Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Ηλεκτρικων Βιομηχανικων Διαταξεων και Συστηματων Αποφασεων

Ταξινόμηση & Προσομοίωση Νοσοκομειακών Δικτυακών Δεδομένων Ροής με χρήση Μοντέλων Μηχανικής Μάθησης

Διπλωματικη Εργασια

του

ΧΡΗΣΤΟΥ ΜΠΕΤΖΕΛΟΥ

Επιβλέπων: Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εργαστηριο Συστηματων Αποφασεων και Διοικησης

Αθήνα, Μάρτιος 2022

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων
Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης

Ταξινόμηση & Προσομοίωση Νοσοκομειακών Δικτυακών Δεδομένων Ροής με χρήση Μοντέλων Μηχανικής Μάθησης

Διπλωματικη Εργασια

του

ΧΡΗΣΤΟΥ ΜΠΕΤΖΕΛΟΥ

Επιβλέπων: Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 8η Μαρτίου 2022.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Δημήτριος Ασκούνης	Ιωάννης Ψαρράς	Χρυσόστομος Δούκας
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2022

(Υπογραφή)

.....

Χρηστος Μπετζελος

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

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων
Εργαστήριο Συστημάτων Αποφάσεων και Διοίκησης

Ευχαριστίες

Ολοκληρώνοντας την εκπόνηση της διπλωματικής μου εργασίας, κλείνει και ένας κύκλος φοίτησης στο Εθνικό Μετσόβιο Πολυτεχνείο. Μια πορεία, κατά την οποία πέρα απο τον προσωπικό μόχθο, συντέλεσαν καθοριστικά και κάποια πρόσωπα. Τα πρόσωπα αυτά, αισθάνομαι την ανάγκη να ευχαριστήσω.

Πρώτον, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Δημήτριο Ασκούνη, ο οποίος με ενέπνευσε να ασχοληθώ με το συγκεκριμένο αντικείμενο, με εμπιστεύτηκε και μου έδωσε την ευκαιρία να εμβαθύνω περαιτέρω. Παράλληλα, ευχαριστώ ιδιαίτερα τον υποψήφιο Δρ. Σωτήριο Πελέκη για τις πολύτιμες συμβουλές που μου παρείχε και για την άψογη συνεργασία που θεμελίωσε, ήδη απο την αρχή. Τέλος, δε θα μπορούσα να παραλείψω και τα υπόλοιπα μέλη του εργαστηρίου, για το θεωρητικό αλλά και πρακτικό υπόβαθρο που μου διαμόρφωσαν, κατά την διάρκεια της προπτυχιακής μου εκπαίδευσης.

Ταυτόχρονα, θα ήθελα να ευχαριστήσω μέσα απο την καρδία μου τους γονείς μου Νικόλαο και Γεωργία, τον αδερφό μου Γεώργιο και γενικότερα την οικογένειά και τους φίλους μου. Η βοήθεια τους ήταν διαρκής, ουσιαστική και πολύτιμη.

Περίληψη

Αντικείμενο της διπλωματικής εργασίας είναι η ανάλυση και η προσομοίωση δικτυακών δεδομένων ροής νοσοκομειακής και υγειονομική περίθαλψης, με χρήση παραγωγικών μοντέλων και μοντέλων μηχανικής μάθησης. Ιδιαίτερα χρήσιμη στην ανάλυση της συμπεριφοράς ενός παρακολουθούμενου δικτύου αποτελεί η δημιουργία προφίλ συμπεριφοράς χρηστών. Δηλαδή, η δημιουργία ομάδων χρηστών με παρόμοιες δικτυακές και επικονωνιακές συμπεριφορές που αντιπροσωπεύουν ένα συγκεκριμένο πρότυπο κίνησης.

Το σύνολο δεδομένων ροής που χρησιμοποιείται στην παρούσα εργασία συλλέχθηκε από πραγματική νοσοκομειακή υποδομή, ανωνυμοποιήθηκε και αναλύθηκε διερευνητικά μέσω διαφόρων τεχνικών εξόρυξης δεδομένων, οπτικοποιήσεων και στατιστικών στοιχείων. Εν συνεχεία, η διπλωματική επικεντρώθηκε στη δημιουργία προφίλ συμπεριφοράς χρηστών και στη παραγωγή τεχνητών ρεαλιστικών δεδομένων με βάση αυτά τα προφίλ. Η προσομοίωση δεδομένων ροής για κάθε ένα ξεχωριστό προφίλ πραγματοποιήθηκε με χρήση μη επιβλεπόμενης μηχανικής μάθησης και συγκεκριμένα μοντέλων μείξης. Προκειμένου όμως να γίνει εφικτή η παραπάνω προσέγγιση, απαραίτητη ήταν η κατηγοριοποίηση των νοσοκομειακών χρηστών με βάση την εργασιακή τους θέση (ιατροί, νοσοκόμοι, γραμματεία κτλ). Η ταξινόμηση αυτή πραγματοποιήθηκε τόσο με εμπειρικούς κανόνες, όσο και με μοντέλα μηχανικής μάθησης και νευρωνικά δίκτυα.

Επομένως, η συνεισφορά της εργασίας είναι τόσο στον τομέα της ανάλυσης των δικτυακών ροών και των χαρακτηριστικών τους, όσο και στον τομέα της παραγωγής τεχνητών δεδομένων ροής μέσω των προφίλ συμπεριφοράς. Οι ρεαλιστικές αυτές κατανομές χαρακτηριστικών ροής, οι οποίες δεν υπάρχουν ευρέως διαθέσιμες, θα μπορούσαν να φανούν ιδιαίτερα χρήσιμες ακόμα και στην αξιολόγηση - αξιοπιστία των μοντέλων ανίχνευσης ανωμαλιών - επιθέσων που κυριαρχούν τα τελευταία χρόνια την επιστημονική κοινότητα.

Λέξεις Κλειδιά

Διερευνητική Ανάλυση Δεδομένων, Εξόρυξη Δεδομένων, Παρακολούθηση Δικτυακών Ροών, Μηχανική Μάθηση, Μοντέλα Μείξης, Παραγωγή - Προσομοίωση, Συσταδοποίηση, Δημιουργία Προφίλ, Ταξινόμηση, Τεχνητά Νευρωνικά Δίκτυα

Abstract

The purpose of this diploma thesis is the analysis and simulation of network flows of hospital and health care data, using generative and machine learning models. Profiling users' behavior is particularly useful in analyzing the behavior of a monitored network. That is, creating user groups with similar network and communication behaviors that represent a specific traffic pattern.

The dataset of NetFlows used in the current work was collected from real hospital infrastructure, anonymized and analyzed exploratory through various data mining techniques, visualizations and statistics. The thesis then focuses on creating user behavior profiles and generating artificial realistic data based on those profiles. The flow-based data simulation for each individual profile was performed using unsupervised machine learning and in particular mixture models. However, in order for the above approach to be achievable, it was necessary to categorize the hospital users based on their job position (doctors, nurses, secretariat, etc.). This classification was performed both with empirical rules and with machine learning models - artificial neural networks.

Therefore, the contribution of the diploma thesis is both in the field of network flow analysis and its characteristics, as well as in the field of simulation of realistic artificial flow-based data through behavioral profiles. These realistic distributions of flow characteristics, which are not widely available, could be particularly useful even in evaluating the reliability of anomaly detection models, that have dominated the scientific community in recent years.

Keywords

Exploratory Data Analysis, Data Mining, Network Flow Monitoring, Machine Learning, Mixture Models, Generation - Simulation, Clustering, Profiling, Classification, Artificial Neural Networks

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η μεταγωγή και η δρομολόγηση **βάσει ροής** (flow-based routing and switching) έχουν κερδίσει την προσοχή των ερευνητών εδώ και αρκετό καιρό [1]. Μπορεί να υπάρξει ιδιαίτερα επωφελής σε σύγκριση με τη μεταγωγή ανά πακέτο, ειδικά όσον αφορά την ποιότητα υπηρεσιών (QoS) [2] και την ασφάλεια [3]. Η αποτελεσματικότητα όμως πολλών λύσεων που βασίζονται στη ροή, εξαρτάται σε μεγάλο βαθμό από τα χαρακτηριστικά δικτυακής κίνησης, και ως εκ τούτου, θα πρέπει να αξιολογούνται βάσει ρεαλιστικών και με ακρίβεια μοντέλων ροής.

Σε δίκτυα IP, το **προφίλ συμπεριφοράς** ενός χρήστη (host) αναφέρεται στην παρατήρηση μετρημένων δεδομένων ροής από τον κορμό του διαδικτύου και στην εξαγωγή πληροφοριών που είναι αντιπροσωπευτικές της επικοινωνιακής συμπεριφοράς ή των προτύπων κίνησης του παρατηρηθέντος χρήστη. Είναι ιδιαίτερα χρήσιμο στην κατανόηση της συμπεριφοράς ενός παρακολουθούμενου δικτύου και στην εξαγωγή συμπερασμάτων σχετικά με φυσιολογικές και μη φυσιολογικές δραστηριότητες.

Η δημιουργία προφίλ μπορεί να γίνει σε τέσσερα επίπεδα [11]: επίπεδο χρήστη, επίπεδο εφαρμογής, επίπεδο υπολογιστή και επίπεδο δικτύου. Η δημιουργία προφίλ σε μεγάλη κλίμακα αντιμετωπίζει πολλές προκλήσεις όπως ο τεράστιος αριθμός ενεργών χρηστών και άρα η σποραδική εμφάνιση του παρατηρηθέντος υπολογιστή στη συνολική κίνηση. Η δημιουργία προφίλ υπολογιστή και η **συσταδοποίηση** στοχεύουν στον εντοπισμό κυρίαρχων και επίμονων συμπεριφορών απομονώνοντας τους ξενιστές. Η δημιουργία, δηλαδή, προφίλ με παρόμοιες συμπεριφορές είναι πολύ χρήσιμη ακόμα και για πολλές εφαρμογές ασφάλειας δικτύου. Ο εντοπισμός επιθέσεων στο δίκτυο θα είναι ευκολότερος αν μπορεί το προφίλ που φιλοξενεί συμπεριφορές κίνησης να εντοπίσει απότομες αλλαγές στους χρήστες του.

Προκειμένου όμως τέτοιες ιδέες να πραγματοποιηθούν και να αξιολογηθούν αξιόπιστα, πρέπει να διασφαλιστούν **ρεαλιστικές κατανομές των ροών**. Δυστυχώς, τέτοια δεδομένα δεν είναι εύκολα διαθέσιμα στη βιβλιογραφία. Η έλλειψη ρεαλιστικών μοντέλων επηρεάζει αρνητικά την αξιοπιστία των αποτελεσμάτων που παρουσιάζονται σε πολλές εργασίες. Επιπλέον, διαφορετικές και αυθαίρετες παραδοχές σε διάφορες εργασίες αποκλείουν τη δυνατότητα αποτελεσματικής σύγκρισης διαφορετικών λύσεων.

1.1 Αντικείμενο της διπλωματικής

Το αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι η δημιουργία προφίλ συμπεριφοράς χρηστών, αναλύοντας δικτυακές ροές νοσοκομειακών δεδομένων υγειονομικής περίθαλψης, με απώτερο σκοπό την παραγωγή τεχνητών ρεαλιστικών δεδομένων ροής. Η συνεισφορά της, δηλαδή, είναι τόσο στον τομέα της ανάλυσης των δικτυακών ροών και των χαρακτηριστικών τους (flow-based analysis), όσο και στον τομέα της παραγωγής διαφορετικών προφίλ συμπεριφοράς χρηστών (profiling and generation).

Αρχικά, η καταγραφή της δικτυακής κίνησης έγινε με το εργαλείο nProbe και τα δεδομένα ροής αποθηκεύτηκαν με βάση το πρότυπο NetFlow v9. Η επεξεργασία και η διερευνητική ανάλυση των δεδομένων πραγματοποιήθηκε με εργαλεία και βιβλιοθήκες της γλώσσας προγραμματισμού Python και περιλάμβανε διάφορες τεχνικές εξόρυξης δεδομένων και στατιστικών στοιχείων. Στη συνέχεια, η συσταδοποίηση, η δημιουργία προφίλ χρηστών και η παραγωγή δεδομένων πραγματοποιήθηκε με χρήση μοντέλων μη επιβλεπόμενης μηχανικής μάθησης και συγκεκριμένα με μοντέλα μείξης (mixture models). Συνδυαστικά, χρησιμοποιήθηκε και ο αλγόριθμος συσταδοποίησης K-Μέσων (K-Means), καθώς και διάφορες μετρικές για τη αξιολόγηση των παραπάνω μοντέλων. Τέλος, μέσω αλγορίθμων επιβλεπόμενης μηχανικής μάθησης, καθώς και τεχνητών νεωρωνικών δικτύων, έγινε μια προσπάθεια δημιουργίας μοντέλων ταξινόμησης των ροών στις αντίστοιχες κατηγορίες-προφίλ των χρηστών.

1.2 Συγγενικές εργασίες

Η συμβολή της δημοσίευσης [5] είναι από τις κυριότερες στη συγκεκριμένη εργασία. Οι συγγραφείς χρησιμοποιούν μοντέλα μείξης για την εκπαίδευση και την παραγωγή νέων χαρακτηριστικών ροών και συγκεκριμένα flow sizes και flow lengths. Στην παρούσα εργασία χρησιμοποιήθηκαν ως χαρακτηριστικά τα flow duration, flow inter-arrivals και flow bytes και επιπλέον δοκιμάστηκαν και τα Generative Adversarial Networks. Από όσο γνωρίζουμε, καμία άλλη δημοσίευση δεν παρέχει από κοινού την εκμάθηση μοντέλων μείξης και GANs για την εξαγωγή ρεαλιστικών χαρακτηριστικών ροής. Η [6] φιτάρει συγκεκριμένες κατανομές στα δεδομένα και παρέχει πλήρεις περιγραφικές παραμέτρους, παρόλο αυτά λαμβάνονται υπόψη μόνο μεμονωμένες κατανομές και όχι μείξεις κατανομών που χρησιμοποιούνται στα μοντέλα μας. Η πρώτη προσπάθεια για παραγωγή δικτυακών δεδομένων μέσω GANs έγινε στην [4], όπου πρατάθηκαν τρεις διαφορετικές προσεγγίσεις προεπεξεργασίας των flow-based δεδομένων προκειμένου να μετατραπούν σε συνεχείς τιμές.

Αξίζει επίσης να σημειωθεί ότι η [5] χωρίζει την κίνηση σε τρεις κατηγορίες (All trafic, TCP only και UDP only), ενώ η [6] την χωρίζει σε peer-to-peer, world-wide-web και TCP-big. Εμείς πάμε ένα βήμα παραπέρα αυτές τις προσεγγίσεις δημιουργώντας προφίλ συμπεριφοράς χρηστών ανάλογα με τις νοσοκομειακές υπηρεσίες που χρησιμοποιούν. Η εργασία [11] κάνει επιλογή και εξαγωγή προτύπων επικοινωνίας με βάση χαρακτηριστικά ροής, προκειμένου να κατηγοριοποιήσει και να δημιουργήσει προφίλ IP χρηστών. Διάφορες άλλες έρευνες έχουν πραγματοποιηθεί για το profiling δικτυακής κίνησης, οι περισσότερες από αυτές είχαν

ως απώτερο σκοπό την ανίχνευση ανωμαλιών-επιθέσεων [12]-[15]. Επιπλέον, αρκετές μελέτες έχουν επικεντρωθεί στην ευπάθεια συσκευών IoT για επιθέσεις ασφαλείας, τονίζοντας την ανάγκη για τον εντοπισμό, την αναγνώριση, και την ανακάλυψη συσκευών IoT μέσω των προτύπων συμπεριφοράς τους. Ένα χαρακτηριστικό παράδειγμα αποτελεί η [16] που χρησιμοποιεί LSTM-autoencoders για εκμάθηση χαρακτηριστικών από την κίνηση των συσκευών, μαθαίνει τις κατηγορίες κίνησης που παρατηρούνται και μοντελοποιεί κάθε συσκευή ως μια συστάδα κατανομής κίνησης.

Όπως αναφέρθηκε και προηγουμένως, ορισμένες έρευνες παρέχουν ανάλυση χαρακτηριστικών ροών δικτυακής κίνησης, χωρίς όμως να χρησιμοποιούν μοντέλα μείξης ή GANs για την παραγωγή δεδομένων, όπως γίνεται στην παρούσα εργασία. Για παράδειγμα, οι δημοσιεύσεις [7] και [8] παρέχουν γραφικές αναπαραστάσεις των κατανομών flow size, duration και rate, αλλά έχουν έλλειψη από αριθμητικά δεδομένα, ενώ στην [9] οι συγγραφείς επικεντρώνονται μόνο στην διάρκεια των ροών. Επιπλέον, εμπειρικές CDFs των flow length, size και duration παρουσιάζονται στην [10] χωρίς όμως να φιτάρουν συγκεκριμένες κατανομές. Τέλος, στην [17] εξετάζονται και συγκρίνονται συγκεκριμένα τα χαρακτηριστικά των WWW και P2P συστημάτων χρησιμοποιώντας τα flow inter-arrival, flow duration, flow size και flow rate. Η ανάλυση και η οπτικοποίηση πραγματοποιείται με CDFs, PDFs, QQ-plots και power spectrum density διαγράμματα.

1.3 Οργάνωση του τόμου

Η παρούσα διπλωματική εργασία διαρθρώνεται σε 8 κεφάλαια. Στο Κεφάλαιο 1, πραγματοποιείται μια εισαγωγή στο θέμα, παρουσιάζεται το αντικείμενο και η συνεισφορά της εργασίας, καθώς και συγγενικές εργασίες σχετικές με το θέμα της διπλωματικής. Στο Κεφάλαιο 2, καλύπτεται το θεωρητικό υπόβαθρο που απαιτείται ώστε να είναι δυνατή η κατανόηση εννοιών γύρω από την δικτυακή κίνηση ροής, την μηχανική μάθηση και τα τεχνητά νευρωνικά δίκτυα. Το Κεφάλαιο 3, παρέχει μια επισκόπηση στα προγραμματιστικά εργαλεία και στις τεχνικές λεπτομέρειες που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας. Στο Κεφάλαιο 4, παρουσιάζονται τα στάδια καταγραφής, επεξεργασίας και διερευνητικής ανάλυσης των δεδομένων. Το Κεφάλαιο 5, περιγράφει μια πρώτη προσπάθεια συσταδοποίησης των νοσοκομειακών χρηστών, ενώ στο Κεφάλαιο 6, αναλύονται αναλυτικά τα προφίλ χρηστών και ο τρόπος παραγωγής των συνθετικών δεδομένων. Η ταξινόμηση των χρηστών σε κατηγορίες-προφίλ μέσω αλγορίθμων επιβλεπόμενης μάθησης και νευρωνικών δικτύων καλύπτεται στο Κεφάλαιο 7. Τέλος, το Κεφάλαιο 8, συνοψίζει και ολοκληρώνει την παρούσα διπλωματική εργασία με ορισμένες κατευθύνσεις και μελλοντικές επεκτάσεις για την εξέλιξη του συγκεκριμένου θέματος.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Εισαγωγή

Στην συγκεκριμένο κεφάλαιο αναλύονται οι μεθοδολογίες και τα θεωρητικά μοντέλα που χρησιμοποιούνται στη διπλωματική και είναι αναγκαία η κατανόησή τους από τον αναγνώστη πριν από την παρουσίαση της ανάλυσης και σχεδίασης του συστήματος. Πρόκειται προφανώς για τεχνικές που έχουν προταθεί από τρίτους και δεν είναι πρωτότυπη δουλειά της διπλωματικής.

2.2 Διαδίκτυο

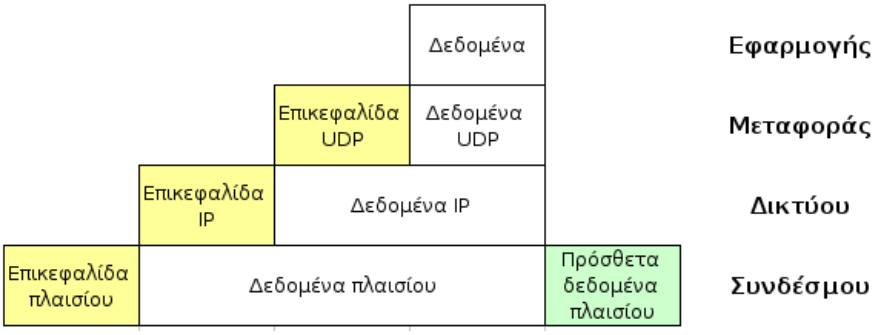
Το Διαδίκτυο (Internet) θα μπορούσε να χαρακτηριστεί ως ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν μια καθιερωμένη ομάδα πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας). Οι διασυνδεδεμένοι χρήστες ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν πακέτα με τη χρήση αυτών των πρωτοκόλλων, τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Αυτή η ομάδα πρωτοκόλλων είναι γνωστή ως μοντέλο αναφοράς TCP/IP.

2.2.1 Μοντέλο Αναφοράς TCP/IP

Η αρχιτεκτονική του μοντέλου αναφοράς TCP/IP είναι μια συλλογή πρωτοκόλλων επικοινωνίας στην οποία βασίζεται το σημερινό διαδίκτυο αλλά και μεγάλο ποσοστό των εμπορικών δικτύων. Η ονομασία του προέρχεται από τις συντομογραφίες των δύο κυριότερων πρωτοκόλλων του: το Πρωτόκολλο Ελέγχου Μετάδοσης (Transmission Control Protocol) και το Πρωτόκολλο Διαδικτύου (Internet Protocol). Το μοντέλο αυτό ορίστηκε αρχικά στο έγγραφο των Cerf and Kahn (1974), και κατόπιν εξελίχθηκε και ορίστηκε ως πρότυπο για την κοινότητα του Διαδικτύου (Braden, 1989). Η σχεδιαστική φιλοσοφία πίσω από το μοντέλο αναλύεται από τον Clark (1988) [18].

Το μοντέλο αναφοράς TCP/IP καθορίζει τον τρόπο με τον οποίο τα δεδομένα τοποθετούνται σε πακέτα, διευθύνονται, μεταδίδονται, δρομολογούνται και λαμβάνονται. Αυτές οι

λειτουργίες είναι οργανωμένες σε τέσσερα επίπεδα (layers), τα οποία ταξινομούν όλα τα σχετικά πρωτόκολλα σύμφωνα με το πεδίο δικτύωσης του κάθε ενός [19] [20]. Για την αφαίρεση πρωτοκόλλων και υπηρεσιών χρησιμοποιείται η ενθυλάκωση. Η ενθυλάκωση συνήθως ταυτίζεται με τη διαίρεση του μοντέλου στα τέσσερα επίπεδα. Γενικά, μια εφαρμογή χρησιμοποιεί ένα σύνολο πρωτοκόλλων για να στείλει τα δεδομένα της στα επίπεδα. Τα δεδομένα ενθυλακώνονται περαιτέρω σε κάθε επίπεδο προσθέτοντας τις απαραίτητες κεφαλίδες κάθε πρωτοκόλλου. Ένα παράδειγμα ενθυλάκωσης δεδομένων σ'ένα δεδομενόγραμμα UDP, ενθυλακωμένο σε ένα πακέτο IP παρουσιάζεται στο Σχήμα 2.1.



Σχήμα 2.1: Παράδειγμα ενθυλάκωσης δεδομένων σε ένα πακέτο IP

Στον Πίνακα 2.1 παρουσιάζονται μερικά χαρακτηριστικά πρωτόκολλα ανά επίπεδο του μοντέλου αναφοράς TCP/IP. Από το χαμηλότερο προς το υψηλότερο, τα επίπεδα είναι τα εξής [18]:

- Επίπεδο συνδέσμου: Περιέχει μεθόδους επικοινωνίας για συνδέσμους όπως οι σειριακές γραμμές και το κλασικό Ethernet, προκειμένου να ικανοποιήσει τις ανάγκες του ασυνδεσμικού επιπέδου διαδικτύου.

- Επίπεδο διαδικτύου: Είναι ο ακρογωνιαίος λίθος ολόκληρης της αρχιτεκτονικής. Η δουλειά του είναι να επιτρέπει στους υπολογιστές υπηρεσίας να εισάγουν τα πακέτα τους σε οποιοδήποτε δίκτυο και αυτά να δρομολογούνται ανεξάρτητα προς τον προορισμό τους. Το συγκεκριμένο επίπεδο ορίζει μια επίσημη μορφή για τα πακέτα και ένα επίσημο πρωτόκολλο, που ονομάζεται Πρωτόκολλο Διαδικτύου ή IP (Internet Protocol).

- Επίπεδο μεταφοράς: Η βασική λειτουργία του είναι να δέχεται δεδομένα από το ανώτερο επίπεδο, να τα διασπά εφόσον χρειάζεται σε μικρότερες μονάδες, να τα μεταβιβάζει στο επίπεδο διαδικτύου και να εξασφαλίζει ότι όλα τα τμήματα φτάνουν σωστά στο άλλο άκρο. Επιπλέον, καθορίζει τον τύπο της υπηρεσίας που θα παρέχεται στο επίπεδο συνδιάλεξης και, τελικά, στους χρήστες του δικτύου. Ορίζει δύο πρωτόκολλα μεταφοράς από άκρο εις άκρο, το Πρωτόκολλο Ελέγχου Μετάδοσης ή TCP (Transmission Control Protocol) και το Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη ή UDP (User Datagram Protocol).

- Επίπεδο εφαρμογών: Παρέχει την ανταλλαγή δεδομένων από διεργασία σε διαδικασία σε επίπεδο εφαρμογής και περιέχει όλα τα πρωτόκολλα ανωτέρου επιπέδου. Μερικά από τα

σημαντικά τα οποία συναντάμε κι εμείς στην ανάλυσή μας είναι το Σύστημα Ονομάτων Περιοχών (DNS), το SMTP, το HTTP/HTTPS, το NTP και το DHCP.

Επίπεδο	**Πρωτόκολλα**
Εφαρμογών	HTTP, SMTP, DNS, DHCP, TLS, NTP
Μεταφοράς	TCP, UDP
Διαδικτύου	IP, ICMP, IGMP, ARP
Συνδέσμου	Ethernet, DSL, 802.11, SONET

Πίνακας 2.1: Μοντέλο TCP/IP με ορισμένα πρωτόκολλα

2.2.2 Χρήσιμα Πρωτόκολλα (HTTP/HTTPS, DHCP, DNS)

Κατά την διάρκεια της καταγραφής και της ανάλυσης των δεδομένων που θα ακολουθήσει στα επόμενα κεφάλαια, γίνεται αναφορά σε διάφορα πρωτόκολλα εφαρμογής, τα οποία αξίζει να επεξηγηθούν αναλυτικότερα.

Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP/HTTPS)

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP) είναι ένα πρωτόκολλο επιπέδου εφαρμογής στο μοντέλο αναφοράς TCP/IP για κατανεμημένα και συνεργατικά συστήματα πληροφοριών και υπερμέσων [21]. Το HTTP είναι το θεμέλιο της επικοινωνίας δεδομένων για τον Παγκόσμιο Ιστό (World Wide Web), όπου τα έγγραφα υπερκειμένου περιλαμβάνουν ακόμα και υπερσυνδέσμους προς άλλες ιστοσελίδες, στις οποίες ο χρήστης μπορεί εύκολα να έχει πρόσβαση. Η πρώτη τεκμηριωμένη έκδοση του ήταν η έκδοση 0.9 [22], ενώ η τωρινή έκδοση είναι η 2.0. Το HTTP/2 είναι μια πιο βελτιωμένη εκδοχή του HTTP/1 που δημοσιεύτηκε το 2015. Χρησιμοποιείται από διακομιστές ιστού μέσω Ασφάλειας Επιπέδου Μεταφοράς (TLS) [23].

Το Ασφαλές Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTPS) είναι μια επέκταση του πρωτοκόλλου μεταφοράς υπερκειμένου (HTTP). Χρησιμοποιείται ευρέως στο διαδίκτυο για ασφαλή επικοινωνία μέσω δικτύου υπολογιστών [24]. Στο HTTPS, το πρωτόκολλο επικοινωνίας κρυπτογραφείται χρησιμοποιώντας το Transport Layer Security (TLS) ή, παλαιότερα, το Secure Sockets Layer (SSL). Ως εκ τούτου, το πρωτόκολλο αναφέρεται επίσης ως HTTP μέσω (over) TLS, ή HTTP μέσω (over) SSL [25]. Τα πλεονεκτήματα του HTTPS είναι ο έλεγχος ταυτότητας του ιστότοπου πρόσβασης και η προστασία του απορρήτου και της ακεραιότητας των δεδομένων που ανταλλάσσονται κατά τη μεταφορά. Προστατεύει από διάφορες επιθέσεις, όπως «man-in-the-middle» και λόγω της αμφίδρομης κρυπτογράφησης των επικοινωνιών μεταξύ πελάτη και διακομιστή, προστατεύει από υποκλοπές και παραβιάσεις [26].

Πρωτόκολλα Κρυπτογράφησης (TLS/SSL)

Αναφερόμενοι προηγουμένως στο Transport Layer Security (TLS) και στον προκάτοχό του, το Secure Sockets Layer (SSL), αξίζει να σημειωθεί ότι είναι Πρωτόκολλα Κρυπτογράφησης τα οποία παρέχουν ασφάλεια επικοινωνίας πάνω από ένα δίκτυο υπολογιστών [27]. Αρκετές από τις εκδόσεις του πρωτοκόλλου χρησιμοποιούνται σε εφαρμογές όπως το ηλεκτρονικό ταχυδρομείο, οι κλήσεις μέσω IP (VoIP), αλλά η χρήση του ως επίπεδο ασφαλείας στο HTTPS παραμένει η πιο ορατή και δημοφιλής.

Πρωτόκολλο DHCP

Το DHCP (Dynamic Host Configuration Protocol) είναι ένα πρωτόκολλο για την αυτόματη εκχώρηση διευθύνσεων IP και άλλων παραμέτρων επικοινωνίας σε συσκευές συνδεδεμένες στο δίκτυο χρησιμοποιώντας αρχιτεκτονική πελάτη-διακομιστή [28]. Είναι δηλαδή ένας μηχανισμός διαχείρισης πρωτοκόλλων TCP/IP. Προκειμένου να πραγματοποιηθεί εύκολα και γρήγορα η διαχείριση αυτή σε πολλά τερματικά, υπάρχει η ανάγκη να αρχικοποιηθεί κάθε μηχάνημα με τις αντίστοιχες παραμέτρους για αυτό και για τη θέση του στο δίκτυο. Η αρχικοποίηση (initialisation) αυτή μπορεί να γίνει κατά τη διάρκεια της εκκίνησης (αν το πρωτόκολλο είναι συγχωνευμένο στο λειτουργικό σύστημα) ή με την κλήση του πρωτοκόλλου από κάποια εφαρμογή (αν το πρωτόκολλο υπάρχει στη συγκεκριμένη εφαρμογή).

Το DHCP όπως αναφέρθηκε και προηγουμένως, παρέχει παραμέτρους ρυθμίσεων για ένα μοντέλο δικτύου πελάτη-διακομιστή. Οι DHCP servers δεσμεύουν τις διευθύνσεις του δικτύου και στέλνουν τις πληροφορίες για αυτές στους clients. Το DHCP αποτελείται από δύο τμήματα. Το πρώτο είναι το πρωτόκολλο που στέλνει παραμέτρους ρυθμίσεων από τον server στον client και το δεύτερο είναι ο μηχανισμός για να αντιστοιχίζει τις διευθύνσεις IP στους clients.

Πρωτόκολλο DNS

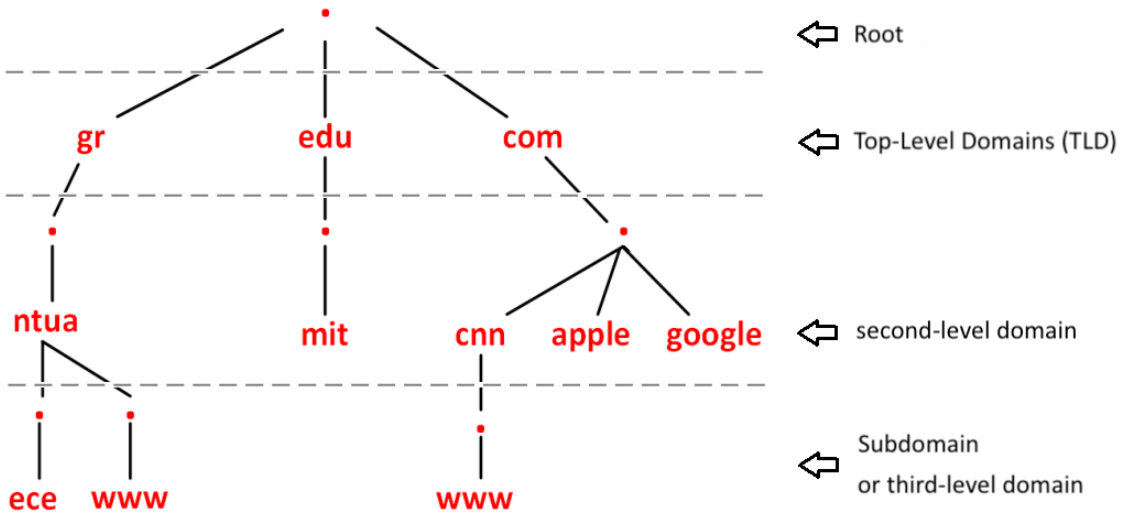
Σε συναλλαγές με το Διαδίκτυο, είναι προφανές ότι είναι δύσκολο να μπορεί να γίνει απομνημόνευση των διευθύνσεων IP έτσι ώστε να επικοινωνήσουμε για παράδειγμα με έναν διακοσμιτή. Γι' αυτό το λόγο έχει αναπτυχθεί ένα σύστημα ονοματοδοσίας των υπολογιστών του Διαδικτύου και μια υπηρεσία καταλόγου για αναζήτηση των ονομάτων. Η υπηρεσία αυτή ονομάζεται DNS (Domain Name Service – Υπηρεσία Ονομασίας Περιοχών) [29].

Το σύστημα ονομασίας περιοχών (DNS) είναι μια κατανεμημένη βάση δεδομένων στο Διαδίκτυο που επιτρέπει τη μετάφραση ανάμεσα σε ονόματα και διευθύνσεις IP. Θα μπορούσαμε να πούμε ότι το DNS είναι κάτι σαν «τηλεφωνικός κατάλογος». Είναι ο μηχανισμός του Διαδικτύου για την αναφορά μέσω ονομάτων σε ό,τι πόρους χρησιμοποιούμε σε αυτό και που μας επιτρέπει τη μετάφραση ονομάτων σε διευθύνσεις IP και το αντίστροφο. Πρόκειται για μία κατανεμημένη βάση δεδομένων που εφαρμόζεται σε μια ιεραρχία πολλών εξυπηρετητών ονομάτων (DNS servers) και περιλαμβάνει τα ακόλουθα:

- Χώρος ονομάτων: Το Διαδίκτυο είναι χωρισμένο νοητά σε πολλές περιοχές (domains)

υψηλού επιπέδου που αναλύονται σε υποπεριοχές (subdomains), με πολλούς υπολογιστές (hosts) η καθεμία. Οι περιοχές μπορεί να παρασταθούν με ένα δέντρο. Τα ονόματα των περιοχών απαρτίζουν μια ιεραρχία κατά τρόπο που τα ονόματα να είναι μοναδικά και να απομνημονεύονται εύκολα. Ένας οργανισμός είναι αρμόδιος για μέρος του χώρου ονομάτων και μπορεί να προσθέσει επιπλέον επίπεδα στην ιεραρχία [30].

- Ιεραρχία DNS: Κάθε κόμβος στο δένδρο DNS αναπαριστά ένα όνομα (DNS name). Κάθε κλαδί κάτω από ένα κόμβο είναι μια περιοχή (DNS domain). Η περιοχή μπορεί να περιέχει hosts ή άλλες υποπεριοχές (subdomains). Η κορυφή του δένδρου είναι η ρίζα (root) και συμβολίζεται με μία τελεία «.». Η IANA (Internet Assigned Numbers Authority) είναι η επίσημη αρχή που διαχειρίζεται τη ρίζα του DNS. Κάτω από την κορυφή υπάρχουν οι περιοχές ανωτάτου επιπέδου (top level domains ή περιοχές 1ου επιπέδου ή βασικές περιοχές). Χαρακτηριστικά παραδείγματα είναι η edu, gov, com, org, mil, net, int, arpa, gr, uk, us κτλ. Κάτω από κάθε περιοχή 1ου επιπέδου, υπάρχει δεύτερο επίπεδο περιοχών, που προσδιορίζει συνήθως τον οργανισμό ή την εταιρεία στην οποία ανήκει το δίκτυο. Οι περιοχές αυτές ονομάζονται περιοχές 2ου επιπέδου και κάθε μία είναι μοναδική. Η διαχείριση του χώρου ονομάτων κάτω από τις περιοχές ανωτάτου επιπέδου έχει εκχωρηθεί σε οργανισμούς, που μπορούν να εκχωρήσουν περαιτέρω τη διαχείριση υποπεριοχών τους (subdomains). Κάθε νέο subdomain αντιστοιχεί σε περιοχή ονομάτων 3ου επιπέδου. Στο Σχήμα 2.2 παρουσιάζεται ένα μικρό παράδειγμα της ιεραρχικής οργάνωσης χώρου ονομάτων του διαδικτύου.



Σχήμα 2.2: Ιεραρχική οργάνωση χώρου ονομάτων διαδικτύου

- Εξυπηρετητές (Name Servers): Βρίσκονται σε διαφορετικά σημεία του Διαδικτύου, συνεργάζονται μεταξύ τους και μέσω αυτών γίνεται διαθέσιμος ο χώρος ονομάτων [31]. Η ιεραρχία του χώρου ονομάτων ανταποκρίνεται σε μία αντίστοιχη ιεραρχία εξυπηρετητών ονομάτων. Κάθε εξυπηρετητής είναι υπεύθυνος για ένα συμπαγές τμήμα του χώρου ονομάτων που αποκαλείται ζώνη (zone). Ο εξυπηρετητής ονομάτων απαντά σε ερωτήσεις

(queries) για τους hosts της ζώνης του. Κάθε ζώνη είναι εμφωλευμένη σε ένα κόμβο του δένδρου. Οι ζώνες δεν είναι περιοχές (domains). Η ζώνη είναι τμήμα του χώρου ονομάτων που εν γένει αποθηκεύεται σε ένα αρχείο. Ο εξυπηρετητής ονομάτων μπορεί να χωρίσει μέρος της ζώνης του και να το εκχωρήσει σε άλλους εξυπηρετητές.

- Αναλυτές (Resolvers): Ερωτούν τους εξυπηρετητές περί του χώρου ονομάτων. Για την ανεύρεση δεδομένων, ο εξυπηρετητής ονομάτων χρειάζεται μόνο το όνομα και τη διεύθυνση IP των εξυπηρετητών ονομάτων κορυφής (ρίζας). Οι εξυπηρετητές κορυφής γνωρίζουν όλες τις περιοχές ανωτάτου επιπέδου και μπορούν να υποδείξουν τους εξυπηρετητές με τους οποίους μπορεί να γίνει επαφή.

Το πρωτόκολλο DNS είναι επιπέδου εφαρμογής που επιτρέπει σε υπολογιστές (hosts), δρομολογητές (routers) και εξυπηρετητές DNS (Name Servers) να επικοινωνούν για να αναλύσουν (resolve) ονόματα αλλά και να ανταλλάξουν επιπλέον πληροφορίες όπως για mail servers. Είναι βασική λειτουργία του κορμού του Διαδικτύου, όπου οι αναζητήσεις DNS γίνονται από οποιοδήποτε μηχάνημα και οποιαδήποτε υπηρεσία. Τα αποτελέσματα από μακρινούς εξυπηρετητές ονομάτων αποθηκεύονται προσωρινά σε τοπική μνήμη, ώστε να βελτιωθεί η επίδοση [18].

2.3 Παρακολούθηση Δικτύου

Ανά τα χρόνια, έχουν προταθεί και αναπτυχθεί διάφορες προσεγγίσεις παρακολούθησης δικτύου, η κάθε μια από αυτές εξυπηρετούσε διαφορετικό σκοπό. Μπορούν γενικά να ταξινομηθούν σε δύο κατηγορίες: ενεργητική και παθητική [32].

- Ενεργητικές προσεγγίσεις: Εγχέουν κίνηση σε ένα δίκτυο για την εκτέλεση διαφορετικών τύπων μετρήσεων. Υλοποιούνται για παράδειγμα από εργαλεία όπως το *Ping* και το *Traceroute*.

- Παθητικές προσεγγίσεις: Παρατηρούν την υπάρχουσα κίνηση καθώς αυτή διαπερνά ένα σημείο μέτρησης. Μια προσέγγιση παθητικής παρακολούθησης είναι η σύλληψη πακέτων.

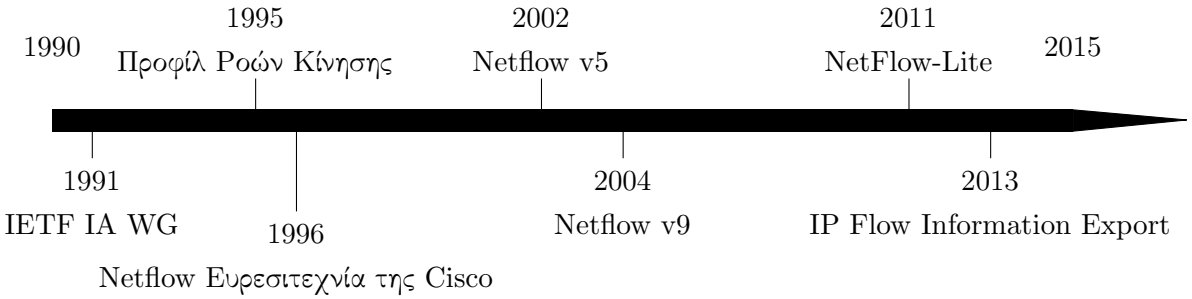
Η παθητική μέθοδος παρέχει γενικά περισσότερες πληροφορίες για το δίκτυο, καθώς ολόκληρα πακέτα μπορούν να συλληφθούν και να αναλυθούν περαιτέρω. Ωστόσο, σε δίκτυα υψηλής ταχύτητας με υψηλούς ρυθμούς γραμμής, η σύλληψη πακέτων απαιτεί ακριβό υλικό και σημαντική υποδομή για αποθήκευση και ανάλυση.

2.3.1 Παρακολούθηση Ροής

Μια επιπλέον προσέγγιση παθητικής παρακολούθησης δικτύου, που είναι περισσότερο επεκτάσιμη για χρήση σε δίκτυα υψηλής ταχύτητας, είναι η εξαγωγή ροής. Τα πακέτα συγκεντρώνονται σε ροές, εξάγονται, αποθηκεύονται και αναλύονται. Μια ροή ορίζεται στο [33] ως «Ένα σύνολο IP πακέτων που διέρχονται από ένα σημείο παρατήρησης στο δίκτυο κατά τη

διάρκεια ένα ορισμένου χρονικού διαστήματος, τέτοιο ώστε όλα τα πακέτα που ανήκουν σε μια συγκεκριμένη ροή να έχουν ένα σύνολο κοινών ιδιοτήτων». Αυτές οι κοινές ιδιότητες μπορεί να περιλαμβάνουν πεδία κεφαλίδας πακέτου, όπως για παράδειγμα διευθύνσεις IP προέλευσης - προορισμού και αριθμοί θυρών, περιεχόμενα πακέτου και μετα-δεδομένα.

2.3.2 Ιστορική Αναδρομή



Οι δημοσιευμένες πηγές εξαγωγών ροής χρονολογούνται από το 1991, όταν η συνάθροιση πακέτων σε ροές μέσω πληροφοριών κεφαλίδας περιγράφηκαν στο [34]. Πραγματοποιήθηκε ως μέρος του Internet Accounting (IA) Working Group (WG) του Internet Engineering Task Force (IETF). Το 1995, το ενδιαφέρον για εξαγωγή ροής δεδομένων για την ανάλυση της δικτυακής κίνησης αναζωογονήθηκαν από το [35], το οποίο παρουσίασε μια μεθοδολογία για τη δημιουργία προφίλ των ροών κίνησης στο διαδίκτυο με βάση τη συνένωση πακέτων.

Ένα χρόνο αργότερα, το 1996, η Cisco, εργαζόμενη στην τεχνολογία εξαγωγής ροής, κατοχυρώνει με δίπλωμα ευρεσιτεχνίας το όνομα NetFlow, το οποίο βρίσκει την προέλευσή του στη μεταγωγή. Η πρώτη έκδοση που υιοθετήθηκε ευρέως ήταν το NetFlow v5 [36], το οποίο έγινε διαθέσιμο στο κοινό γύρω στο 2002. Το NetFlow v5 καταργήθηκε από το πιο ευέλικτο NetFlow v9, το οποίο περιγράφεται στο [37] από το 2004. Το NetFlow v9 υποστηρίζει, μεταξύ άλλων χαρακτηριστικών, προσαρμόσιμες μορφές δεδομένων μέσω προτύπων, καθώς και IPv6, εικονικά τοπικά δίκτυα (VLAN) και Multiprotocol Label Switching (MPLS). Αργότερα, το 2011, η Cisco παρουσίασε το NetFlow-Lite, μια τεχνολογία που βασίζεται στο Flexible NetFlow, που χρησιμοποιεί ένα εξωτερικό μηχάνημα συγκέντρωσης πακέτων, για τη διευκόλυνση της εξαγωγής ροής σε συσκευές προώθησης πακέτων [38].

Παράλληλα με την ανάπτυξη του NetFlow, η IETF αποφάσισε το 2004 να τυποποιήσει ένα πρωτόκολλο εξαγωγής ροής και οριστικοποίησε το IP Flow Information Export (IPFIX) WG [39]. Οι πρώτες προδιαγραφές τελειοποιήθηκαν στις αρχές του 2008, τέσσερα χρόνια μετά τη οριστικοποίηση του IPFIX WG. Αυτές οι προδιαγραφές ήταν η βάση του IPFIX Internet Standard [33] στα τέλη του 2013. Σύντομο ιστορικό σχετικά με την εξαγωγή ροής και λεπτομέρειες της ανάπτυξης του IPFIX παρέχονται στο [40].

2.3.3 NetFlow v9

Μια εγγραφή NetFlow μπορεί να περιέχει μια μεγάλη ποικιλία πληροφοριών σχετικά με την κίνηση σε μια δεδομένη ροή. Το NetFlow έκδοση 9 περιέχει τα εξής [37]:

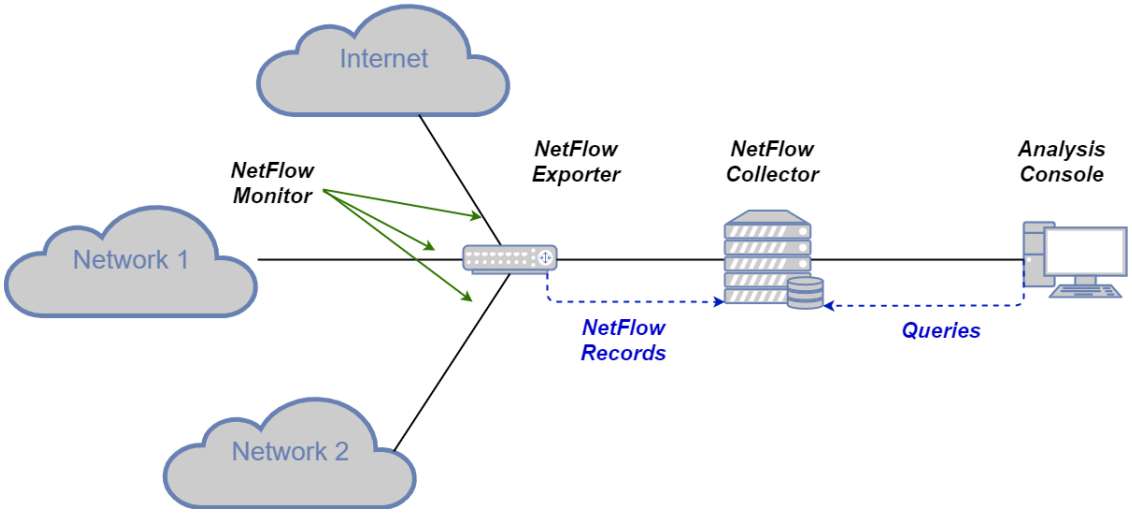
- Δείκτης διεπαφής εισόδου που χρησιμοποιείται από το SNMP (ifIndex στο IF-MIB).
- Δείκτης διεπαφής εξόδου ή μηδέν εάν το πακέτο απορριφθεί.
- Χρονικές σημάνσεις για το χρόνο έναρξης και λήξης ροής, σε χιλιοστά του δευτερολέπτου από την τελευταία εκκίνηση.
- Επικεφαλίδες επιπέδου 3.
  - Διευθύνσεις IP πηγής και προορισμού.
  - Τύπος και κωδικός ICMP.
  - Πρωτόκολλο IP.
  - Τιμή τύπου υπηρεσίας (ToS).
- Αριθμοί θύρας πηγής και προορισμού για TCP, UDP, SCTP.
- Για τις ροές TCP, η ένωση όλων των σημαιών TCP που παρατηρούνται κατά τη διάρκεια ζωής της ροής.
- Πληροφορίες δρομολόγησης επιπέδου 3.
  - Διεύθυνση IP του άμεσου επόμενου βήματος.
  - BGP nexthop.
  - Μάσκες IP πηγής και προορισμού.
  - Αριθμός πηγής και προορισμού Αυτόνομου Συστήματος (AS).
- Αριθμός byte και πακέτων που παρατηρήθηκαν στη ροή.
- Ετικέτες Multiprotocol Label Switching (MPLS).
- Πληροφορίες για τα εικονικά τοπικά δίκτυα (VLAN).
- Διευθύνσεις και θύρες IPv6.

