 493–500. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00074.
[54] G. Yu, P. Chen, Y. Li, H. Chen, X. Li, and Z. Zheng, “Nezha: Interpretable Fine-Grained Root Causes Analysis for Microservices on Multi-modal Observability Data,” in Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, San Francisco CA USA: ACM, Nov. 2023, pp. 553–565. doi: 10.1145/3611643.3616249.
[55] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: identifying density-based local outliers,” ACM SIGMOD Rec., vol. 29, no. 2, pp. 93–104, Jun. 2000, doi: 10.1145/335191.335388.
[56] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy: IEEE, Dec. 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.
[57] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, “Drain: An Online Log Parsing Approach with Fixed Depth Tree,” in 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA: IEEE, Jun. 2017, pp. 33–40. doi: 10.1109/ICWS.2017.13.
[58] “Welcome to Python.org,” Python.org. Accessed: Oct. 02, 2024. [Online]. Available: https://www.python.org/
[59] “Welcome to Flask — Flask Documentation (3.0.x).” Accessed: Oct. 02, 2024. [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/
[60] “Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (3.1.x).” Accessed: Oct. 02, 2024. [Online]. Available: https://flask-sqlalchemy.readthedocs.io/en/3.1.x/
[61] “SQLite Home Page.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.sqlite.org/
[62] “Redis - The Real-time Data Platform,” Redis. Accessed: Oct. 02, 2024. [Online]. Available: https://redis.io/
[63] “fluentbit.” Accessed: Oct. 02, 2024. [Online]. Available: https://fluentbit.io/
[64] “Grafana Loki OSS | Log aggregation system,” Grafana Labs. Accessed: Oct. 02, 2024. [Online]. Available: https://grafana.com/oss/loki/
[65] “Grafana: The open observability platform,” Grafana Labs. Accessed: Oct. 02, 2024. [Online]. Available: https://grafana.com/
[66] “OpenTelemetry,” OpenTelemetry. Accessed: Oct. 02, 2024. [Online]. Available: https://opentelemetry.io/
[67] “OpenZipkin · A distributed tracing system.” Accessed: Oct. 02, 2024. [Online]. Available: https://zipkin.io/
[68] Prometheus, “Prometheus - Monitoring system & time series database.” Accessed: Oct.

02, 2024. [Online]. Available: https://prometheus.io/
[69] “Docker: Accelerated Container Application Development.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.docker.com/
[70] “scikit-learn: machine learning in Python — scikit-learn 1.5.2 documentation.” Accessed: Oct. 02, 2024. [Online]. Available: https://scikit-learn.org/stable/
[71] “NumPy -.” Accessed: Oct. 02, 2024. [Online]. Available: https://numpy.org/
[72] “SciPy -.” Accessed: Oct. 02, 2024. [Online]. Available: https://scipy.org/
[73] “Matplotlib — Visualization with Python.” Accessed: Oct. 02, 2024. [Online]. Available: https://matplotlib.org/
[74] “Birch,” scikit-learn. Accessed: Oct. 02, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.cluster.Birch.html
[75] “LocalOutlierFactor,” scikit-learn. Accessed: Oct. 02, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html
[76] “IsolationForest,” scikit-learn. Accessed: Oct. 02, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.ensemble.IsolationForest.html
[77] “GridSearchCV,” scikit-learn. Accessed: Oct. 02, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[78] “RobustScaler,” scikit-learn. Accessed: Oct. 03, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.RobustScaler.html
[79] “StandardScaler,” scikit-learn. Accessed: Oct. 03, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[80] “MinMaxScaler,” scikit-learn. Accessed: Oct. 03, 2024. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[81] L. Wang, X. Zhang, H. Su, and J. Zhu, “A Comprehensive Survey of Continual Learning: Theory, Method and Application,” Feb. 06, 2024, arXiv: arXiv:2302.00487. Accessed: Oct. 06, 2024. [Online]. Available: http://arxiv.org/abs/2302.00487
[82] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, “Online Learning: A Comprehensive Survey,” Oct. 22, 2018, arXiv: arXiv:1802.02871. Accessed: Oct. 06, 2024. [Online]. Available: http://arxiv.org/abs/1802.02871
[83] “What is reinforcement learning? | IBM.” Accessed: Oct. 06, 2024. [Online]. Available: https://www.ibm.com/topics/reinforcement-learning
[84] “What Is a Root Cause Analysis? | IBM.” Accessed: Oct. 06, 2024. [Online]. Available: https://www.ibm.com/topics/root-cause-analysis