2.3.4 Στάδια παρακολούθησης ροής με NetFlows

Με την ανάλυση των δεδομένων ροής, μπορεί να δημιουργηθεί μια εικόνα της ροής και του όγκου κίνησης σε ένα δίκτυο. Η μορφή εγγραφής NetFlow έχει εξελιχθεί με την πάροδο του χρόνου, για αυτό και η συμπερίληψη των διαφορετικών εκδόσεων. Η Cisco διατηρεί αναλυτικές λεπτομέρειες για τις διαφορετικές εκδόσεις.

Μια τυπική ρύθμιση παρακολούθησης ροής με χρήση NetFlow αποτελείται από τα παρακάτω τρία κύρια στοιχεία:[32]

- Εξαγωγή ροής: Συγκεντρώνει πακέτα σε ροές και εξάγει αρχεία ροής προς έναν ή περισσότερους συλλέκτες ροής.
- Συλλογή ροής: Υπεύθυνος για τη λήψη, αποθήκευση και προεπεξεργασία των δεδομένων ροής που λαμβάνονται από έναν εξαγωγέα ροής.
- Ανάλυση ροής: Αναλύει τα δεδομένα ροής που λαμβάνονται στο πλαίσιο για παράδειγμα της ανίχνευσης εισβολής (intrusion detection) ή του προφίλ κίνησης (traffic profiling).



Σχήμα 2.3: Τυπική Τοπολογία NetFlow

Ένα χαρακτηριστικό παράδειγμα τοπολογίας παρακολούθησης NetFlow παρατηρείται στο Σχήμα 2.3. Στην παρούσα διπλωματική εργασία ασχολούμαστε και αναλύουμε τα δεδομένα ροής με χρήση του NetFlow v9 στο πλαίσιο του προφίλ κίνησης και της κατηγοριοποίησης των χρηστών. Το εργαλείο που χρησιμοποιήθηκε και πραγματοποίησε την εξαγωγή και την συλλογή των ροών είναι το nProbe, το οποίο αναλύεται περαιτέρω στο επόμενο κεφάλαιο.

2.4 Εισαγωγή στη Μηχανική Μάθηση

2.4.1 Εισαγωγή

Η Μάθηση (Learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Επί χρόνια, πολλές μελέτες και έρευνες έχουν πραγματοποιηθεί από τους επιστήμονες του πεδίου της γνωστικής ψυχολογίας και της φιλοσοφίας, παρόλο αυτά, η έννοια της μάθησης δεν έχει γίνει πλήρως κατανοητή. Πώς, λοιπόν, θα μπορούσαν οι επιστήμονες του χώρου της Τεχνητής Νοημοσύνης να δημιουργήσουν υπολογιστικά συστήματα ικανά να μάθουν, να επιτύχουν, δηλαδή, τη λεγόμενη Μηχανική Μάθηση (Machine Learning) [41].

Η Μηχανική Μάθηση μπορεί να οριστεί ως το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μιας συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να προγραμματιστεί εκ νέου. Το 1959, ο Άρθουρ Σάμουελ ορίζει τη μηχανική μάθηση ως «Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μάθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί» [42]. Βάσει του ορισμού αυτού, η Μηχανική Μάθηση έχει ως σκοπό τη δημιουργία μηχανών ικανών να μαθαίνουν, να βελτιώνουν, δηλαδή, την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας.

Ένας σχετικός γενικός ορισμός Μηχανικής Μάθησης δίνεται από τον Mitchell (1997) και αναφέρει «Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P, αν η απόδοσή του σε εργασίες από το T, όπως μετριέται από το P, βελτιώνεται μέσω της εμπειρίας E.» [43]. Αυτός ο ορισμός είναι σημαντικός για τον καθορισμό της μηχανικής μάθησης σε βασικό λειτουργικό πλαίσιο παρά με γνωστικούς όρους, ακολουθώντας έτσι την πρόταση του Alan Turing στην εργασία του «Υπολογιστικές μηχανές και Νοημοσύνη», ότι το ερώτημα αν μπορούν οι μηχανές να σκεφτούν, μπορεί να αντικατασταθεί με το ερώτημα αν μπορούν οι μηχανές να κάνουν αυτό που εμείς (ως σκεπτόμενες οντότητες) μπορούμε να κάνουμε [44].

Ως κλάδος της Τεχνητής Νοημοσύνης, η Μηχανική Μάθηση ασχολείται με τη μελέτη αλγορίθμων που βελτιώνουν τη συμπεριφορά τους σε κάποια εργασία που τους έχει ανατεθεί χρησιμοποιώντας την εμπειρία τους. Όσον αφορά τη σχεδίαση των συστημάτων Μηχανικής Μάθησης, για τα συστήματα που ανήκουν στη συμβολική Τεχνητής Νοημοσύνης, η δυνατότητα μάθησης προσδιορίζεται ως η ικανότητα πρόσκτησης επιπλέον γνώσης, που επιφέρει μεταβολές στην υπάρχουσα καταχωρημένη γνώση είτε αλλάζοντας χαρακτηριστικά της είτε με αυξομείωσή της. Στην περίπτωση των συστημάτων Τεχνητής Νοημοσύνης που ανήκουν στη Μη Συμβολική Τεχνητής Νοημοσύνης (όπως η περίπτωση των Τεχνητών Νευρωνικών Δικτύων), ως μάθηση προσδιορίζεται η δυνατότητα που διαθέτουν τα συστήματα στο να μετασχηματίζουν την εσωτερική τους δομή, παρά στο να μεταβάλλουν κατάλληλα τη γνώση που έχει καταχωρηθεί μέσα σε αυτά κατά το σχεδιασμό τους [41].

Εκτός της ίδιας της Τεχνητής Νοημοσύνης, μεταξύ των επιστημονικών κλάδων που επωφελούνται από τα επιτεύγματα στον τομέα της Μηχανικής Μάθησης συγκαταλέγονται οι: Εξόρυξη Δεδομένων, Πιθανότητες και Στατιστική, Θεωρία της Πληροφορίας, Αριθμητική Βελτιστοποίηση, Θεωρία της Πολυπλοκότητας, Θεωρία Ελέγχου (προσαρμοστική), Ψυχολο-

γία (εξελικτική, γνωστική), Νευροβιολογία και Γλωσσολογία [41].

2.4.2 Είδη Μηχανικής Μάθησης

Ο τομέας της Μηχανικής Μάθησης αναπτύσσει τρεις τρόπους μάθησης, ανάλογους με τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος: επιβλεπόμενη μάθηση (supervised learning), μη επιβλεπόμενη μάθηση (unsupervised learning και ενισχυτική μάθηση (reinforcement learning). Αναλυτικότερα,

- **Επιβλεπόμενη Μάθηση**: Είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Χρησιμοποιείται σε προβλήματα:
  - Ταξινόμησης (Classification)
  - Πρόγνωσης (Prediction)
  - Διερμηνείας (Interpretation)
- **Μη Επιβλεπόμενη Μάθηση**: Ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων υπό μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους. Χρησιμοποιείται σε προβλήματα:
  - Ανάλυσης Συσχετισμών (Association Analysis)
  - Ομαδοποίησης - Συσταδοποίησης (Clustering)
- **Ενισχυτική Μάθηση**: Ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα σχεδιασμού (planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

2.5 Αλγόριθμοι Επιβλεπόμενης Μάθησης

Όπως αναφέρθηκε και προηγουμένως, η επιβλεπόμενη μάθηση είναι μία κατηγορία μηχανικής μάθησης, στόχος της οποίας είναι ο χαρακτηρισμός δεδομένων με βάση κάποια δεδομένα εκπαίδευσης. Τα δεδομένα εκπαίδευσης αποτελούνται από ένα σύνολο παραδειγμάτων τα οποία χρησιμοποιούνται για εκπαίδευση μοντέλων. Στην επιβλεπόμενη μάθηση, κάθε παράδειγμα αποτελείται από ένα σύνολο εισόδου (συνήθως ένα διάνυσμα από χαρακτηριστικά) και μια επιθυμητή τιμή εξόδου.

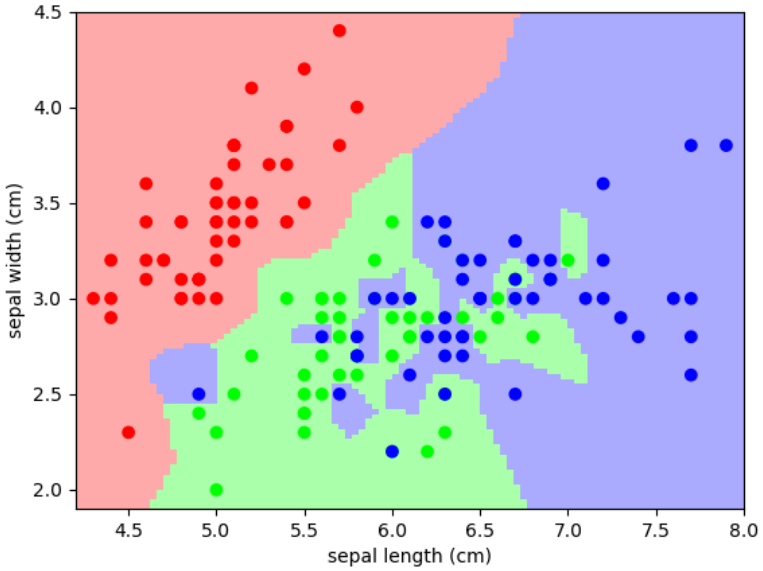
Οι αλγόριθμοι επιβλεπόμενης μάθησης αναλύουν τα δεδομένα εκπαίδευσης και παράγουν ένα μοντέλο το οποίο μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει νέα παραδείγματα. Το βέλτιστο σενάριο επιτρέπει στον αλγόριθμο να καθορίσει σωστά την ετικέτα της κατηγορίας για άγνωστα μέχρι τώρα παραδείγματα. Για να επιτευχθεί αυτό, απαιτείται ο αλγόριθμος μάθησης να γενικεύει από τα δεδομένα εκπαίδευσης σε αθέατες καταστάσεις με ένα «λογικό» τρόπο [45].

2.5.1 Αλγόριθμος kNN

Ο αλγόριθμος Κ κοντινότερων γειτόνων (k-Nearest Neighbors - kNN) είναι μία πολύ γνωστή και ευρεία χρησιμοποιούμενη τεχνική ταξινόμησης που στηρίζεται στη χρήση μέτρων βασισμένων στην απόσταση. Η κεντρική ιδέα είναι πως η τιμή της συνάρτησης-στόχου για ένα νέο στιγμιότυπο βασίζεται αποκλειστικά και μόνο στις αντίστοιχες τιμές των k πιο «κοντινών» στιγμιότυπων εκπαίδευσης, τα οποία αποτελούν τους «γείτονες» του.

Τα παραδείγματα εκπαίδευσης είναι διανύσματα σε έναν πολυδιάστατο χώρο χαρακτηριστικών, το καθένα με μια ετικέτα κλάσης. Η φάση εκπαίδευσης του αλγορίθμου αποτελείται μόνο από την αποθήκευση των διανυσμάτων χαρακτηριστικών και των ετικετών κλάσεων των δειγμάτων εκπαίδευσης. Στη φάση ταξινόμησης, ορίζεται η σταθερά k από τον χρήστη και ταξινομείται ένα μη επισημασμένο διάνυσμα με την ανάθεση της ετικέτας που είναι πιο συχνή μεταξύ των k δειγμάτων εκπαίδευσης που είναι πλησιέστερα σε αυτό το σημείο ερωτήματος.

Μια μετρική απόστασης που χρησιμοποιείται συνήθως για συνεχείς μεταβλητές είναι η Ευκλείδεια απόσταση. Για διακριτές μεταβλητές, όπως για την ταξινόμηση κειμένου, μπορεί να χρησιμοποιηθεί μια άλλη μετρική, όπως η μετρική επικάλυψης (απόσταση Hamming). Ένα μειονέκτημα της βασικής ταξινόμησης «ψηφοφορία της πλειοψηφίας» εμφανίζεται όταν η κατανομή της τάξης είναι λοξή. Δηλαδή, παραδείγματα μιας πιο συχνής τάξης τείνουν να κυριαρχούν στην πρόβλεψη του νέου παραδείγματος, επειδή τείνουν να είναι κοινά μεταξύ των k πλησιέστερων γειτόνων λόγω του μεγάλου αριθμού τους.



Σχήμα 2.4: Περιοχές απόφασης για k=1

Στο παραπάνω παράδειγμα του Σχήματος 2.4, επιθυμούμε να ταξινομήσουμε άνθη σε τρεις κατηγορίες με βάση κάποια χαρακτηριστικά που διαθέτουν. Τα χαρακτηριστικά είναι στον δυσδιάστατο χώρο, δηλαδή είναι της μορφής $x = [x1, x2]$, όπου το x1 είναι το χαρακτηριστικό sepal width (πλάτος φύλλου ανθού) και x2 το χαρακτηριστικό sepal length (μήκος φύλλου

ανθού).

Οι κουκίδες αναφέρονται στα δείγματα του συνόλου εκπαίδευσης, ενώ το χρώμα τους υποδηλώνει την κλάση στην οποία ανήκουν. Με αντίστοιχο χρώμα έχουν επισημειωθεί και οι περιοχές απόφασης. Η περιοχή απόφασης για μια κατηγορία Α ορίζεται η περιοχή του n-διάστατου χώρου χαρακτηριστικών, στην οποία, αν βρεθεί ένα νέο δείγμα, θα ταξινομηθει στην κατηγορία Α. Για παράδειγμα, ένα νέο δείγμα με χαρακτηριστικά sepal length = 4.5 cm και sepal width = 4 cm δηλαδή το $x = [4.5, 4]$ θα ταξινομηθει στην κατηγορία με κόκκινο χρώμα, καθώς ο ένας (k = 1) κοντινότερος του γείτονας ανήκει σε αυτή την κατηγορία (έχει σημειωθεί με κόκκινο χρώμα).

2.5.2 Δέντρα Απόφασης

Τα Δένδρα Απόφασης (Decision Trees) είναι ο γνωστότερος αλγόριθμος επιβλεπόμενης επαγωγικής μάθησης και έχει εφαρμοστεί με επιτυχία σε πολλούς τομείς όπου απαιτείται ταξινόμηση, όπως στην αναγνώριση προσώπων σε εικόνες, στην ιατρική για διάγνωση περιστατικών, σε προβλέψεις απαραίτητες για τη διαφήμιση, σε προώθηση προϊόντων και, γενικότερα, σε εξόρυξη γνώσης.

Ένα δέντρο απόφασης, μπορεί να προβλέψει την κλάση που ανήκει ένα δείγμα, διενεργώντας μια σειρά ερωτήσεων στα χαρακτηριστικά του. Πρόκειται για ένα διάγραμμα ροής (flowchart) το οποίο έχει ανάποδη δενδρική δομή. Πιο ειδικά, σε κάθε εσωτερικό κόμβο πραγματοποιείται ένας έλεγχος (ερώτηση) για κάποιο χαρακτηριστικό. Κάθε κλαδί αυτού του κόμβου αναπαριστά και διαφορετική απάντηση (έξοδο) απο τον έλεγχο που πραγματοποιήθηκε. Τα φύλλα αυτού του δέντρου, αντιπροσωπεύουν τις ετικέτες των κλάσεων (labels).

Στο παρακάτω παράδειγμα του Σχήματος 2.5, έχουμε ένα πρόβλημα δυαδικής ταξινόμησης (yes/now) στο ερώτημα: Play Golf. Για την κατασκευή του δέντρου απόφασης, χρησιμοποιήθηκαν 14 δείγματα εκπαίδευσης, κάθε ένα απο τα οποία διέθεται 4 χαρακτηριστικά (Outlook, Temp, Humidity, Windy).

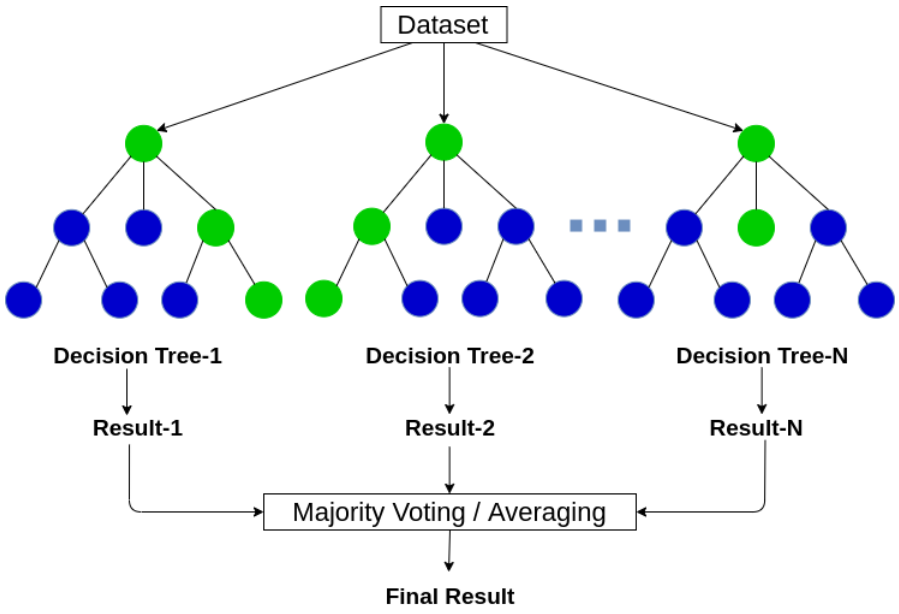
Σχήμα 2.5: Κατασκευή Δέντρου Απόφασης

Τα δέντρα απόφασης συνιστούν ένα ιδιαίτερα δημοφιλές εργαλείο, λόγω της ταχύτητας τους και της ευκολίας που παρέχουν τόσο σε επίπεδο κατανόησης, όσο και υλοποίησης. Ω-

στόσο, όπως αναφέρεται δεν μπορούν να γίνουν το καλύτερο εργαλειο ταξινόμησης, λόγω ανακρίβειας [46]. Με άλλα λόγια, σε κατάλληλα σύνολα δεδομένων (όπου δεν υπάρχουν δείγματα με πανομοιότυπα χαρακτηριστικά και διαφορετικές ετικέτες), είναι πάντα πιθανόν να κατασκευαστούν δενδρικές δομές με μηδενικό σφάλμα ταξινόμησης, στο σύνολο εκπαίδευσης. Ωστόσο κάτι τέτοιο εγείρει τον κίνδυνο overfit, δηλαδή την αδυναμία του δέντρου να ανταποκριθεί σε δείγματα που δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση και να γενικεύσει κατάλληλα. Η ιδέα ωστόσο ότι για ένα σύνολο δεδομένων μπορούν να κατασκευαστούν διαφορετικά δέντρα απόφασης (δίνοντας το κάθε ένα διαφορετική βαρύτητα σε ορισμένα χαρακτηριστικά), οδήγησε στην μετάβαση απο τα δέντρα απόφασης στα δάση από δέντρα (Random Forest).

2.5.3 Τυχαία Δάση - Random Forest

Τα Τυχαία Δάση, ή τα Δάση Τυχαίας Απόφασης, είναι μια μέθοδος εκμάθησης συνόλου για ταξινόμηση και άλλες εργασίες που λειτουργεί κατασκευάζοντας ένα πλήθος δέντρων αποφάσεων κατά το χρόνο εκπαίδευσης. Για εργασίες ταξινόμησης, η έξοδος του τυχαίου δάσους είναι η κλάση που επιλέγεται από τα περισσότερα δέντρα (Ensemble Models).



Σχήμα 2.6: Κατασκευή Τυχαίου Δάσους

Μια μέθοδος, για τη δημιουργία ensemble μοντέλων είναι η Bootstrap AGGregating (BAGG). Πιο αναλυτικά, όταν δίνεται ένα σύνολο δεδομένων, εξάγονται bootstrapped σύνολα. Ένα τέτοιο σύνολο δημιουργείται επιλέγοντας τυχαία δείγματα από το αρχικό dataset, χωρίς να αποκλείεται να επιλεγεί το ίδιο δείγμα περισσότερες από μία φορές. Στην συνέχεια, κατασκευάζεται ένα δέντρο απόφασης, το οποίο όμως περιλαμβάνει ένα υποσύνολο των αρχικών χαρακτηριστικών στους κόμβους του. Η διαδικασία αυτή (δημιουργία bootstrapped συνόλου - εκπαίδευση δέντρου με ορισμένα χαρακτηριστικά), επαναλαμβάνεται έως ότου συμπληρωθεί ο επιθυμητός αριθμός δέντρων. Κάθε ένα απο τα διαφορετικά δέντρα, δίνει μια

πρόβλεψη για την κλάση στην οποία ανήκει το δείγμα, η οποία μπορει να είναι διαφορετική για κάθε δέντρο απόφασης. Οι προβλέψεις αυτές συλλέγονται (aggregated), και η τελική απόφαση προκύπτει με βάση την πλειοψηφία των προβλέψεων, των επιμέρους δέντρων απόφασης. Η διαδικασία αυτή περιγράφεται σχηματικά στο παραπάνω Σχήμα 2.6.

2.5.4 Μηχανές Διανυσμάτων Υποστήριξης

Οι μηχανές διανυσμάτων υποστήριξης (SVMs) είναι μία ομάδα αλγορίθμων επιβλεπόμενης μάθησης που αρχικά χρησιμοποιήθηκαν για την κατηγοριοποίηση, ενώ αργότερα εφαρμόστηκαν και σε προβλήματα παλινδρόμησης. Αναπτύχθηκαν για πρώτη φορά από τον Vapnik και τους συνεργάτες του στο AT&T Bell Labs το 1992 [47]. Απέσπασαν γρήγορα το ενδιαφέρον, καθώς παρουσίασαν μεγάλη ικανότητα γενίκευσης σε σχέση με άλλες παραδοσιακές μεθόδους ταξινόμησης.

Η βασική ιδέα της κατασκευής τους στηρίζεται στην αρχή ελαχιστοποίησης του κατασκευαστικού ρίσκου (SRM), που έχει αποδειχθεί πως υπερτερεί έναντι της παραδοσιακής ελαχιστοποίησης του εμπειρικού ρίσκου (ERM), στην οποία στηρίζονται τα νευρωνικά δίκτυα. Η κατηγοριοποίηση των δεδομένων στηρίζεται στην εύρεση ενός βέλτιστου υπερεπιπέδου που διαχωρίζει τα δεδομένα δημιουργώντας το μέγιστο περιθώριο. Στην περίπτωση που ο γραμμικός διαχωρισμός είναι αδύνατος, γίνεται χρήση κατάλληλων απεικονίσεων που μεταφέρουν το σύνολο των δεδομένων σε μεγαλύτερη διάσταση ώστε να επιτευχθεί τελικά ο διαχωρισμός τους. Η ικανότητα γενίκευσης της χρήσης των SVM σε μη γραμμικά δεδομένα στηρίζεται στο τέχνασμα του πυρήνα (kernel trick). Κάθε μηχανή διανυσμάτων υποστήριξης είναι ένας δυαδικός ταξινομητής, έχει δηλαδή τη δυνατότητα κατηγοριοποίησης σε δύο κλάσεις. Εάν οι κλάσεις είναι περισσότερες, τότε κρίνεται απαραίτητη η χρήση περισσότερων μηχανών διανυσμάτων υποστήριξης.

Έστω ότι δίνεται ένα σύνολο δεδομένων εκπαίδευσης με n σημεία

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \tag{2.1}$$

,όπου το y_i είναι είτε 1 είτε -1, δηλώνοντας την κλάση στην οποία το σημείο $\mathbf{x}_i$ ανήκει. Κάθε $\mathbf{x}_i$ είναι ένα p-διάστατο πραγματικό διάνυσμα. Σκοπός είναι η εύρεση ενός «κυπερεπίπεδου μέγιστου περιθωρίου» που διαιρεί την ομάδα σημείων $\mathbf{x}_i$ για τα οποία $y_i = 1$, από την ομάδα σημείων για τα οποία $y_i = -1$, η οποία ορίζεται έτσι ώστε η απόσταση μεταξύ των υπερεπίπεδων και του πλησιέστερου σημείου $\mathbf{x}_i$ από οποιαδήποτε ομάδα να μεγιστοποιείται. Οποιοδήποτε υπερεπίπεδο μπορεί να γραφτεί ως το σύνολο των σημείων $\mathbf{x}$ που ικανοποιούν την

$$\mathbf{w}^T\mathbf{x} - b = 0 \tag{2.2}$$

,όπου $\mathbf{w}$ είναι το κανονικό διάνυσμα στο υπερεπίπεδο.

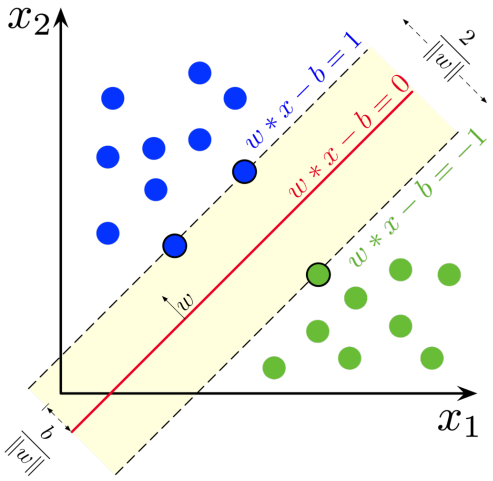
Αν τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρίσιμα, μπορεί να επιλεγούν δύο παράλληλα υπερεπίπεδα που χωρίζουν τις δύο κλάσεις δεδομένων, έτσι ώστε η απόσταση μεταξύ τους να είναι όσο το δυνατόν μεγαλύτερη. Η περιοχή που οριοθετείται από αυτά τα δύο

υπερεπίπεδα ονομάζεται «περιθώριο» και το υπερεπίπεδο μέγιστου περιθωρίου είναι το υπερεπίπεδο που βρίσκεται στη μέση της απόστασης μεταξύ τους. Με ένα κανονικοποιημένο σύνολο δεδομένων, αυτά τα υπερεπίπεδα μπορούν να περιγραφούν από τις εξισώσεις

$$\mathbf{w}^T\mathbf{x} - b \geq 1, y_i = 1 \tag{2.3}$$

$$\mathbf{w}^T\mathbf{x} - b \leq -1, y_i = -1 \tag{2.4}$$

Τα $\mathbf{w}$ και b που λύνουν αυτό το πρόβλημα καθορίζουν τον ταξινομητή.

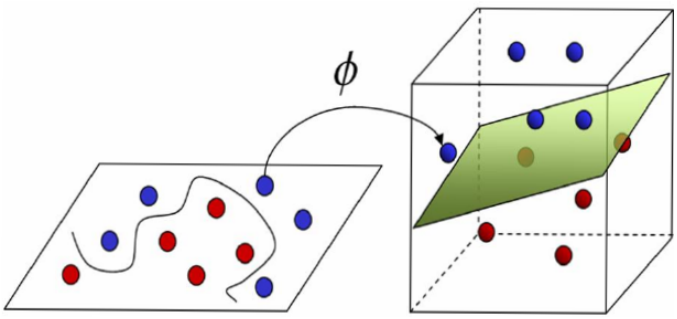


Σχήμα 2.7: Υπερεπίπεδο μέγιστου περιθωρίου για ένα SVM

Σε περιπτώσεις όπου το σύνολο των δεδομένων δεν είναι γραμμικά διαχωρίσιμο, πραγματοποιείται αναγωγή των δεδομένων εισόδου σε έναν νέο χώρο δεδομένων μεγαλύτερων διαστάσεων. Αναλυτικότερα ο χώρος Μ-διαστάσεων των δεδομένων μετασχηματίζεται σε έναν χώρο Μ′ μεγαλύτερων διαστάσεων χρησιμοποιώντας τον παρακάτω μετασχηματισμό.

$$x_i x_j^T = \phi(x_i)\phi^T(x_j) \tag{2.5}$$

Στο παράδειγμα του Σχήματος 2.8, παρατηρείται ότι τα μη γραμμικά διαχωρίσιμα δεδομένα σε χώρο δύο διαστάσων, μετασχηματίζονται σε γραμμικός διαχωρίσιμα στις τρεις διαστάσεις μέσω της συνάρτησης απεικόνισης ϕ.



Σχήμα 2.8: Μετασχηματισμός για μη γραμμικά διαχωρίσιμα δεδομένα (Kernel Trick)

Το εσωτερικό γινόμενο στον νέο υψηλότερο διαστάσεων χώρο ορίζεται από μια συνάρτηση γνωστή ως πυρήνα (Kernel)

$$K(x_i, x_j) = \phi(x_i)\phi^T(x_j) \tag{2.6}$$

Οι πιο ευρέως χρησιμοποιούμενοι πυρήνες είναι ο γραμμικός, ο πολυωνυμικός, ο RBF και ο σιγμοειδής πυρήνας. Τα μοντέλα ταξινόμησης που χρησιμοποιούν τους πυρήνες εμφανίζουν μεγάλη ικανότητα γενίκευσης και μπορούν να χρησιμοποιηθούν σε πολλά σύνολα δεδομένων της καθημερινής ζωής, αφού υπάρχουν πολλές ελεύθερες παράμετροι οι οποίες μπορούν να καθορίζονται ανάλογα με τη φύση του προβλήματος.

2.5.5 Naive Bayes Classifier

Η βασική ιδέα λειτουργίας του ταξινομητή είναι πρώτον ο γνωστός νόμος του Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.7}$$

και δεύτερον η (naive) υπόθεση ότι τα χαρακτηριστικά είναι όλα ανεξάρτητα μεταξύ τους (δεν ισχύει γενικά, αλλά ο ταξινομητής είναι πρακτικά καλός σε πολλές περιπτώσεις). Για παράδειγμα, την υπόθεση αν θα βρέξει σήμερα, θα την προβλέψει με βάση το παρελθόν θεωρώντας ότι τα χαρακτηριστικά θερμοκρασία, νεφοκάλυψη και ατμοσφαιρική πίεση είναι όλα ανεξάρτητα μεταξύ τους. Με δεδομένα μια μεταβλητή κατηγορίας (κλάσης) y και ένα εξαρτώμενο διάνυσμα χαρακτηριστικών x_1 μέχρι x_n , σύμφωνα με το θεώρημα του Bayes θα ισχύει:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)} \tag{2.8}$$

$$P(x_1, \ldots, x_i, \ldots, x_n \mid y) = \prod_{i=1}^{n} P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \tag{2.9}$$

Κάνοντας την αφελή υπόθεση ότι το χαρακτηριστικό x_i για κάθε i εξαρτάται μόνο από την κλάση y και όχι από οποιοδήποτε άλλο χαρακτηριστικό:

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y) \tag{2.10}$$

αυτό οδηγεί στην απλοποίηση

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)} \tag{2.11}$$

Με δεδομένη είσοδο, το $P(x_1, \ldots, x_n)$ είναι σταθερό. Συνεπώς μπορούμε να χρησιμοποιήσουμε τον ακόλουθο κανόνα ταξινόμησης

$$P(y \mid x_1, \ldots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i \mid y) \longrightarrow \hat{y} = \arg\max_y P(y)\prod_{i=1}^{n} P(x_i \mid y) \tag{2.12}$$

Το $P(y)$ είναι η υπόθεσή και ισούται με τη σχετική συχνότητα της κλάσης y στο τραινινγ σετ. Το $P(x_i \mid y)$ είναι η πιθανοφάνεια δηλαδή η πιθανότητα του δείγματος με δεδομένη την υπόθεση και μπορεί επίσης να υπολογιστεί απλά από το training set. Οι διάφοροι Naive Bayes

classifiers διαφοροποιούνται κυρίως από τις υποθέσεις που κάνουν ως προς την κατανομή $P(x_i \mid y)$. Η κλάση $\hat{y}$ που ανατίθεται σε ένα νέο δείγμα είναι αυτή που μεγιστοποιεί το δεξί μέλος της σχέσης.

Σε περίπτωση που έχουμε συνεχείς μεταβλητές, θα θεωρηθεί ότι η κατανομή κάθε χαρακτηριστικού ως προς κάθε κλάση ακολουθεί την κανονική κατανομή:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{2.13}$$

Ο συγκεκριμένος ταξινομητής είναι ο **Gaussian Naive Bayes**. Πρακτικά, για κάθε κλάση υπολογίζεται η μέση τιμή μ_y και η διακύμανση σ_y^2 κάθε χαρακτηριστικού για τη συγκεκριμένη κλάση. Όσο πιο κοντά στη μέση τιμή του είναι ένα χαρακτηριστικό ενός δείγματος, τόσο πιο κοντά στη μονάδα θα είναι η πιθανοφάνια του χαρκτηριστικού και αντιθετοαντίστροφα.

2.6 Αλγόριθμοι Μη Επιβλεπόμενης Μάθησης

Στην επιβλεπόμενη μάθηση μας δίνεται ένα σύνολο δεδομένων με τις αντίστοιχες κλάσεις-ετικέτες κάθε εγγραφής. Στόχος είναι η δημιουργία ενός μοντέλου, το οποίο να μπορεί να κατηγοριοποιήσει νέα δεδομένα σε κάποια από τις προϋπάρχουσες κλάσεις. Αντίθετα, στη μη επιβλεπομένη μάθηση μας δίνεται ένα σύνολο δεδομένων, χωρίς τις αντίστοιχες κλάσεις-ετικέτες κάθε εγγραφής και στόχος είναι η χρήση κάποιου αλγορίθμου, ώστε αυτόματα να ανακαλύψουμε κάποια ενδεχομένως ενδιαφέρουσα δομή των δεδομένων. Για παράδειγμα, η συσταδοποίηση είναι μια από τις τεχνικές μη επιβλεπόμενης μάθησης. Δοθέντων κάποιων δεδομένων χωρίς κλάσεις, οι αλγόριθμοι συσταδοποίησης ομαδοποιούν τα δεδομένα σε συστάδες, έτσι ώστε εγγραφές, οι οποίες ανήκουν στην ίδια συστάδα, να έχουν όμοια ή παραπλήσια χαρακτηριστικά.

2.6.1 Έννοια της Συστάδας

Στο πρόβλημα της συσταδοποίησης δίνεται ένα σύνολο δεδομένων, χωρίς τις αντίστοιχες κλάσεις ή ετικέτες και χρειαζόμαστε κάποιον αλγόριθμο, ο οποίος θα ομαδοποιήσει αυτόματα τα δεδομένα σε συστάδες. Οι συστάδες που δημιουργούνται διαχωρίζουν ορθά τα δεδομένα. Αυτό πρακτικά σημαίνει ότι μια συστάδα απαρτίζεται από αντικείμενα, όπου κάθε αντικείμενο είναι πιο κοντά σε κάθε άλλο αντικείμενο της ίδιας συστάδας απ' ότι σε κάποιο άλλο αντικείμενο διαφορετικής συστάδας.

2.6.2 Αλγόριθμος K-Μέσων (K-Means)

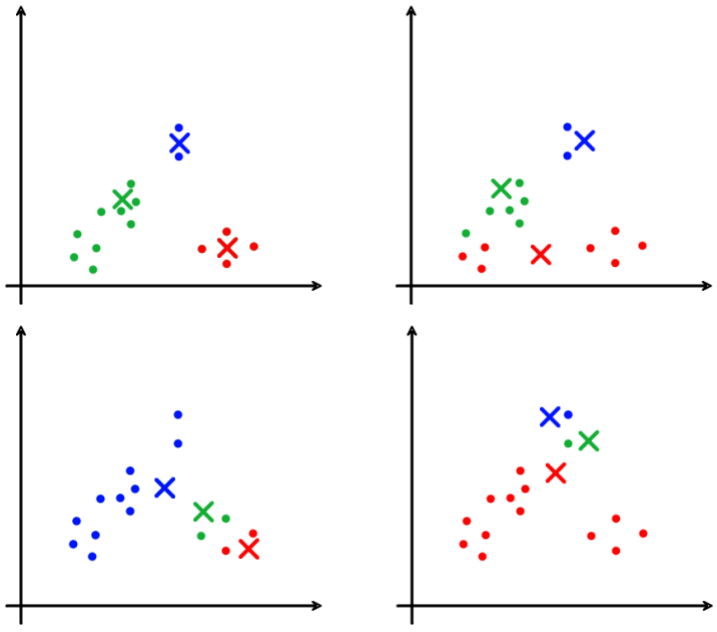
Ο αλγόριθμος k-means ξεκινάει με k τυχαία σημεία, τα οποία ονομάζονται κεντροειδή της συστάδας και δηλώνουν το κέντρο βάρους της συστάδας. Το k υποδηλώνει σε πόσες συστάδες ο αλγόριθμος θα χωρίσει τα δεδομένα. Ο αλγόριθμος εκτελεί επαναληπτικά δύο βήματα. Το πρώτο βήμα αφορά την ανάθεση σε κάποια συστάδα, ενώ το δεύτερο βήμα αφορά τον επαναπροσδιορισμό και τη μετατόπιση του κεντροειδούς κάθε συστάδας [48].

Πιο αναλυτικά, όσον αφορά στο πρώτο βήμα, δηλαδή την ανάθεση σε κάποια συστάδα, ο αλγόριθμος εξετάζει κάθε δείγμα σε σχέση με τα κεντροειδή των συστάδων. Με χρήση κάποιου μέτρου απόστασης (συνήθως Ευκλείδειας), αναθέτει το εξεταζόμενο δείγμα στη συστάδα, της οποίας το κεντροειδές είναι το πλησιέστερο ως προς το συγκεκριμένο δείγμα. Στο δεύτερο βήμα, παίρνοντας τον μέσο όρο των δειγμάτων κάθε συστάδας, επανυπολογίζονται τα κεντροειδή της κάθε συστάδας, ώστε το κεντροειδές να είναι πιο αντιπροσωπευτικό στην πρόσφατα διαμορφωμένη συστάδα.

Ο αλγόριθμος εκτελεί επαναληπτικά αυτά τα δύο βήματα, μέχρις ότου τα κεντροειδή των συστάδων να μετατοπίζονται ελάχιστα και σε απόσταση μικρότερη από κάποια δοθείσα τιμή κατωφλίου. Ως εναλλακτικό κριτήριο τερματισμού του αλγορίθμου μπορεί να χρησιμοποιηθεί και ο αριθμός επαναλήψεων.

Τυχαία Αρχικοποίηση Κεντροειδών

Το πρώτο βήμα του αλγορίθμου k-means είναι η τυχαία αρχικοποίηση των k κεντροειδών των συστάδων. Παρόλο που το συγκεκριμένο βήμα φαίνεται απλό και ασήμαντο, αξίζει να σημειωθεί ότι αρκετές φορές μια «κακή» αρχικοποίηση μπορεί να οδηγήσει σε κακής ποιότητας συστάδες στην πορεία. Στο Σχήμα 2.9 βλέπουμε ένα παράδειγμα τεσσάρων τυχαίων αρχικοποιήσεων των κεντροειδών, που εν τέλη καταλήγουν να διαφοροποιούν τις συστάδες που δημιουργεί ο αλγόριθμος.



Σχήμα 2.9: Τυχαία αρχικοποίηση κέντροειδών

Επιλογή του Αριθμού Συστάδων

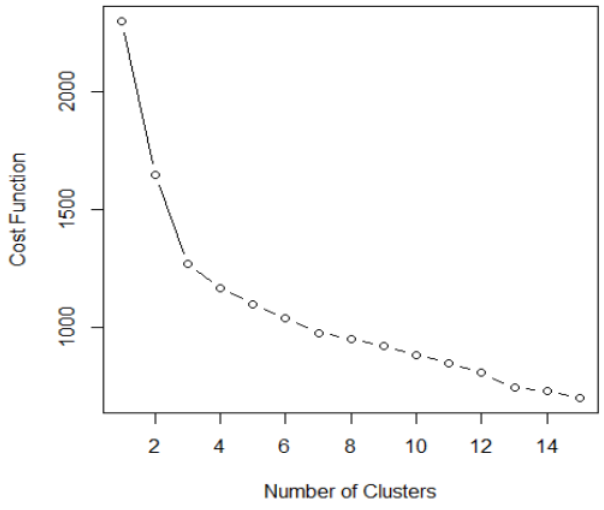
Ένα από τα μειονεκτήματα του αλγορίθμου Κ-Μέσων είναι το γεγονός ότι δεν υπάρχει κάποιος αυτοματοποιημένος τρόπος επιλογής του κ, δηλαδή του αριθμού των συστάδων. Ο

αριθμός των συστάδων δίνεται ως είσοδος από τον χρήστη και η επιλογή του σωστού αριθμού επαφίεται στη δική του γνώση και εμπειρία. Να υπενθυμίσουμε ότι κατά τη συσταδοποίηση δεν δίνεται το επιπλέον χαρακτηριστικό κλάσης των δειγμάτων. Συνεπώς, η διαδικασία επιλογής του αριθμού συστάδων, ενδεχομένως, να απαιτήσει την εξερεύνηση και μελέτη των δεδομένων, για παράδειγμα, μέσα από οπτικοποιήσεις, προκειμένου να καταλήξουμε στον σωστό αριθμό συστάδων.

Ένα απλό και πρακτικό τέχνασμα, το οποίο μπορεί να βοηθήσει σε ορισμένες περιπτώσεις και χρησιμοποιήθηκε σε αυτή την εργασία, είναι «ο κανόνας του αγκώνα» (the Elbow Method). Εκτελούμε τον αλγόριθμο για πολλές τιμές του k, από πολύ μικρές έως πολύ μεγάλες. Για κάθε εκτέλεση, υπολογίζεται ένας δείκτης εκτίμησης των συστάδων. Εν προκειμένω, αυτός ο δείκτης είναι το άθροισμα του τετραγώνου των σφαλμάτων (SSE ή Inertia), που ορίζεται ως το άθροισμα των τετραγώνων των αποστάσεων ανάμεσα σε κάθε δεδομένο και το κεντροειδές του cluster στο οποίο ανήκει. Δηλαδή:

$$SSE = \sum_{i=1}^{k} \sum_{x \epsilon c_i} dist(x, c_i)^2 \tag{2.14}$$

Αν σχεδιαστεί η γραφική παράσταση του SSE συναρτήσει του αριθμού των συστάδων k, θα παρατηρηθεί ότι ο δείκτης SSE, δηλαδή το σφάλμα, μειώνεται όσο το k αυξάνεται, το οποίο είναι αναμενόμενο με βάση τα όσα αναφέρθηκαν παραπάνω. Για κάποιο k, η γραφική παράσταση μπορεί να παρουσιάζει μια έντονη γωνία (αγκώνα), εξ' ου και το όνομα της μεθόδου. Αυτή η τιμή του k είναι που επιλέγεται ως βέλτιστη για την εκτέλεση του αλγορίθμου. Ένα χαρακτηριστικό παράδειγμα παρουσιάζεται στο Σχήμα 2.10, όπου ο κανόνας του αγκώνα υποδυκνύει ότι η επιλογή k=3 είναι αρκετά ικανοποιητική. Ωστόσο, αξίζει να σημειωθεί ότι υπάρχουν περιπτώσεις, όπου η γραφική παράσταση είναι πιο ομαλή και δεν έχει τον τύπο σχήματος του αγκώνα, με αποτέλεσμα η επιλογή και πάλι να μην είναι ξεκάθαρη και να επαφίεται στη γνώση και στην εμπειρία του χρήστη.



Σχήμα 2.10: Η μέθοδος του αγκώνα

2.6.3 Μοντέλα Μείξης (Mixture Models)

Η Μείξη Γενικών Μοντέλων (General Mixture Models) είναι ένα μη επιβλεπόμενο πιθανοτικό μοντέλο που αποτελείται από πολλαπλές κατανομές, που συνήθως αναφέρονται ως συνιστώσες (components), και αντίστοιχα βάρη. Αυτό επιτρέπει να μοντελοποιηθούν πιο σύνθετες κατανομές που αντιστοιχούν σε ένα μοναδικό υποκείμενο φαινόμενο.

Στη στατιστική, ένα μοντέλο μείξης είναι ένα πιθανό μοντέλο για την αναπαράσταση της παρουσίας υποπληθυσμών σε έναν συνολικό πληθυσμό, χωρίς να απαιτείται ότι ένα παρατηρούμενο σύνολο δεδομένων πρέπει να προσδιορίζει τον υποπληθυσμό στον οποίο ανήκει μια μεμονωμένη παρατήρηση. Τυπικά, ένα mixture model αντιστοιχεί στην κατανομή του μείγματος που αντιπροσωπεύει την κατανομή πιθανοτήτων των παρατηρήσεων στο συνολικό πληθυσμό. Ωστόσο, τα προβλήματα που σχετίζονται με τις «μίξεις κατανομών» αφορούν την εξαγωγή των ιδιοτήτων του συνολικού πληθυσμού από εκείνες των υποπληθυσμών. Επομένως, τα «mixture models» χρησιμοποιούνται για την εξαγωγή στατιστικών συμπερασμάτων σχετικά με τις ιδιότητες των υποπληθυσμών, δίνοντας μόνο παρατηρήσεις για συγκεντρωτικό πληθυσμό, χωρίς στοιχεία ταυτότητας υποπληθυσμού.

Προκειμένου να το αναπαραστήσουμε μαθηματικά, διατυπώνουμε το μοντέλο με όρους λανθανουσών μεταβλητών (latent model analysis), που συνήθως συμβολίζονται με z. Αυτές είναι μεταβλητές που δεν μπορούν να μετρηθούν ή να παρατηρηθούν άμεσα. Ο αλγόριθμος μάθησης πρέπει δηλαδή να καταλάβει τι αντιπροσωπεύουν, χωρίς καθορισμό με το χερί από άνθρωπο.

Στα mixture models, η λανθάνουσα μεταβλητή αντιστοιχεί στη συνιστώσα μείξης (mixture component). Παίρνει τιμές σε ένα διακριτό σύνολο, το οποίο συμβολίζεται $\{1, ..., K\}$. Γενικά, ένα μοντέλο μείξης υποθέτει ότι τα δεδομένα δημιουργούνται από την ακόλουθη διαδικασία: πρώτα δειγματίζουμε το z και μετά δειγματίζουμε τα παρατηρήσιμα στοιχεία x από μια κατανομή που εξαρτάται από το z, δηλ.

$$p(z, x) = p(z)p(x|z) \tag{2.15}$$

,όπου το $p(z)$ είναι μια πολυωνυμική κατανομή. Το $p(x|z)$ μπορεί να πάρει μια ποικιλία από παραμετρικές μορφές όπως: Gaussian, Uniform, Bernoulli, Normal, Log-Normal, Exponential, Poisson, Beta-Bernoulli, Gamma, Multivariate Gaussian, Dirichlet Distribution κτλ.

Η **Συνάρτηση Πυκνότητας Πιθανότητας** (PDF) πάνω στο x υπολογίζεται περιθωριοποιώντας ή αθροίζοντας το z:

$$p(\mathbf{x}) = \sum_{z} p(z)p(\mathbf{x}|z) \tag{2.16}$$

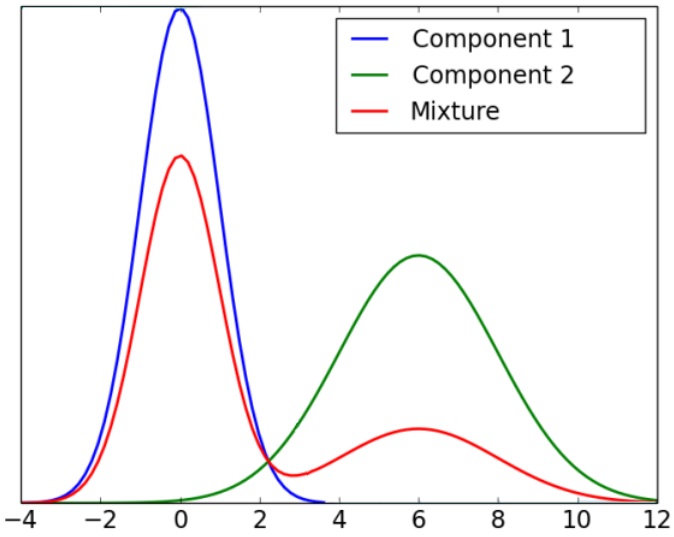
$$p(\mathbf{x}) = \sum_{k=1}^{K} Pr(z = k)p(\mathbf{x}|z = k) \tag{2.17}$$

Οι εξισώσεις 2.16 και 2.17 είναι δύο διαφορετικοί τρόποι σύνταξης της PDF. Η πρώτη είναι γενικότερη (αφού ισχύει για άλλα τα μοντέλα λανθάνουσας μεταβλητής), ενώ η δεύτερη τονίζει

το νόημα του μοντέλου για συσταδοποίηση. Η PDF είναι ένας convex combination (γραμμικός συνδυασμός που οι συντελεστές αθροίζουν στην μονάδα) των PDF των διαφορετικών συνιστωσών.

Μοντέλα Μείξης Γκαουσιανών (Gaussian Mixture Models)

Υποθέτοντας ότι έχουμε μόνο Γκαουσιανές κατανομές, τότε το μοντέλο ονομάζεται Gaussian Mixture Model (GMM).



Σχήμα 2.11: Παράδειγμα Gaussian Mixture Model

Για παράδειγμα, το Σχήμα 2.11 δείχνει ένα παράδειγμα μείξης μοντέλου γκαουσιανών με 2 συνιστώσες (components) που έχει την ακόλουθη διαδικασία παραγωγής:

- Με πιθανότητα 0.65, επιλέγεται η συνιστώσα 1, διαφορετικά επιλέγεται η συνιστώσα 2 με πιθανότητα 0.35.
- Εάν επιλεχθεί η συνιστώσα 1, τότε το δείγμα x είναι από μια Γκαουσιανή κατανομή με μέση τιμή 0 και απόκλιση 1.
- Εάν επιλεχθεί η συνιστώσα 2, τότε το δείγμα x είναι από μια Γκαουσιανή κατανομή με μέση τιμή 6 και απόκλιση 2.

Αυτό μπορεί να αναπαρασταθεί με μια πιο συμπαγή μαθηματική έκφραση:

$$z \sim Multinomial(0.65, 0.35) \tag{2.18}$$

$$x|z = 1 \sim Gaussian(0, 1) \tag{2.19}$$

$$x|z = 2 \sim Gaussian(6, 2) \tag{2.20}$$

Επομένως, στη γενική περίπτωση έχουμε,

$$z \sim Multinomial(\pi) \tag{2.21}$$

$$x|z = k \sim Gaussian(\mu_k, \sigma_k) \tag{2.22}$$

,όπου το π είναι ένα διάνυσμα πιθανοτήτων (με μη αρνητικές τιμές που αθροίζονται στο 1), γνωστό ως αναλογίες μείξης (mixing proportions). Στο συγκεκριμένο παράδειγμα του Σχήματος 2.11 η PDF είναι με βάση τα παραπάνω:

$$p(x) = 0.65 \cdot Gaussian(0, 1) + 0.35 \cdot Gaussian(6, 2) \tag{2.23}$$

Εκπαίδευση - Μάθηση Μοντέλων Μείξης

Η εκμάθηση ενός μοντέλου μείξης είναι μια προσέγγιση για την συσταδοποίηση και το μοντέλο οφείλει να μάθει δύο σετ παραμέτρων:

- Τις παραμέτρους που σχετίζονται με την κατανομή κάθε στοιχείου k. Για Γκαουσιανές, για παράδειγμα, τη μέση τίμη μ_k και την τυπική απόκλιση σ_k που σχετίζονται με κάθε στοιχείο. (Για άλλα είδη μοντέλων μείξης υπάρχουν άλλα σύνολα παραμέτρων).
- Τις αναλογίες μείξης π_k, που ορίζονται ως $Pr(z = k)$.

Η εκπαίδευση των General Mixture Models αντιμετωπίζει το κλασικό chicken-and-egg πρόβλημα που αντιμετωπίζουν οι περισσότεροι αλγόριθμοι μη επιβλεπόμενης μάθησης.

Από τη μία πλευρά, εάν γνωρίζουμε σε ποια συνιστώσα ανήκει ένα δείγμα, μπορούμε να χρησιμοποιήσουμε εκτίμηση μέγιστης πιθανότητας MLE (**Maximum Likelihood Estimation**) για να ενημερώσουμε τη συνιστώσα. Σύμφωνα με τη στατιστική, αυτό επιτυγχάνεται με τη μεγιστοποίηση μιας συνάρτησης πιθανότητας, έτσι ώστε, σύμφωνα με το υποτιθέμενο στατιστικό μοντέλο, τα παρατηρούμενα δεδομένα να είναι πιο πιθανά. Το σημείο στο χώρο των παραμέτρων που μεγιστοποιεί τη συνάρτηση πιθανότητας ονομάζεται εκτίμηση μέγιστης πιθανότητας [49]. Η λογική της μέγιστης πιθανότητας είναι τόσο διαισθητική όσο και ευέλικτη, και ως εκ τούτου η μέθοδος έχει γίνει κυρίαρχο μέσο στατιστικής εξαγωγής συμπερασμάτων.[50] [51]

Από την άλλη πλευρά, εάν γνωρίζουμε τις παραμέτρους των συνιστωσών, μπορούμε εύκολα να προβλέψουμε ποιο δείγμα ανήκει σε ποια συνιστώσα.

Το παραπάνω πρόβλημα επιλύεται χρησιμοποιώντας τον αλγόριθμο EM (**Expectation Maximization**), ο οποίος επαναλαμβάνεται μεταξύ των δύο μέχρι τη σύγκλιση. Ουσιαστικά, επιλέγεται ένα σημείο αρχικοποίησης που συνήθως δεν είναι μια πολύ καλή αρχή, αλλά μέσω διαδοχικών βημάτων επανάληψης, οι παράμετροι συγκλίνουν σε ένα καλό αποτέλεσμα.

Ο αλγόριθμος EM αποτελεί μία αριθμητική μέθοδο που χρησιμοποιείται για την εύρεση εκτιμητριών μεγίστων πιθανοφανιών για συγκεκριμένα προβλήματα. Επειδή έχει κατασκευαστεί με στατιστικές τεχνικές προσφέρει αρκετή πληροφορία στην επίλυση στατιστικών προβλημάτων. Ο αλγόριθμος EM μελετήθηκε διεξοδικά από τους Dempster το 1977, αλλά προϋπήρχε σε διάφορες μορφές πριν από αυτή τη χρονολογία (πρωτοεμφανίστηκε στις αρχές του 1900). Ο αλγόριθμος πήρε το όνομά του επειδή η εκτέλεσή του περιλαμβάνει δύο βήματα. Πρώτον, τον υπολογισμό των εκτιμήσεων (expectations) και δεύτερον την εφαρμογή της ενημέρωσης μέγιστης πιθανότητας με βάση αυτές τις εκτιμήσεις. Πιο συγκεκριμένα εφαρμόζονται τα εξής μέχρι να επέλθει η σύγκλιση:

- E-step: Υπολογίζει τις εκτιμήσεις των λανθάνουσσων μεταβλητών, που συνήθως ορίζονται ως οι μεταγενέστερες πιθανότητες (posterior probabilities):

$$r_k^{(i)} \leftarrow Pr(z^{(i)} = k|x^{(i)}) \tag{2.24}$$

- M-step: Υπολογίζει τις παραμέτρους μέγιστης πιθανότητας με βάση αυτές τις εκτιμήσεις:

$$\theta \leftarrow \arg\max_{\theta} \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} [\log Pr(z^{(i)} = k) + \log p(\mathbf{x}^{(i)}|z^{(i)} = k)] \tag{2.25}$$

Οι συνθήκες τερματισμού μπορεί να είναι με βάση τη μεταβολή της πιθανοφάνειας ή με βάση τις μεταβολές των παραμέτρων, υποδεικνύοντας περισσότερο έλλειψη προόδου παρά σύγκλιση. Γενικά, χρειάζεται ένας μέτριος αριθμός επαναλήψεων (για παράδειγμα 20-50) για να καταλήξει ο αλγόριθμος κοντά σε ένα τοπικό βέλτιστο, αλλά όταν είναι κοντά στη βέλτιστη λύση μπορεί να χρειαστεί πολύς χρόνος για να την εντοπίσει. Παρά την ευρετική αιτιολόγηση του αλγορίθμου, στην πραγματικότητα έχει αρκετά ισχυρές εγγυήσεις, καθώς κάθε επανάληψη του αυξάνει σίγουρα την πιθανοφάνεια.

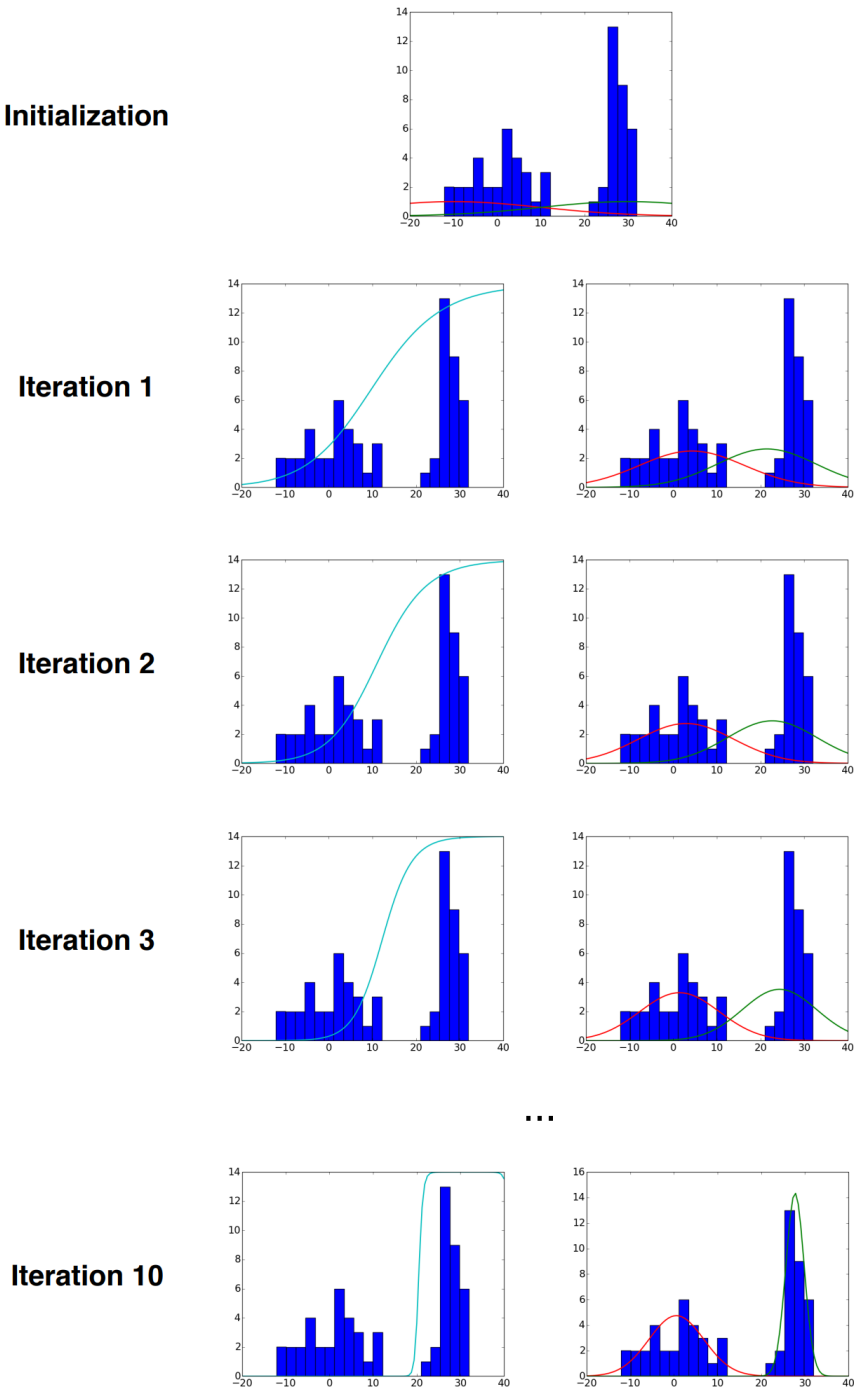
Στο Σχήμα 2.12 παρουσιάζεται ένα παράδειγμα οπτικοποίησης των βημάτων του αλγορίθμου E-M. Αριστερά είναι το E-step, όπου εμφανίζεται η posterior πιθανότητα ως συνάρτηση του x και δεξιά το M-step.

Επιλογή του Αριθμού Συστάδων-Συνιστωσών

Ομοίως με τον αλγόριθμο K-Μέσων, δεν υπάρχει κάποιος αυτοματοποιημένος τρόπος επιλογής του αριθμού των συνιστωσών (components), δηλαδή του αριθμού των συστάδων. Όπως αναφέρθηκε και προηγουμένως, ο ο αλγόριθμος Expectation Maximization εγγυάται μόνο ότι προσεγγίζει ένα τοπικό βέλτιστο σημείο, αλλά δεν εγγυάται ότι αυτό το τοπικό βέλτιστο είναι επίσης και το ολικό. Δηλαδή, εάν ο αλγόριθμος ξεκινά από διαφορετικά σημεία αρχικοποίησης, γενικά μπορεί να καταλήξει και σε διαφορετικά αποτελέσματα. Ένας τρόπος για να αντιμετωπιστεί αυτό είναι να εκτελεστεί η διαδικασία εκτίμησης πολλές φορές και να ληφθεί υπόψη η μέση τιμή και η τυπική απόκλιση για κάθε εκτέλεση. Δεδομένου ότι δεν είναι γνωστή η βασική αλήθεια των γεννητριών κατανομών, δηλαδή δεν γνωρίζουμε την αρχική κατανομή που δημιούργησε τα δεδομένα, οι επιλογές σχετικά με την αξιολόγηση απόδοσης της διαδικασίας συσταδοποίησης είναι περιορισμένες και αρκετά δύσκολες. Μια τεχνική που χρησιμοποιήθηκε και στην συγκεκριμένη εργασία είναι το Bayesian information criterion (BIC).

Στη στατιστική, το **Bayesian Information Criterion** είναι ένα κριτήριο για την επιλογή μοντέλου μεταξύ ενός πεπερασμένου συνόλου μοντέλων. Τα μοντέλα με το χαμηλότερο BIC προτιμώνται συνήθως σε σχέση με τα υπόλοιπα. Βασίζεται, εν μέρει, στη συνάρτηση πιθανότητας και συνδέεται στενά με το Akaike information criterion (AIC).

Κατά την εκπαίδευση των μοντέλων, είναι δυνατό να αυξηθεί η πιθανότητα προσθέτοντας παραμέτρους, αλλά κάτι τέτοιο μπορεί να οδηγήσει σε υπερβολικό φιτάρισμα (overfitting).



Σχήμα 2.12: Παράδειγμα βημάτων αλγορίθμου E-M

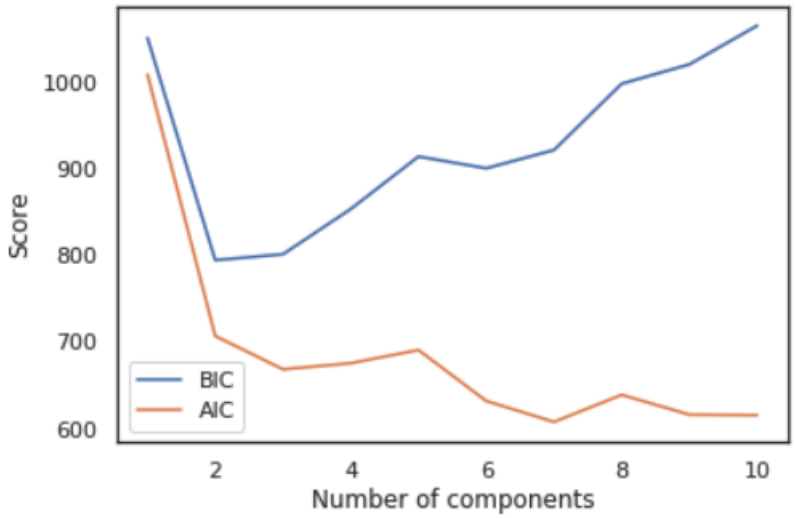
Τόσο το BIC, όσο και το AIC προσπαθούν να επιλύσουν αυτό το πρόβλημα εισάγοντας έναν όρο ποινής για τον αριθμό των παραμέτρων στο μοντέλο. Αυτός ο όρος ποινής είναι μεγαλύτερος στο BIC από ότι στο AIC [52] και για αυτό το λόγο προτιμήθηκε. Το BIC αναπτύχθηκε από τον Gideon E. Schwarz και δημοσιεύτηκε σε μια εργασία του [53] το 1978, όπου έδωσε ένα Μπεϋζιανό επιχείρημα για την υιοθέτησή του.

Το BIC ορίζεται επίσημα [54] ως:

$$BIC = k \ln n - 2 \ln \hat{L} \tag{2.26}$$

,όπου:

- $\hat{L}$: Η μέγιστη τιμή της συνάρτησης πιθανότητας του μοντέλου M, $\hat{L} = p(x|\hat{\theta}, M)$, με $\hat{\theta}$ να είναι οι τιμές των παραμέτρων που μεγιστοποιούν τη συνάρτηση πιθανότητας.
- x: Τα παρατηρήσιμα δεδομένα.
- n: Ο αριθμός των δεδομένων x, δηλαδή το μέγεθος του δείγματος.
- k: Ο αριθμός των παραμέτρων που υπολογίζονται από το μοντέλο.



Σχήμα 2.13: AIC και BIC scores συναρτήσει του αριθμού των components

Ένα χαρακτηριστικό παράδειγμα παρουσιάζεται στο Σχήμα 2.13, όπου δείχνει το AIC και το BIC ως συνάρτηση του αριθμού των συνιστωσών για ένα σύνολο δεδομένων. Ο βέλτιστος αριθμός θα αντιστοιχεί στην ελάχιστη τιμή AIC ή BIC. Το BIC υποδεικνύει ότι ο βέλτιστος αριθμός είναι περίπου 2 με 3. Από την άλλη πλευρά, το AIC υποδεικνύει ότι ο βέλτιστος αριθμός components θα έπρεπε να ήταν περίπου 6 με 7. Η μέθοδος Elbow, που αναφέρθηκε και προηγουμένως, επιτρέπει να προσδιοριστεί ο βέλτιστος αριθμός components χρησιμοποιώντας τα AIC και BIC. Επιλέγεται το σημείο δηλαδή στο οποίο η μείωση του σκορ γίνεται σημαντικά μικρότερη. Αυτή είναι μια ευρετική μέθοδος για τον προσδιορισμό του αριθμού των συνιστωσών. Στην συγκεκριμένη γραφική παράσταση, ο βέλτιστος αριθμός που προκύπτει είναι 3.

Εκτίμηση Μοντέλων Μείξης

Τα μοντέλα μείξης, όπως είναι φανερό, είναι και παραγωγικά μοντέλα, δηλαδή μπορούν να παράξουν νέα τεχνητά δεδομένα με βάση την εκπαιδευμένη τους κατανομή. Όμως, για να μπορέσει να εκτιμηθεί το αποτέλεσμα της προσομοίωσης χρειάζονται ορισμένες μέθοδοι είτε μέσω μετρικών, είτε μέσω οπτικών διαγραμμάτων. Στην παρούσα εργασία, οι τεχνικές που ακολουθήθηκαν για το evaluation των Mixture Models είναι οι εξής:

- CDF Plots
- Q-Q Plots
- Kullback-Leibler Divergence
- Maximum Mean Discrepancy

Η **Kullback-Leibler Divergence**, ή αλλιώς και relative entropy, ποσοτικοποιεί πόσο μια κατανομή πιθανότητας διαφέρει από την άλλη [55]. Η απόκλιση KL μεταξύ δύο κατανομών Q και P δηλώνεται συχνά χρησιμοποιώντας τον συμβολισμό $KL(P \parallel Q)$, όπου το " $\parallel$ " υποδεικνύει απόκλιση της P από τη Q. Η απόκλιση KL μπορεί να υπολογιστεί ως το θετικό άθροισμα της πιθανότητας κάθε συμβάντος στη P πολλαπλασιασμένο με το λογάριθμο της πιθανότητας του συμβάντος στη P έναντι της πιθανότητας του γεγονότος στη Q.

$$KL(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right). \tag{2.27}$$

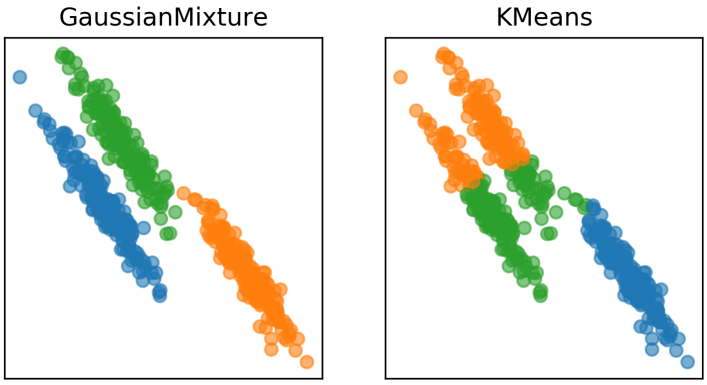
Η διαίσθηση για την απόκλιση KL είναι ότι όταν η πιθανότητα για ένα γεγονός από το P είναι μεγάλη, αλλά η πιθανότητα για το ίδιο γεγονός στη Q είναι μικρή, υπάρχει μεγάλη απόκλιση. Όταν η πιθανότητα από τη P είναι μικρή και η πιθανότητα από τη Q είναι μεγάλη, υπάρχει επίσης μεγάλη απόκλιση, αλλά όχι τόσο μεγάλη όσο η πρώτη περίπτωση. Η βιβλιοθήκη SciPy της Python παρέχει τη συνάρτηση rel_entr για τον υπολογισμό της σχετικής εντροπίας, η οποία ταιριάζει με τον ορισμό της απόκλισης KL. Για να αντιμετωπιστούν τα μηδενικά στις PDF, χρησιμοποιούμε μια διόρθωση Laplace. Ουσιαστικά, προσθέτει ένα count σε όλα τα bins και κανονικοποιεί εκ νέου. Επιλέγουμε να προσθέσουμε 0.5 με βάση και τον εκτιμητή Krichevsky-Trofimov.

Η **Maximum Mean Discrepancy (MMD)** είναι μια στατιστική μετρική βασισμένη στον πυρήνα που χρησιμοποιείται για να προσδιορίσει εάν δύο δεδομένες κατανομές είναι παρόμοιες [56]. Η MMD μπορεί να χρησιμοποιηθεί ως συνάρτηση απώλειας/κόστους σε διάφορους αλγόριθμους μηχανικής μάθησης, όπως εκτίμηση πυκνότητας, παραγωγικά μοντέλα και επίσης σε αναστρέψιμα νευρωνικά δίκτυα. Σε αντίθεση με τα παραγωγικά ανταγωνιστικά δίκτυα (GAN) που απαιτούν μια λύση σε ένα σύνθετο πρόβλημα βελτιστοποίησης ελάχιστης μέγιστης τιμής, τα κριτήρια MMD μπορούν να χρησιμοποιηθούν ως απλούστεροι παράγοντες διάκρισης. Στην παρούσα εργασία χρησιμοποιήθηκε η MMD με **polynomial kernel**, με το μόνο μειονέκτημα ότι δεν μπορεί να διακρίνει κατανομές με τον ίδιο μέσο όρο και διακύμανση, αλλά διαφορετική κύρτωση.

2.6.4 Σύγκριση Gaussian Mixture Models με K-Means

Η ιδέα της συσταδοποίησης με τον αλγόριθμο K-Means είναι αρκετά απλή στην κατανόηση, σχετικά εύκολη στην υλοποίηση και μπορεί να εφαρμοστεί σε πολλές περιπτώσεις χρήσης. Υπάρχουν όμως ορισμένα μειονεκτήματα και περιορισμοί που αξίζει να σημειωθούν. Συγκεκριμένα, ένα από τα πιο σημαντικά είναι ότι οι συστάδες που δημιουργούνται από τον αλγόριθμο έχουν κυκλικό σχήμα. Αυτό συμβαίνει επειδή τα κεντροειδή των συστάδων ενημερώνονται επαναληπτικά χρησιμοποιώντας τη μέση τιμή. Εξετάζοντας για παράδειγμα το ακόλουθο Σχήμα 2.14, όπου η κατανομή των σημείων δεν είναι σε κυκλική μορφή, ο K-Means αποτυγχάνει να προσδιορίσει τις σωστά συστάδες. Σε αυτό το σημείο εισέρχονται να δώσουν τη λύση τα Gaussian Mixture Models, τα οποία αντί να βασίζονται στην απόσταση, βασίζονται στη κατανομή και έτσι μπορούν να προσδιορίσουν ακόμα και ελλειπτικές συστάδες. Με τα Gaussian Mixtures υπάρχει η δυνατότητα διαφορετικών επιλογών για τον περιορισμό της συνδιακύμανσης των εκτιμώμενων κατηγοριών διαφοράς όπως σφαιρική, διαγώνια, συνδεδεμένη ή πλήρης συνδιακύμανση.

Συμπερασματικά, ο πιο κλασικός αλγόριθμος συσταδοποίησης είναι ο K-Means, λόγω του ότι είναι αρκετά απλός και εύκολος στην χρήση και στην κατανόηση. Παρόλο αυτά, είναι πολύ περιορισμένος ως προς το σχήμα κάθε κέντρου συστάδας. Επίσης, είναι ένας κατά προσέγγιση αλγόριθμος που εξαρτάται από την αρχικοποίηση των κεντροειδών. Τα μοντέλα μείξης Γκαουσιανών, είναι ένα είδος αναβάθμισης του αλγορίθμου K-Μέσων, λόγω του ότι μπορούν να μοντελοποιήσουν και τη συνδιακύμανση και έτσι δεν περιορίζονται μόνο σε σφαιρικές συστάδες.



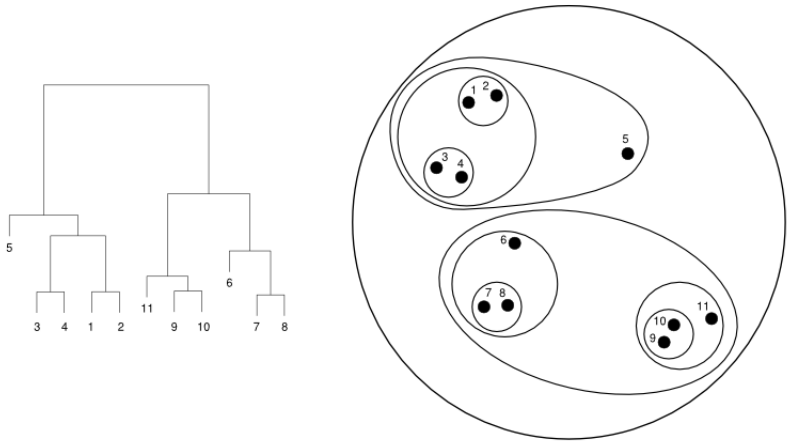
Σχήμα 2.14: Παράδειγμα σύγκρισης K-Means με GMM

2.6.5 Ιεραρχικοί Αλγόριθμοι Συσταδοποίησης

Οι ιεραρχικοί αλγόριθμοι συσταδοποίσης, όπως δηλώνει και το όνομά τους, δημιουργούν μια ιεραρχία εμφωλιασμένων συσταδοποιήσεων. Δηλαδή, συστάδες περιέχουν μεμονωμένα στοιχεία και άλλες συστάδες, οι οποίες με τη σειρά τους μπορεί να περιέχουν και αυτές άλλες, μικρότερες συστάδες, δημιουργώντας έτσι τα επίπεδα της ιεραρχίας [48].

Οι ιεραρχικοί αλγόριθμοι διακρίνονται σε δύο υποκατηγορίες: τους συσσωρευτικούς και

τους διαιρετικούς. Οι αλγόριθμοι μπορούν να αναπαρασταθούν πλήρως με δενδρογράμματα, δηλαδή με δενδρικά διαγράμματα, τα οποία παρουσιάζουν τη διάταξη των συστάδων που δημιουργήθηκαν από την ιεραρχική συσταδοποίηση και καταγράφουν τις ακολουθίες από συγχωνεύσεις (merges) και διαχωρισμούς (splits). Ουσιαστικά, κάθε επίπεδο ενός δενδρογράμματος ορίζει ένα βήμα του αλγορίθμου. Το βασικό πλεονέκτημα των ιεραρχικών αλγορίθμων είναι ότι δεν χρειάζεται να υποθέσουμε ένα συγκεκριμένο αριθμό συστάδων, αφού οποιοσδήποτε αριθμός μπορεί να επιτευχθεί, απλά κόβοντας το δενδρόγραμμα στο κατάλληλο επίπεδο. Ένα παράδειγμα ιεραρχικής συσταδοποίησης με δενδόγραμμα παρουσιάζεται στο Σχήμα 2.15.



Σχήμα 2.15: Δενδόγραμμα Ιεραρχικής Συσταδοποίησης

Συσσωρευτικοί Αλγόριθμοι (Agglomerative Algorithms)

Οι συσσωρευτικοί αλγόριθμοι ξεκινάνε με κάθε ένα από τα n δείγματα να ανήκει σε μια ξεχωριστή συστάδα, δηλαδή ξεκινάνε με n συστάδες. Σε κάθε βήμα, συγχωνεύονται οι δύο πιο κοντινές συστάδες, δηλαδή το πλήθος των συστάδων μειώνεται κατά ένα. Αυτή η διαδικασία επαναλαμβάνεται, μέχρις ότου ο αλγόριθμος καταλήξει σε μια μοναδική συστάδα, η οποία θα εμπεριέχει όλα τα n δείγματα. Η όλη διαδικασία του αλγορίθμου μπορεί να αναπαρασταθεί με δενδρόγραμμα ανομοιότητας. Το δενδρόγραμμα περιέχει n-1 επίπεδα και το κάθε επίπεδο αντιστοιχεί σε ένα βήμα του αλγορίθμου

Διαιρετικοί Αλγόριθμοι (Divisive Algorithms)

Οι διαιρετικοί αλγόριθμοι ξεκινάνε με όλα τα δείγματα να ανήκουν σε μια ενιαία συστάδα. Σε κάθε βήμα, μια ομάδα διασπάται σε δύο. Αυτό γίνεται επαναληπτικά, μέχρι να καταλήξουμε σε n ομάδες. Η πολυπλοκότητα των διαιρετικών αλγορίθμων είναι μεγαλύτερη από αυτή των συσσωρευτικών, αφού η διάσπαση μιας ομάδας σε δύο μπορεί να γίνει κατά 2n-1-1 τρόπους. Η επιλογή της βέλτιστης διάσπασης πρακτικά είναι αδύνατη ακόμα και για μικρό n. Στην πράξη η διάσπαση γίνεται, αλλά όχι κατά τον βέλτιστο τρόπο. Η όλη διαδικασία του αλγορίθμου μπορεί να αναπαρασταθεί, όπως και στους συσσωρευτικούς, με δενδρόγραμμα.

Μετρική Wasserstein

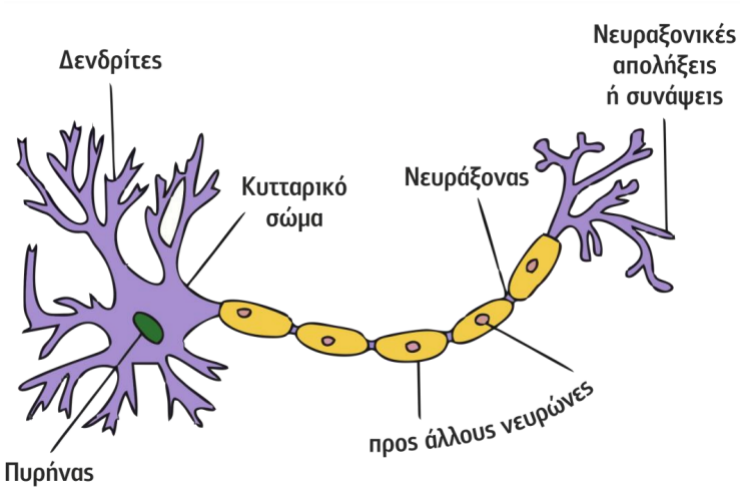
Στην παρούσα διπλωματική εργασία χρησιμοποιείται συσσωρευτική ιεραρχική συσταδοποίηση ορίζοντας ως απόσταση την μετρική Wasserstein. Στα μαθηματικά, η απόσταση Wasserstein ή η μέτρηση Kantorovich–Rubinstein είναι μια συνάρτηση απόστασης που ορίζεται μεταξύ των κατανομών πιθανοτήτων σε έναν δεδομένο μετρικό χώρο M. Πήρε το όνομά του από τον Leonid Vaserštein. Η μετρική αυτή είναι γνωστή επίσης στην επιστήμη των υπολογιστών ως η απόσταση του μετακινούμενου της γης earth mover's distance. Η μέτρηση ορίστηκε για πρώτη φορά από τον Leonid Kantorovich στο The Mathematical Method of Production Planning and Organization [57] (1939) στο πλαίσιο του βέλτιστου σχεδιασμού μεταφοράς αγαθών και υλικών. Η απόσταση p^{th} Wasserstein μεταξύ δύο μέτρων πιθανότητας μ και ν στο $P_p(M)$ ορίζεται ως:

$$W_p(\mu,\nu) := \left(\inf_{\gamma\in\Gamma(\mu,\nu)}\int_{M\times M} d(x,y)^p \,\mathrm{d}\gamma(x,y)\right)^{1/p} \tag{2.28}$$

,όπου το $\Gamma(\mu,\nu)$ υποδηλώνει τη συλλογή όλων των μέτρων στο $M\times M$ με περιθώρια μ και ν στον πρώτο και δεύτερο παράγοντα αντίστοιχα.

2.7 Νευρωνικά Δίκτυα

Ο όρος Νευρωνικά Δίκτυα (Neural Networks, Connectionist Networks, Parallel Distributed Processing Models) περιγράφει έναν αριθμό από διαφορετικά μαθηματικά μοντέλα, εμπνευσμένα από αντίστοιχα βιολογικά μοντέλα, δηλαδή μοντέλα που προσπαθούν να μιμηθούν τη συμπεριφορά των νευρώνων του ανθρώπινου εγκεφάλου. Ήδη από τον 19ο αιώνα οι επιστήμονες παραδέχονται ότι ο εγκέφαλος αποτελείται από διακριτά στοιχεία, τους νευρώνες (neurons), που επικοινωνούν το ένα με το άλλο.



Σχήμα 2.16: Σχηματικό διάγραμμα ενός τυπικού νευρώνα

2.7.1 Φυσικός Ανθρώπινος Νευρώνας

Οι νευρώνες συνιστούν το βασικό δομικό κομμάτι του ανθρώπινου εγκεφάλου. Υπολογίζεται ότι ο εγκέφαλος περιέχει 10 δισ. περίπου νευρώνες τοποθετημένους σε ομάδες, καθεμία από τις οποίες συνιστά ένα φυσικό νευρωνικό δίκτυο. Έτσι, ο ανθρώπινος εγκέφαλος περιέχει εκατοντάδες φυσικά νευρωνικά δίκτυα, καθένα από τα οποία περιέχει χιλιάδες διασυνδεδεμένους νευρώνες με μέσο αριθμό διασυνδέσεων ανά νευρώνα 1000 με 10.000. Ένας νευρώνας διαχωρίζεται από τα υπόλοιπα κύτταρα με μια μεμβράνη και έχει την ικανότητα να μεταφέρει ηλεκτρικά σήματα προς τους υπόλοιπους νευρώνες με τους οποίους επικοινωνεί. Όπως παρατηρείται και στο Σχήμα 2.16 αποτελείται από τα παρακάτω τρία κύρια τμήματα:

- Τους δενδρίτες (dendrites), οι οποίοι λειτουργούν ως κανάλια εισόδου για το νευρώνα.
- Το κυρίως κυτταρικό σώμα (cell body)
- Τον άξονα του κυττάρου-νευροάξονα (axon), που συνδέει ένα νευρώνα με άλλους νευρώνες

Ο άξονας του νευρώνα μεταφέρει σήματα στους δενδρίτες γειτονικών νευρώνων μέσω του σημείου ένωσης που ονομάζεται νευροαξονική απόληξη ή σύναψη (synapse). Ένας νευρώνας μπορεί να λάβει σήματα από ένα σύνολο γειτονικών νευρώνων μέσω των δενδριτών, να τα επεξεργαστεί και να τροφοδοτήσει την έξοδό του μέσω του άξονα προς ένα άλλο σύνολο γειτονικών νευρώνων. Τα σήματα που έρχονται μέσω των δενδριτών «ζυγίζονται» και τα αποτελέσματα αθροίζονται. Όταν το άθροισμα ξεπεράσει το οριακό επίπεδο (τιμή κατωφλίου), ο νευρώνας δημιουργεί μια έξοδο (με τη μορφή νευρικής ώσης ή ηλεκτρικού σήματος) στον άξονά του, η οποία εν συνεχεία μέσω των συνάψεων θα μεταφερθεί στους γειτονικούς νευρώνες.

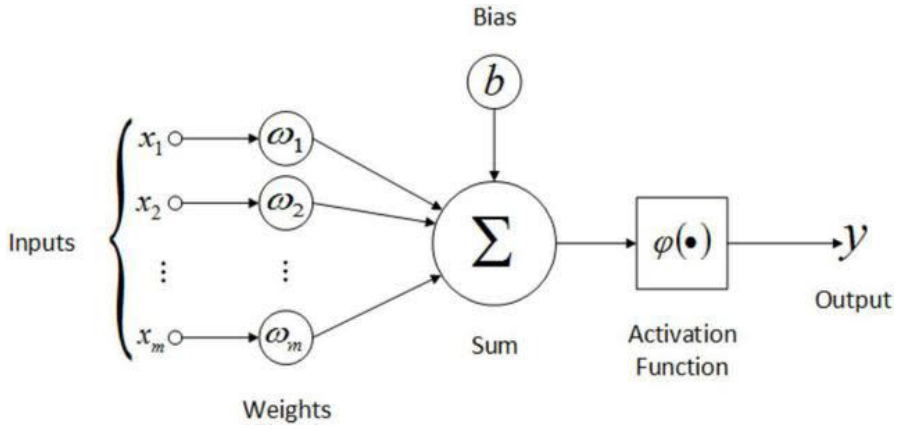
2.7.2 Τεχνητός Νευρώνας - Perceptron

Τα μαθηματικά μοντέλα των τεχνητών νευρωνικών δικτύων, σε πλήρη αντιστοιχία με τα βιολογικά, αποτελούνται από έναν αριθμό απλών και με υψηλό βαθμό εσωτερικής διασύνδεσης επεξεργαστικών μονάδων, οργανωμένων σε στρώματα. Τα Τεχνητά Νευρωνικά Δίκτυα - ΤΝΔ (Artificial Neural Networks) επεξεργάζονται πληροφορίες ανταποκρινόμενα δυναμικά σε εξωτερικά ερεθίσματα (εισόδους). Κάθε τεχνητός νευρώνας αποτελείται από πολλές εισόδους x_i και μία μόνο έξοδο y. Κάθε είσοδος x_i «ζυγίζεται» με ένα βάρος w_i και τα αποτελέσματα αθροίζονται μαζί με το bias b μέσω της συνάρτησης αθροίσματος F:

$$F = \sum_{i}^{m} x_i w_i + b \tag{2.29}$$

Η συνάρτηση ενεργοποίησης $\phi()$ περιορίζει το πλάτος του σήματος εξόδου ενός νευρώνα σε κάποια πεπερασμένη τιμή. Δηλαδή ο τεχνητός νευρώνας δίνει έξοδο μέσω της συνάρτησης αυτής, μόνο όταν το ζυγισμένο άθροισμα των εισόδων είναι μεγαλύτερο μιας ορισμένης τιμής κατωφλίου. Επομένως, η έξοδος του νευρώνα είναι:

$$y = \phi(F) = \phi(\sum_{i}^{m} x_i w_i + b) \tag{2.30}$$

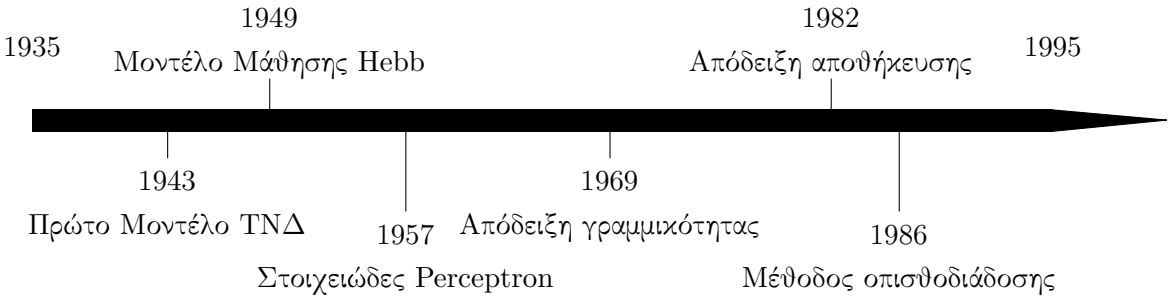


Σχήμα 2.17: Δομή Τεχνητού Νευρώνα - Perceptron

Ένας τεχνητός νευρώνας αποτελεί απλοποιημένο μοντέλο του φυσικού νευρώνα κατά το ότι τα βάρη διασύνδεσης σχηματίζουν τα ηλεκτρικά χαρακτηριστικά της επαφής της σύναψης και η τιμή κατωφλίου προσομοιώνει τη συμπεριφορά κορεσμού του φυσικού νευρώνα. Ένα από τα απλούστερα τεχνητά νευρωνικά δίκτυα που προσομοιώνουν τον φυσικό νευρώνα είναι ο στοιχειώδης **Perceptron**, δηλαδή ένα ΤΝΔ που αποτελείται από έναν μόνο νευρώνα.

Τα Τεχνητά Νευρωνικά Δίκτυα συνήθως οργανώνονται σε επίπεδα (layers) τα οποία καλούνται και στρώματα. Τα ενδιάμεσα επίπεδα καλούνται κρυμμένα επίπεδα (hidden layers) και δεν είναι απαραίτητο να υπάρχουν. Τα επίπεδα αποτελούνται από έναν αριθμό μονάδων (units) ή κόμβων (nodes) που είναι έτσι συνδεδεμένες μεταξύ τους, ώστε μία μονάδα να έχει συνδέσμους με πολλές άλλες μονάδες του ίδιου ή άλλου επιπέδου. Οι μονάδες επιδρούν σε άλλες μονάδες με το να τις διεγείρουν ή να αναστέλλουν την ενεργοποίησή τους. Για να επιτευχθεί αυτό η μονάδα λαμβάνει το σταθμισμένο άθροισμα όλων των εισόδων μέσω των συνδέσμων που καταλήγουν σε αυτήν και παράγει μέσω της συνάρτησης μετάβασης μία μοναδική έξοδο, εάν το άθροισμα υπερβαίνει μία τιμή κατωφλίου. Οι είσοδοι παρουσιάζονται στο δίκτυο μέσω του επιπέδου εισόδου (input layer) το οποίο επικοινωνεί με έναν ή περισσότερα κρυμμένα επίπεδα. Τα κρυμμένα επίπεδα συνδέονται με το επίπεδο εξόδου (output layer) από το οποίο εξάγεται η απάντηση.

2.7.3 Μικρή Ιστορική Αναδρομή



Το 1943 οι McCulloch και Pitts δημιουργούν το πρώτο μοντέλο ΤΝΔ, ενώ ο Hebb το 1949 δημιουργεί το μοντέλο μάθησης που πήρε το όνομά του στο οποίο κάθε φορά που ενεργοποιείται μια σύναψη αυτή ενισχύεται, με αποτέλεσμα το δίκτυο να μαθαίνει «λίγο περισσότερο» το πρότυπο που του παρουσιάζεται εκείνη τη στιγμή. Το 1957 ο Rosenblatt προτείνει το στοιχειώδες ΤΝΔ του απλού αισθητήρα που ονόμασε Perceptron, ενώ το 1969 οι Minsky και Papert αποδεικνύουν μαθηματικά ότι τα ΤΝΔ ενός επιπέδου δεν μπορούν να λύσουν μη γραμμικά προβλήματα. Το 1982 έρχεται η μαθηματική απόδειξη ότι ένα ΤΝΔ πολλών επιπέδων μπορεί να αποθηκεύσει οποιαδήποτε πληροφορία και το 1986 οι Werbos και Rumelhart προτείνουν τη μέθοδο οπισθοδιάδοσης (backpropagation) για την εκπαίδευση ΤΝΔ.

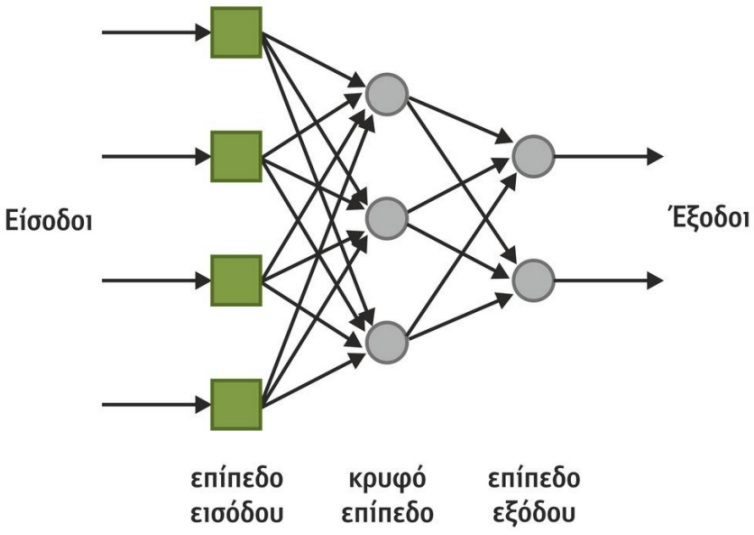
2.7.4 Αρχιτεκτονικές ΤΝΔ

Πολυεπίπεδα ΤΝΔ

Το κοινό χαρακτηριστικό της δομής των πολυεπίπεδων τεχνητών νευρωνικών δικτύων είναι ότι διαθέτουν τουλάχιστον ένα κρυφό επίπεδο. Οι κόμβοι των διάφορων επιπέδων μπορεί να είναι πλήρως συνδεδεμένοι (fully connected), δηλαδή κάθε κόμβος του ενός επιπέδου συνδέεται με όλους τους κόμβους του επόμενου, όπως στο σχήμα 4.19, ή μερικώς συνδεδεμένοι (partially connected).

Ανάλογα με το πώς είναι συνδεδεμένες οι μονάδες μεταξύ τους σε ένα τεχνητό νευρωνικό δίκτυο, υπάρχουν δυο βασικές κατηγορίες:

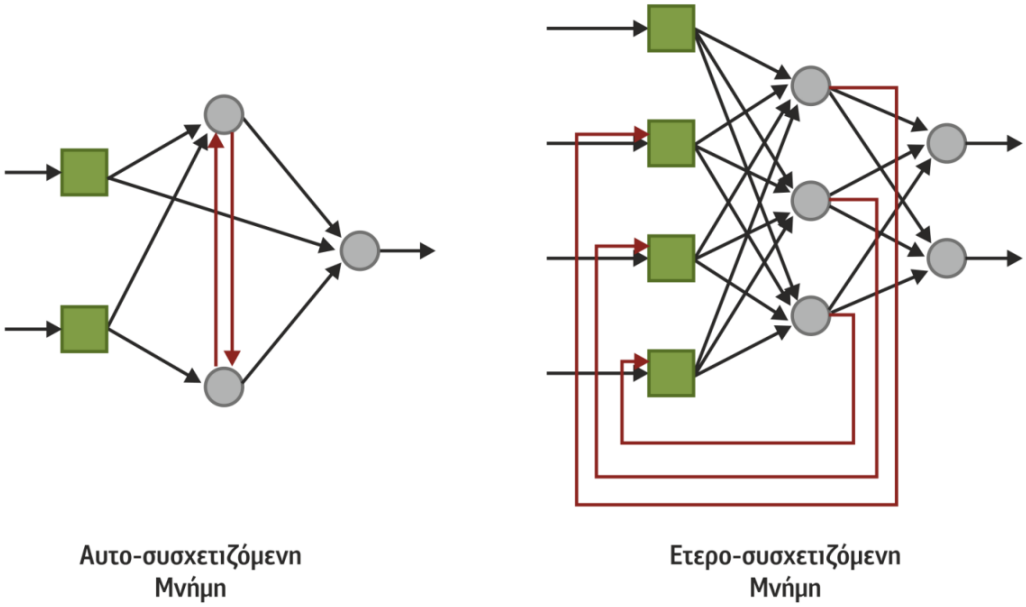
- Πρόσθιας Τροφοδότησης (Feed Forward)
- Οπίσθιας Τροφοδότησης (Feed Backward)



Σχήμα 2.18: Δίκτυο Πρόσθιας Τροφοδότησης

Στα δίκτυα πρόσθιας τροφοδότησης, οι μονάδες είναι οργανωμένες σε διαφορετικά επίπεδα, ώστε οι μονάδες του ενός επιπέδου να τροφοδοτούν τις μονάδες του επόμενου επιπέδου, έως

ότου τροφοδοτηθούν και οι μονάδες του τελευταίου επιπέδου, όπως φαίνεται και στο Σχήμα 2.18. Στα οπισθίως τροφοδοτούμενα δίκτυα του Σχήματος 2.19, που καλούνται και ανατροφοδοτούμενα (recurrent), επιτρέπεται στις μονάδες ενός επιπέδου να τροφοδοτούν και μονάδες του ιδίου επιπέδου ή και προηγούμενων επιπέδων. Αν η ανατροφοδότηση αφορά κόμβους στο ίδιο επίπεδο, τότε τα δίκτυα καλούνται αυτοσυσχετιζόμενες μνήμες (autoassociated memories) διαφορετικά, καλούνται ετεροσυσχετιζόμενες μνήμες (heteroassociated memories).



Σχήμα 2.19: Ανατροφοδοτούμενα Δίκτυα

2.7.5 Συναρτήσεις Ενεργοποίησης

Όσον αφορά τις συναρτήσεις ενεργοποίησεις (activation functions) οι πιο απλές είναι οι γραμμικές, όπως οι βηματικές συναρτήσεις ή συναρτήσεις κατωφλίου (threshold functions), οι συναρτήσεις προσήμου (sign functions), οι συναρτήσεις βηματικής μεταβολής (hard limiter functions), οι συναρτήσεις αναρρίχησης (ramping functions) κτλ. Στην περίπτωση αυτή, το νευρωνικό δίκτυο είναι πιο εύκολο να εκπαιδευτεί, ωστόσο είναι δύσκολη η εκμάθηση περίπλοκων σχέσεων και απεικονίσεων μεταξύ εισόδου - εξόδου. Για το λόγο αυτό, στους κόμβους των κρυφών επιπέδων χρησιμοποιούνται μη γραμμικές συναρτήσεις, όπως η ReLU, οι σιγμοειδείς συναρτήσεις (sigmoid functions) και οι Γκαουσιανές συναρτήσεις (Gaussian functions). Χαρακτηριστικά παραδείγματα συναρτήσεων ενεργοποίησης παρουσιάζονται στο Σχήμα 2.20.

Ορισμένες από τις συναρτήσεις που χρησιμοποιήθηκαν και στην συγκεκριμένη διπλωματική εργασία είναι:

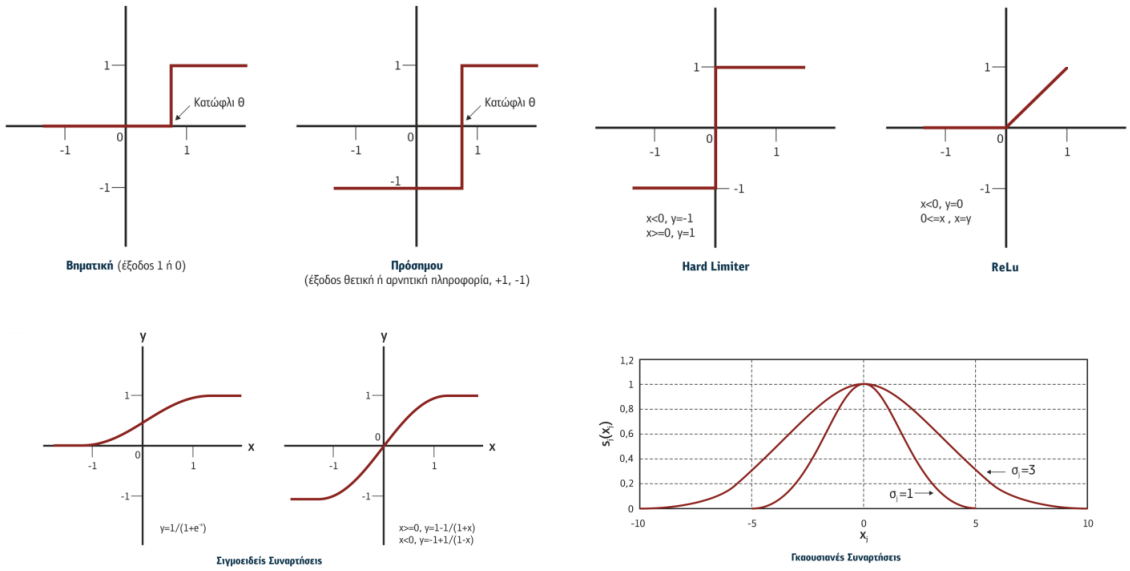
- ReLU: Η συνάρτηση ReLU μηδενίζει τις αρνητικές εισόδους και εφαρμόζει ένα γραμμικό μετασχηματισμό στις θετικές. Το πεδίο τιμών της είναι το διάστημα $[0, \infty)$. Πρόκει-

ται για μια ευρύτατα διαδεδομένη συνάρτηση ενεργοποίησης που δεν αντιμετωπίζει το πρόβλημα της εξαφάνισης κλίσης (gradient vanishing).

- Tanh-sigmoid: Η tanh είναι μια μη γραμμική συνάρτηση που αντιστοιχίζει τις τιμές εισόδου στο σύνολο $(-1, 1)$, ενώ η σιγμοειδής συνάρτηση, που ονομάζεται και logistic, μετατρέπει την είσοδο στο διάστημα $(0, 1)$. Και οι δύο συναρτήσεις, οι οποίες έχουν παρόμοιο σχήμα, οδηγούν σε κορεσμό. Αυτό πρακτικά σημαίνει ότι οι τιμές που είναι πολύ μεγάλες αντιπροσωπεύονται απλά από την μονάδα, ενώ οι πολύ μικρές τιμές αντιπροσωπεύονται από το -1 για tanh (αντίστοιχα το 0 για sigmoid). Για το λόγο αυτό, οι συναρτήσεις αυτές είναι ευαίσθητες ως προς αλλαγές τιμών γύρω από τα κέντρα τους (0 για tanh και 0.5 για sigmoid). Αυτές οι συναρτήσεις χρησιμοποιούνται συχνά στους κόμβους των κρυφών επιπέδων.

- softmax: Κάποιες συναρτήσεις ενεργοποίησης όπως και η softmax, χρησιμοποιούνται συχνότερα στο επίπεδο εξόδου (output layer). Οι συναρτήσεις αυτές, δεν δέχονται σαν είσοδο μια τιμή x, αλλά ένα διάνυσμα τιμών που προέκυψαν απο το προηγούμενο επίπεδο. Για την softmax, με βάση το διάνυσμα εισόδου $x = [x_1, x_2, x_3, ..., x_K]$ υπολογίζεται για κάθε $0 \leq i \leq K$ η τιμή:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.31}$$



Σχήμα 2.20: Συναρτήσεις Ενεργοποίησης

Επομένως, στην περίπτωση που το πρόβλημα προς επίλυσης είναι δυαδικό, χρησιμοποιείται ένας κόμβους εξόδου με συνάρτηση ενεργοποίησης σιγμοειδή (αντί tanh). Έτσι αν η τιμή που προκύπτει είναι μεγαλύτερη από ένα κατώφλι (συνήθως 0.5 για tanh και αντίστοιχα 0 για σιγμοειδή), τότε ανήκει στην κατηγορία A, ενώ σε αντίθετη περίπτωση στην κατηγορία B. Στην περίπτωση που το πρόβλημα προς επίλυση είναι ταξινόμηση πολλών κλάσεων, χρησιμοποιούνται τόσοι κόμβοι εξόδου όσες και οι κλάσεις. Χρησιμοποιώντας συνάρτηση ενεργοποίησης

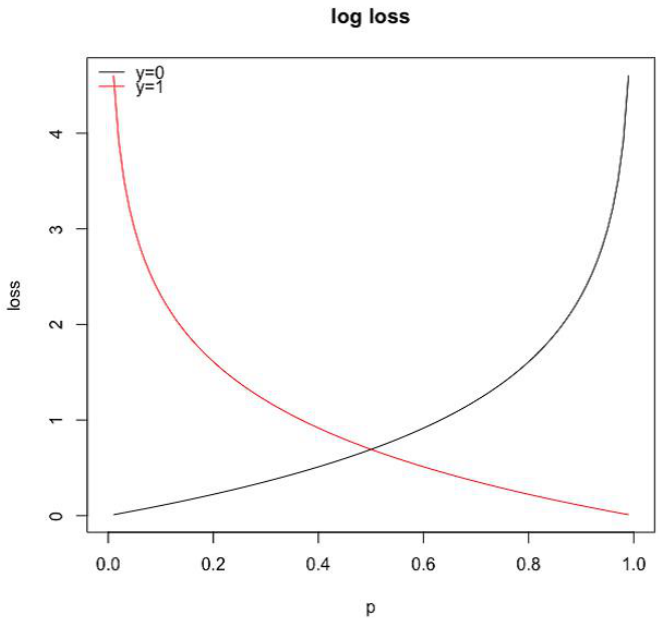
softmax, δίνεται για κάθε κατηγορία (κόμβο) ένας αριθμός που δηλώνει την πιθανότητα να ανήκει το δείγμα στην κλάση αυτή. Βέβαια μπορεί να χρησιμοποιηθεί σε κάθε κόμβο και η tanh ή σιγμοειδή και το δείγμα να αποδοθεί στον κόμβο-κλάση με την μεγαλύτερη τιμή εξόδου.

2.7.6 Συναρτήσεις Κόστους

Οι συναρτήσεις κόστους, αποτελούν μια μέθοδο αποτίμησης της απόδοσης ενός νευρωνικού δικτύου, κατά την διάρκεια της εκπαίδευσης. Συγκεκριμένα, για κάθε δείγμα εκπαίδευσης, συγκρίνεται η έξοδος του δικτύου (prediction) με την αναμενόμενη τιμή (label). Τις περισσότερες φορές, καθώς και στη συγκεκριμένη εργασία, χρησιμοποείται το κόστος διασταυρούμενης εντροπίας (Cross Entropy Loss), το οποίο αρχικά εξηγείται για ένα πρόβλημα δυαδικής ταξινόμησης.

Cross Entropy Loss

$$L = -(y \log p + (1 - y) \log(1 - p)) \tag{2.32}$$



Σχήμα 2.21: Συνάρτηση κόστους log loss για πρόβλημα δυαδικής ταξινόμησης

- Έστω ότι ένα δείγμα που ανήκει στην κατηγορία A, αποδίδεται από το νευρωνικό δίκτυο ότι ανήκει στην κατηγόρία A, με πιθανότητα p. Στην περίπτωση αυτή, η πρόβλεψη είναι σωστή ($y = 1$) και ο τύπος μετασχηματίζεται σε $-\log p$ που απεικονίζεται με κόκκινο χρώμα στο Σχήμα 2.21. Επιθυμητό είναι να ληφθεί $L = 0$, δηλαδή οι σωστές προβλέψεις να δίνονται με μεγάλη πιθανότητα. Με τον τρόπο αυτό, «τιμωρούνται» προβλέψεις που ήταν σωστές αλλά δόθηκαν με μικρή πιθανότητα.

- Έστω ότι ένα δείγμα που ανήκει στην κατηγορία A, αποδίδεται από το νευρωνικό δίκτυο ότι ανήκει στην κατηγορία B, με πιθανότητα p. Στην περίπτωση αυτή, η πρόβλεψη είναι λάθος ($y = 0$) και ο τύπος μετασχηματίζεται σε $-log(1-p)$ που απεικονίζεται με μαύρο χρώμα στο Σχήμα 2.21. Επιθυμητό είναι να ληφθεί $L = 0$, δηλαδή οι λάθος προβλέψεις να δίνονται με μικρή πιθανότητα. Με τον τρόπο αυτό, «τιμωρούνται» προβλέψεις που ήταν λάθος, αλλά δόθηκαν με μεγάλη πιθανότητα.

Για προβλήματα ταξινόμησης περισσότερων των δύο κλάσεων (έστω M), τα παραπάνω μπορούν να γενικευτούν ως εξής:

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_o, c) \tag{2.33}$$

2.7.7 Εκπαίδευση ΤΝΔ

Η εκπαίδευση των τεχνητών νευρωνικών δικτύων έχει ως βασικό στόχο την εύρεση ενός τρόπου αλλαγής των συνδεσμικών βαρών w που θα έχει ως αποτέλεσμα την αλλαγή της γενικής συμπεριφοράς του δικτύου με την αύξηση της ικανότητας του να παρέχει στο μέλλον μία επιθυμητή έξοδο μετά από μία δεδομένη είσοδο. Όταν η επιθυμητή έξοδος είναι εκ των προτέρων γνωστή λέμε ότι το δίκτυο μαθαίνει με επίβλεψη (supervised learning), αλλιώς μαθαίνει χωρίς επίβλεψη (unsupervised learning).

Βασικό στοιχείο της αρχιτεκτονικής ενός ΤΝΔ είναι ο τρόπος ελέγχου της αλλαγής των βαρών κατά την εκπαίδευση, δηλαδή ο αλγόριθμος εκπαίδευσης (training algorithm) που υλοποιείται αποκλειστικά από το ίδιο το δίκτυο χωρίς εξωτερική επέμβαση. Γνωστότεροι αλγόριθμοι εκπαίδευσης είναι οι ακόλουθοι:

- Αλγόριθμος οπισθοδιάδοσης λάθους (backpropagation)
- Ανταγωνιστική μάθηση (competitive learning)
- Τυχαία μάθηση (random learning)

Σε πρώτη φάση αρχικοποιούνται τυχαία τα βάρη του μοντέλου. Στην συνέχεια, με βαση την συνάρτηση κόστους, τροποποιούνται κατάλληλα, ώστε να ελαχιστοποιηθεί η συνάρτηση αυτή. Γίνεται λοιπόν αντιληπτό ότι η τιμή της συνάρτηση κόστους για μια δεδομένη παρατήρηση, εξαρτάται απο τις τιμές των βαρών w. Η γνώσης της παραγώγου της συνάρτησης κόστους (μιας συνάρτησης πολλών μεταβλητών), μπορεί να δώσει χρήσιμη πληροφορία για την «κατεύθυνση», όπου ελαχιστοποιείται η συνάρτηση. Συγκεκριμένα, σε συναρτήσεις πολλών μεταβλητών, η παράγωγος της συνάρτησης (grad) δίνει την κατεύθυνση κατά την οποία η συνάρτηση αυξάνεται πιο απότομα. Απο τη στιγμή που υπολογίζεται το gradient, χρησιμοποιείται για να ανανεωθούν οι τιμές των βαρών. Η διαδικασία αυτή (ανανέωση βαρών) μπορεί να γίνει, είτε μεμονωμένα για κάθε δείγμα, είτε σε ομάδες (batches) δειγμάτων, είτε για όλο το σύνολο εκπαίδευσης. Έτσι διακρίνονται οι παρακάτω τρεις κατηγορίες εκπαίδευσης:

- Stochastic Gradient Descent: Στην περίπτωση αυτή, για κάθε νέο δείγμα εκπαίδευσης, υπολογίζεται το gradient της συνάρτησης και πραγματοποιείται ανανέωση των βαρών.

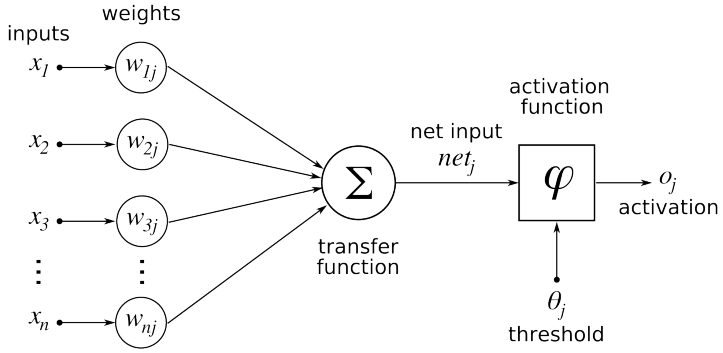
- Mini Batch Gradient Descent: Στην περίπτωση αυτή, τα δείγματα χωρίζονται σε ομάδες δεδομένων και όταν μια ομάδα τροφοδοτηθεί στο νευρωνικό, τότε λαμβάνεται ο μέσος όρος των gradients και ανανεώνονται τα βάρη.

- Gradient Descent: Στην περίπτωση αυτή, λαμβάνεται ο μέσος όρος των gradients για κάθε κατηγορία ξεχωριστά, αφού έχουν τροφοδοτηθεί όλα τα δείγματα εκπαίδευσης στο νευρωνικό δίκτυο (μια εποχή).

Ωστόσο, αυτό που δεν αναφέρθηκε είναι ο τρόπος με τον οποίο υπολογίζεται το gradient της συνάρτησης κόστους για ένα βάρος. Ο συνηθέστερος αλγόριθμος που χρησιμοποιείται στην πράξη είναι με οπισθοδιάδοση (backpropagation), που σχεδιαστηκε για τον υπολογισμό παραγώγων σε δομές γράφων, ξεκινώντας απο την έξοδο προς την είσοδο. Η διαδικασία αυτή γίνεται εφικτή χρησιμοποιώντας τον κανόνα της αλυσίδας (Chain Rule).

Αλγόριθμος Backpropagation

Ο αλγόριθμος οπισθοδιάδοσης backpropagation υπολογίζει το gradient της συνάρτησης κόστους σε σχέση με κάθε βάρος από τον κανόνα της αλυσίδας, υπολογίζοντας την κλίση σε ένα επίπεδο τη φορά. Επαναλαμβάνεται προς τα πίσω από το τελευταίο επίπεδο προς το πρώτο για να αποφευχθούν περιττοί υπολογισμοί ενδιάμεσων όρων στον κανόνα της αλυσίδας. Αποτελεί δηλαδή ένα χαρακτηριστικό παράδειγμα δυναμικού προγραμματισμού [58].



Σχήμα 2.22: Διάγραμμα τεχνητού νευρωνικού δικτύου

Η τιμή του βάρους w_{ij} ανανεώνεται σύμφωνα με τη σχέση $w_{ij} = w_{ij} - \frac{\partial L}{\partial w_{ij}}$. Η μερική παράγωγος $\frac{\partial L}{\partial w_{ij}}$ υπολογίζεται με βάση τον κανόνα της αλυσίδας:

$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} = \frac{\partial L}{\partial o_j}\frac{\partial o_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} \tag{2.34}$$

Στον τελευταίο παράγοντα το άθροισμα net_j όταν παραγωγιστεί μερικώς ως προς w_{ij} θα δώσει:

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left(\sum_{k=1}^{n} w_{kj}x_k\right) = \frac{\partial}{\partial w_{ij}} w_{ij}x_i = x_i \tag{2.35}$$

Η έξοδος o_i προκύπτει απο την ενεργοποίηση του αθροίσματος net_j με βάση κάποια συνάρτηση ϕ. ($o_i = \phi(net_j)$):

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial \varphi(net_j)}{\partial net_j} \tag{2.36}$$

Αντικαθιστώντας τις εξισώσεις 2.35, 2.36 στην εξίσωση 2.34, αποτιμάται η τιμή της μερικής παραγώγου $\frac{\partial L}{\partial w_{ij}}$.

2.7.8 Αλγόριθμοι Βελτιστοποίησης

Ο όρος βελτιστοποίησης αναφέρεται στην εύρεση των τιμών μιας συνάρτησης που την μεγιστοποιούν ή αντίστοιχα την ελαχιστοποιούν. Η εκπαίδευση ενός νευρωνικού είναι στην ουσία ένα πρόβλημα ελαχιστοποίησης της συνάρτησης κόστους. Το gradient που υπολογίστηκε με τον αλγόριθμο backpropagation, μπορεί να χρησιμοποιηθεί από τους ακόλουθους αλγορίθμους βελτιστοποίησης, κάποιοι από τους οποίους αναγράφονται ακολούθως:

- SGD: Η νέα τιμή του βάρους διαφέρει απο την προηγούμενη κατά ένα παράγοντα $\eta \frac{\partial L}{\partial w_i}$. Με η συμβολίζουμε το ρυθμό εκμάθησης (learning rate).

- Adam: Το όνομα του, προέρχεται απο τη φράση «adaptive moment estimation». Πρόκειται για μια επέκταση-τροποποίηση του αλγορίθμου SGD. Στον αλγόριθμο SGD, για όλες τις ανανεώσεις βαρών, χρησιμοποιείται το ίδιο learning rate, και μάλιστα δεν τροποποιείται κατά την διάρκεια της εκπαίδευσης. Αντιθέτως με τη χρήση του αλγορίθμου Adam, κάθε βάρος ανανεώνεται με διαφορετικό learning rate και μάλιστα αυτό μπορεί να αλλάξει κατά τη διάρκεια της εκπαίδευσης, δηλαδή για κάθε νέο δείγμα εκπαίδευσης.

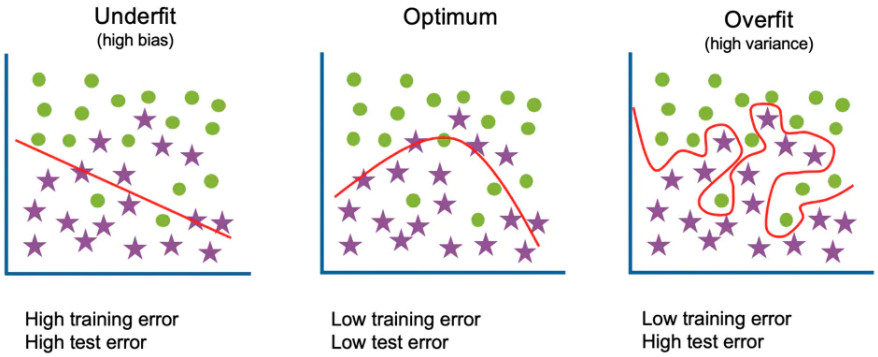
- Adamax: Πρόκειται για μια επέκταση του αλγορίθμου Adam. Η ανανέωση των βαρών πραγματοποιείται με αντίστοιχο τρόπο με τον αλγόριθμο Adam. Όταν η τιμή του gradient είναι πολύ μικρή, (κοντά στο μηδέν), αγνοείται και στη θέση της λαμβάνεται μια άλλη ποσότητα. Στην περίπτωση αυτή τα βάρη επηρεάζονται λιγότερες φορές από gradients. Κάτι τέτοιο είναι αναγκαίο ορισμένες φορές που τα gradients αναπαριστούν θόρυβο, και όχι κάποια σημαντική πληροφορία για την κατεύθυνση του ελαχίστου κάποιας συνάρτησης κόστους.

2.7.9 Προβλήματα Εκπαίδευσης και Τρόποι Αντιμετώπισης

Vanishing Gradients

Κατά την διάρκεια του backpropagation, η τιμή του κάθε βάρους προκύπτει αφαιρώντας από την τρέχουσα τιμή την μερική παράγωγο της συνάρτησης κόστους αναφορικά με το συγκεκριμένο βάρος. Η τιμή τελευταία τιμή υπολογίζεται με τον κανόνα της αλυσίδας. Εξαιτίας των διαδοχικών πολλαπλασιασμών, είναι πιθανό η τιμή που θα προκύψει να είναι αρκετά κοντά στο μηδέν. Αυτό πρακτικά σημαίνει, ότι η νέα τιμή βάρους θα είναι ίση με την προηγούμενη. Γενικεύοντας αυτό το πρόβλημα και για τα υπόλοιπα βάρη του δικτύου, το νευρωνικό δίκτυο πλέον δεν εκπαιδεύεται.

Υποπροσαρμογή - Υπερπροσαρμογή



Σχήμα 2.23: Underfit - Optimum - Overfit

Στόχος των αλγορίθμων βελτιστοποίησης, μεταξύ άλλων, είναι να επιλύσουν το ζήτημα της υποπροσαρμογής (underfitting). Για παράδειγμα σε ένα πρόβλημα δυαδικής ταξινόμησης που φαίνεται στο Σχήμα 2.23, το πρόβλημα (underfitting) συναντάται όταν η διαχωριστική καμπύλη δεν διαμερίζει πλήρως τα πρότυπα του συνόλου εκπαίδευσης. Η κατάσταση αυτή χαρακτηρίζεται απο υψηλό bias, ενώ στην αντίθετη περίπτωση, όπου τα πρότυπα εκπαίδευσης διαμερίζονται πλήρως, είμαστε σε μια κατάσταση low βιας. Επιλέγοντας ωστόσο τη δεύτερη περίπτωση (με low bias), θα παρατηρήσουμε ότι στο σύνολο ελέγχου (test set), τα πρότυπα δεν διαχωρίζονται επιτυχώς. Η διαφορά (variation) αυτή που παρατηρείται, από την επιλογή μιας καμπύλης, στο διαχωρισμό προτύπων που ανήκουν στο σύνολο εκπαίδευσης και προτύπων που ανήκουν στο σύνολο ελέγχου, καλείται variance. Υψηλό variance έχουμε στην εικόνα δεξιά, όπου η καμπύλη, μπορεί να προσαρμόζεται άψογα στο σύνολο εκπαίδευσης, ωστόσο είναι πιθανόν να μην πετύχει στο σύνολο ελέγχου.

Πρόωρη Διακοπή Εκπαίδευσης - Early Stopping

Η πρόωρη διακοπή της εκπαίδευσης είναι ένας από τους πιο συνηθισμένους τρόπους αντιμετώπισης του overfitting. Κατά την διάρκεια της εκπαίδευσης, χρησιμοποιείται ένα σύνολο δεδομένων (validation set) για την εκτίμηση της συμπεριφοράς του μοντέλου σε άγνωστα δεδομένα. Υπάρχει περίπτωση το μοντέλο να εκπαιδεύεται επιτυχώς στο σύνολο εκπαίδευση (η συνάρτηση κόστους να μειώνεται ανά εποχή), αλλά να μη συμβαίνει το ίδιο για το σύνολο επικύρωσης. Στη περίπτωση αυτή, το μοντέλο αδυνατεί να γενικεύσει και καλό είναι να διακόπτεται η διαδικασία της εκπαίδευσης, ώστε να μην παρατηρηθεί υπερπροσαρμογή στα δεδομένα εκπαίδευσης. Η διαδικασία διακοπής της εκπαίδευσης καλείται Early Stopping.

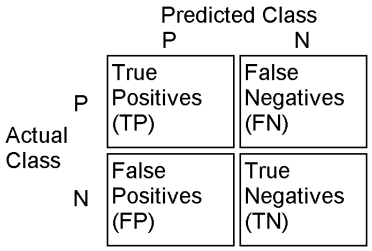
2.7.10 Μετρικές Απόδοσης

Προκειμένου να ποσοτικοποιηθεί το πόσο επιτυχής ήταν η εκπαίδευση ενός νευρωνικού δικτύου, χρησιμοποιούνται κάποιες μετρικές αξιολόγησης στο σύνολο ελέγχου (test set).

Πίνακας Σύγχυσης - Confusion Matrix

Ο Πίνακας Σύγχυσης είναι μια σύνοψη των αποτελεσμάτων των προβλέψεων σε συγκεκριμένη διάταξη πίνακα που επιτρέπει την οπτικοποίηση της μέτρησης απόδοσης του μοντέλου μηχανικής μάθησης. Πρόκειται για έναν τετραγωνικό πίνακα μεγέθους NxN, που δίνει χρήσιμη πληροφορία για τα λάθη που συμβαίνουν μεταξύ των N κλάσεων. Συγκεκριμένα, κάθε τιμή αυτού του πίνακα αναφέρεται στον αριθμό των προβλέψεων που αποδόθηκαν σε μια κλάση (Predicted Class), δεδομένου ότι τα δείγματα άνηκαν σε μια άλλη κλάση (Actual Class).

Έστω ένα δυαδικό πρόβλημα ταξινόμησης (Θετικό - Αρνητικό). Οι όροι που χρησιμοποιούνται για τον ορισμό ενός πίνακα σύγχυσης, όπως φαίνεται και στο Σχήμα 2.24, είναι οι TP, TN, FP και FN.



Σχήμα 2.24: Πίνακας Σύγχυσης Δυαδικής Ταξινόμησης

- True Positives (TP): Αυτές είναι οι σωστές προβλέψεις θετικών τιμών, που σημαίνει ότι η τιμή της πραγματικής κλάσης είναι ναι και η τιμή της προβλεπόμενης κλάσης είναι επίσης ναι.

- True Negatives (TN): Αυτές είναι οι σωστές προβλέψεις αρνητικών τιμών, που σημαίνει ότι η τιμή της πραγματικής κλάσης είναι όχι και η τιμή της προβλεπόμενης κλάσης είναι επίσης όχι.

- False Positives (FP): Αυτές είναι οι λάθος προβλέψεις θετικών τιμών, που σημαίνει ότι η τιμή της πραγματικής κλάσης είναι όχι και η τιμή της προβλεπόμενης κλάσης είναι ναι.

- False Negatives (FN): Αυτές είναι οι λάθος προβλέψεις αρνητικών τιμών, που σημαίνει ότι η τιμή της πραγματικής κλάσης είναι ναι και η τιμή της προβλεπόμενης κλάσης είναι όχι.

Ορθότητα - Accuracy

Η ορθότητα (accuracy) είναι μια απο τις βασικότερες μετρικές αξιολόγησης ενός μοντέλου. Ορίζεται ως το κλάσμα των ορθών προβλέψεων προς τον αριθμό των συνολικών εκτιμήσεων που πραγματοποιήθηκαν.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.37}$$

Η μετρική αυτή, καλό είναι να μην χρησιμοποιείται όταν το dataset δεν είναι ισορροπημένο (balanced), δηλαδή όπου οι τιμές των FP και των FN δεν είναι σχεδόν ίδιες. Επομένως, πρέπει να εξεταστούν και άλλες μετρικές για την σωστή αξιολόγηση της απόδοση ενός μοντέλου.

Ακρίβεια - Precision

Η ακρίβεια (precision) ορίζεται ως ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων (TP) προς τις συνολικές προβλεπόμενες θετικές παρατηρήσεις.

$$Precision = \frac{TP}{TP + FP} \tag{2.38}$$

Το ερώτημα που η συγκεκριμένη μετρική απαντάει είναι από όλες τις θετικές προβλέψεις που έγιναν, πόσες όντως πραγματικά ήταν θετικές.

Ανάκληση - Recall

Η ανάκληση (recall), ή αλλιώς και ευαισθησία - sensitivity, είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων (TP) προς όλες τις παρατηρήσεις στην πραγματική θετική τάξη.

$$Recall = \frac{TP}{TP + FN} \tag{2.39}$$

Το ερώτημα που η συγκεκριμένη μετρική απαντάει είναι από όλες τις πραγματικά θετικές παρατηρήσεις, πόσες τελικά προέβλεψε το μοντέλο.

F1 Score

Το F1 score είναι ο σταθμισμένος μέσος όρος ακρίβειας και ανάκλησης. Επομένως, αυτή η βαθμολογία λαμβάνει υπόψη τόσο τα ψευδώς θετικά όσο και τα ψευδώς αρνητικά. Διαισθητικά δεν είναι τόσο εύκολο να γίνει κατανοητό όσο η ακρίβεια, αλλά το Φ1 είναι συνήθως πιο χρήσιμο από την ακρίβεια, ειδικά εάν υπάρχει άνιση κατανομή κλάσης (imbalanced dataset). Η ακρίβεια λειτουργεί καλύτερα εάν τα ψευδώς θετικά και τα ψευδώς αρνητικά έχουν παρόμοια τιμή. Εάν η τιμή των ψευδώς θετικών και των ψευδών αρνητικών είναι πολύ διαφορετική, είναι καλύτερο να εξεταστεί τόσο η ακρίβεια όσο και η ανάκληση.

$$F1Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.40}$$

Ταξινόμηση πολλών κλάσεων (Multiclass Classification)

Οι παραπάνω ορισμοί των παραμέτρων και των μετρικών αφορούν δυαδικά προβλήματα. Στην περίπτωση που υπάρχουν περισσότερες από δύο κλάσεις εξόδου, τότε οι παραπάνω παράμετροι υπολογίζονται χωριστά για κάθε κλάση. Ουσιαστικά για κάθε κλάση αξιολογείται το πρόβλημα ως δυαδικό, όπου δηλαδή η πρώτη έξοδος είναι η ίδια η κλάση και η δεύτερη έξοδος όλες οι υπόλοιπες. Αξίζει να σημειωθεί ότι αντίθετα με την περίπτωση πραγματικά δυαδικού προβλήματος, εάν το σύστημα προβλέψει σωστά ότι ένα στιγμιότυπο δεν ανήκει στην κλάση που ελέγχουμε (περίπτωση TN), αυτό δεν σημαίνει απαραίτητα ότι το σύστημα

το πρόβλεψε και στην σωστή. Για αυτόν τον λόγο η παράμετρος TN και όσες μετρικές την χρησιμοποιούν (ορθότητα, ανάκληση) χάνουν την αξιοπιστία τους.

Έχοντας υπολογίσει τις παραπάνω μετρικές για κάθε κλάση, μπορούμε να πάρουμε τον μέσο όρο. Από τους πιο συνηθισμένους και αυτούς που χρησιμοποιούνται στην συγκεκριμένη εργασία είναι το **micro average** και το **macro average**.

Ο macro average θα υπολογίσει τη μετρική ανεξάρτητα για κάθε κλάση και στη συνέχεια θα λάβει το μέσο όρο αντιμετωπίζοντας εξίσου όλες τις κατηγορίες. Αντίθετα, ο micro average θα συγκεντρώσει τις συνεισφορές όλων των κλάσεων για τον υπολογισμό του μέσου όρου της μετρικής. Η διαφορά δηλαδή μεταξύ micro και macro είναι ότι ο macro ζυγίζει κάθε κλάση εξίσου, ενώ ο micro ζυγίζει κάθε δείγμα εξίσου. Ο micro average είναι συνήθως προτιμότερος εάν υπάρχει υποψία ότι μπορεί να υπάρχει ανισορροπία κλάσης. Εάν υπάρχει ίσος αριθμός δειγμάτων για κάθε κλάση, τότε θα βγάλουν το ίδιο αποτέλεσμα.

Κεφάλαιο 3

Τεχνικές Λεπτομέρειες - Εργαλεία

3.1 Εργαλείο nProbe

Οι μετρήσεις κίνησης είναι απαραίτητες για τη λειτουργία όλων των τύπων δικτύων IP. Ο διαχειριστής δικτύων χρειάζεται μια λεπτομερή προβολή της κίνησης του δικτύου για λόγους ασφάλειας, λογιστικών και διαχείρισης. Οι συνθέσεις της κίνησης πρέπει να αναλύονται με ακρίβεια κατά την εκτίμηση των μετρήσεων επισκεψιμότητας ή κατά την εύρεση προβλημάτων δικτύου. Όλες αυτές οι μετρήσεις πρέπει να γίνουν αναλύοντας όλα τα πακέτα που ρέουν στα κεντρικά σημεία του δικτύου (όπως δρομολογητές και μεταγωγείς). Η ανάλυση θα μπορούσε να γίνει είτε εν κινήσει, είτε με την καταγραφή όλων των πακέτων και μετά την επεξεργασία τους. Όμως, λόγω αυξανόμενης χωρητικότητας του δικτύου και του όγκο της κίνησης, αυτού του είδους η προσέγγιση δεν είναι πολύ αποτελεσματική. Αντίθετα, παρόμοια πακέτα (πακέτα με ένα σύνολο κοινών ιδιοτήτων) μπορούν να ομαδοποιηθούν συνθέτοντας ροές. Για παράδειγμα, μια ροή μπορεί να αποτελείται από όλα τα ρέοντα πακέτα που μοιράζονται την ίδια διεύθυνση προέλευσης και προορισμού, έτσι ώστε μια ροή να μπορεί να προκύψει χρησιμοποιώντας μόνο ορισμένα πεδία ενός πακέτου δικτύου. Με αυτόν τον τρόπο, παρόμοιοι τύποι επισκεψιμότητας μπορούν να αποθηκευτούν σε πιο συμπαγή μορφή χωρίς να χάσουμε τις πληροφορίες που μας ενδιαφέρουν. Αυτές οι πληροφορίες μπορούν να συγκεντρωθούν σε ένα διάγραμμα ροής και να εξαχθούν σε έναν συλλέκτη ικανό να αναφέρει μετρήσεις δικτύου σε μορφή φιλική προς το χρήστη. Όταν συλλέγονται αυτές οι πληροφορίες παρέχουν μια λεπτομερή προβολή της κίνησης του δικτύου.

Οι ακριβείς μετρήσεις δικτύου είναι μια πρόκληση, που έχει απασχολήσει εδώ και πολλά χρόνια την επιστυμονική κοινότητα. Σε εμπορικά περιβάλλοντα, το NetFlow είναι πιθανώς το πιο διαδεδομένο πρότυπο για τη λογιστική και τη διαχείριση της κίνησης δικτύου.

Το **nProbe** είναι ένα λογισμικό NetFlow v5/v9/IPFIX ικανό να συλλέγει, να αναλύει και να εξάγει αναφορές δικτυακής κίνησης χρησιμοποιώντας την τυπική μορφή Cisco NetFlow v5/v9/IPFIX. Είναι διαθέσιμο για τα περισσότερα λειτουργικά συστήματα της αγοράς, όπως Windows, BSD, Linux, MacOSX.

Τα πεδία πληροφορίας ροής που υποστηρίζονται αυτή τη στιγμή από το nProbe είναι αυτά που καθορίζονται στο NetFlow v9 RFC. Χαρακτηριστικά παραδείγματα παρουσιάζονται στον Πίνακα 3.1.

NetFlow Label	**IPFIX Label**	**Description**
%IPV4_SRC_ADDR	%sourceIPv4Address	IPv4 source address
%IPV4_SRC_MASK	%sourceIPv4PrefixLength	IPv4 source subnet mask
%INPUT_SNMP	%ingressInterface	Input interface SNMP idx
%IPV4_DST_ADDR	%destinationIPv4Address	IPv4 destination address
%IPV4_DST_MASK	%destinationIPv4PrefixLength	IPv4 dest subnet mask
%OUTPUT_SNMP	%egressInterface	Output interface SNMP idx
%SRC_AS	%bgpSourceAsNumber	Source BGP AS
%DST_AS	%bgpDestinationAsNumber	Destination BGP AS
%IN_BYTES	%octetDeltaCount	Incoming flow bytes (src→dst)
%IN_PKTS	%packetDeltaCount	Incoming flow packets (src→dst)
%OUT_BYTES	%postOctetDeltaCount	Outgoing flow bytes (dst→src)
%OUT_PKTS	%postPacketDeltaCount	Outgoing flow packets (dst→src)
%FLOW_START_MILLISECONDS	%flowStartMilliseconds	Msec of the first flow packet
%FLOW_END_MILLISECONDS	%flowEndMilliseconds	Msec of the last flow packet

Πίνακας 3.1: Στοιχεία πληροφορίας ροής

Το nProbe έχει σχεδιαστεί ως μηχανή που επεξεργάζεται πακέτα και υπολογίζει βασικά στατιστικά στοιχεία, καθώς και πρόσθετα που επεκτείνουν τον πυρήνα με πρόσθετες δυνατότητες. Κάθε πρόσθετο (plugin) αναλύει ένα συγκεκριμένο είδος κίνησης όπως για παράδειγμα DNS, DHCP, SMTP, BGP και HTTP, προσθέτοντας επιπλέον πεδία (HTTP_URL, DNS_QUERY, DHCP_CLIENT_IP, κτλ).

3.2 Εργαλεία Python - Βιβλιοθήκες

Η υλοποίηση της παρούσας διπλωματικής εργασίας έγινε με χρήση της γλώσσας προγραμματισμού Python. Τα προγραμματιστικά περιβάλλοντα που χρησιμοποιήθηκαν ήταν το Visual Studio Code, σε συνδυασμό με εικονικό περιβάλλον και το Google Colaboratory.

- Visual Studio Code: Είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα που εκτελείται στην επιφάνεια εργασίας και είναι διαθέσιμο για λειτουργικά συστήματα Windows, macOS και Linux.
- Google Colaboratory: Είναι ένα προϊόν από την Google το οποίο επιτρέπει σε οποιονδήποτε να γράψει και να εκτελέσει κώδικα Python μέσω του προγράμματος περιήγησης. Είναι ιδιαίτερα κατάλληλο για μηχανική μάθηση, ανάλυση δεδομένων και εκπαίδευση. Συγκεκριμένα, το Google Colaboratory είναι μια φιλοξενούμενη υπηρεσία Jupyter notebook που δεν απαιτεί εγκατάσταση για τη χρήση της, ενώ παρέχει δωρεάν πρόσβαση σε υπολογιστικούς πόρους, συμπεριλαμβανομένων και των GPU.

- Εικονικό Περιβάλλον: Είναι ένα απομονωμένο περιβάλλον Python όπου οι εξαρτήσεις και οι απαιτούμενες βιβλιοθήκες ενός έργου εγκαθίστανται σε διαφορετικό κατάλογο από εκείνους που είναι εγκατεστημένοι στην προεπιλεγμένη διαδρομή Python του συστήματος και σε άλλα εικονικά περιβάλλοντα.

Οι βασικές βιβλιοθήκες που συμπεριλήφθηκαν συνοψίζονται ακολούθως:

- **NumPy**: Θεμελιώδες βιβλιοθήκη για την εκτέλεση υπολογισμών, προσθέτοντας υποστήριξη για μεγάλους, πολυδιάστατους πίνακες μαζί με μια μεγάλη συλλογή μαθηματικών συναρτήσεων υψηλού επιπέδου.

- **MatplotLib**: Είναι μια βιβλιοθήκη δημιουργίας δισδιάστατων γραφικών παραστάσεων, που παράγει υψηλής ποιότητας γραφικά. Τα γραφικά αυτά έχουν διάφορους τύπους, δημιουργούνται εύκολα και μπορούν να αποθηκευτούν σε αρχεία διαφόρων τύπων. Η matplotlib συνήθως χρησιμοποιείται σε συνεργασία με την NumPy, επειδή αυτή χρειάζεται για το κομμάτι των μαθηματικών που χρησιμοποιούνται στα γραφήματα.

- **Pandas**: Είναι μια βιβλιοθήκη που επιτρέπει τον χειρισμό, την ταξινόμηση, το φιλτράρισμα και την τροποποίηση των δεδομένων. Η pandas παρέχει υψηλής απόδοσης δομές δεδομένων για τον χειρισμό, τον καθαρισμό και την προετοιμασία δεδομένων ώστε να αποτελέσσουν input για την διαδικασία της μηχανικής μάθησης.

- **Pomegranate**: Είναι ένα πακέτο Python που υλοποιεί γρήγορα και ευέλικτα πιθανοτικά μοντέλα που κυμαίνονται από μεμονωμένες κατανομές πιθανοτήτων, έως μοντέλα σύνθεσης, όπως τα δίκτυα Βαψες και τα κρυφά μοντέλα Μαρκο. Η βασική φιλοσοφία πίσω από την pomegranate είναι ότι όλα τα πιθανοτικά μοντέλα μπορούν να θεωρηθούν ως κατανομή πιθανοτήτων, καθώς παράγουν εκτιμήσεις πιθανοτήτων για δείγματα και μπορούν να ενημερώνονται για τα δείγματα και τα σχετικά βάρη τους.

- **Scikit-Learn**: Είναι μια βιβλιοθήκη που χρησιμοποιείται συνήθως σε προγράμματα μηχανικής μάθησης. Επικεντρώνεται στα εργαλεία μηχανικής μάθησης, συμπεριλαμβανομένων μαθηματικών, στατιστικών και γενικών αλγορίθμων που αποτελούν τη βάση για πολλές τεχνολογίες εκμάθησης μηχανών. Μερικά από τα βασικά στοιχεία της Scikit-Learn που είναι χρήσιμα για την εκμάθηση μηχανών, περιλαμβάνουν τους αλγόριθμους ταξινόμησης, παλινδρόμησης και συσταδοποίησης.

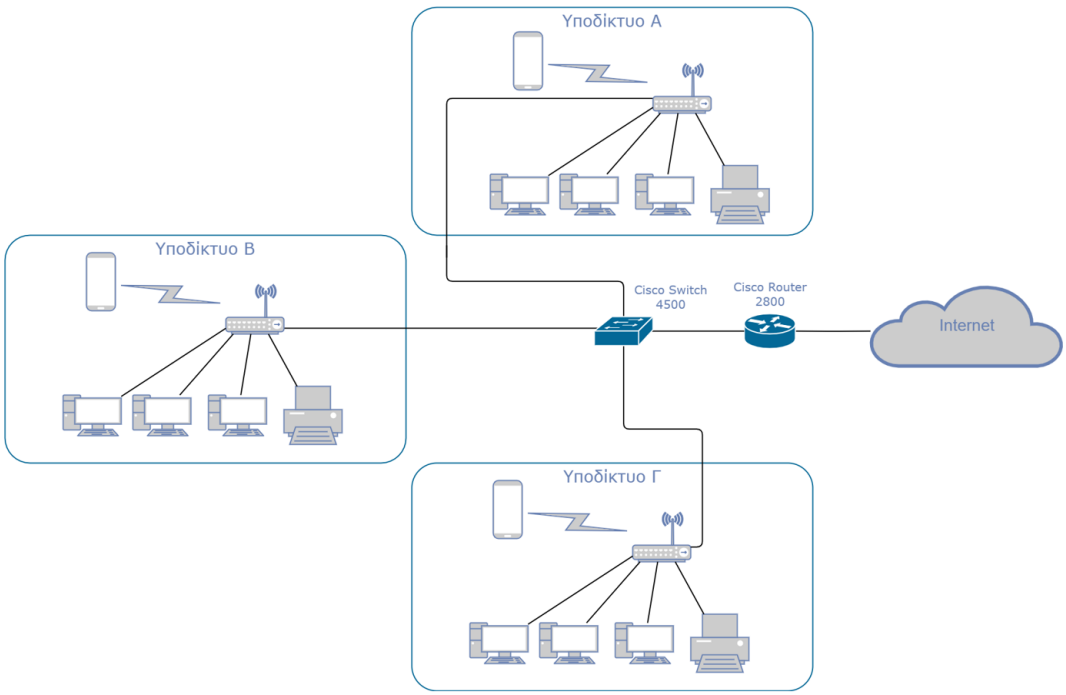
- **PyTorch**: Είναι μια βιβλιοθήκη μηχανικής εκμάθησης ανοιχτού κώδικα βασισμένη στη βιβλιοθήκη Torch, που χρησιμοποιείται για εφαρμογές όπως η όραση υπολογιστή και η επεξεργασία φυσικής γλώσσας, που αναπτύχθηκε κυρίως από το εργαστήριο AI Research του Facebook. Η PyTorch ορίζει μια κλάση που ονομάζεται Tensor για αποθήκευση και λειτουργία σε ομοιογενείς πολυδιάστατους ορθογώνιους πίνακες αριθμών. Οι τανυστές PyTorch είναι παρόμοιοι με τους πίνακες NumPy, αλλά εν αντιθέση, μπορούν επίσης να λειτουργήσουν σε GPU Nvidia με δυνατότητα CUDA.

Η επιτάχυνση της εκτέλεσης αριθμητικών πράξεων, και γενικότερα η εκπαίδευση των μοντέλων, έγινε εφικτή με τη χρήση κάρτας γραφικών GPU. Το περιβάλλον Google Colaboratory παρέχει διαφορετικούς τύπους GPU κάθε φορά, χωρίς ωστόσο να δίνεται η δυνατότητα επιλογής. Οι διαθέσιμες κάρτες γραφικών είναι οι Nvidia K80s, T4s, P4s και P100s.

Κεφάλαιο 4

Διερευνητική Ανάλυση Δεδομένων

Στην συγκεκριμένη ενότητα περιγράφονται τα στάδια καταγραφής, επεξεργασίας και ανάλυσης των νοσοκομειακών δεδομένων δικτυακής κίνησης.



Σχήμα 4.1: Ενδεικτικό παράδειγμα της δικτυακής νοσοκομειακής υποδομής

4.1 Καταγραφή - Περιγραφή Συνόλου Δεδομένων

Η καταγραφή του συνόλου δεδομένων πραγματοποιήθηκε σε πραγματική νοσοκομειακή υποδομή, μέρος της οποίας παρουσιάζεται στο Σχήμα 4.1. Ένας φορητός υπολογιστής συνδέθηκε με καλώδιο δικτύου στον Cisco μεταγωγέα (switch), αφού ενεργοποιήθηκε η SPAN

λειτουργικότητα, έτσι ώστε όλη η εξερχόμενη κίνηση του νοσοκομείου να περνάει μέσα από το μηχάνημα. Χρησιμοποιήθηκε το εργαλείο καταγραφής **nProbe**, σε συνδυασμό με το ntopng και μια τοπική βάση δεδομένων, προκειμένου να γίνει η καταγραφή και η συλλογή της δικτυακής κίνησης σε μορφή **NetFlow v9**.

NetFlow v9 Label	**Description**
%IPV4_SRC_ADDR	IPv4 source address
%IPV4_DST_ADDR	IPv4 destination subnet mask
%OUT_DST_MAC	Post Destination MAC Address
%IN_DST_MAC	Destination MAC Address
%OUT_SRC_MAC	Post Source MAC Address
%L4_SRC_PORT	IPv4 source port
%PROTOCOL	IP protocol byte
%L7_PROTO_NAME	Layer 7 protocol name
%L7_PROTO_CATEGORY	Layer 7 protocol category
%IN_BYTES	Incoming flow bytes (src→dst)
%IN_PKTS	Incoming flow packets (src→dst)
%OUT_BYTES	Outgoing flow bytes (dst→src)
%OUT_PKTS	Outgoing flow packets (dst→src)
%FLOW_END_REASON	The reason for flow termination
%TCP_FLAGS	Cumulative of all flow TCP flags
%SERVER_TCP_FLAGS	Cumulative of all server TCP flags
%DIRECTION	Flow direction [0=RX, 1=TX]
%FLOW_START_MILLISECONDS	Msec of the first flow packet
%FLOW_END_MILLISECONDS	Msec of the last flow packet
%ICMP_TYPE	ICMP Type * 256 + ICMP code
%HTTP_URL	HTTP URL (IXIA URI)
%HTTP_METHOD	HTTP METHOD
%HTTP_RET_CODE	HTTP return code (e.g. 200, 304...)
%DNS_QUERY	DNS query
%DNS_QUERY_TYPE	DNS query type (e.g. 1=A, 2=NS..)
%DNS_RET_CODE	DNS return code (e.g. 0=no error)
%DNS_RESPONSE	DNS response(s)

Πίνακας 4.1: Τα πεδία NetFlow v9 που καταγράφηκαν

Τα αρχεία ροής που συλέχθηκαν ήταν κάθε ένα λεπτό (μονόλεπτα) και ολόκληρη η διαδικασία καταγραφής διήρκησε 33 ημέρες, από τις 07/04/2021 έως τις 09/05/2021. Τα πεδία πληροφορίας ροής που συμπεριλήφθηκαν εν τέλη και καταγράφηκαν παρουσιάζονται στον Πίνακα 4.1 μαζί με μια μικρή περιγραφή.

4.2 Επεξεργασία Συνόλου Δεδομένων

Προκειμένου να υπάρχει πλήρης ανωνυμοποίηση του συνόλου δεδομένων, αρκετά πεδία πληροφορίας αφαιρέθηκαν οριστικά, όπως για παράδειγμα η διεύθυνση IP πηγής, το πλήρες HTTP URL κτλ. Τα στάδια επεξεργασίας που ακολουθήθηκαν παρουσιάζονται παρακάτω:

- Κρατήθηκαν τα πεδία πληροφορίας που δίνουν κάποια σημαντική πληροφορία και διαγράφηκαν τα υπόλοιπα.
- Κρατήθηκαν οι εγγραφές που η IP πηγής ανήκει στο υποδίκτυο 10.10.0.0/16, διότι στους συγκεκριμένους χρήστες θα γίνει η συσταδοποίηση και το profiling.
- Δημιουργήθηκε νέο χαρακτηριστικό (στήλη στα flows) με βάση το vlan που ανήκει ο κάθε χρήστης πηγή (10.10.X.0/24 → vlan X), το οποίο, με βάση τις πληροφορίες που λήφθηκαν, αντιπροσωπεύει κτηριακά συγκροτήματα στο νοσοκομείο.
- Οι διευθύνσεις IP πηγής αφαιρέθηκαν και αντικαταστάθηκαν από γενικές κατηγοριοποιήσεις σύμφωνα με το τμήμα που ανήκει ο χρήστης, όπως ακτινολογικό, καρδιολογικό κτλ. Αυτές οι επιπλέον πληροφορίες για την κατηγοριοποίηση προέκυψαν από τα logs του DHCP server του νοσοκομείου. Προκειμένου να αποφευχθεί το πρόβλημα των πολλαπλών διευθύνσεων IP ανά συσκευή διαχρονικά λόγω ανανεώσεων μισθώσεων, τα DHCP logs έπρεπε να διερευνηθούν αναλόγως χρονικά, ώστε να προκύψει η σωστή αντιστοίχηση IP με συσκευή για κάθε χρονικό πλαίσιο.
- Η μοναδικότητα κάθε χρήστη διατηρήθηκε χρησιμοποιώντας μοναδικά IDs πηγής και προορισμού.
- Επομένως, με βάση τα δύο παραπάνω bullets, η διεύθυνση IP αντικαταστήθηκε με δύο επιπλέον πεδία, SRC_MACHINE/DST_MACHINE και SRC_ID/DST_ID.
- Οι συσκευές που δεν αντιστοιχίστηκαν σε ένα συγκεκριμένο τμήμα (παθολογική, γραμματεία, κτλ.), επισημάνθηκαν ως «user desktop», «mobile» ή «network device» ανάλογα με το host-name που προέκυπτε από τα DHCP logs.
- Όσον αφορά τον «server» ως ετικέτα στα πεδία SRC_MACHINE/DST_MACHINE, αναφέρεται σε όλους τους DHCP, DNS, Secondary DNS, Active Directory, NAS που επισημάνθηκαν ως «server» στο πεδίο SRC_MACHINE.
- Το πεδίο HTTP_URL χρησιμοποιήθηκε μόνο για την εξαγωγή των ονομάτων των ιστοσελίδων που επισκέφτηκε ο χρήστης, αγνοώντας τυχόν παραμέτρους.

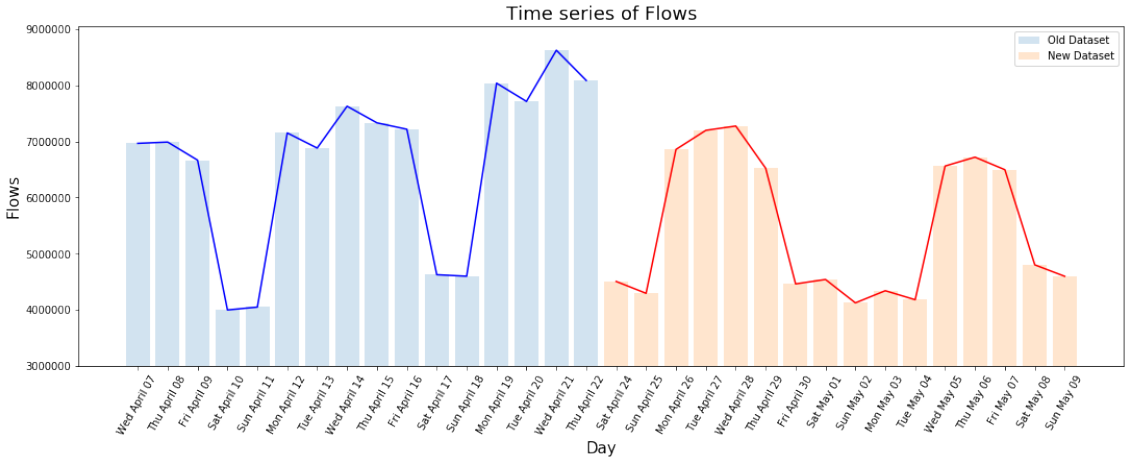
- Ένα γενικό αναγνωριστικό για τον τύπο κίνησης (π.χ. DICOM, LIS, BMS κτλ.) διατηρήθηκε σε ένα νέο πεδίο με το όνομα «Traffic».

4.3 Ανάλυση Συνόλου Δεδομένων

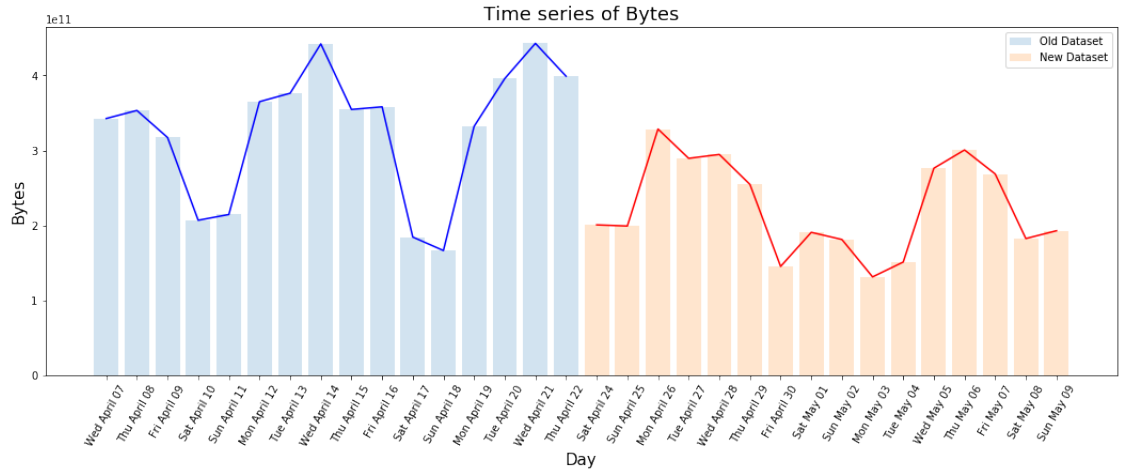
Έπειτα από την πρώτη επεξεργασία και ανωνυμοποίηση του συνόλου δεδομένων, πραγματοποιήθηκε λεπτομερής διερευνητική ανάλυση δεδομένων, που περιλαμβάνει διάφορες τεχνικές εξόρυξης δεδομένων, πολλαπλές οπτικοποιήσεις, διαγράμματα, καθώς και στατιστικά στοιχεία.

4.3.1 Διαγράμματα Χρονοσειρών

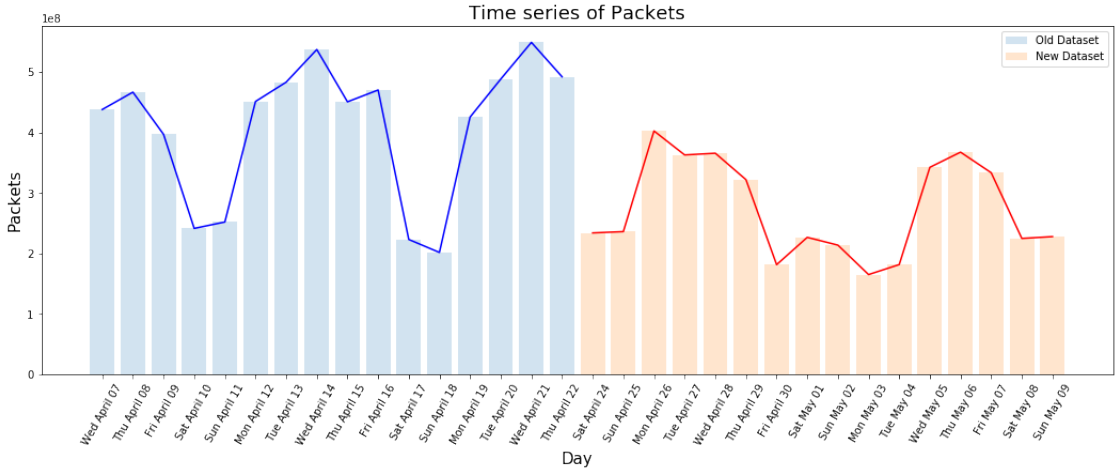
Αρχικά, οι χρονοσειρές ανά ημέρα του αριθμού των ροών (Σχήμα 4.2), των συνολικών bytes (Σχήμα 4.3) και των πακέτων (Σχήμα 4.4) απεικονίζονται στα παρακάτω διαγράμματα.



Σχήμα 4.2: Αριθμός Flows ανά ημέρα



Σχήμα 4.3: Αριθμός Bytes ανά ημέρα



Σχήμα 4.4: Αριθμός Packets ανά ημέρα

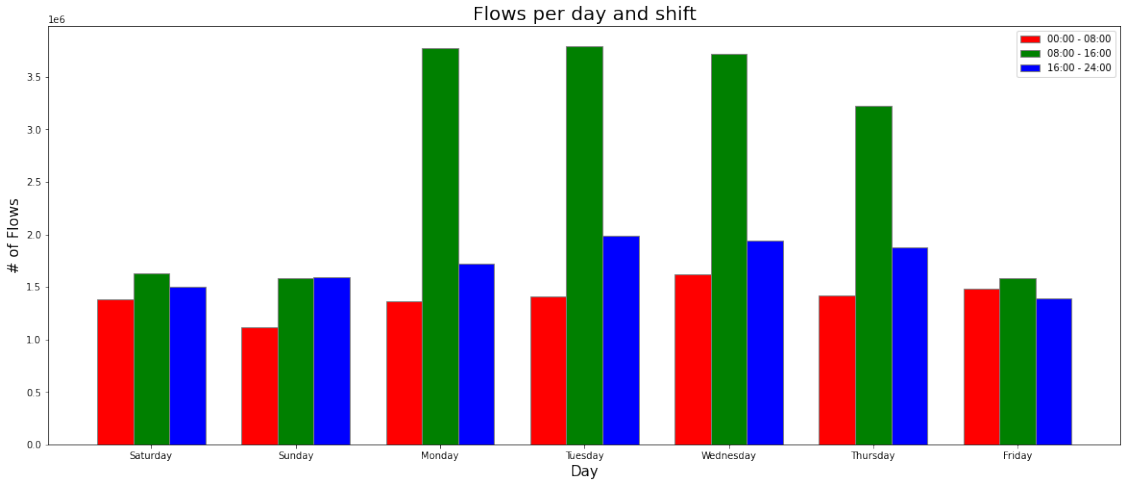
Με βάση τα παραπάνω διαγράμματα παρατηρείται ξεκάθαρα ότι το Σαββατοκύριακο έχει πολύ μικρότερη επισκεψιμότητα στο διαδίκτυο σε σχέση με άλλες ημέρες, γεγονός που οφείλεται κυρίως στο μικρότερο αριθμό διοικητικών καθηκόντων. Επιπλέον, παρόμοια συμπεριφορά παρατηρείται και την εβδομάδα του Πάσχα που ξεκινάει τη Δευτέρα 26 Απριλίου και τελειώνει Κυριακή 2 Μαΐου. Η χαμηλότερη επισκεψιμότητα καταγράφεται σε όλες τις χρονοσειρές (flows, bytes, packets), ειδικά την Μεγάλη Παρασκευή έως την Κυριακή του Πάσχα, που θεωρούνται και εθνικές εορτές. Επίσης, τη Δευτέρα 3 Μαΐου και την Τρίτη 4 Μαΐου που είναι εθνικές αργίες (Καθαρά Δευτέρα και Πρωτομαγιά αντίστοιχα), η κίνηση είναι ακόμη χαμηλότερη από τα κανονικά Σαββατοκύριακα. Αυτό συμβαίνει για διάφορους λόγους, που κυμαίνονται από το γεγονός ότι μεγάλο μέρος του προσωπικού του νοσοκομείου παίρνει ρεπό (εκτός από το προσωπικό που χειρίζεται επείγοντα περιστατικά), λιγότερα άτομα έχουν ραντεβού στο νοσοκομείο αυτές τις ημέρες (υποτίθεται ότι ήταν επείγουσα περιστατικά νοσηλείας συμβαίνουν στον ίδιο όγκο όπως τις κανονικές ημέρες) κτλ. Τέλος, παρατηρείται μια ομοιότητα που διακατέχει τα διαγράμματα των bytes και των packets, λόγω της αναλογικότητας που συνήθως κυριαρχεί τη σχέση τους.

Κατανομή της κίνησης μεταξύ των βάρδιων

Στο Σχήμα 4.5, ο όγκος ροής (συνολικός αριθμός ροών) ανά βάρδια την εβδομάδα μεταξύ 24/04/2021 και 30/04/2021.

Παρατηρείται ότι από τις τρεις βάρδιες, η δεύτερη (08:00 - 16:00) είναι αυτή με τη μεγαλύτερη κίνηση. Μεταξύ των άλλων δύο, η τρίτη (16:00 - 24:00) έχει συνήθως τη μεγαλύτερη κίνηση. Αυτό προκύπτει από το γεγονός ότι σχεδόν όλες οι διοικητικές-γραμματειακές εργασίες γίνονται το πρωί και νωρίς το απόγευμα. Το ιατρικό προσωπικό, που εργάζεται σε βάρδιες, δημιουργεί ροές 24/7, παρόλο που ο αριθμός των ροών το πρωί είναι μεγαλύτερος, με εξαίρεση τα Σαββατοκύριακα.

Όσον αφορά τις ημέρες της εβδομάδας, η περισσότερη δικτυακή κίνηση παρατηρείται από Δευτέρα έως Πέμπτη. Η Παρασκευή και το Σαββατοκύριακο παρουσιάζουν σημαντική μείωση

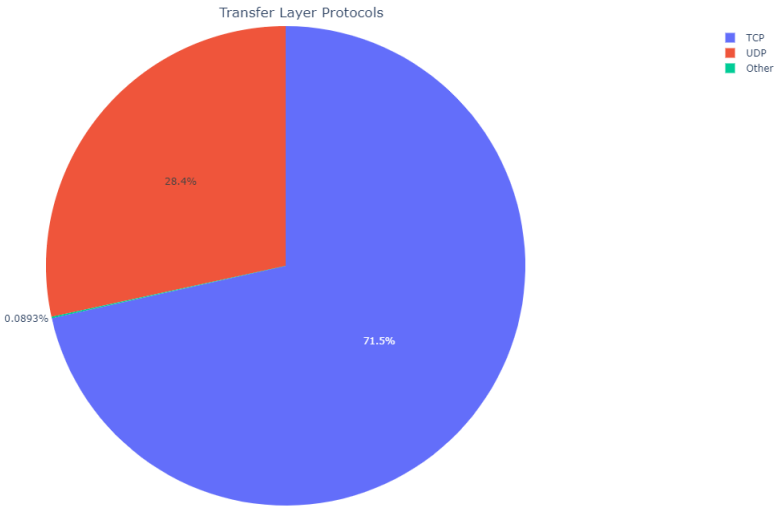


Σχήμα 4.5: Αριθμός ροών ανά βάρδια για μια εβδομάδα

της κίνησης σχεδόν και στις τρεις βάρδιες.

Αξίζει επίσης να σημειωθεί και πάλι ότι η Παρασκευή 30/04/2021 θεωρείται εθνική εορτή στην Ελλάδα, καθώς είναι η Μεγάλη Παρασκευή της Ορθοδοξίας. Έτσι, αναμένεται να έχει χαμηλότερη από το μέσο όρο επισκεψιμότητα. Επίσης, η προηγούμενη ημέρα (Μεγάλη Πέμπτη), είναι μέρος της Μεγάλης Εβδομάδας του Πάσχα (αν και δεν είναι εθνική εορτή) και επομένως παρατηρείται αισθητή μείωση του συνολικού αριθμού ροών.

4.3.2 Στατιστικά πρωτόκολλου επιπέδου μεταφοράς



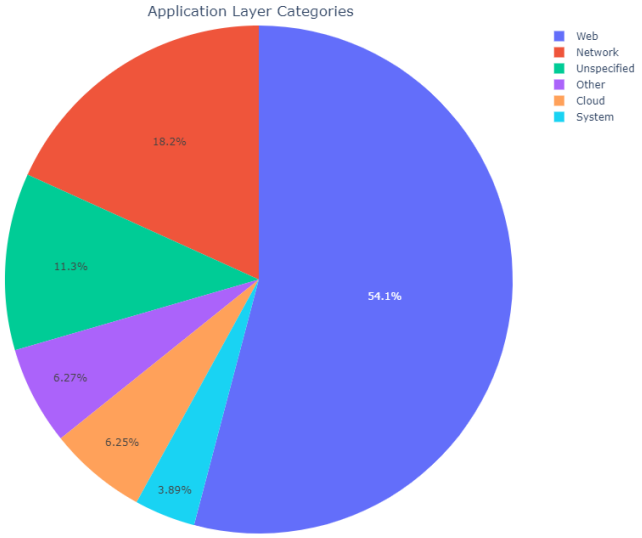
Σχήμα 4.6: Κατανομή πρωτοκόλλων στρώματος μεταφοράς

Στο Σχήμα 4.6 απεικονίζεται η κατανομή των πρωτοκόλλων του επιπέδου μεταφοράς μεταξύ των ροών. Όπως αναμενόταν, το TCP είναι το κυρίαρχο πρωτόκολλο επιπέδου μεταφοράς, καθώς οι περισσότερες επικοινωνίες απαιτούν ανταλλαγή δεδομένων προσανατολισμένη στη

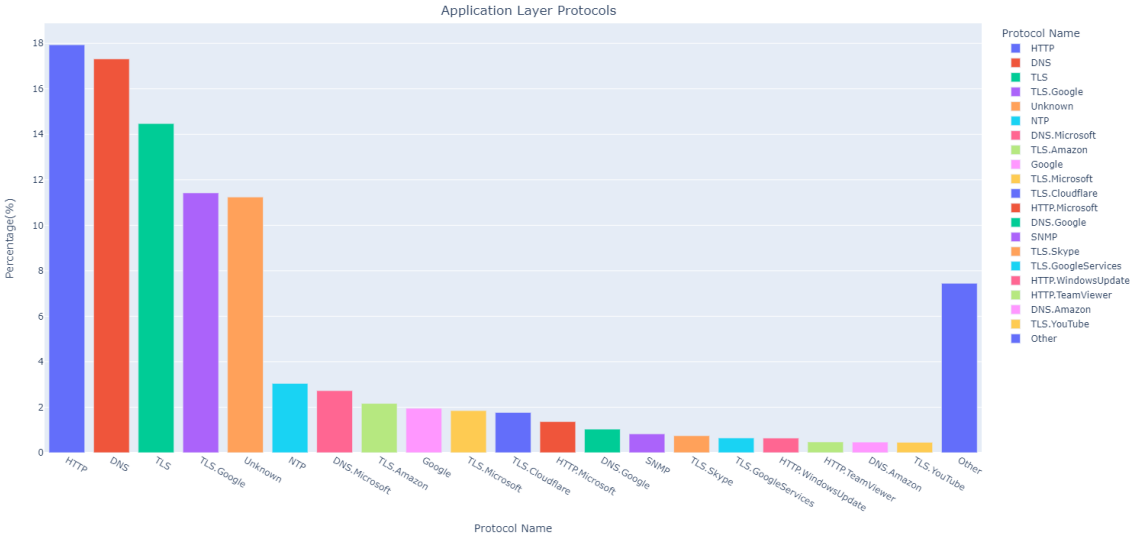
σύνδεση. Ωστόσο, το μερίδιο του UDP είναι σημαντικό, αφού χρησιμοποιείται σε λειτουργίες, όπως η ροή ήχου και βίντεο και η απάντηση ερωτημάτων (πχ. ερωτήματα DNS), τα οποία είναι εξαιρετικά κοινά στις σημερινές υποδομές επιχειρηματικών δικτύων.

4.3.3 Στατιστικά πρωτόκολλου επιπέδου εφαρμογής

Στο Σχήμα 4.7 παρουσιάζεται η κατανομή των κατηγοριών πρωτοκόλλων στρώματος εφαρμογής. Όπως ήταν αναμενόμενο, τα πιο δημοφιλή είναι αυτά της κατηγορίας web και ακολουθούν αυτά του network. Σε μικρότερα ποσοστά παρατηρούνται οι κατηγορίες cloud και system.



Σχήμα 4.7: Κατανομή κατηγοριών πρωτοκόλλων στρώματος εφαρμογής

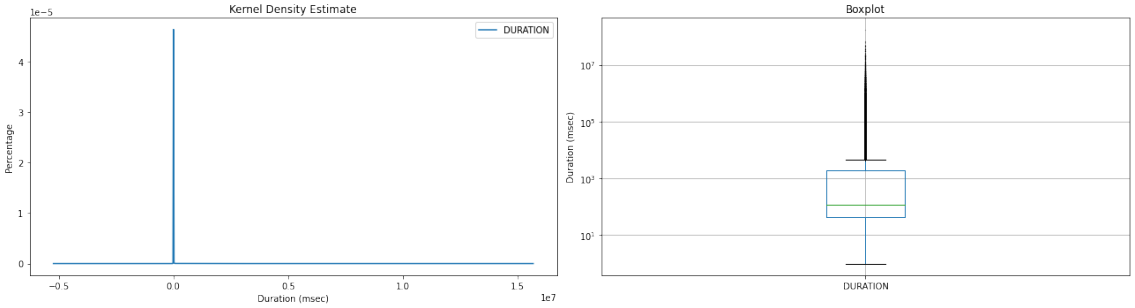


Σχήμα 4.8: Κατανομή πρωτοκόλλων στρώματος εφαρμογής

Η γραφική παράσταση του Σχήματος 4.8 απεικονίζει την κατανομή των πρωτοκόλλων του επιπέδου εφαρμογής στις παρατηρούμενες ροές. Είναι προφανές ότι το TLS είναι από τα πιο δημοφιλή (όπως θα έπρεπε, καθώς η περισσότερη κίνηση είναι κρυπτογραφημένη στις μέρες μας). Ακολουθεί το DNS και το HTTP (μαζί με κάποιο πρωτόκολλο άγνωστου επιπέδου εφαρμογής), και μετά το NTP και το SNMP.

4.3.4 Στατιστικά διάρκειας ροής

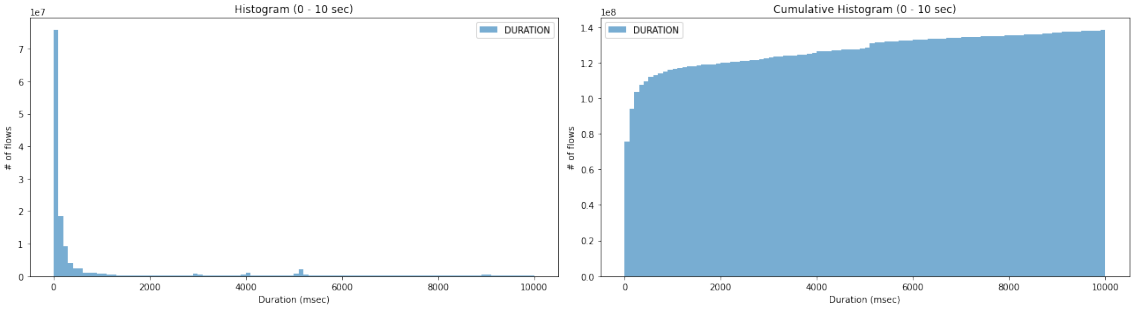
Το Σχήμα 4.9 απεικονίζει την κατανομή της διάρκειας ροής σε όλη τη δικτυακή κίνηση. Παρέχεται η πυκνότητα πιθανότητας (ως αποτέλεσμα της μεθόδου εκτίμησης πυκνότητας πυρήνα) και ένα boxplot.



Σχήμα 4.9: Κατανομή διάρκειας ροής

Είναι προφανές από τα διαγράμματα ότι η πλειονότητα των διάρκειων ροής κυμαίνεται από 0.1 έως 10 δευτερόλεπτα. Αυτό οφείλεται στο γεγονός ότι πολλές ροές δικτύου δημιουργούνται αυτόματα ακόμη και από μόνες τους (πχ. αίτημα και αποκρίσεις DNS, πακέτα DHCP, πακέτα NTP, ενημερώσεις των Windows, Active Directory πακέτα που είναι Radius ή LDAP). Επιπλέον, οι μικρής διάρκειας ροές δημιουργούνται από τις αλληλεπιδράσεις πραγματικών χρηστών (πχ. η περιήγηση σε μια ιστοσελίδα έχει συχνά ένα ερώτημα DNS ως επακόλουθο ή η παρακολούθηση ενός βίντεο YouTube έχει διάφορες μικρότερες ροές που δημιουργούνται για τη συλλογή δεδομένων, όπως διαφημίσεις).

Τα διαγράμματα του Σχήματος 4.10 εστιάζουν σε μικρότερες ροές, έως και 10 δευτερολέπτων, καθώς αποτελούν το 87% της συνολικής δικτυακής κίνησης.

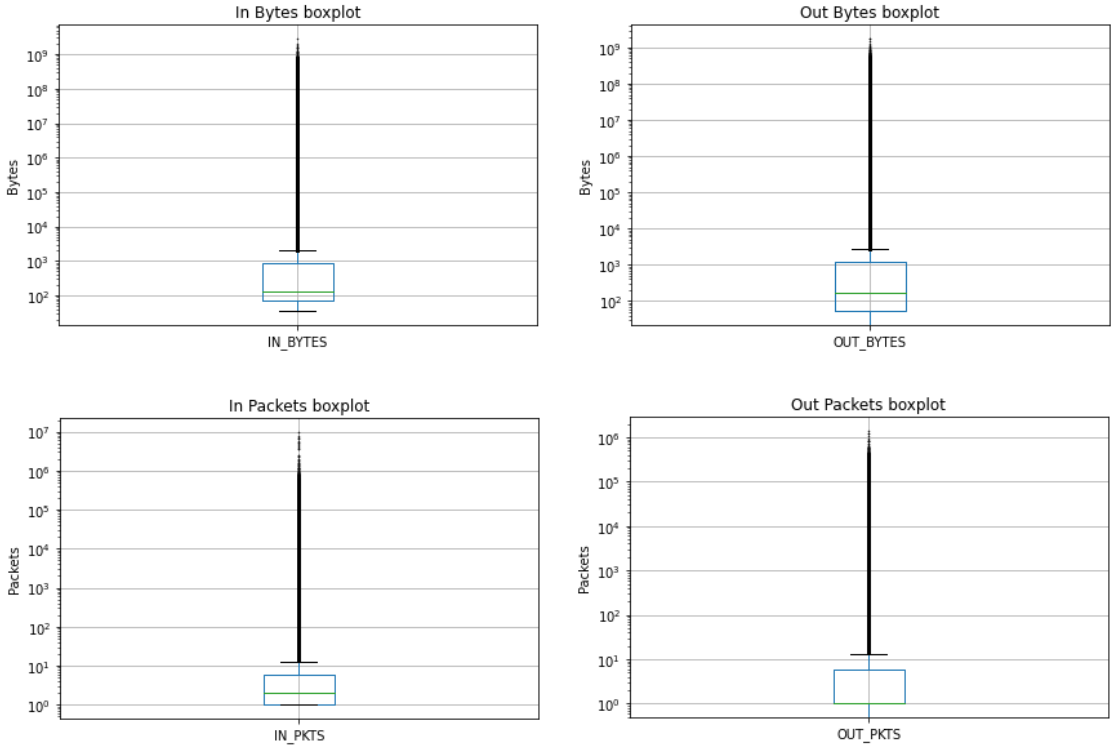


Σχήμα 4.10: Κατανομή διάρκειας ροής (μικρότερη των 10 sec)

Από την άλλη πλευρά, μεγαλύτερες ροές μπορεί να φτάσουν έως και αρκετές ώρες. Τέτοιες μεγάλες ροές μπορεί να είναι ιστοσελίδες ραδιοφώνου, βίντεο YouTube, ζωντανές μεταδόσεις, λήψεις μεγάλων αρχείων κτλ.

4.3.5 Στατιστικά bytes και πακέτων ροής

Το Σχήμα 4.11 απεικονίζει την κατανομή των εισερχόμενων και εξερχόμενων bytes και πακέτων σε όλη τη δικτυακή κίνηση μέσω boxplots.



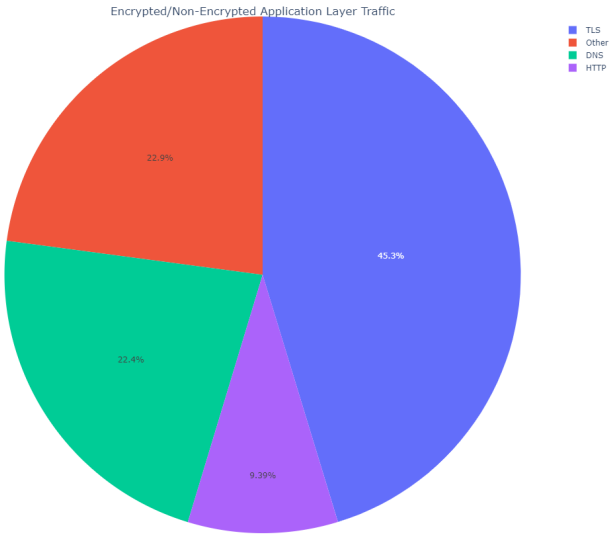
Σχήμα 4.11: Κατανομή in/out bytes και packets

Με βάση τα παραπάνω διαγράμματα, παρατηρείται ότι ο μέσος αριθμός bytes ανά ροή κυμαίνεται μεταξύ 100 και 1000. Όσον αναφορά τα πακέτα, η πλειοψηφία των ροών έχει από 5 μέχρι 10 πακέτα εισερχόμενα και εξερχόμενα. Αυτές οι παρατηρήσεις είναι παρόμοιες με αυτές των διαρκειών που μελετήθηκαν παραπάνω και οφείλονται στις ίδιες αφορμές. Δηλαδή, οι πλειοψηφία των ροών όπως DNS, DHCP, NTP δημιουργούνται αυτόματα ακόμη και από μόνες τους. Παρόλο αυτά, όπως και προηγουμένως, υπάρχουν πολλές ακραίες τιμές (outliers) που φτάνουν μέχρι και τα 10^9 bytes και 10^6 packets, που πιθανόν να προέρχονται από βίντεο, ζωντανές μεταδόσεις, λήψεις μεγάλων αρχείων, μουσική κτλ.

4.3.6 Στατιστικά Κρυπτογράφησης

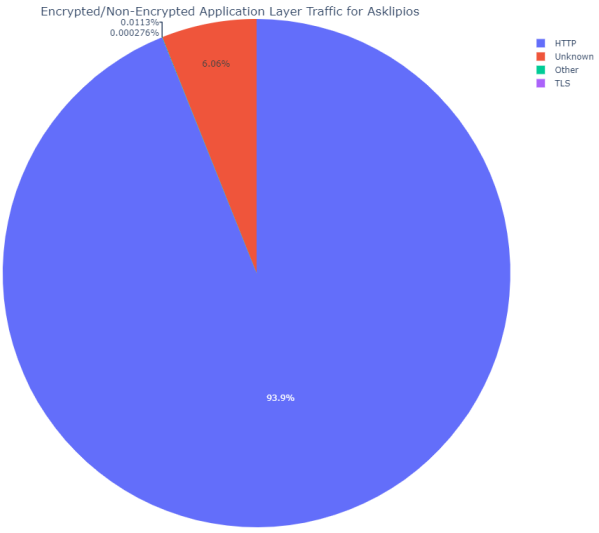
Όπως αναφέρθηκε και σε προηγούμενη υποενότητα, το πρωτόκολλο TLS διασπάται και σε περαιτέρω layer 7 πρωτόκολλα όπως το TLS.Google, TLS.Facebook, TLS.YouTube,

TLS.Amazon κτλ. Επομένως, ήταν χρήσιμο να ενοποιηθεί και εν τέλη να παρουσιαστεί ένα γενικό διάγραμμα με τα τρία πιο δημοφιλή πρωτόκολλα εφαρμογής, όπως φαίνεται και στο Σχήμα 4.12.



Σχήμα 4.12: Τα τρία δημοφιλέστερα πρωτόκολλα εφαρμογής

Με βάση το παραπάνω διάγραμμα αξίζει να παρατηρηθεί η αναλογία μεταξύ TLS και HTTP, δηλαδή η αναλογία μεταξύ κρυπτογραφημένης και μη κρυπτογραφημένης κίνησης. Οι ροές με το πρωτόκολλο TLS είναι σχεδόν 4.5 φορές περισσότερες από αυτές με HTTP. Αυτό το νούμερο θα μπορούσε να θεωρηθεί μικρό στις μέρες μας, διότι σχεδόν όλη η κίνηση θα έπρεπε να είναι κρυπτογραφημμένη. Παρόλο αυτά, στην συγκεκριμένη περίπτωση, οι ροές προς τις εσωτερικές ιατρικές υπηρεσίες του HIS είναι σχεδόν όλες μέσω του HTTP, γι αυτό και παρατηρείται μια τέτοια αναλογία TLS-HTTP.



Σχήμα 4.13: Κατανομή πρωτόκολλων εφαρμογής μόνο προς HIS

Αυτό ακριβώς το φαινόμενο παρουσιάζεται στο Σχήμα 4.13, όπου το διάγραμμα αποκαλύπτει την επείγουσα ανάγκη για αναβάθμιση του νοσοκομειακού συστήματος και δημιουργίας ασφαλής σύνδεσης μέσω κρυπτογραφημένων επικοινωνιών. Παρόλο που η υπηρεσία HIS βρίσκεται σε εωτερικό δίκτυο, με private IP, αυτό δεν αναιρεί τον κίνδυνο για παρακολουθήσεις κυκλοφορίας από κακόβουλους χρήστες του νοσοκομείου. Αυτό ερμηνεύεται ως μια σημαντική ευπάθεια, ειδικά για υποδομές ζωτικής σημασίας όπως οργανισμοί υγειονομικής περίθαλψης και κρίσιμες υπηρεσίες που περιέχουν ευαίσθητα προσωπικά δεδομένα υγειονομικής περίθαλψης και ασθενών. Αξίζει επίσης να σημειωθεί ότι η κίνηση προς το HIS είναι περίπου το 1/3 των συνολικών ροών HTTP (4.76 εκατομμύρια από 15.73 εκατομμύρια ροές HTTP).

4.3.7 Διερεύνηση δικτυακής συμπεριφοράς διαφορετικών χρηστών

Τα μηχανήματα (machines) του νοσοκομείου έχουν κατηγοριοποιηθεί ανάλογα με τη λειτουργικότητά τους, το τμήμα του νοσοκομείου και τους τύπους χρηστών σε σχέση με τους πελάτες. Γι αυτό το λόγο ορίστηκαν και τα εξής:

- Public IP: Περιέχει οποιαδήποτε διεύθυνση IP είναι άμεσα ορατή στο διαδίκτυο, όπως τις IP του YouTube, της Google, κλπ. Ωστόσο, η IP του promitheus.gov δεν συμπεριλήφθηκε σε αυτή την κατηγορία καθώς πρόκειται για ιατρικό site το οποίο είναι χρήσιμο να μελετηθεί ξεχωριστά.

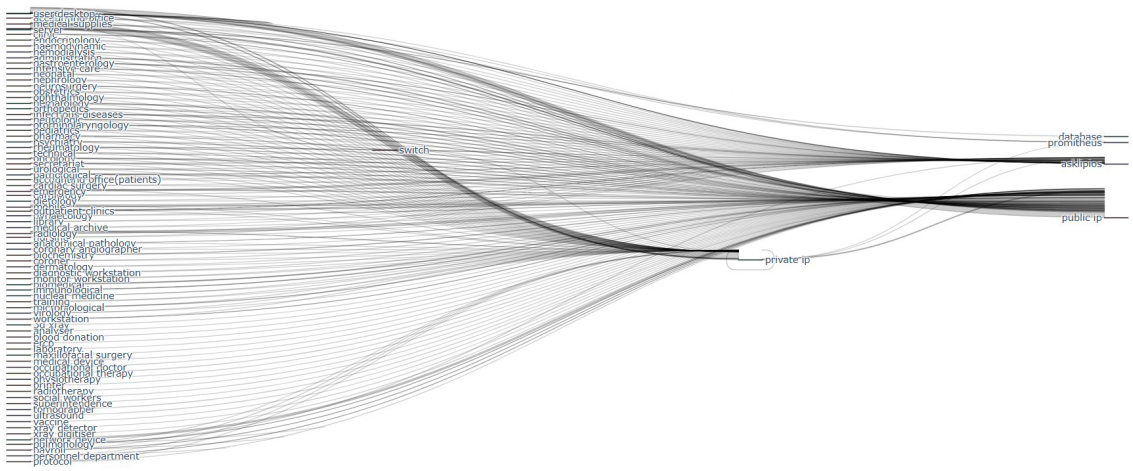
- Private IP: Περιέχει τις διευθύνσεις IP που χρησιμοποιούνται σε εσωτερικά δίκτυα και βρίσκονται σε διάφορα υποδίκτυα ανά τον κόσμο που δεν διέρχονται μέσα από δρομολογητές. Ωστόσο, η IP του HIS εξαιρείται από τις ιδιωτικές IP (παρόλο που είναι) καθώς πρόκειται για ιατρικό σιτε το οποίο είναι χρήσιμο να μελετηθεί σαν ξεχωριστή υπηρεσία.

- Servers: Αυτή η κατηγορία ομαδοποιεί τους DHCP, DNS, Secondary DNS, NAS και Active Directory διακομιστές της νοσοκομειακής υποδομής.

Ποιοτική Έρευνα

Στο Σχήμα 4.14 παρουσιάζεται ένα πλήρες διάγραμμα ροής (Sankey diagram) της κίνησης του δικτύου συγκεντρωτικά για τις πρώτες 10 ημέρες. Απεικονίζονται δηλαδή οι σχέσεις μεταξύ των κόμβων του νοσοκομειακού δικτύου. Είναι αρκετά περίπλοκο καθώς περιέχει τα 200 κορυφαία ζεύγη IP πηγής και προορισμού, όσον αφορά τον αριθμό των ροών μεταξύ τους. Ήταν σημαντικό να πραγματοποιηθεί αυτή η μείωση των ζεύγων, προκειμένου να αποφευχθεί μια περαιτέρω περίπλεξη του διαγράμματος.

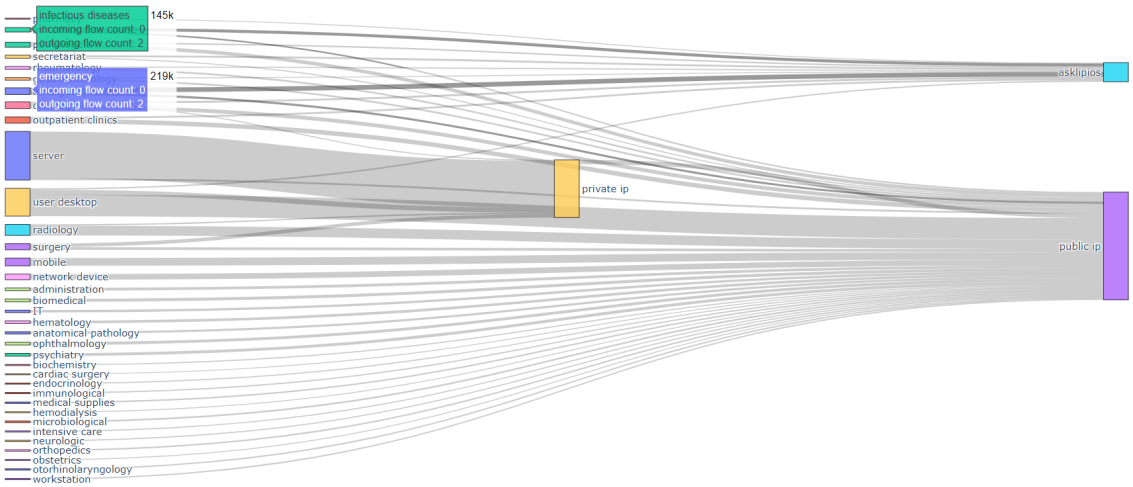
Με βάση λοιπόν το διάγραμμα παρατηρείται ότι σχεδόν όλες οι συσκευές που σχετίζονται με κλινικές όπως η παθολογία η καρδιολογία κτλ., συμπεριφέρονται ως πελάτες (clients) στους διακομιστές του HIS. Τα διοικητικά τμήματα όπως το τμήμα μισθοδοσίας και προσωπικού αλληλεπιδρούν με τις αμιγώς διοικητικές υπηρεσίες του νοσοκομειακού συστήματος. Επιπλέον, το τμήμα ιατρικών προμηθειών μαζί με μερικούς προσωπικούς υπολογιστές χρηστών και συσκευές λογιστηρίου, είναι τα μόνα που αλληλεπιδρούν με την ιατρική υπηρεσία Προμηθέας, η



Σχήμα 4.14: Διάγραμμα Sankey για τις πρώτες 10 ημέρες

οποία αφορά ιατρικό εφοδιασμό. Όπως ήταν αναμενόμενο, όλες οι συσκευές αλληλεπιδρούν με τυχαίες δημόσιες IP του διαδικτύου.

Μια αξιοσημείωτη παρατήρηση είναι ότι η κίνηση διακομιστή (server) εμφανίζεται ως κίνηση πελάτη στο διάγραμμα Sankey και αυτό επιβεβαιώθηκε ακόμη και στην περίπτωση των ροών DNS (που περιέχουν πάνω από το 90% των ροών διακοσμιτών). Το φαινόμενο αυτό είναι εξαιρετικά περίεργο, καθώς οι συσκευές δικτύου συνήθως εκκινούν τις συνδέσεις (και άρα τις ροές) όταν αναζητάνε ένα όνομα προς επίλυσης DNS. Παρτηρήθηκαν δηλαδή μόνο οι απαντήσεις των ερωτημάτων DNS και όχι οι ερωτήσεις. Το γεγονός αυτό αποδόθηκε στην ανικανότητα δημιουργίας δικτυακών ροών από το εργαλείο nProbe για την σωστή καταγραφή αυτών των συνδέσεων. Επομένως, οι ροές που έχουν διακομιστή στα αριστερά θα πρέπει να θεωρηθούν ότι ρέουν προς την αντίθετη κατεύθυνση από αυτήν που απεικονίζεται στα διαγράμματα.

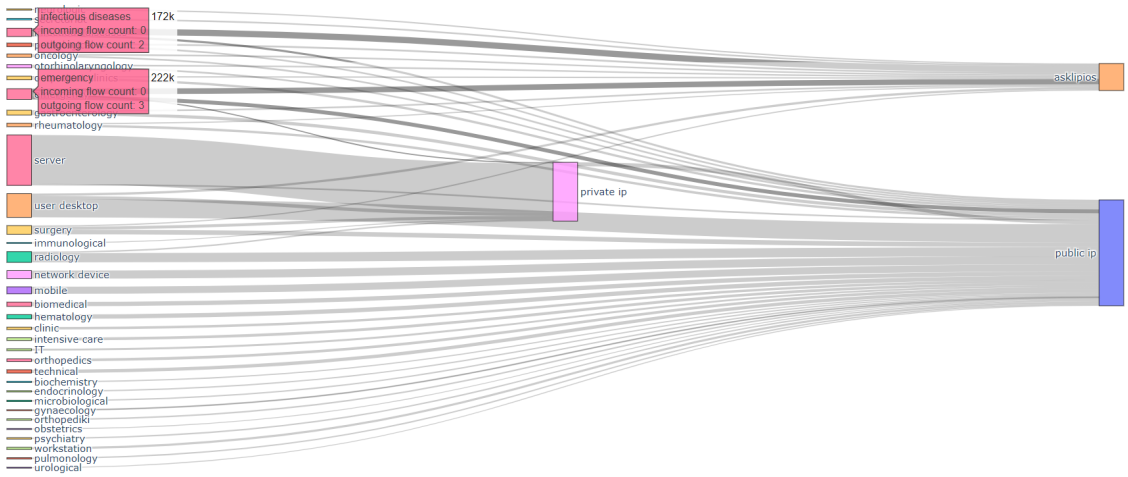


Σχήμα 4.15: Διάγραμμα Sankey Τετάρτη 28 Απριλίου 2021 (Καθημερινή)

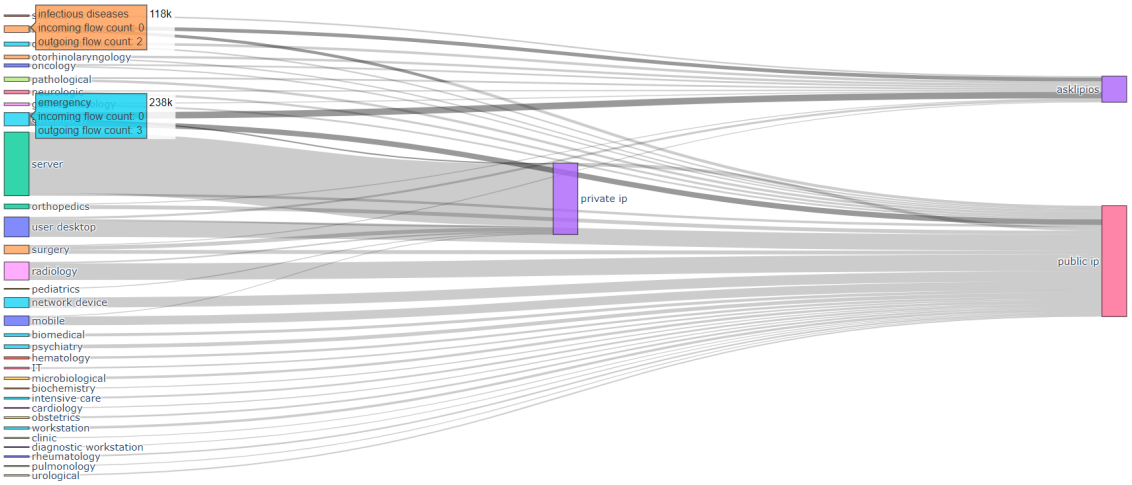
Προκειμένου να απλοποιηθεί το γενικό διάγραμμα και να διερευνηθεί καλύτερα η δικτυακή

συμπεριφορά, επιλέχθηκαν μια εργάσιμη ημέρα, μια μέρα Σαββατοκύριακου και μια αργία για να σχεδιαστούν τα αντίστοιχα διαγράμματα Sankey. Όπως και προηγουμένως, για να αποφευχθεί μια μεγάλη πολυπλοκότητα και να εξασφαλιστεί η ορατότητα των διαγραμμάτων, επιλέχθηκαν οι κορυφαίοι 50 συνδυασμοί πηγής-προορισμού από άποψη των αριθμών ροών.

Με βάση τα τρία ημερήσια διαγράμματα Sankey, μπορεί να παρατηρηθεί μια σαφώς χαμηλότερη δικτυακή κίνηση τις ημέρες του Σαββατοκύριακου (Εικόνα 13) και τις επίσημες αργίες (Εικόνα 15). Αυτή η μείωση οφείλεται κυρίως στη μειωμένη δραστηριότητα του διοικητικού προσωπικού (προσωπικοί υπολογιστές χρηστών και προσωπικό πληροφορικής) καθώς είναι αυτά που υπόκεινται σε συμβατικές ώρες λειτουργίας.



Σχήμα 4.16: Διάγραμμα Sankey Σάββατο 24 Απριλίου 2021 (Σαββατοκύριακο)



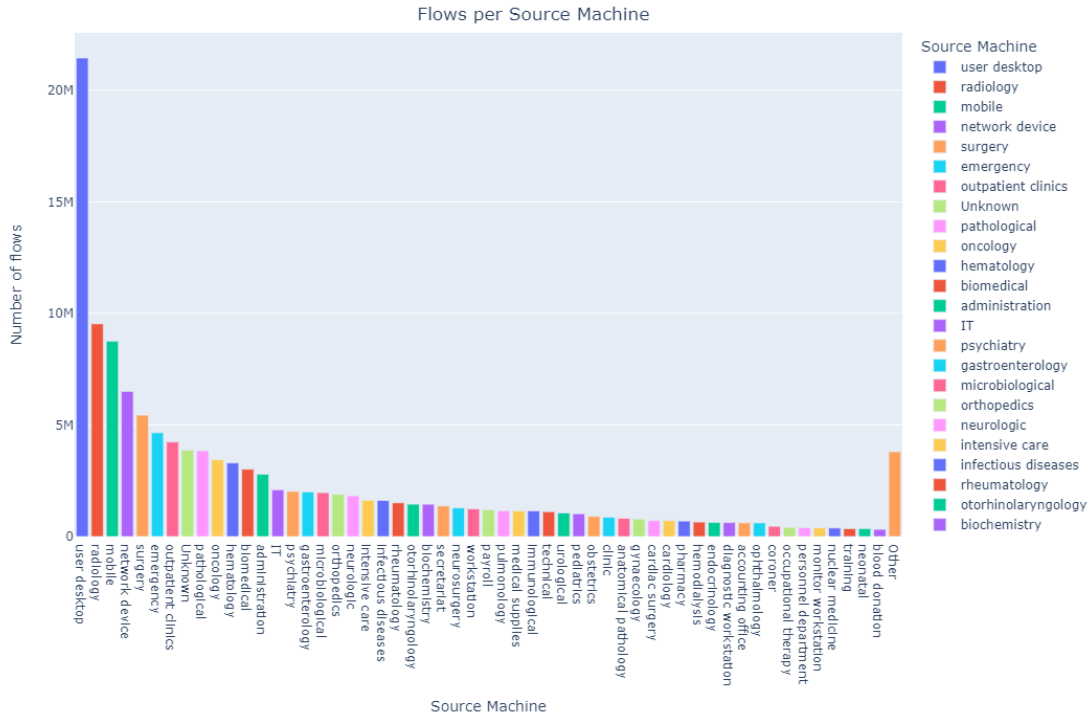
Σχήμα 4.17: Διάγραμμα Sankey Τρίτη 04 Μαΐου 2021 (Αργία)

Παρόλο αυτά, αρκετές είναι οι SRC_MACHINE που παρουσιάζουν παρόμοιες συμπεριφορές μεταξύ των τριών επιλεγμένων ημερών καθώς, ειδικά στην τρέχουσα περίοδο του κορονοϊού, δεν υφίστανται αργίες και Σαββατοκύριακα για τους νοσοκομειακούς εργαζόμενους. Ενδεικτικά παραδείγματα είναι τα τμήματα των επειγόντων, του γαστρεντερολογικού και της

παθολογικής. Μαζί με τα προηγούμενα, είναι και η γραμματεία με τα εξωτερικά ιατρεία που έχουν μεγάλο όγκο εξερχόμενων ροών που κατευθύνονται προς την υπηρεσία του HIS.

Ποσοτική Έρευνα

Στο Σχήμα 4.18 απεικονίζεται ο αριθμός των ροών ανά συσκευή (SRC_MACHINE). Στο σύνολο δεδομένων υπάρχουν συνολικά 91 μηχανήματα πηγής, εκ των οποίων τα δημοφιλέστερα 55 οπτικοποιούνται προκειμένου να επιτύχουν σαφή και ορατά αποτελέσματα στο διάγραμμα. Οι υπόλοιπες 36 κατηγορίες επισημάνθηκαν ως «Other» και αποτελούν λιγότερο από το 4% των συνολικών ροών.



Σχήμα 4.18: Αριθμός ροών ανά κατηγορία συσκευής για όλα τα δεδομένα

Αξίζει να σημειωθεί ότι κατηγορία «server» απορρίφθηκε και δεν λήφθηκε υπόψη στην συγκεκριμένη έρευνα, καθώς, όπως αναφέρθηκε και προηγουμένως, αποτελείται κυρίως από ροές DNS, οι οποίες δεν μεταφέρουν άμεσα πληροφορίες σχετικά με τη δικτυακή συμπεριφορά των χρηστών. Παρόλο αυτά, πρέπει να τονιστεί ότι η συγκεκριμένη κατηγορία είχε κατά πολύ τον μεγαλύτερο αριθμό ροών και σχεδόν διπλάσιο από το δεύτερο που είναι το «user desktop». Από τις κλινικές, παρατηρείται ότι οι πιο δημοφιλείς είναι η χειρουργική, τα επείγοντα και τα εξωτερικά ιατρεία.

Κεφάλαιο 5

Συσταδοποίηση Χρηστών

5.1 Ορισμοί - Παραδοχές

Στην συγκεκριμένη ενότητα γίνεται μια πρώτη προσπάθεια συσταδοποίησης των χρηστών - συσκευών του νοσοκομείου σχετικά τόσο με κάποια χαρακτηριστικά ροών τους, όσο και με την υπηρεσία που χρησιμοποιούν. Ο **σκοπός** της συσταδοποίησης αυτής είναι η ανάλυση όμοιων και διαφορετικών δικτυακών συμπεριφορών διαφόρων χρηστών και ο έλεγχος της ορθότητας των κατηγοριών SRC_MACHINE που προέκυψαν από την προηγούμενη επεξεργασία μέσω των DHCP logs. Προτού όμως αναλυθεί η διαδικασία της συσταδοποίησης, πρέπει να επισημανθούν ορισμένες παραδοχές που έγιναν τόσο σε επίπεδο χρηστών, όσο και σε επίπεδο υπηρεσιών.

Η έννοια **χρήστης** που θα χρησιμοποιείται από εδώ και στο εξής θα αναφέρεται σε μια μοναδική νοσοκομειακή συσκευή για ένα ολόκληρο οχτάωρο (βάρδια). Γίνεται η υπόθεση ότι δεν θα αλλάζει ο χρήστης της συσκεύης μέσα στο οχτάωρό του. Θεωρήθηκε λοιπόν ότι οι χρήστες μπορούν να αλλάζουν μόνο ανά βάρδια, δηλαδή ότι μπορεί να χρησιμοποιήσει διαφορετικός άνθρωπος την ίδια συσκευή αφού αλλάξει η βάρδιά του. Αυτή η παραδοχή έπρεπε να γίνει προκειμένου να μελετηθούν και οι διαφορετικές συμπεριφορές ανάλογα με το οχτάωρο του κάθε ενός χρήστη.

Όσον αναφορά τις **υπηρεσίες**, η συσταδοποίηση δεν θα μπορούσε να εφαρμοστεί σε όλα τα δεδομένα συνολικά, καθώς αυτό θα ήταν υπερβολικά περίπλοκο, δυσνόητο και δεν θα οδηγούσε σε σαφή συμπεράσματα. Επομένως, έγινε διαχωρισμός των χρηστών ανάλογα με την υπηρεσία που χρησιμοποιούν. Μερικές ειδικές υπηρεσίες που χρησιμοποιήθηκαν και αξίζει να επισημανθούν περαιτέρω είναι οι εξής:

- HIS: Οι περισσότεροι χρήστες που έχουν πρόσβαση στο πληροφοριακό σύστημα του νοσοκομείου (Hospital Information System), συχνά επικοινωνούν με μια συγκεκριμένη εσωτερική υπηρεσία που ονομάζεται Ασκληπιός. Είναι μια υπηρεσία που διαχειρίζεται διάφορους τομείς του νοσοκομείου όπως αιτήματα παραγγελιών, εγγραφές, κλινικές, επείγοντα περιστατικά, λογιστικά, προμήθειες, εξωτερικά ιατρεία, γραμματεία κτλ. Το HIS ακούει σε δύο διαφορετικές πόρτες την 7778 και την 51001. Η πόρτα 7778 αφορά περισσότερο κίνηση από τα επείγοντα, τα εξωτερικά ιατρεία και τη γραμματεία, ενώ

η 51001 από τις κλινικές, το κτίριο λοιμωδών, το παθολογικό τμήμα και τα αιτήματα παραγγελιών.

- DICOM: Το Digital Imaging and Communications in Medicine (DICOM) είναι το πρότυπο για την επικοινωνία και τη διαχείριση πληροφοριών ιατρικής απεικόνισης και σχετικών δεδομένων. Χρησιμοποιείται συχνότερα για την αποθήκευση και τη μετάδοση ιατρικών εικόνων που επιτρέπουν την ενσωμάτωση ιατρικών συσκευών απεικόνισης όπως σαρωτές, διακομιστές, σταθμοί εργασίας, εκτυπωτές, τομογράφοι, υλικό δικτύου κτλ. Στα δεδομένα που καταγράφηκαν, βρέθηκαν τρεις κατηγορίες συσκευών που στοχεύουν έναν διακομιστή DICOM του νοσοκομείου. Συγκεκριμένα πρόκειται για τομογράφους, ανιχνευτές ακτίνων Χ και διαγνωστικούς σταθμούς εργασίας.

- BMS: Το Σύστημα Διαχείρισης Κτιρίων (Building Management System), γνωστό και ως Σύστημα Αυτοματισμού Κτιρίου (Building Automation System), είναι ένα σύστημα ελέγχου εγκατεστημένο σε κτίρια που ελέγχει και παρακολουθεί τον μηχανικό και ηλεκτρικό εξοπλισμό του κτιρίου, όπως αερισμό, φωτισμό, συστήματα ισχύος, συστήματα πυρκαγιάς και συστήματα ασφαλείας. Στα δεδομένα που καταγράφηκαν, βρέθηκαν δύο κατηγορίες συσκευών που χρησιμοποιούν BMS. Συγκεκριμένα, πρόκειται για διακομιστές και σταθμούς παρακολούθησης εργασίας.

- LIS: Το Laboratory Information System είναι ένα εργαστηριακό σύστημα διαχείρισης πληροφοριών, που υποστηρίζει τις λειτουργίες ενός σύγχρονου εργαστηρίου. Στα δεδομένα που καταγράφηκαν, βρέθηκαν τρείς κατηγορίες συσκευών που χρησιμοποιούν LIS. Συγκεκριμένα, πρόκειται για διακομιστές, σταθμούς εργασίας και αναλυτές.

Τα **χαρακτηριστικά** (πεδία) των ροών στα οποία βασίστηκε η συσταδοποίηση των χρηστών είναι:

- Η χρονική διάρκεια των ροών (**flow duration**).
- Η χρονική διάρκεια μεταξύ διαδοχικών αφίξεων ροών του ίδιου χρήστη (**flow inter-arrival**).
- Τα εισερχόμενα και εξερχόμενα byte που ανταλλάχθηκαν στις ροές (**in/out bytes**).
- Τα εισερχόμενα και εξερχόμενα πακέτα που ανταλλάχθηκαν στις ροές (**in/out packets**).

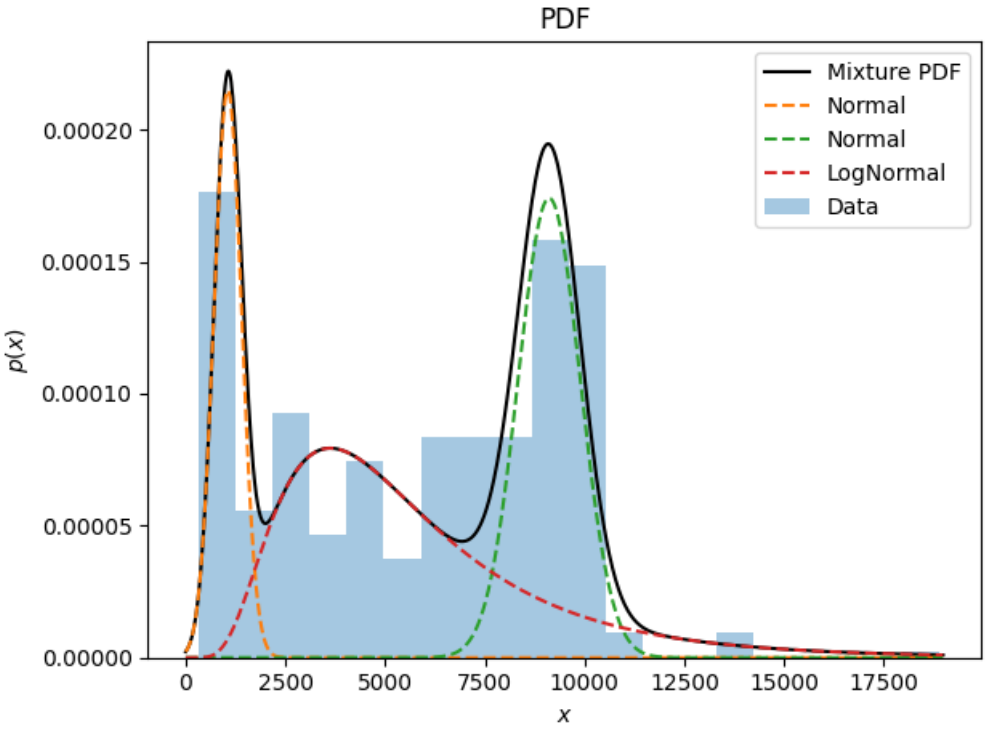
Η εκτίμηση και η αξιολόγηση των συστάδων έγινε προφανώς με βάση τις κατηγορίες συσκευών που προέκυψαν από τα αρχεία του DHCP server. Σκοπός δηλαδή είναι να φανεί πόσο «κοντά», από άποψη δικτυακής συμπεριφοράς, είναι χρήστες-συσκευές του νοσοκομείου που ανήκουν στην ίδια ή σε παρόμοιες κατηγορίες SRC_MACHINE.

Χρησιμοποιήθηκαν δύο διαφορετικοί τρόποι για την συσταδοποίηση. Ο ένας είναι με Μοντέλα Μείξης (Mixture Models) και τον αλγόριθμο K-Μέσων (K-Means), ενώ ο άλλος είναι με ιεραρχική μέθοδο και τη μετρική απόστασης Wasserstein.

5.2 Μέθοδος 1η - Μοντέλα Μείξης

Αρχικά, χρησιμοποιήθηκαν **General Mixture Models** από τη βιβλιοθήκη Pomegranate της Python για να μοντελοποιήσουν την κατανομή των διάρκειων (duration), των χρονικών διαστημάτων αφίξεων (inter-arrivals) και των bytes των ροών κάθε ενός χρήστη (για κάθε βάρδια). Δεν χρησιμοποιήθηκαν τα πακέτα, διότι όπως παρατηρήθηκε και στην προηγούμενη ενότητα της ανάλυσης, έχουν παρόμοια συμπεριφορά με τα bytes. Επειδή δεν ήταν γνωστές εκ των προτέρων οι παράμετροι των κατανομών και έπρεπε να μαθευτούν εξ ολοκλήρου από τα δεδομένα, χρησιμοποιήθηκε η μέθοδος κλάσης «*from_samples*». Οι διαφορετικές κατανομές πιθανοτήτων που χρησιμοποιήθηκαν για τη μοντελοποίηση των δεδομένων είναι:

- 2 Normal Distributions
- 1 Log-Normal Distribution

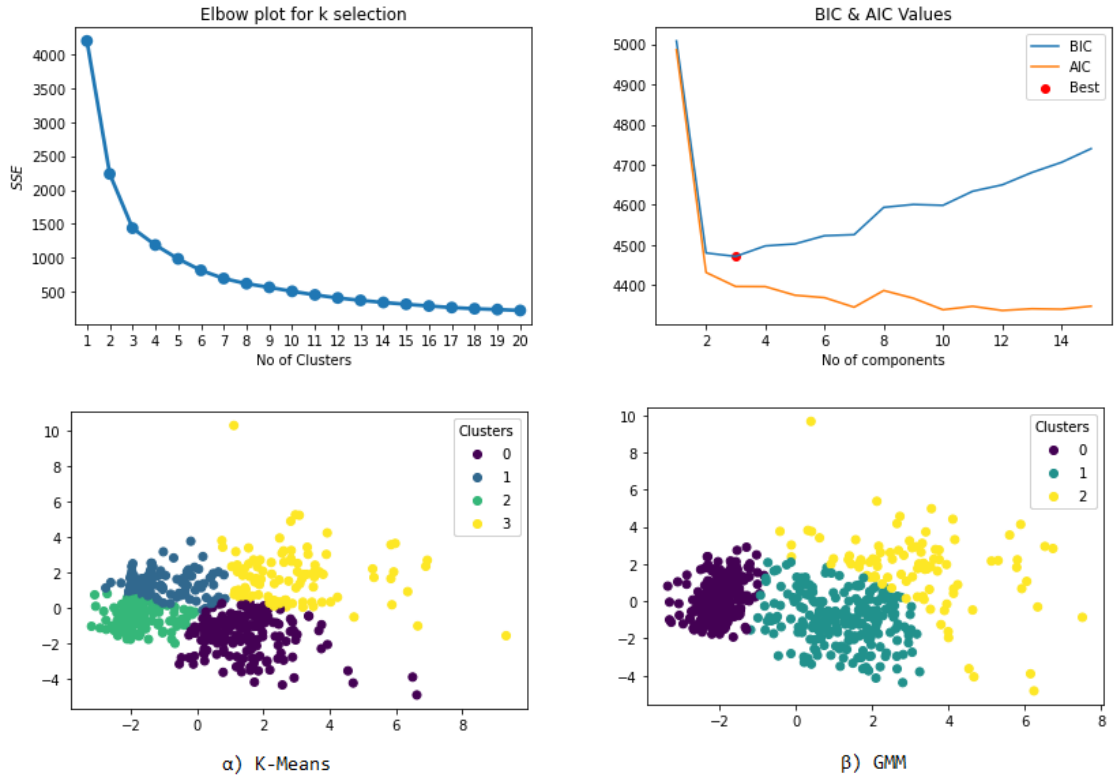


Σχήμα 5.1: Παράδειγμα εκπαιδευμένου μοντέλου μείξης σε κατανομή διαρκειών ροών

Ένα χαρακτηριστικό παράδειγμα εκπαιδευμένου μοντέλου μείξης σε κατανομή διαρκειών ροών παρουσιάζεται στο Σχήμα 5.1. Στη συνέχεια, χρησιμοποιήθηκαν οι μαθημένες παράμετροι του μοντέλου μείξης ως νέα χαρακτηριστικά προκειμένου να συσταδοποιηθούν οι χρήστες. Οι δύο διαφορετικοί αλγόριθμοι που χρισιμοποιήθηκαν είναι ο Gaussian Mixture και ο K-Means από τη βιβλιοθήκη scikit-learn της Python. Για την εύρεση του βέλτιστου αριθμού των συστάδων χρησιμοποιήθηκαν η μέθοδος του αγκώνα και το Bayesian Information Criterion για τον K-Means και το GMM αντίστοιχα.

Δοκιμάστηκε η συσταδοποίηση τόσο με, όσο και χωρίς τη χρήση της ανάλυσης κύριων συνιστωσών (Principal Component Analysis). Παρόλο αυτά, παρατηρήθηκε ότι τα αποτελέσματα ήταν πιο ακριβή όταν εφαρμόστηκε PCA (με n_components = 2) πριν από την συσταδοποίηση, για αυτό και διατηρήθηκε έτσι στην παρακάτω ανάλυση. Όπως αναφέρθηκε και προηγουμένως, για ευκολότερο χειρισμό των δεδομένων και σαφέστερα αποτελέσματα, χωρίστικαν τα δεδομένα ανάλογα με την εφαρμογή-υπηρεσία.

HIS

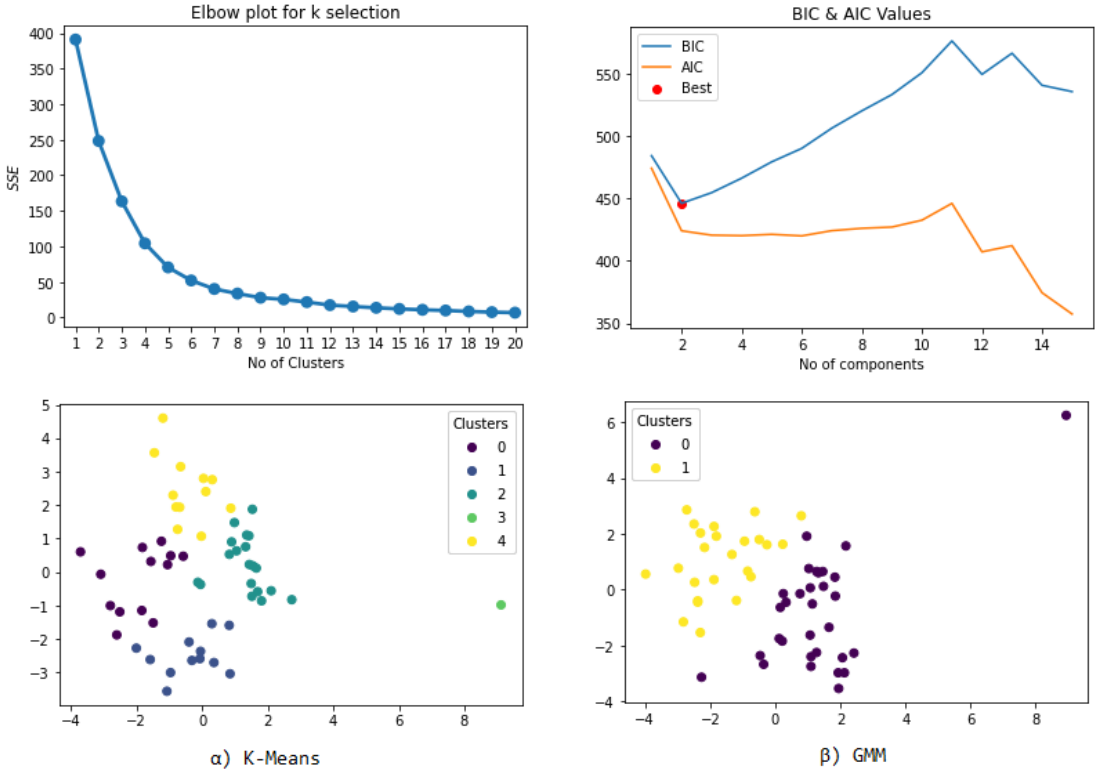


Σχήμα 5.2: Συσταδοποίηση για την υπηρεσία του HIS (Ασκληπιού)

Με βάση τα διαγράμματα του Σχήματος 5.2 παρατηρείται ότι οι δύο αλγόριθμοι έχουν κάνει παρόμοι διαχωρισμό. Η μόνη διαφορά έγκειται στο γεγονός ότι η μέθοδος GMM έχει μια ομάδα λιγότερη (3 αντί για 4) και στην ουσία έχει συγχωνεύσει το cluster 1 και 2 του K-Μεανς σε ένα ενιαίο (το αντίσοιχο 0). Όσο αναφορά τις κατηγορίες συσκευών σε κάθε συστάδα, οι χρήστες στην αριστερή πλευρά του χάρτη απεικόνισης, σχετίζονται περισσότερο με την κίνηση από την παθολογική και το λοιμοδόν και χρησιμοποιούν τη θύρα 51001 του HIS. Αντίθετα, οι χρήστες στη δεξιά πλευρά του χάρτη απεικόνισης, σχετίζονται περισσότερο με την κίνηση από έκτακτα περιστατικά και τη γραμματεία και χρησιμοποιούν τη θύρα 7778 του HIS. Επομένως, γίνεται αντιλυπτό ότι χρήστες της ίδιας κατηγορίας SRC_MACHINE έχουν παρόμοια δικτυακή συμπεριφορά και χρησιμοποιούν την ίδια θύρα της υπηρεσίας του Ασκληπιού. Αυτό είναι ιδιαίτερα σημαντικό για την επαλήθευση της εγκυρότητας αυτών των κατηγοριών.

DICOM

Σύμφωνα με τα διαγράμματα του Σχήματος 5.3 παρατηρείται ότι οι δύο αλγόριθμοι έχουν επιλέξει διαφορετικό βέλτιστο αριθμό συστάδων, χωρίς βέβαια να είναι εμφανής ο «αγκώνας» στον K-Means. Ιδανικά, θα θέλαμε τρεις συστάδες, λόγω του ότι έχουμε τρεις κατηγορίες συσκευών που χρησιμοποιούν DICOM. Παρόλο αυτά ο διαχωρισμός με GMM φαίνεται αρκετά ικανοποιητικός διότι έχει χωρίσει τις συσκευές στις δύο δημοφιλέστες κατηγορίες. Στην δεξιά συστάδα 0 βρίσκονται όλοι οι ανιχνευτές ακτίνων X, ενώ στην αριστερή 1 βρίσκονται όλοι οι τομογράφοι. Οι διαγνωστικοί σταθμοί εργασίας επειδή είναι λίγοι στον αριθμό, παρατηρήθηκαν και στα δύο clusters.

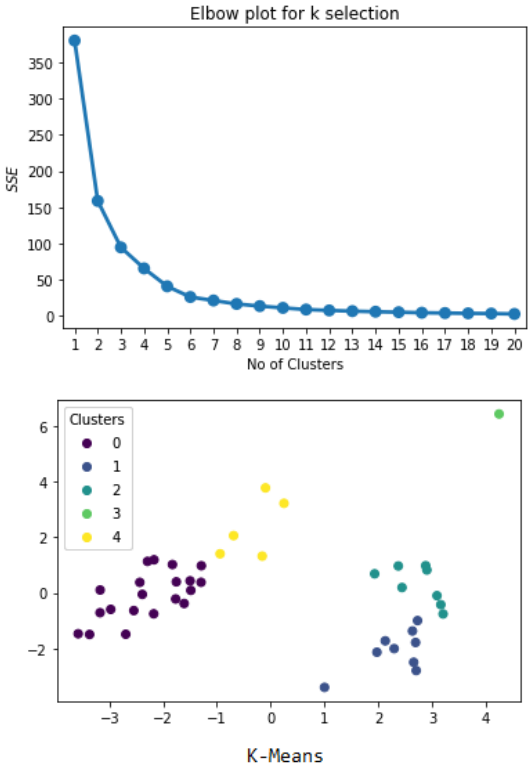


Σχήμα 5.3: Συσταδοποίηση για την υπηρεσία DICOM

LIS

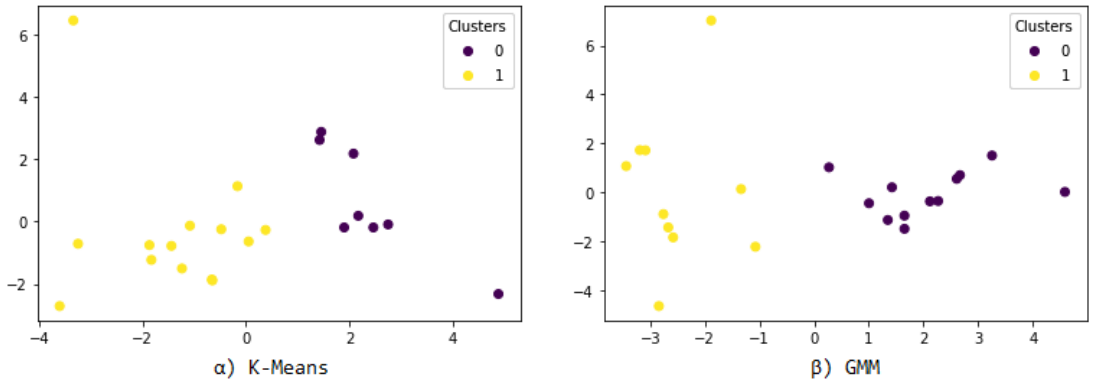
Με βάση τη συνάρτηση γονάτου K-Means που φαίνεται στο Σχήμα 5.4, ο βέλτιστος αριθμός συστάδων είναι πέντε. Στην αριστερή πλευρά του χάρτη (cluster 0 και 4), είναι οι διακομιστές (servers) που χρησιμοποιούν το πρωτόκολλο Oracle. Στη δεξιά πλευρά, παρατηρείται η συστάδα 1 που έχει σταθμούς εργασίας και η συστάδα 2 που έχει περισσότερους αναλυτές και λιγότερους σταθμούς εργασίας. Και οι δύο αυτές ομάδες έχουν κοινό το πρωτόκολλο TLS, σε σύγκριση με τις ομάδες της αριστερής πλευράς στον χάρτη που έχουν το Oracle. Τέλος, υπάρχει ένα απομακρυσμένο cluster, το τρίτο, το οποίο έχει έναν μόνο διακομιστή με το πρωτόκολλο TLS, το οποίο θεωρείται outlier. Συμπερασματικά, παρατηρείται μια

αρκετά καλή ομαδοποίηση χρηστών που χρησιμοποιούν την υπηρεσία LIS, τόσο με βάση την κατηγορία στην οποία ανήκουν, όσο και με το πρωτόκολλο που χρησιμοποιούν.



Σχήμα 5.4: Συσταδοποίηση για την υπηρεσία LIS

BMS



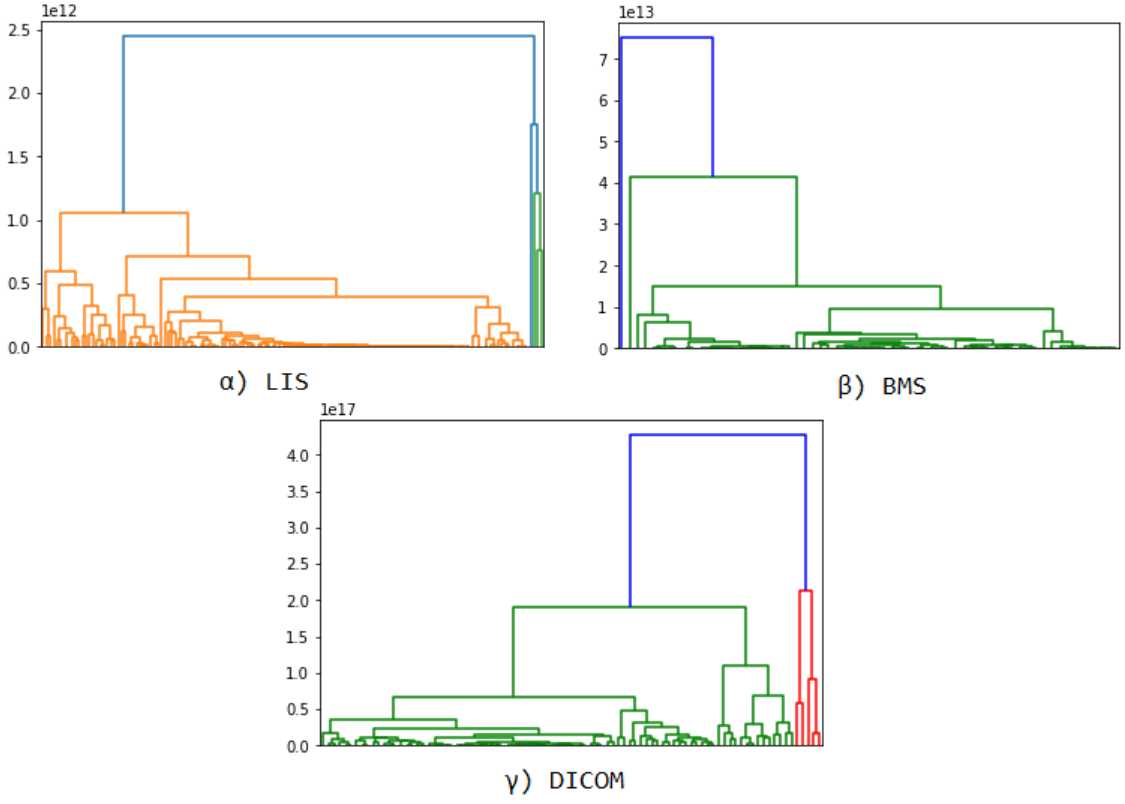
Σχήμα 5.5: Συσταδοποίηση για την υπηρεσία BMS

Επειδή ήταν ήδη γνωστός ο μικρός αριθμός των κατηγοριών που χρησιμοποιούν την υπηρεσία BMS (μόνο διακομιστές και σταθμοί παρακολούθησης εργασίας), προτιμήθηκε να εφαρμοστούν κατευθείαν οι αλγόριθμοι συσταδοποίησης για δύο ομάδες. Ο K-Means κατάφερε να ομαδοποιήσει τους χρήστες αρκετά ικανοποιητικά (μόνο δύο επιλέχθηκαν σε διαφορετικό

cluster), ενώ η μέθοδος Gaussian Mixture πέτυχε τον τέλειο διαχωρισμό των χρηστών σε δύο συστάδες, ανάλογα με την κατηγορία (SRC_MACHINE) που ανήκουν.

5.3 Μέθοδος 2η - Wasserstein Distance

Μια άλλη μέθοδος που αποφασίστηκε να χρησιμοποιηθεί για τη συσταδοποίηση των χρηστών είναι η μετρική Wasserstein (επίσης γνωστή ως Earth Mover's Distance), η οποία είναι μια συνάρτηση απόστασης μεταξύ των κατανομών πιθανοτήτων σε έναν δεδομένο μετρικό χώρο. Οι κατανομές για κάθε χρήστη αποτελούνται από τρισδιάστατα σημεία δεδομένων (διάρκεια ροών, διάστημα μεταξύ αφίξεων ροών, byte ροών) ομοιόμορφα κατανεμημένα προκειμένου να μην εφαρμόζεται προκατάληψη στα αποτελέσματα της απόστασης Wasserstein. Με βάση τους πίνακες απόστασης Wasserstein συσταδοποιούνται οι χρήστες χρησιμοποιώντας τη Συσσωρευτική Ιεραρχική Συσταδοποίηση (Agglomerative Hierarchical Clustering). Προτού γίνει αυτό, οπτικοποιούνται με δενδρογράμματα προκειμένου να βρεθεί ο βέλτιστος αριθμός των συστάδων. Και σε αυτήν τη μέθοδο, τα δεδομένα έχουν χωριστεί ανάλογα με την εφαρμογή-υπηρεσία. Το Σχήμα 5.6 παρουσιάζει τα δενδρογράμματα για τις υπηρεσίες LIS, BMS και DICOM.



Σχήμα 5.6: Δενδρογράμματα για βέλτιστο αριθμό συστάδων

Για την εύρεση του βέλτιστου αριθμού συστάδων σε δενδρογράμματα, βρίσκεται αρχικά η μεγαλύτερη κάθετη γραμμή και στη συνέχεια παρατηρούνται πόσα συμπλέγματα θα δημιουρ-

γηθούν αν «κόψουμε» οριζόντια. Για παράδειγμα, στην υπηρεσία LIS προκύπτουν 2-4, στην υπηρεσία BMS 2-3 και στην υπηρεσία DICOM 2-4 clusters.

Τα αποτελέσματα της συσταδοποίησης με τον παραπάνω τρόπο προέκυψαν σχετικά ικανοποιητικά, ιδιαίτερα όταν οι χρήστες μεταξύ των διαφορετικών κατηγοριών είχαν σημαντική διαφορά στον αριθμό των ροών. Αυτό είναι το πλεονέκτημα και ταυτόχρονα ο περιορισμός της συγκεκριμένης μεθόδου. Δηλαδή, μπορεί να μετρήσει ότι η απόσταση Wasserstein μεταξύ ενός χρήστη με 10000 ροές και ενός άλλου με 100 είναι μεγάλη, αλλά επίσης αυτό μπορεί να προκαλέσει ορισμένες ακραίες τιμές (outliers), οι οποίες να παραμορφώσουν τον αριθμό των χρηστών ανά συστάδα. Για παραδείγματα στο BMS και στο DICOM, η πλειοψηφία των συστάδων έχει πολύ μικρό αριθμό χρηστών, αλλά αυτό δεν είναι απαραίτητα και κακό. Το πρόβλημα δημιουργείται όταν δύο χρήστες με παρόμοιο αριθμό ροών, παρόλο που μπορεί να έχουν μεγάλες διαφορές, οι ακραίες τιμές τους ωθούν στο ίδιο σύμπλεγμα. Εν κατακλείδι, από τις παραπάνω παρατηρήσεις συμπεραίνεται ότι ο αριθμός των σημείων δεδομένων είναι τουλάχιστον εξίσου σημαντικός με τα ίδια τα σημεία δεδομένων.

5.4 Σύγκριση Μεθόδων

Το μεγαλύτερο μειονέκτημα της δεύτερης προσέγγισης έγγυται στη πολυπλοκότητα τόσο του αλγορίθμου όσο και του αριθμού των χρηστών που πρέπει να εξεταστούν. Ο αλγόριθμος που χρησιμοποιήθηκε για να υπολογίσει τη βέλτιστη απόσταση μεταξύ δύο κατανομών έχει πολυπλοκότητα $O(n^3)$, όπου n είναι ο αριθμός των σημείων δεδομένων της μεγαλύτερης κατανομής. Ο αριθμός των χρηστών προσθέτει ένα επιπλέον $O(k2)$, όπου k είναι ο αριθμός των χρηστών. Επομένως, προκύπτει μια συνδυασμένη πολυπλοκότητα $O(k^2 * n^3)$, η οποία για μεγάλες κατανομές, όπως στην συγκεκριμένη περίπτωση, απαιτεί τεράστιο χρόνο.

Συμπερασματικά, με βάση τα αποτελέσματα και των δύο μεθόδων, παρατηρείται ότι οι χρήστες που έχουν την ίδια κατηγορία συσκευής (SRC_MACHINE) έχουν παρόμοια δικτυακή συμπεριφορά μεταξύ τους και άρα συνήθως ομαδοποιούνται στην ίδια συστάδα χρηστών. Ως δικτυακή συμπεριφορά, αξίζει να σημειωθεί ξανά, ότι ορίστικε η διάρκεια των ροών, τα χρονικά διαστήματα αφίξεων ροών, τα bytes και τα πακέτα που ανταλλάχθηκαν στις ροές. Επομένως, πραγματοποιήθηκε και μια επαλήθευση των δεδομένων που λήφθηκαν για αυτές τις κατηγορίες συσκευών από τα logs του DHCP server.

Κεφάλαιο 6

Προφίλ Συμπεριφοράς Χρηστών - Προσομοίωση

Προκειμένου να δημιουργηθούν ρεαλιστικά δεδομένα παρόμοια με αυτά που καταγράφηκαν από το νοσοκομειακό περιβάλλον, απαιτούνταν τα παρακάτω στάδια:

1. Προσδιορισμός των κλάσεων των χρηστών, που θα οδηγήσει σε δεδομένα με ετικέτες (labeled data).

2. Ορισμός προφίλ χρήστη, το οποίο καθορίζεται από την επιλογή των μεταβλητών που θεωρούνται πρότυπα και αντιπροσωπευτικά της συμπεριφοράς του χρήστη.

3. Εξαγωγή των προφίλ από το σύνολο δεδομένων.

4. Μοντελοποίηση προφίλ με παραγωγικά μοντέλα μηχανικής μάθησης με βάση τα δεδομένα καταγραφής.

5. Δημιουργία νέων ρεαλιστικών προφίλ με βάση τη μάθηση των εκπαιδευμένων μοντέλων του προηγούμενου σταδίου.

6. Αξιολόγηση της διαδικασίας προσομοίωσης με στατιστικές μεθόδους, γραφήματα και μετρικές.

6.1 Προσδιορισμός κλάσεων - Κατηγοριοποίηση

Έχοντας ως στόχο την δημιουργία προφίλ χρηστών, ήταν απαραίτητο να υπάρχουν πληροφορίες για τον κάθε τύπο (κατηγορία) χρήστη που αντιστοιχεί σε κάθε συσκευή του δικτύου. Λόγω της ανωνυμοποίησης των δεδομένων απουσιάζουν οι διευθύνσεις IP, επομένως η αντιστοίχιση θα μπορούσε να επιτευχθεί μόνο με στατιστική ανάλυση. Έπειτα από επικοινωνία με ερευνητές και προσωπικό πληροφορικής του νοσοκομείου, λήφθηκαν κάποιοι κανόνες που συνδέουν τις υπηρεσίες που χρησιμοποιεί κάποιος χρήστης και τον ρόλο του στο νοσοκομείο (π.χ. γιατρός, διοικητικό προσωπικό, νοσηλευτής, γραμματεία κτλ).

Οι βασικές υπηρεσίες που αναφέρθηκαν και εν τέλη υπάρχουν και στα δεδομένα καταγραφής είναι οι εξής:

- **HIS:** Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, είναι το κύριο Πληροφοριακό Σύστημα Νοσοκομείου (Hospital Information System), έχει ιδιωτική (private) IP και είναι προσβάσιμο μόνο από το εσωτερικό δίκτυο. Ακούει σε δύο πόρτες, την 7778 και την 51001. Συγκεκριμένα, η πόρτα 51001 ικανοποιεί αιτήματα κλινικών, παραγγελιών κτλ, επομένως είναι κατά κύριο λόγο προσβάσιμη από γιατρούς και νοσηλευτές. Από την άλλη πλευρά, η πόρτα 7778 ικανοποιεί θέματα του λογιστηρίου, των προμηθειών, της διαχείρισης κτλ, επομένως είναι κυρίως προσβάσιμη από τη διοίκηση και τη γραμματεία.

- **Προμηθέας:** Το promitheus.gov και το e-procurement.gov είναι οι κύριοι ιστότοποι που χρησιμοποιούνται για ιατρικές προμήθειες. Είναι προσβάσιμοι μόνο από τη διοίκηση και κυρίως από το τμήμα προμηθειών.

- **Γαληνός**: Ο galinos είναι ένας ιστότοπος που χρησιμοποιείται ως φαρμακευτικός οδηγός. Είναι κυρίως προσβάσιμος από φαρμακοποιούς και γιατρούς.

- **Eopyy:** Ο eopyy.gov είναι ένας ιστότοπος του Εθνικού Οργανισμού Υγείας και χρησιμοποιείται για πολλές ιατρικές ενέργειες τόσο των γιατρών και των νοσηλευτών, όσο και της διοίκησης.

- **VPN Πανεπιστήμιο Θεσσαλίας:** Το συγκεκριμένο vpn μπορεί να χρησιμοποιηθεί μόνο από υπαλλήλους γραμματείας, φοιτητές του πανεπιστημίου Θεσσαλίας και ορισμένους γιατρούς.

- **Ιστότοποι Διαχείρισης:** Το τμήμα διαχείρισης στοχεύει συγκεκριμένες ιστοσελίδες που σχετίζονται με θέματα λογιστικής και διαχείρισης, όπως το idika, apografi.gov, idika.org/EfkaServices, ebaby.ypes και diavgeia.gov.

Με βάση τις παραπάνω πληροφορίες για τις υπηρεσίες και τον ρόλο του κάθε χρήστη, προκύπτει εν τέλη ο συγκεντρωτικός Πίνακας 6.1.

Services / Categories	Asklipios (51001)	Asklipios (7778)	Promitheus	Galinos	Admin Sites	e-Prescription	VPN	Eopyy
Doctor								
Pharmacist								
Nurse								
Administration								
VPN Users								

Πίνακας 6.1: Κατηγορίες - Υπηρεσίες

Μια περαιτέρω διερεύνηση των υπηρεσιών και των χρηστών, οδήγησε στο συμπέρασμα ότι πρέπει να δημιουργηθούν ορισμένες πρόσθετες κατηγορίες. Συγκεκριμένα, χωρίστηκε η κατηγορία της διοίκησης σε Central Administration και Clinic Administration για προσδιορισμό περισσότερων λεπτομερειών. Επίσης, δημιουργήθηκε η Doctor/Clinic Administration για ορισμένους ασαφείς χρήστες. Επομένως, οι τελικές επτά (7) κατηγορίες παρουσιάζονται παρακάτω:

- Doctor
- Pharmacist
- Nurse
- Central Administration
- Clinic Administration
- Doctor/Clinic Administration
- VPN Users

Οι κανόνες που χρησιμοποιήθηκαν για την κατηγοριοποίηση των χρηστών στις παραπάνω επτά κλάσεις αριθμούνται παρακάτω:

1. Η κίνηση στο E-prescription γίνεται μόνο από ιατρούς.
2. Η κίνηση στον Προμηθέα διενεργείται μόνο από τη διοίκηση.
3. Η κίνηση που περιορίζεται μόνο στον Γαληνό εκτελείται μόνο από φαρμακοποιούς.
4. Η κίνηση που περιορίζεται σε ιστότοπους διαχειριστή (apografi, eservices.yeka, idika/e-fkaservices κτλ.) πραγματοποιείται μόνο από τη διοίκηση.
5. Η επισκεψιμότητα που περιλαμβάνει αλλά δεν περιορίζεται σε ιστοτόπους διαχείρισης και εξαιρουμένων των HIS, Γαληνό, Προμηθέα, E-prescription, VPN πραγματοποιείται από τη διοίκηση.
6. Η κίνηση που περιορίζεται στο VPN Πανεπιστήμιο Θεσσαλίας εκτελείται από χρήστες VPN που δυνητικά μπορεί να είναι γραμματείς κλινικών, πανεπιστημιακοί γιατροί ή φοιτητές. Ωστόσο, δεν μπορούν να συλλεχθούν περαιτέρω πληροφορίες σχετικά με αυτά, καθώς αλληλεπιδρούν μόνο μέσω VPN. Ο κανόνας 7 επιθεωρεί περαιτέρω αυτήν την ταξινόμηση για αυτούς που επίσης αλληλεπιδρούν με άλλες υπηρεσίες.
7. Η επισκεψιμότητα, συμπεριλαμβανομένης του VPN Πανεπιστημίου Θεσσαλίας μπορεί να εκτελείται από διαφορετικές κατηγορίες, συγκεκριμένα ιατρός ή διοίκηση κλινικής. Ως εκ τούτου, οι σχετικές ροές διερευνήθηκαν περαιτέρω στατιστικά με βάση το πεδίο SRC_MACHINE.
8. Η κίνηση που περιορίζεται σε μία πόρτα του HIS, διενεργείται από τη διοίκηση (πόρτα 7778) ή από νοσηλευτές (πόρτα 51001).
9. Η κίνηση που περιορίζεται και στις δύο πόρτες του HIS μπορεί να διαφέρει μεταξύ των κατηγοριών του κανόνα 8 και ως εκ τούτου διερευνήθηκε περαιτέρω με βάση τα στατιστικά στοιχεία κίνησης ως εξής:
    - Αριθμός ροών (πόρτα 51001) $>$ 2 x Αριθμός ροών (πόρτα 7778) $\rightarrow$ Νοσηλευτής

- Αριθμός ροών (πόρτα 7778) > 2 x Αριθμός ροών (πόρτα 51001) → Διοίκηση

10. Η κίνηση που περιλαμβάνει αλλά δεν περιορίζεται στις πόρτες του HIS μπορεί να αναφέρεται σε διαφορετικές κατηγορίες. Συγκεκριμένα διαχείριση (που χρησιμοποιούσε ως επί το πλείστον διοικητικές υπηρεσίες και SRC_MACHINE διαχειριστής φύσης), γιατρός (που χρησιμοποιούσε ως επί το πλείστον υπηρεσίες γιατρού και SRC_MACHINE ιατρικής φύσης), ή γιατρός/διοίκηση (που χρησιμοποιούσε εξίσου ιατρικές και διοικητικές υπηρεσίες και το SRC_MACHINE το επικύρωσε).

11. Η επισκεψιμότητα, συμπεριλαμβανομένου του Γαληνού και εξαιρουμένων του HIS, του Προμηθέα, του VPN και του E-prescription μπορεί να πραγματοποιήθηκε από γιατρό, γιατρό/διοίκηση ή διοίκηση και γι αυτό διερευνήθηκε περαιτέρω χειροκίνητα σύμφωνα με το SRC_MACHINE και τις υπηρεσίες που χρησιμοποιούνται πιο συχνά.

12. Η κίνηση συμπεριλαμβανομένης του Γαληνού και του HIS και εξαιρουμένου του Προμηθέα, του VPN και του E-prescription μπορεί να πραγματοποιηθεί από γιατρό ή διοίκηση και ως εκ τούτου διερευνήθηκε περαιτέρω βάσει στατιστικών επισκεψιμότητας ως εξής:

    - Αριθμός ροών (πόρτα 51001) > 2 x Αριθμός ροών (πόρτα 7778) → Ιατρός
    - Αριθμός ροών (πόρτα 7778) > 2 x Αριθμός ροών (πόρτα 51001) → Διοίκηση
    - Αλλιώς → Ιατρός/Διοίκηση

13. Η κίνηση συμπεριλαμβανομένης του HIS και εξαιρουμένων του Γαληνού, του E-prescription, του Προμηθέα και του VPN μπορεί να εκτελεστεί από γιατρό ή διοίκηση και ως εκ τούτου διερευνήθηκε περαιτέρω βάσει στατιστικών επισκεψιμότητας ως εξής:

    - Αριθμός ροών (πόρτα 51001) > 2 x Αριθμός ροών (πόρτα 7778) → Ιατρός
    - Αριθμός ροών (πόρτα 7778) > 2 x Αριθμός ροών (πόρτα 51001) → Διοίκηση
    - Αλλιώς → Ιατρός/Διοίκηση

14. Η κίνηση που περιορίζεται στον Eoppy μπορεί να εκτελείται από διαφορετικούς χρήστες (γιατρούς, διοίκησης) και επομένως διερευνήθηκε περαιτέρω χειροκίνητα σύμφωνα με το πεδίο SRC_MACHINE και τις υπηρεσίες που χρησιμοποιούνται πιο συχνά.

15. Η κίνηση συμπεριλαμβανομένης του Eoppy και εξαιρουμένων του HIS, του Γαληνού, του Προμηθέα, του VPN και του E-prescription μπορεί να εκτελείται από διαφορετικές κατηγορίες (ιατρός, διοίκηση) και επομένως διερευνήθηκε περαιτέρω χειροκίνητα σύμφωνα με το πεδίο SRC_MACHINE και τις υπηρεσίες που χρησιμοποιούνται πιο συχνά.

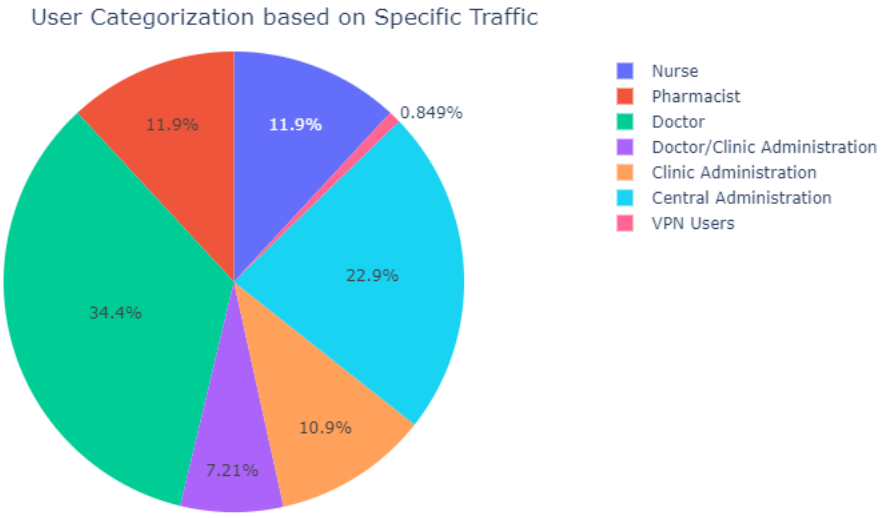
16. Η επισκεψιμότητα από τη διοίκηση επισημάνθηκε περαιτέρω ως διοίκηση κλινικής, εάν εκτελούνταν από μια συσκευή κλινικής (πεδίο SRC_MACHINE).

17. Σε συνδυασμό με τον παραπάνω κανόνα, η κίνηση από τη διοίκηση, εάν το SRC_MACHINE ανήκει σε γραμματεία, λογιστικό γραφείο, ιατρικές προμήθειες κτλ, χαρακτηρίζεται ως κεντρική διοίκηση.

18. Πρόσθετη κατηγοριοποίηση για τους υπόλοιπους εναπομείναντες χρήστες (περίπου 20) πραγματοποιήθηκε χειροκίνητα, ακολουθώντας την εμπειρικούς κανόνες που απορρέουν από τους παραπάνω πίνακες.

19. Για διφορούμενους χρήστες στους οποίους δεν υπήρχαν επαρκή στοιχεία που να τους χαρακτηρίζουν ρητά ως ιατρό ή διοίκηση, δημιουργήθηκε μια επιπλέον ετικέτα ιατρός/διοίκηση κλινικής.

Ο συνολικός αριθμός των χρηστών που έχουν πρόσβαση σε τουλάχιστον μία από τις υπηρεσίες είναι 707. Η κατανομή τους στις παραπάνω επτά κλάσεις παρουσιάζονται στο διάγραμμα του Σχήματος 6.1.



Σχήμα 6.1: Κατανομή χρηστών στις 7 κλάσεις

6.2 Ορισμός Προφίλ Χρήστη

Προκειμένου να επιτευχθούν τα καλύτερα δυνατά αποτελέσματα για την κατανόηση της ανθρώπινη συμπεριφορά στις διαδικτυακές υπηρεσίες, ακολουθήθηκε μια πειραματική προσέγγιση για να καταγραφεί η συσχέτιση μεταξύ των δραστηριοτήτων των χρηστών και των δημιουργούμενων δικτυακών ροών. Σε αυτή την προσπάθεια, διάφορες δραστηριότητες των χρηστών αναπαράχθηκαν και καταγράφηκαν μέσω του nProbe εργαλείου ανιχνευτή δικτυακής ροής. Αυτό είχε ως αποτέλεσμα την παρατήρηση της διαδικασίας καταγραφής της κίνησης σε συνδυασμό με πραγματικές αλληλεπιδράσεις χρηστών.

Με βάση τα αποτελέσματα της παραπάνω πειραματικής προσέγγισης αποφασίστηκε ότι για την επαρκή μοντελοποίηση και ως εκ τούτου προσομοίωση της κίνησης δικτύου (που είναι ο

τελικός σκοπός αυτής της εργασίας), τα χαρακτηριστικά ροής που παίζουν καθοριστικό ρόλο, μαζί με τις υπηρεσίες που χρησιμοποιούνται, είναι η **διάρκεια ροής, η χρονική διάρκεια μεταξύ διαδοχικών αφίξεων ροής και το μέγεθος (bytes) ροής**. Επομένως, αυτές οι τρείς μεταβλητές είναι ιδιαίτερα χρήσιμες για τη μοντελοποίηση της αλληλεπίδρασης ενός χρήστη - συσκευής με μια υπηρεσία. Παρόμοιες προσεγγίσεις μοντελοποίησης έχουν επίσης ακολουθηθεί σε διάφορες συγγενικές έρευνες - εργασίες που αναλύθηκαν στο Κεφάλαιο 1.2, επιβεβαιώνοντας τη χρησιμότητα μιας τέτοιας προσέγγισης και προσομοίωσης.

Αναλόγως με την εξάρτηση ή όχι των τριών αυτών χαρακτηριστικών, ένα προφίλ χρήστη μπορεί να οριστεί είτε ως η κοινή κατανομή στην Εξίσωση 6.2, είτε ως η συλλογή τριών ξεχωριστών πυκνοτήτων στην Εξίσωση 6.3 αντίστοιχα:

$$Profile = p(behaviour|user) \tag{6.1}$$

$$Profile = p(durations, interarrivals, bytes|user) \tag{6.2}$$

$$Profile = (p(durations|user), p(interarrivals|user), p(bytes|user)) \tag{6.3}$$

Ωστόσο, υπάρχει η ανάγκη να μοντελοποιηθούν οι χρήστες σε ομάδες, προκειμένου να μάθουν τη συλλογική δικτυακή συμπεριφορά όλης της ομάδας (πχ. των γιατρών ή των νοσηλευτών κατά τη διάρκεια μιας βάρδιας εργασίας τους σχετική με τη λειτουργία του HIS). Για να επιτευχθεί αυτό, αρκεί να συλλεχθούν κάθε ένα τα προφίλ χρήστη που ικανοποιούν τις απαιτούμενες προϋποθέσεις και να σχηματιστούν έτσι νέες συλλογικές κατανομές για κάθε μεταβλητή συμπεριφοράς.

6.3 Εξαγωγή των Προφίλ

Προκειμένου να μοντελοποιηθούν οι μεταβλητές προφίλ που αναφέρθηκαν προηγουμένως, είναι απαραίτητο να γίνει η κατάλληλη επεξεργασία του συνόλου δεδομένων, ώστε κάθε συσκευή να αντιμετωπίζεται ξεχωριστά με βάση το αναγνωριστικό της ID στο σύνολο δεδομένων. Επίσης, το φιλτράρισμα των ροών που πληρούν τις προϋποθέσεις ενός προφίλ έγινε με βάση τα παρακάτω χαρατηριστικά:

- Υπηρεσία (Service)
- Κλάση - Κατηγορία (Category)
- Κατηγορία Συσκευής (Machine)
- Βάρδια (Shift)

Συγκεντρωτικά, τα προφίλ χρηστών που δημιουργήθηκαν και φαίνονται στον Πίνακα 6.2 είναι τα εξής:

- Προφίλ 1: Αποτελείται από γιατρούς της παθολογικής κλινικής, που επισκέπτονται την υπηρεσία του HIS την πρωινή βάρδια.

Features / Profiles	Service	Category	Machine	Shift
Profile 1	HIS	Doctor	Pathological	1
Profile 2	HIS	Central Administration	Secretariat	2
Profile 3	Promitheus	Central Administration	Supplies	All
Profile 4	LIS	Doctor	Workstation	1
Profile 5	BMS	All	Workstation	3
Profile 6	DICOM	All	Workstation	All
Profile 7	Google	Clinic Administration	Outpatient Clinics	3

Πίνακας 6.2: Προφίλ Χρηστών

- Προφίλ 2: Αποτελείται από διοικητικούς του τμήματος της γραμματείας, που επισκέπτονται την υπηρεσία του HIS την απογευματινή βάρδια.

- Προφίλ 3: Αποτελείται από διοικητικούς του τμήματος των ιατρικών προμηθειών, που επισκέπτονται την υπηρεσία του Προμηθέα όλες (ανεξαρτήτως) τις βάρδιες.

- Προφίλ 4: Αποτελείται από γιατρούς που χρησιμοποιούν το σταθμό εργασίας LIS στην πρωινή βάρδια.

- Προφίλ 5: Αποτελείται από υπαλλήλους (ανεξαρτήτως κατηγορίας), οι οποίοι χρησιμοποιούν το σταθμό εργασίας BMS την βραδινή βάρδια.

- Προφίλ 6: Αποτελείται από υπαλλήλους (ανεξαρτήτως κατηγορίας), οι οποίοι χρησιμοποιούν το σταθμό εργασίας DICOM ανεξάρτητα από τη βάρδια (όλες).

- Προφίλ 7: Αποτελείται από διοικητικούς κλινικών των εξωτερικών ιατρείων, που επισκέπτονται το Google την βραδινή βάρδια.

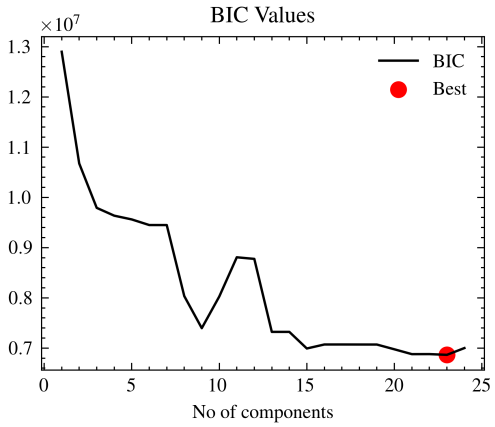
6.4 Μοντελοποίηση Προφίλ Χρηστών

Η μοντελοποίηση με παραγωγικά μοντέλα είναι απαραίτητη στην συγκεκριμένη περίπτωση, καθώς παρέχει τη δυνατότητα παραγωγής - προσομοίωσης νέων παρατηρήσεων με δειγματοληψία από τη μαθημένη κατανομή. Αξίζει να σημειωθεί ότι το χαρακτηριστικό ροής flow size (bytes) είναι χωρισμένο, όπως καταγράφηκε και από το εργαλείο, σε εισερχόμενα (in) και εξερχόμενα (out) bytes. Η προσέγγιση που ακολουθήθηκε χωρίζεται σε δύο μεθόδους:

6.4.1 Μοντελοποίηση ανά χαρακτηριστικό

Η πρώτη μέθοδος θεωρεί τις μεταβλητές των προφίλ, δηλαδή τα χαρακτηριστικά των ροών, **ανεξάρτητα** μεταξύ τους και μοντελοποιεί κάθε χαρακτηριστικό ξεχωριστά, όπως περιγράφηκε μαθηματικά και στην Εξίσωση 6.3. Σε αυτήν την περίπτωση, τα δεδομένα έχουν μία διάσταση, οπότε μπορούν να χρησιμοποιηθούν τα **General Mixture Models** από τη βιβλιοθήκη Pomegranate της Python. Αντίθετα, για multivariate δεδομένα, μόνο οι μείξεις Gaussian κατανομών είναι εφικτές. Έτσι, δίνεται η δυνατότητα να χρησιμοποιηθούν και άλλες κατανομές, εκτός από τις **Normal** και συγκεκριμένα η **Log-Normal** κατανομή, η οποία βιβλιογραφικά εμφανίζεται να προσεγγίζει αρκετά ικανοποιητικά τέτοιου είδους κατανομές.

Το κριτήριο βελτιστοποίησης, που οδηγεί στην επιλογή του καλύτερου συνδυασμού (Normal και Log-Normal κατανομών) και επομένως το μοντέλο με την καλύτερη προσαρμογή στα δεδομένα, ήταν το **Bayesian Information Criterion (BIC)**. Οι χαμηλότερες τιμές BIC δείχνουν καλύτερη προσαρμογή. Οι μείξεις των κατανομών, δηλαδή οι παράμετροι και το βάρος της κάθε συνιστώσας, για κάθε προφίλ και για κάθε χαρακτηριστικό παρατίθενται στο Παράρτημα Α΄.



Σχήμα 6.2: Τιμές BIC για το προφίλ 7

6.4.2 Μοντελοποίηση όλων των χαρακτηριστικών

Η δεύτερη μέθοδος θεωρεί τις μεταβλητές των προφίλ, δηλαδή τα χαρακτηριστικά των ροών, **εξαρτημένα μεταξύ τους** και μοντελοποιεί όλα τα χαρακτηριστικό μαζί, όπως

περιγράφηκε μαθηματικά και στην Εξίσωση 6.2. Σε αυτήν την περίπτωση, τα δεδομένα είναι **multivariate**, έχουν τέσσερις διαστάσεις, οπότε μπορούν να χρησιμοποιηθούν μόνο οι μείξεις Gaussian κατανομών. Στην συγκεκριμένη περίπτωση, χρησιμοποιήθηκε η κλάση **Gaussian Mixture** της βιβλιοθήκης sklearn της Python.

Το κριτήριο βελτιστοποίησης, που οδηγεί στην επιλογή του καλύτερου συνδυασμού Γκαουσιανών και επομένως το μοντέλο με την καλύτερη προσαρμογή στα δεδομένα, ήταν και πάλι το **Bayesian Information Criterion (BIC)**. Στο σχήμα 6.2 παρουσιάζεται η γραφική παράσταση των τιμών BIC όπως προέκυψαν για το προφίλ 7.

Οι μείξεις των κατανομών, δηλαδή οι παράμετροι και το βάρος της κάθε συνιστώσας, για κάθε προφίλ παρατίθενται στο Παράρτημα Β΄.

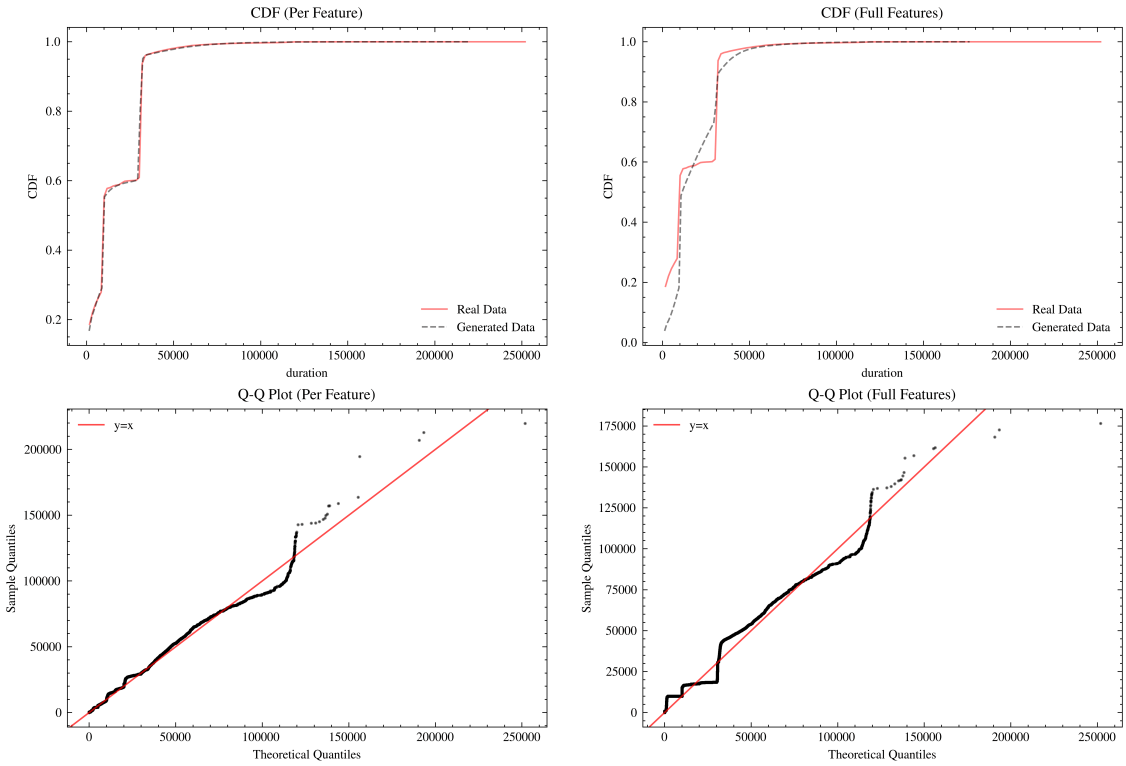
6.5 Προσωμοίωση & Αξιολόγηση Νέων Προφίλ

Με βάση τα παραπάνω αποτελέσματα της εκπαίδευσης των μοντέλων μείξης, δηλαδή των παραμέτρων και των βαρών της κάθε συνιστώσας για κάθε προφίλ, μπορούν να παραχθούν νέα τεχνητά δεδομένα για κάθε μία από τις δύο προσεγγίσεις. Στην πρώτη μέθοδο, παράγωγονται τα δεδομένα ανά στήλη, δηλαδή ξεχωριστά η κατανομή ανά χαρακτηριστικό, ενώ στη δεύτερη παράγωνται τα δεδομένα ανά γραμμή, δηλαδή και τα τέσσερα χαρακτηριστικά μαζί, εξαρτημένα το ένα από το άλλο.

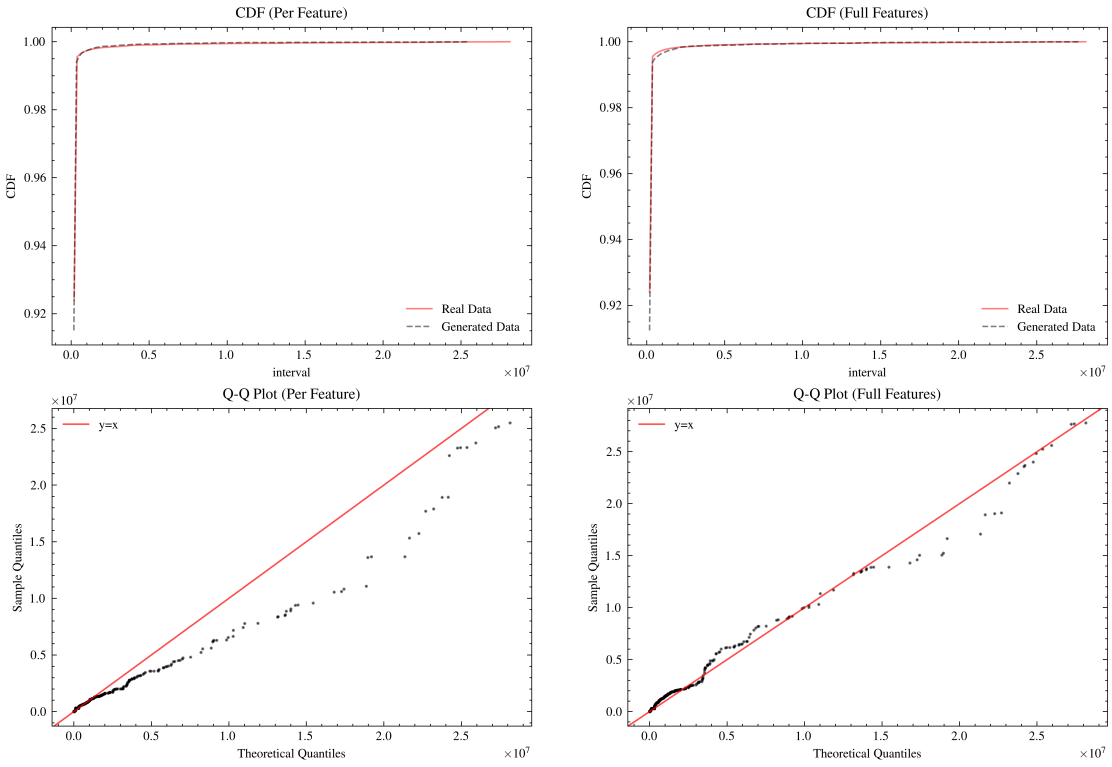
Όμως, για να μπορέσει να εκτιμηθεί το αποτέλεσμα της προσομοίωσης, χρειάζονται ορισμένες μέθοδοι είτε μέσω μετρικών, είτε μέσω οπτικών διαγραμμάτων. Στην παρούσα εργασία, οι τεχνικές που ακολουθήθηκαν για το evaluation των Mixture Models είναι οι εξής:

- CDF Plots
- Q-Q Plots
- Kullback-Leibler Divergence
- Maximum Mean Discrepancy

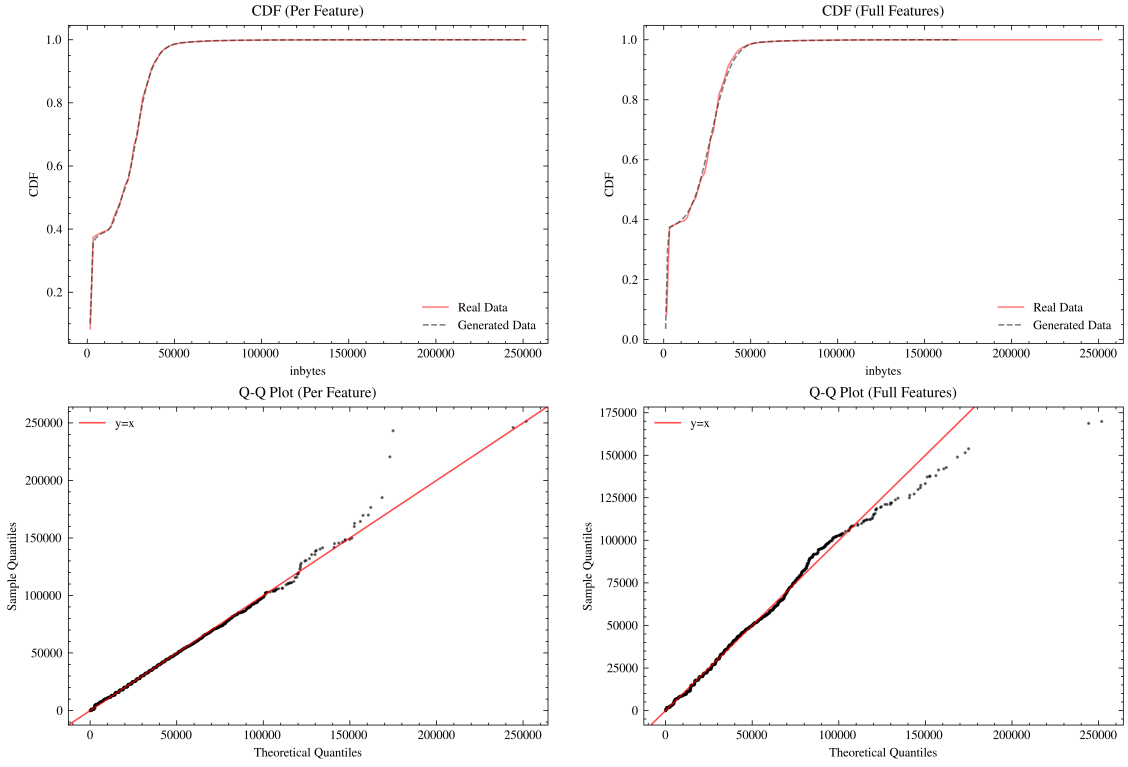
Προφίλ 1 Διαγράμματα



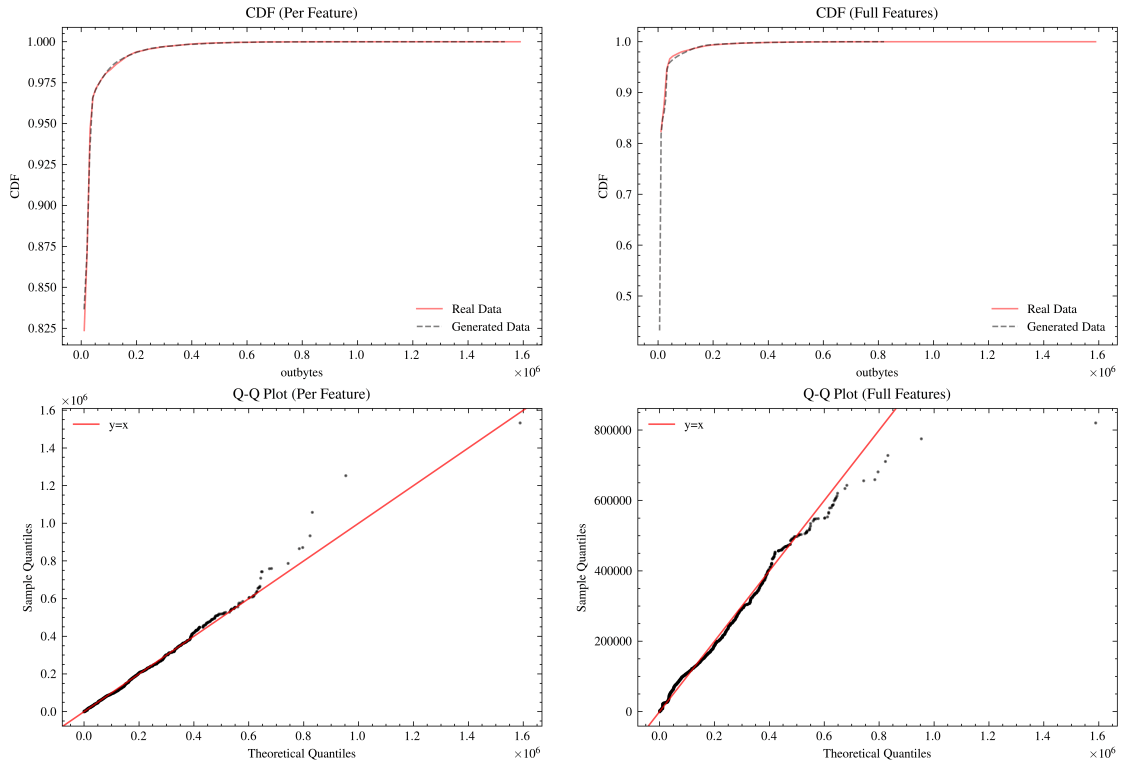
Σχήμα 6.3: Προφίλ 1 Duration



Σχήμα 6.4: Προφίλ 1 Inter-Arrival



Σχήμα 6.5: Προφίλ 1 In-Bytes



Σχήμα 6.6: Προφίλ 1 Out-Bytes

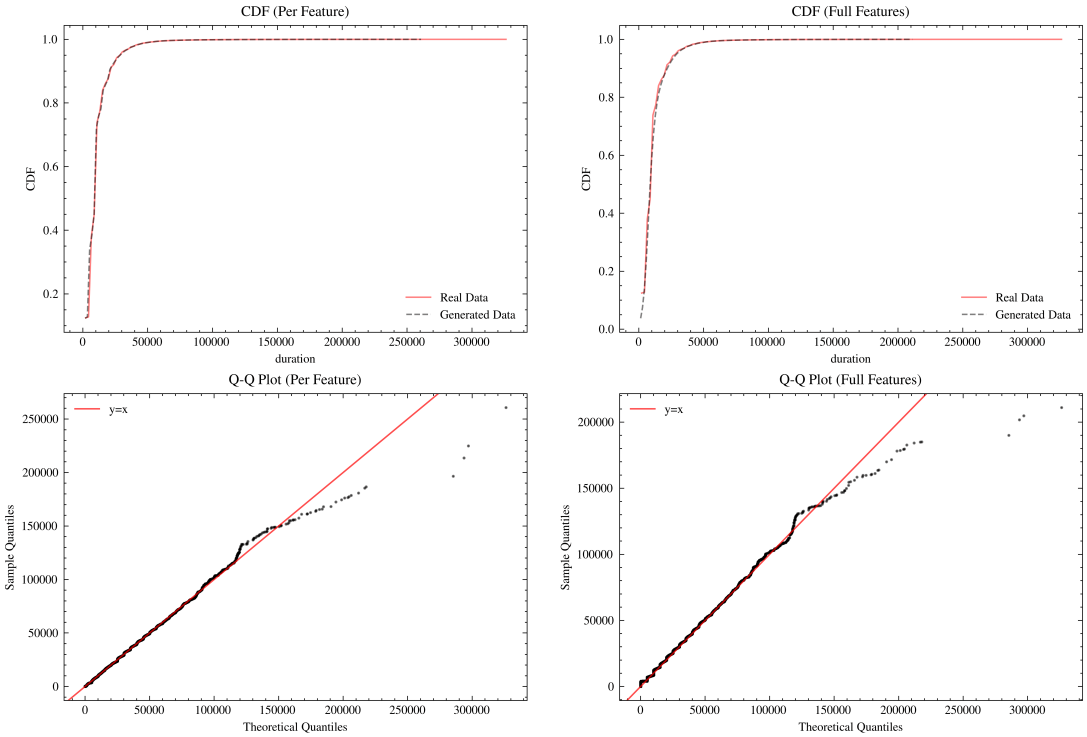
Προφίλ 1 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 0.0031115793 nats
Duration Per Feature KL(Generated || Real): 0.0029838117 nats
Duration Full Features KL(Real || Generated): 0.0030689768 nats
Duration Full Features KL(Generated || Real): 0.0029638182 nats

Interval Per Feature KL(Real || Generated): 1.0235e-06 nats
Interval Per Feature KL(Generated || Real): 1.0255e-06 nats
Interval Full Features KL(Real || Generated): 1.7256e-06 nats
Interval Full Features KL(Generated || Real): 1.7301e-06 nats

Inbytes Per Feature KL(Real || Generated): 4.14362e-05 nats
Inbytes Per Feature KL(Generated || Real): 4.14746e-05 nats
Inbytes Full Features KL(Real || Generated): 0.0002485325 nats
Inbytes Full Features KL(Generated || Real): 0.000248021 nats

Outbytes Per Feature KL(Real || Generated): 7.325e-06 nats
Outbytes Per Feature KL(Generated || Real): 7.3086e-06 nats
Outbytes Full Features KL(Real || Generated): 0.0019995489 nats
Outbytes Full Features KL(Generated || Real): 0.0020635472 nats
```

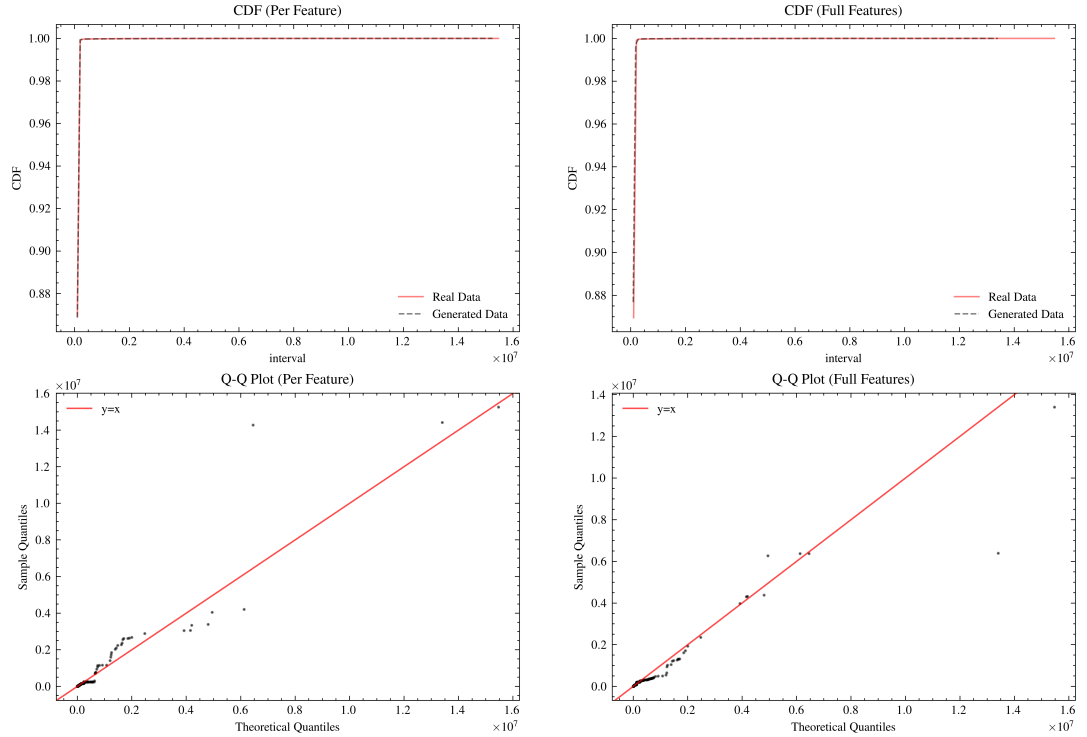
Προφίλ 1 Maximum Mean Discrepancy

```
Profile 1 Per Feature Bootstrapping MMD Mean:      2.017683184814168e+22
Profile 1 Per Feature Bootstrapping MMD Variance: 3.3628250767391385e+44

Profile 1 Full Features Bootstrapping MMD Mean:      1.795658891635674e+22
Profile 1 Full Features Bootstrapping MMD Variance: 9.257915433749856e+44
```

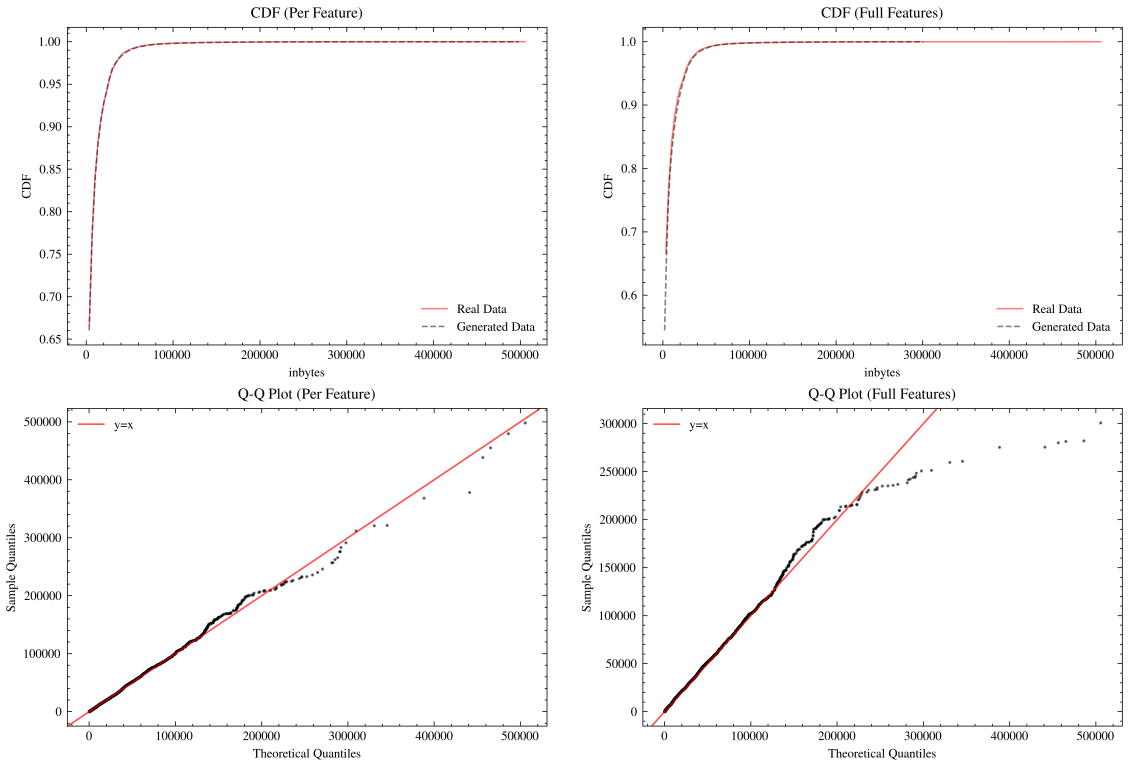
Προφίλ 2 Διαγράμματα



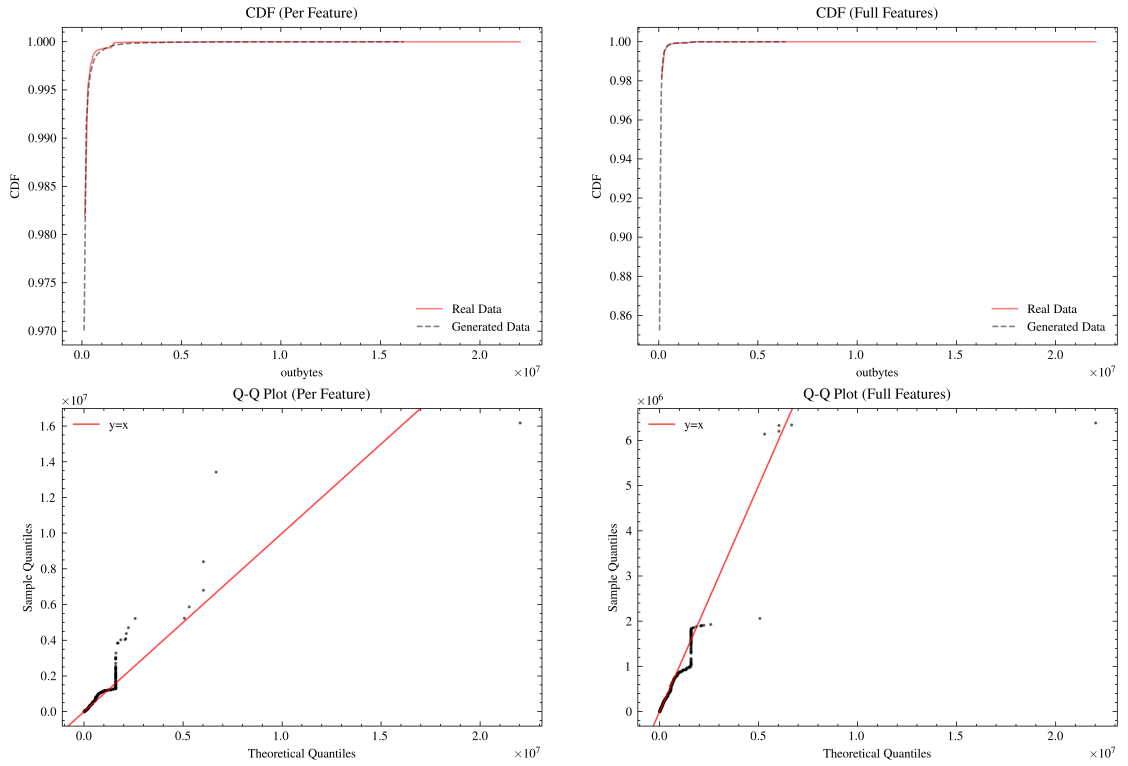
Σχήμα 6.7: Προφίλ 2 Duration



Σχήμα 6.8: Προφίλ 2 Inter-Arrival



Σχήμα 6.9: Προφίλ 2 In-Bytes



Σχήμα 6.10: Προφίλ 2 Out-Bytes

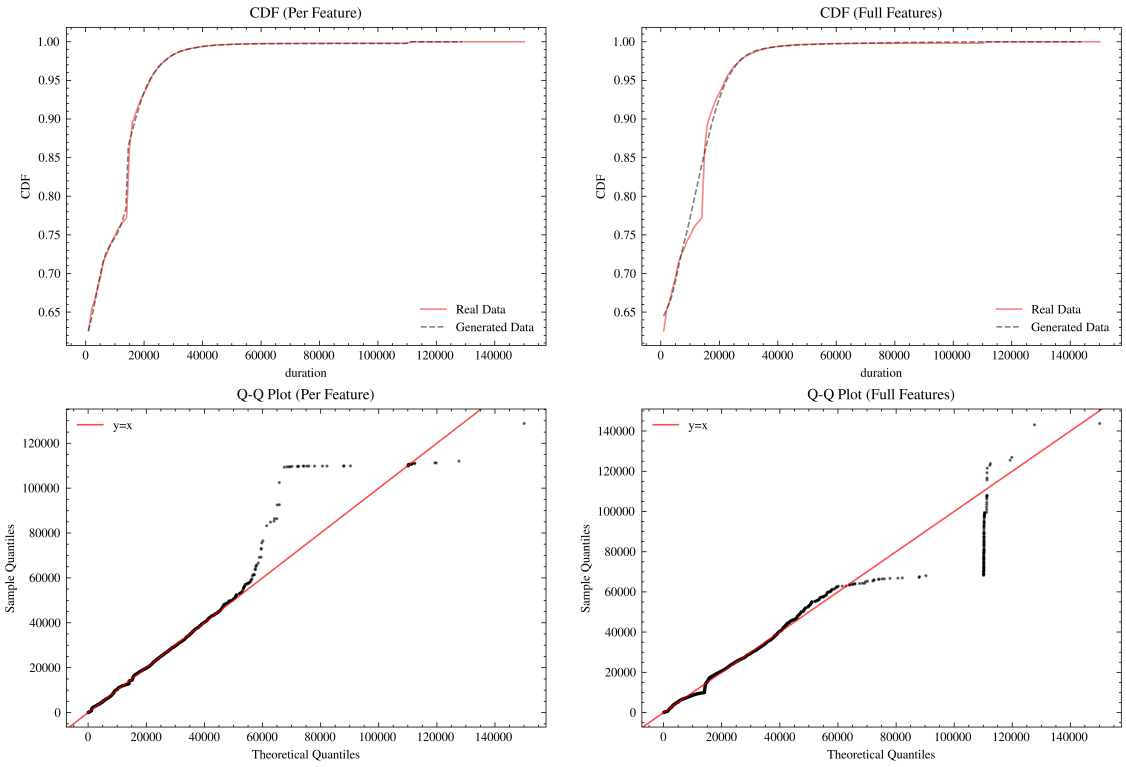
Προφίλ 2 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 0.0011044687 nats
Duration Per Feature KL(Generated || Real): 0.0011186032 nats
Duration Full Features KL(Real || Generated): 0.0009942205 nats
Duration Full Features KL(Generated || Real): 0.0009404235 nats

Interval Per Feature KL(Real || Generated): 1.56e-08 nats
Interval Per Feature KL(Generated || Real): 1.56e-08 nats
Interval Full Features KL(Real || Generated): 1.2059e-06 nats
Interval Full Features KL(Generated || Real): 1.2019e-06 nats

Inbytes Per Feature KL(Real || Generated): 8.654e-07 nats
Inbytes Per Feature KL(Generated || Real): 8.652e-07 nats
Inbytes Full Features KL(Real || Generated): 0.0001007117 nats
Inbytes Full Features KL(Generated || Real): 9.74845e-05 nats

Outbytes Per Feature KL(Real || Generated): 1.5074e-06 nats
Outbytes Per Feature KL(Generated || Real): 1.5102e-06 nats
Outbytes Full Features KL(Real || Generated): 0.0001528339 nats
Outbytes Full Features KL(Generated || Real): 0.0001530978 nats
```

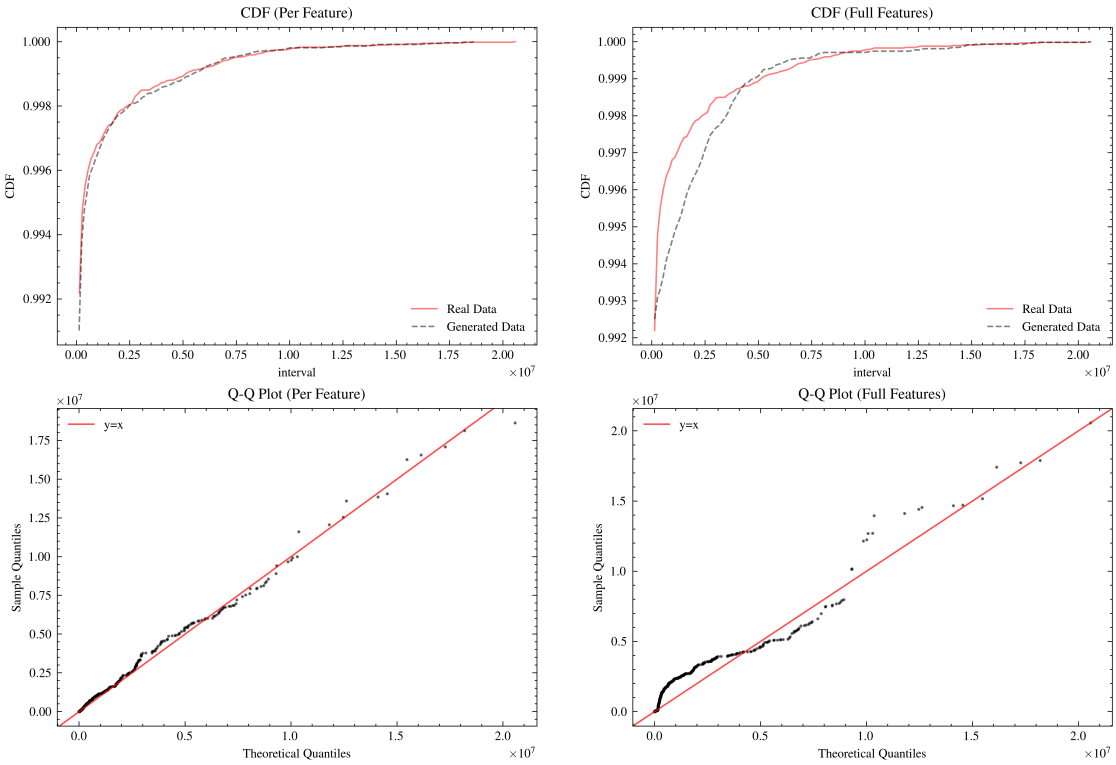
Προφίλ 2 Maximum Mean Discrepancy

```
Profile 2 Per Feature Bootstrapping MMD Mean:      2.4418458279565135e+20
Profile 2 Per Feature Bootstrapping MMD Variance: 2.173133849245799e+41

Profile 2 Full Features Bootstrapping MMD Mean:      6.605324309097 63e+19
Profile 2 Full Features Bootstrapping MMD Variance: 1.3714106031874303e+40
```

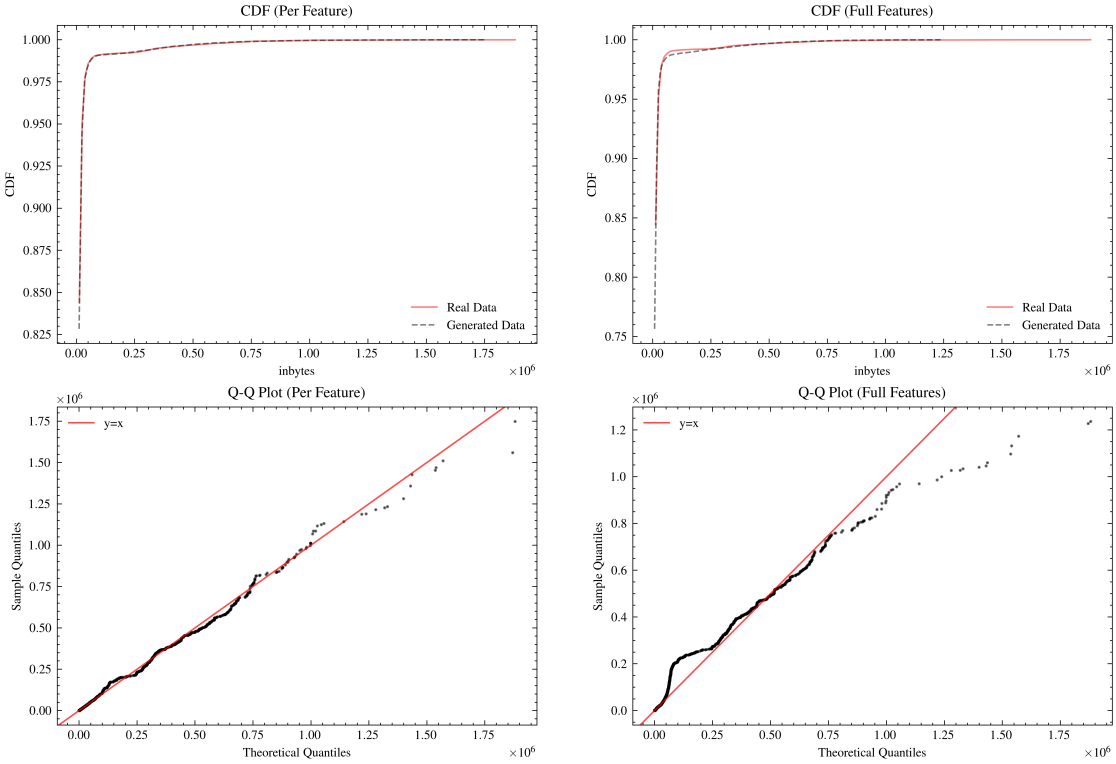
Προφίλ 3 Διαγράμματα



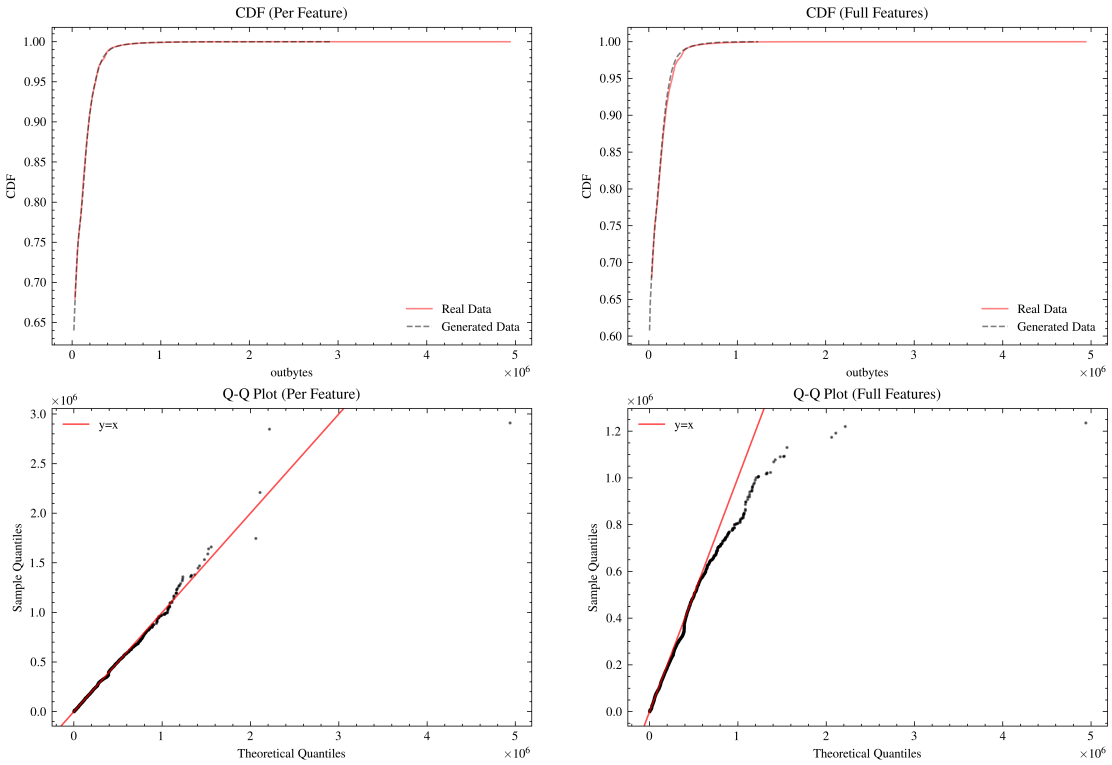
Σχήμα 6.11: Προφίλ 3 Duration



Σχήμα 6.12: Προφίλ 3 Inter-Arrival



Σχήμα 6.13: Προφίλ 3 In-Bytes



Σχήμα 6.14: Προφίλ 3 Out-Bytes

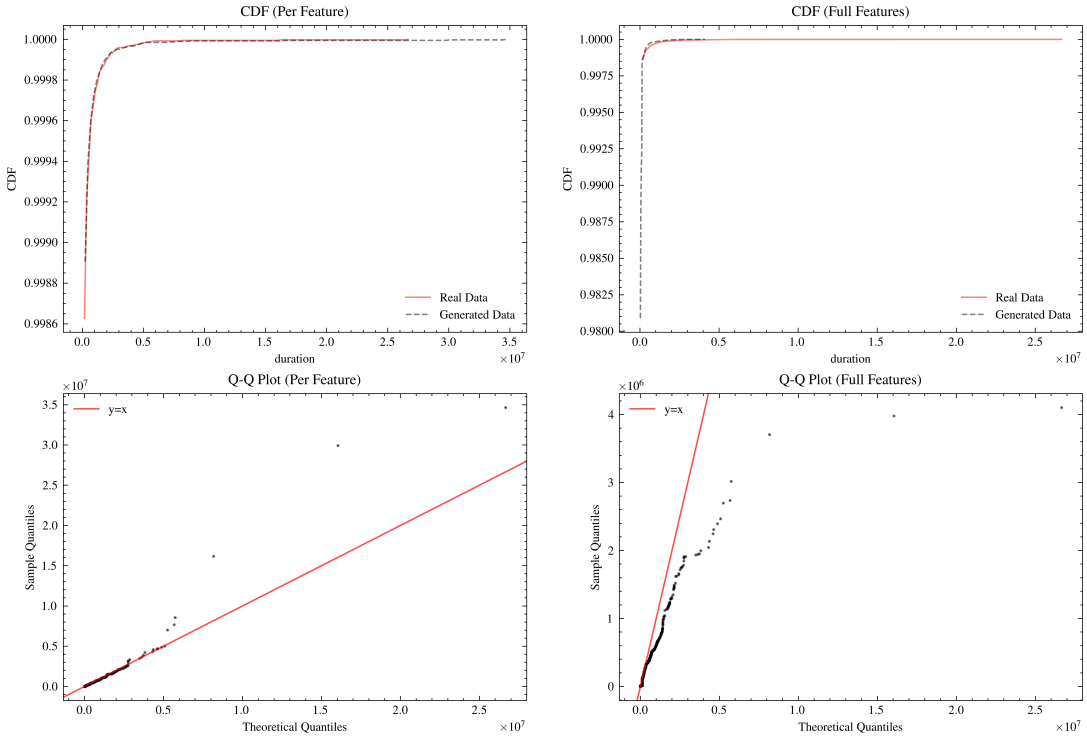
Προφίλ 3 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 0.0001517639 nats
Duration Per Feature KL(Generated || Real): 0.0001504903 nats
Duration Full Features KL(Real || Generated): 8.84819e-05 nats
Duration Full Features KL(Generated || Real): 8.54751e-05 nats

Interval Per Feature KL(Real || Generated): 9.5e-09 nats
Interval Per Feature KL(Generated || Real): 9.5e-09 nats
Interval Full Features KL(Real || Generated): 6.12e-08 nats
Interval Full Features KL(Generated || Real): 6.11e-08 nats

Inbytes Per Feature KL(Real || Generated): 2.2957e-06 nats
Inbytes Per Feature KL(Generated || Real): 2.2946e-06 nats
Inbytes Full Features KL(Real || Generated): 6.76738e-05 nats
Inbytes Full Features KL(Generated || Real): 6.7404e-05 nats

Outbytes Per Feature KL(Real || Generated): 3.79805e-05 nats
Outbytes Per Feature KL(Generated || Real): 3.77214e-05 nats
Outbytes Full Features KL(Real || Generated): 8.91732e-05 nats
Outbytes Full Features KL(Generated || Real): 8.80634e-05 nats
```

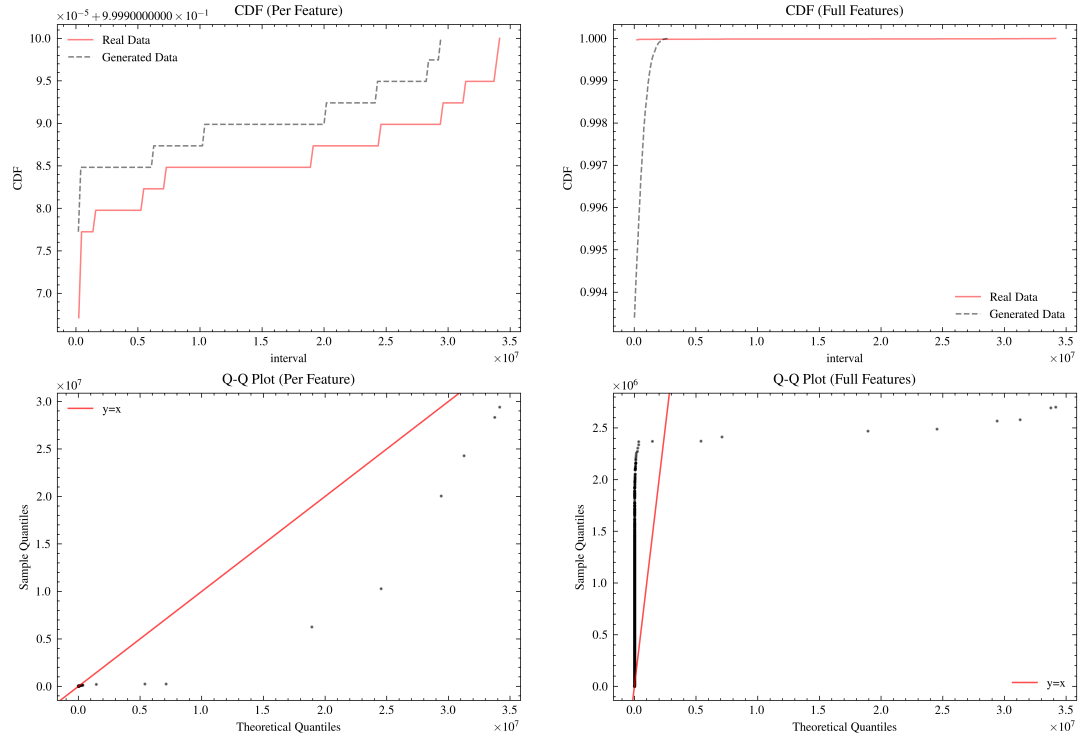
Προφίλ 3 Maximum Mean Discrepancy

```
Profile 3 Per Feature Bootstrapping MMD Mean:      4.3766784945544815e+21
Profile 3 Per Feature Bootstrapping MMD Variance: 5.962749230916159e+43

Profile 3 Full Features Bootstrapping MMD Mean:      8.029894591783679e+21
Profile 3 Full Features Bootstrapping MMD Variance: 2.4186658589652166e+44
```

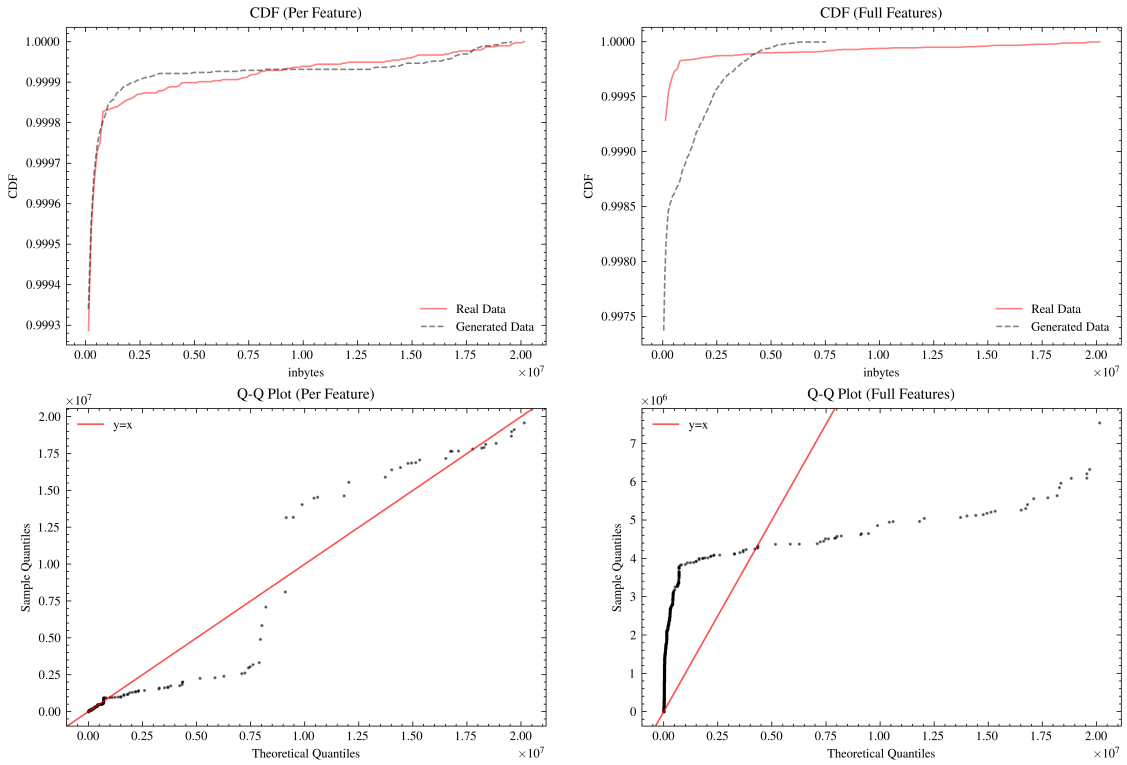
Προφίλ 4 Διαγράμματα



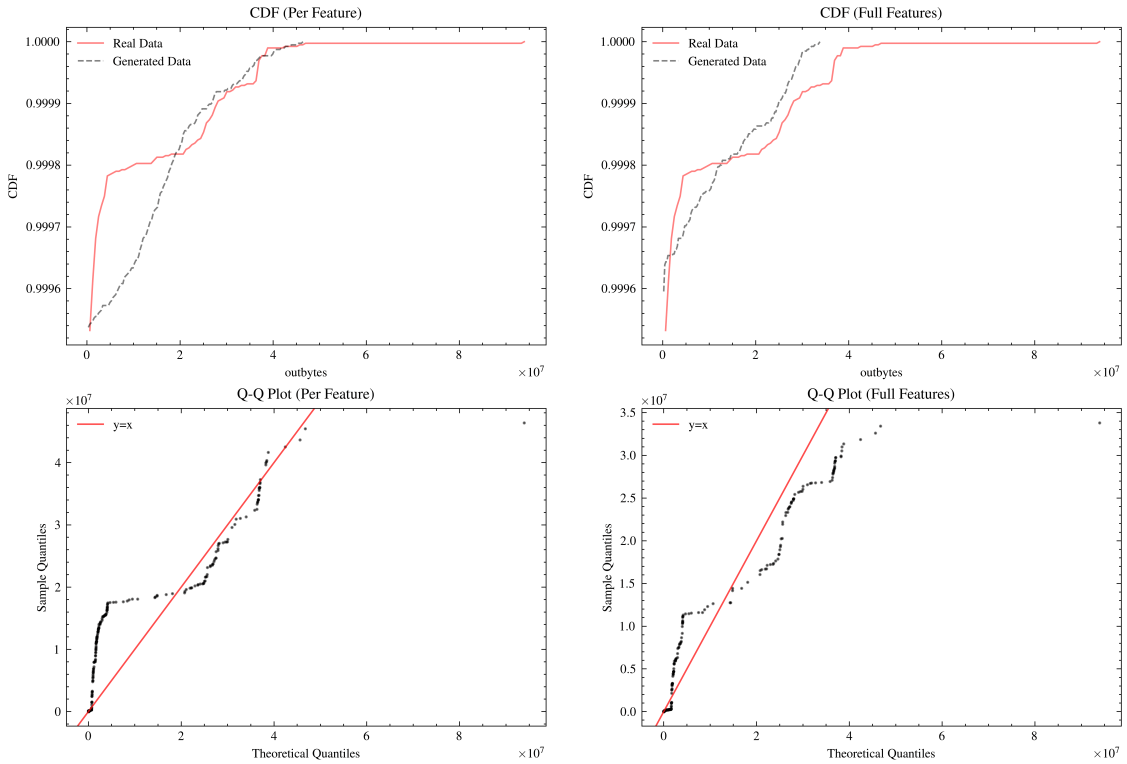
Σχήμα 6.15: Προφίλ 4 Duration



Σχήμα 6.16: Προφίλ 4 Inter-Arrival



Σχήμα 6.17: Προφίλ 4 In-Bytes



Σχήμα 6.18: Προφίλ 4 Out-Bytes

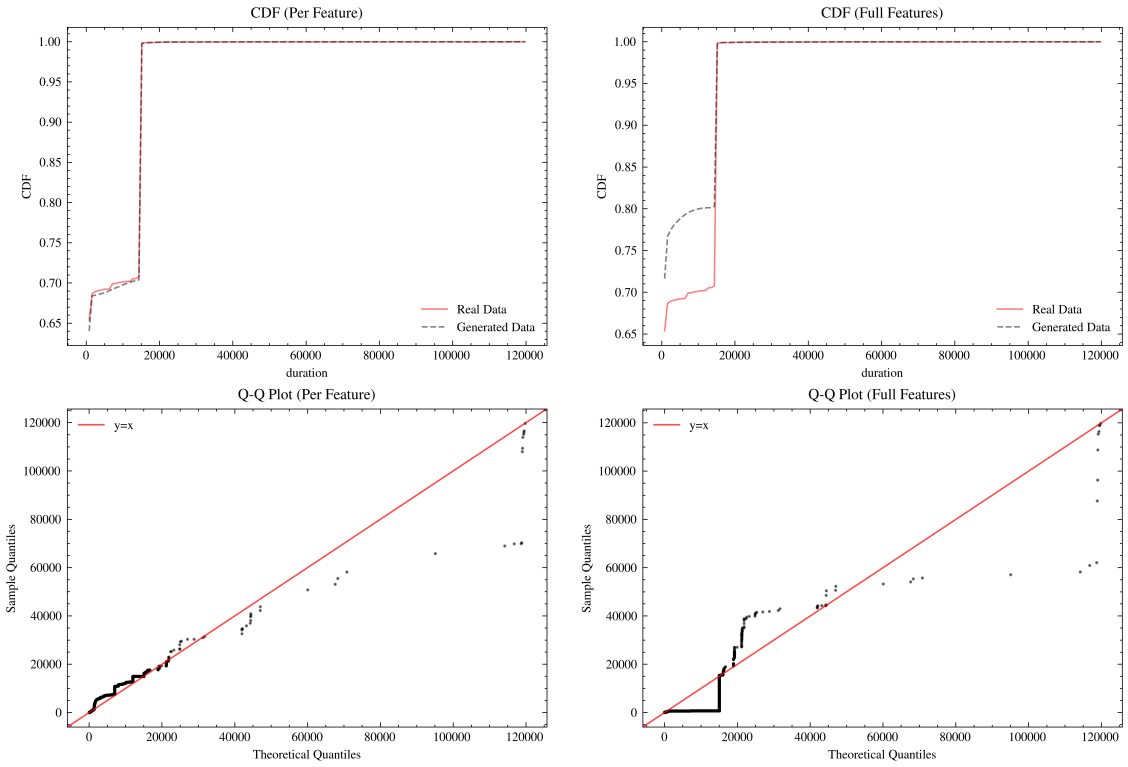
Προφίλ 4 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 6e-10 nats
Duration Per Feature KL(Generated || Real): 6e-10 nats
Duration Full Features KL(Real || Generated): 2.6531e-06 nats
Duration Full Features KL(Generated || Real): 2.6528e-06 nats

Interval Per Feature KL(Real || Generated): 1e-10 nats
Interval Per Feature KL(Generated || Real): 1e-10 nats
Interval Full Features KL(Real || Generated): 1.975e-07 nats
Interval Full Features KL(Generated || Real): 1.972e-07 nats

Inbytes Per Feature KL(Real || Generated): 1e-10 nats
Inbytes Per Feature KL(Generated || Real): 1e-10 nats
Inbytes Full Features KL(Real || Generated): 1.79e-08 nats
Inbytes Full Features KL(Generated || Real): 1.79e-08 nats

Outbytes Per Feature KL(Real || Generated): 2e-10 nats
Outbytes Per Feature KL(Generated || Real): 2e-10 nats
Outbytes Full Features KL(Real || Generated): 2e-10 nats
Outbytes Full Features KL(Generated || Real): 2e-10 nats
```

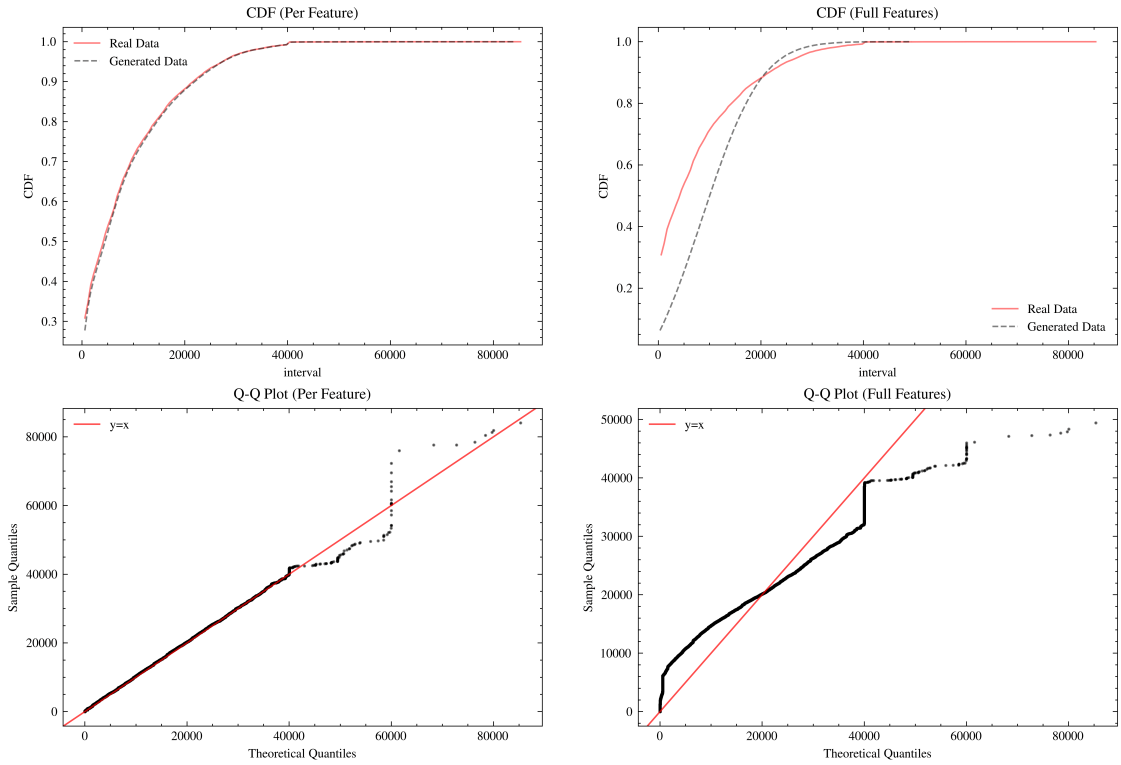
Προφίλ 4 Maximum Mean Discrepancy

```
Profile 4 Per Feature Bootstrapping MMD Mean:     6.138850615921232e+22
Profile 4 Per Feature Bootstrapping MMD Variance: 3.5001678506082515e+45

Profile 4 Full Features Bootstrapping MMD Mean:     3.7590065636204698e+22
Profile 4 Full Features Bootstrapping MMD Variance: 1.1817641351822023e+45
```

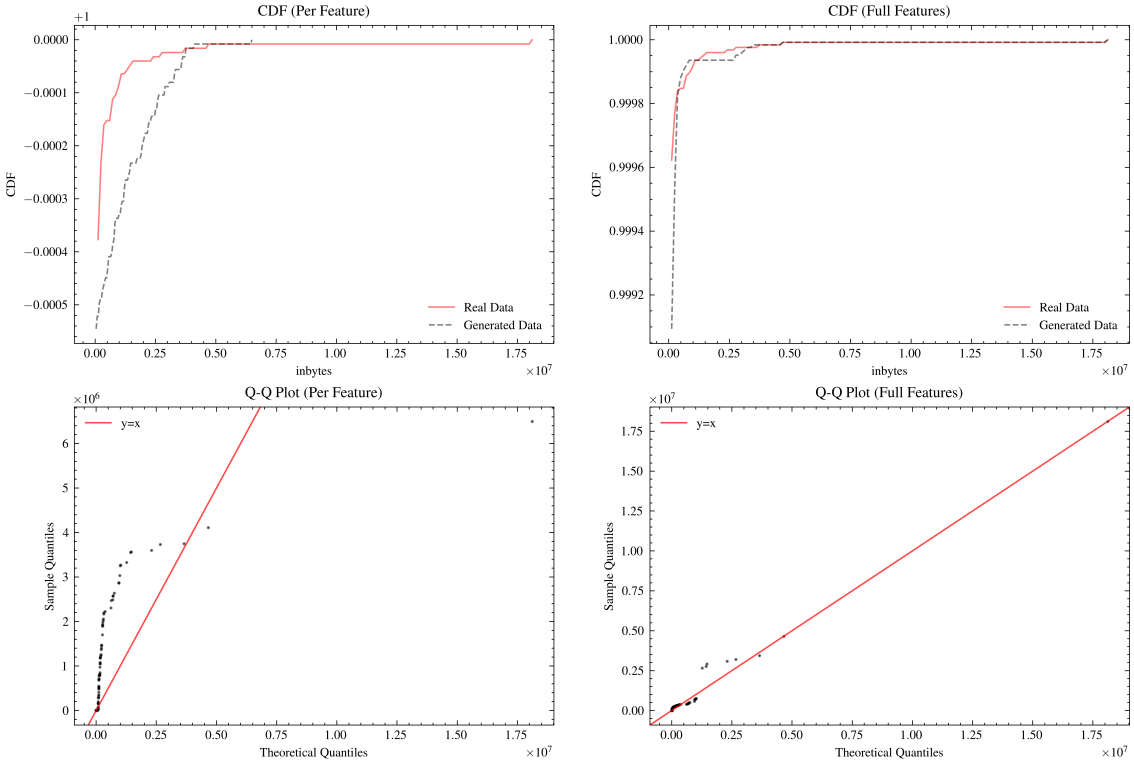
Προφίλ 5 Διαγράμματα



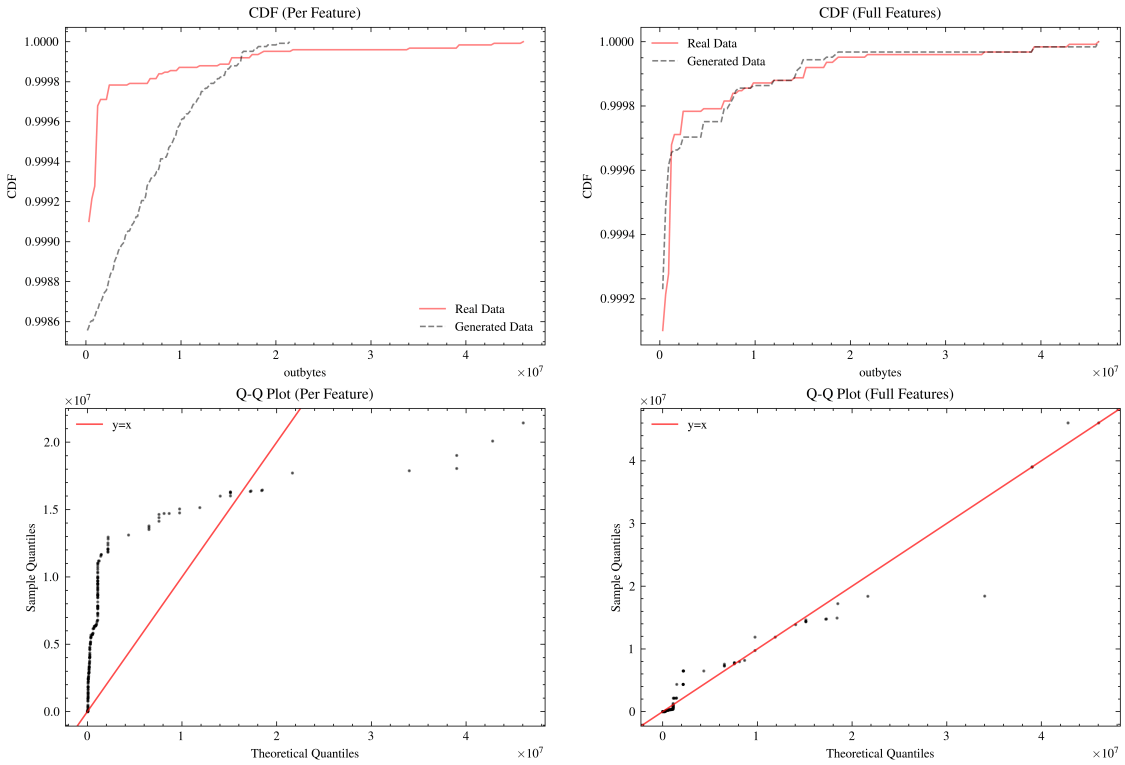
Σχήμα 6.19: Προφίλ 5 Duration



Σχήμα 6.20: Προφίλ 5 Inter-Arrival



Σχήμα 6.21: Προφίλ 5 In-Bytes



Σχήμα 6.22: Προφίλ 4 Out-Bytes

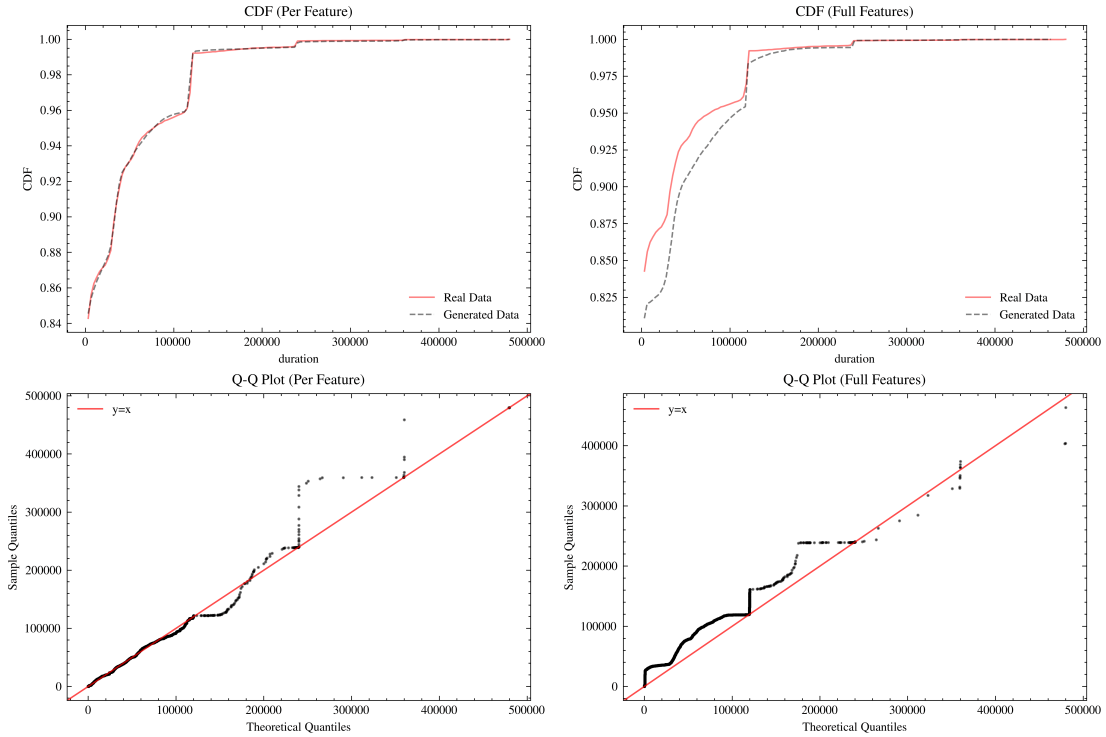
Προφίλ 5 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 3.0398e-06 nats
Duration Per Feature KL(Generated || Real): 3.044e-06 nats
Duration Full Features KL(Real || Generated): 0.0001105514 nats
Duration Full Features KL(Generated || Real): 0.0001074248 nats

Interval Per Feature KL(Real || Generated): 1.54796e-05 nats
Interval Per Feature KL(Generated || Real): 1.54163e-05 nats
Interval Full Features KL(Real || Generated): 0.0006988151 nats
Interval Full Features KL(Generated || Real): 0.000627684 nats

Inbytes Per Feature KL(Real || Generated): 5e-10 nats
Inbytes Per Feature KL(Generated || Real): 5e-10 nats
Inbytes Full Features KL(Real || Generated): 3.1e-09 nats
Inbytes Full Features KL(Generated || Real): 3.1e-09 nats

Outbytes Per Feature KL(Real || Generated): 3.9e-09 nats
Outbytes Per Feature KL(Generated || Real): 3.9e-09 nats
Outbytes Full Features KL(Real || Generated): 2.2e-09 nats
Outbytes Full Features KL(Generated || Real): 2.2e-09 nats
```

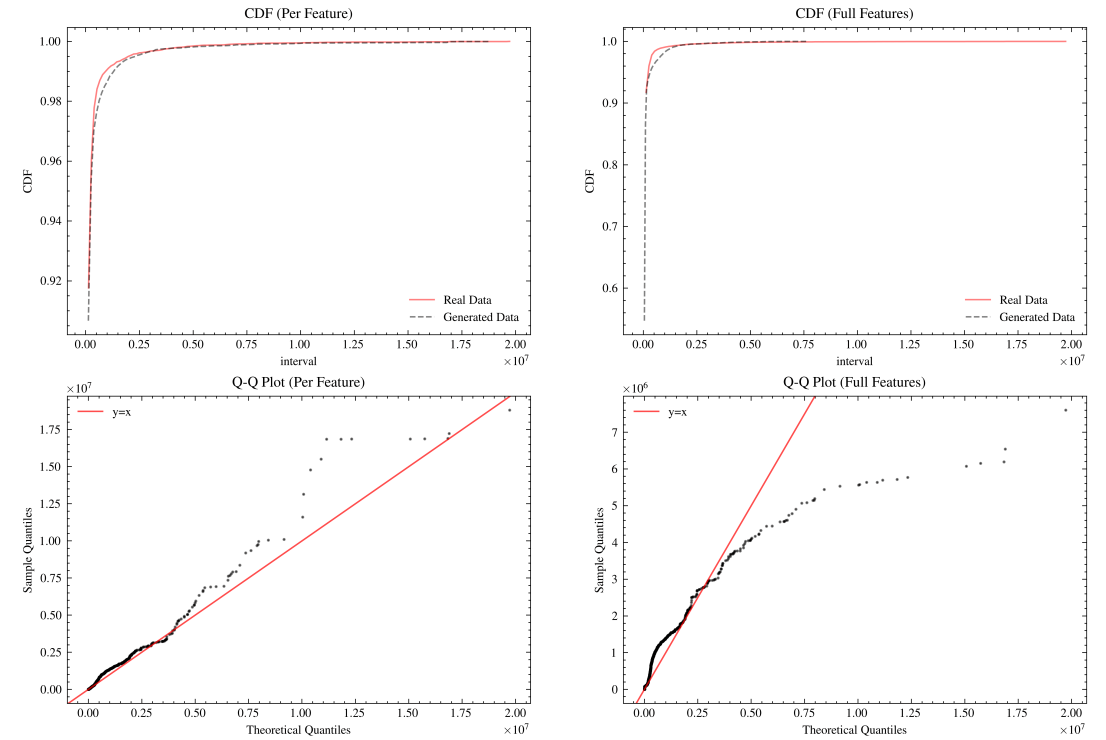
Προφίλ 5 Maximum Mean Discrepancy

```
Profile 5 Per Feature Bootstrapping MMD Mean:     7.784736175812109e+21
Profile 5 Per Feature Bootstrapping MMD Variance: 1.407031520111093e+43

Profile 5 Full Features Bootstrapping MMD Mean:     8.928665006726331e+21
Profile 5 Full Features Bootstrapping MMD Variance: 5.977512103131669e+44
```

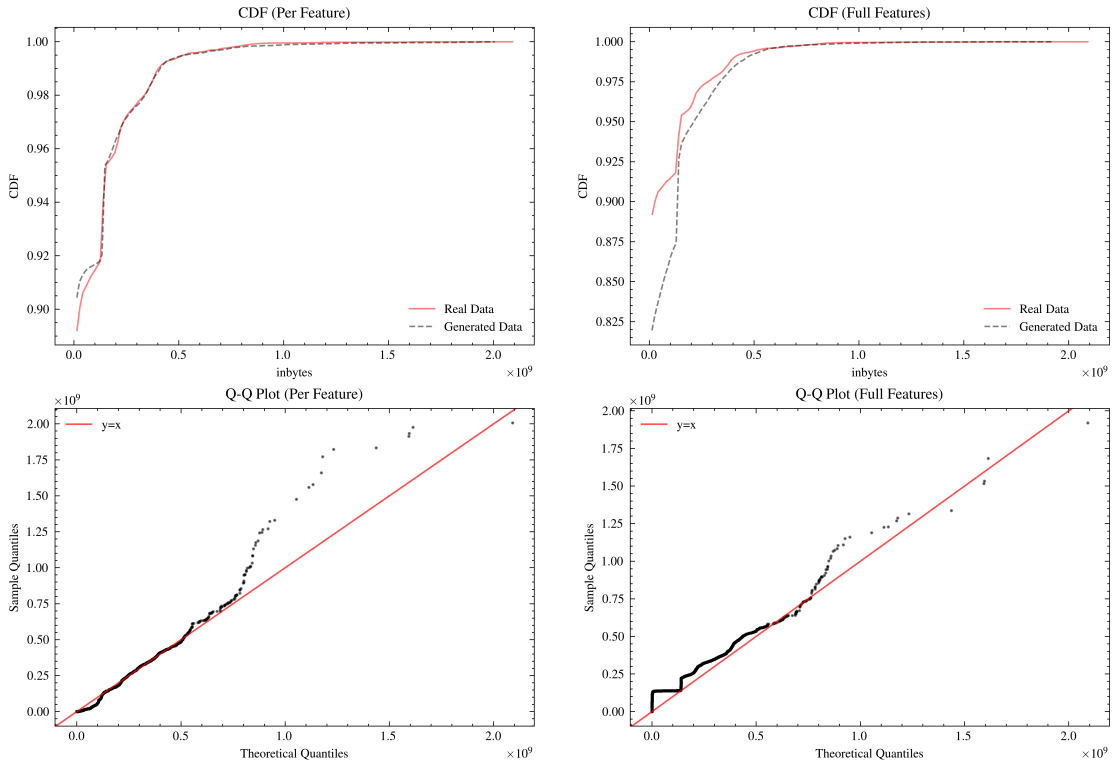
Προφίλ 6 Διαγράμματα



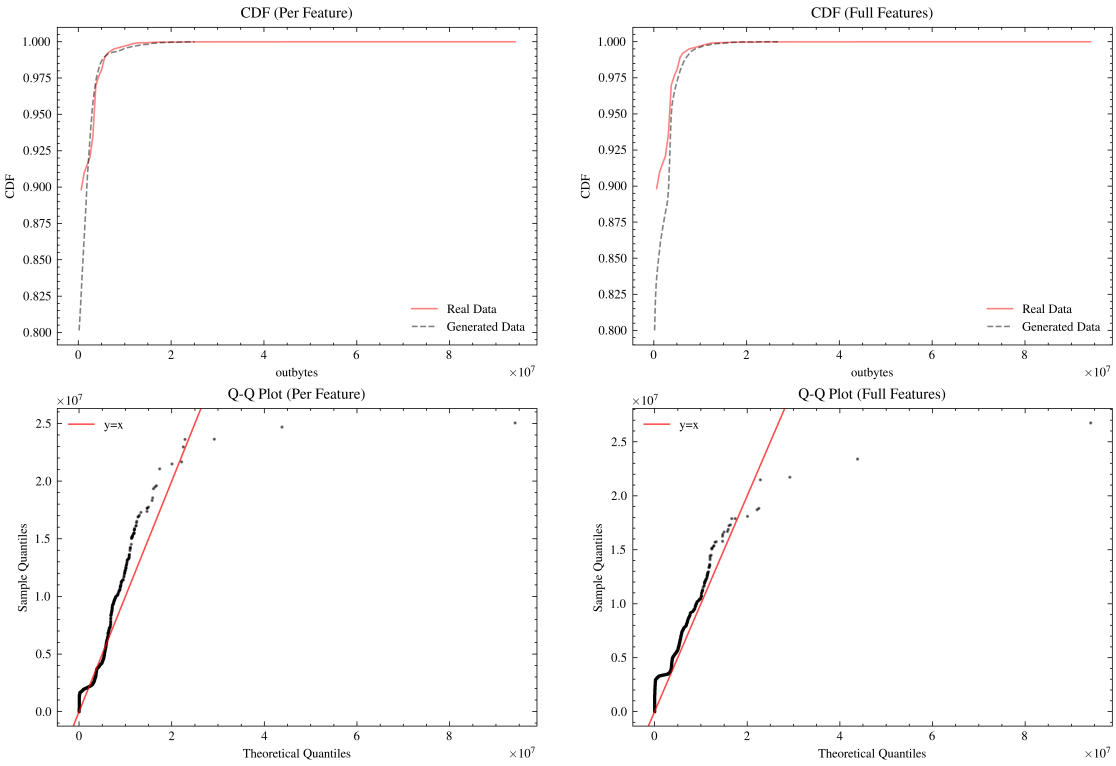
Σχήμα 6.23: Προφίλ 6 Duration



Σχήμα 6.24: Προφίλ 6 Inter-Arrival



Σχήμα 6.25: Προφίλ 6 In-Bytes



Σχήμα 6.26: Προφίλ 6 Out-Bytes

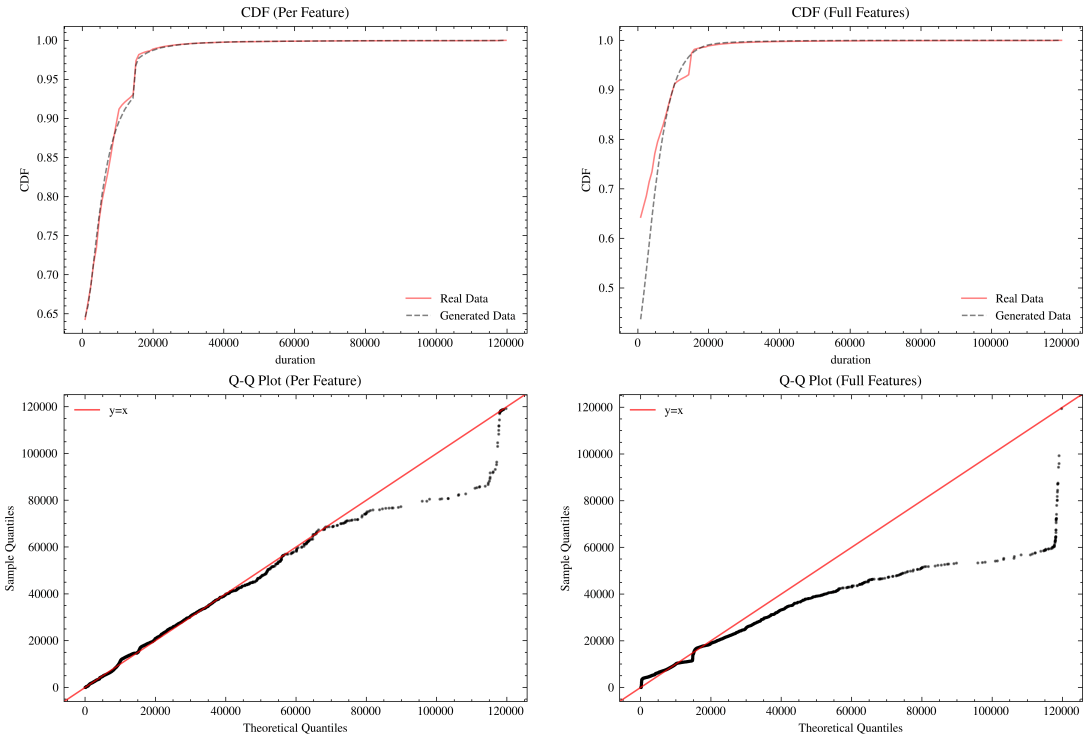
Προφίλ 6 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 2.4952e-06 nats
Duration Per Feature KL(Generated || Real): 2.4939e-06 nats
Duration Full Features KL(Real || Generated): 2.49224e-05 nats
Duration Full Features KL(Generated || Real): 2.50019e-05 nats

Interval Per Feature KL(Real || Generated): 8.306e-07 nats
Interval Per Feature KL(Generated || Real): 8.296e-07 nats
Interval Full Features KL(Real || Generated): 0.001493958 nats
Interval Full Features KL(Generated || Real): 0.0015225797 nats

Inbytes Per Feature KL(Real || Generated): 1.12067e-05 nats
Inbytes Per Feature KL(Generated || Real): 1.12037e-05 nats
Inbytes Full Features KL(Real || Generated): 5.19355e-05 nats
Inbytes Full Features KL(Generated || Real): 5.18974e-05 nats

Outbytes Per Feature KL(Real || Generated): 6.34011e-05 nats
Outbytes Per Feature KL(Generated || Real): 6.22128e-05 nats
Outbytes Full Features KL(Real || Generated): 7.43319e-05 nats
Outbytes Full Features KL(Generated || Real): 7.31032e-05 nats
```

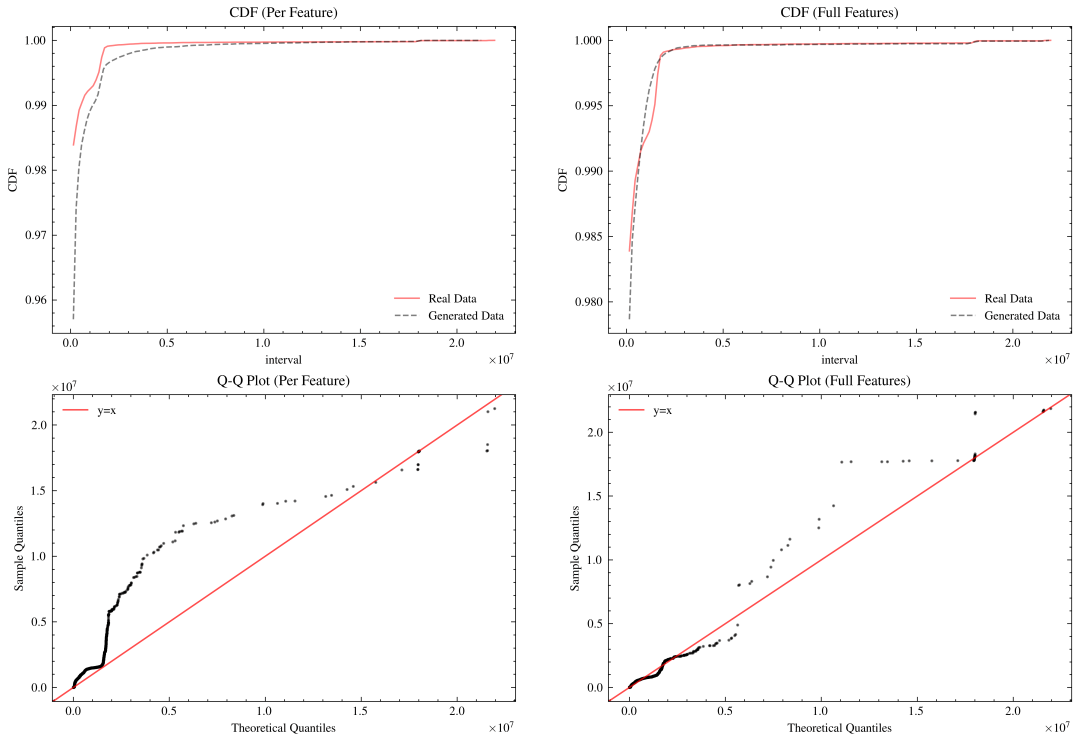
Προφίλ 6 Maximum Mean Discrepancy

```
Profile 6 Per Feature Bootstrapping MMD Mean:     6.413812905667291e+30
Profile 6 Per Feature Bootstrapping MMD Variance: 1.0157711866159439e+62

Profile 6 Full Features Bootstrapping MMD Mean:     1.418689504163915e+31
Profile 6 Full Features Bootstrapping MMD Variance: 3.2541395055998127e+62
```

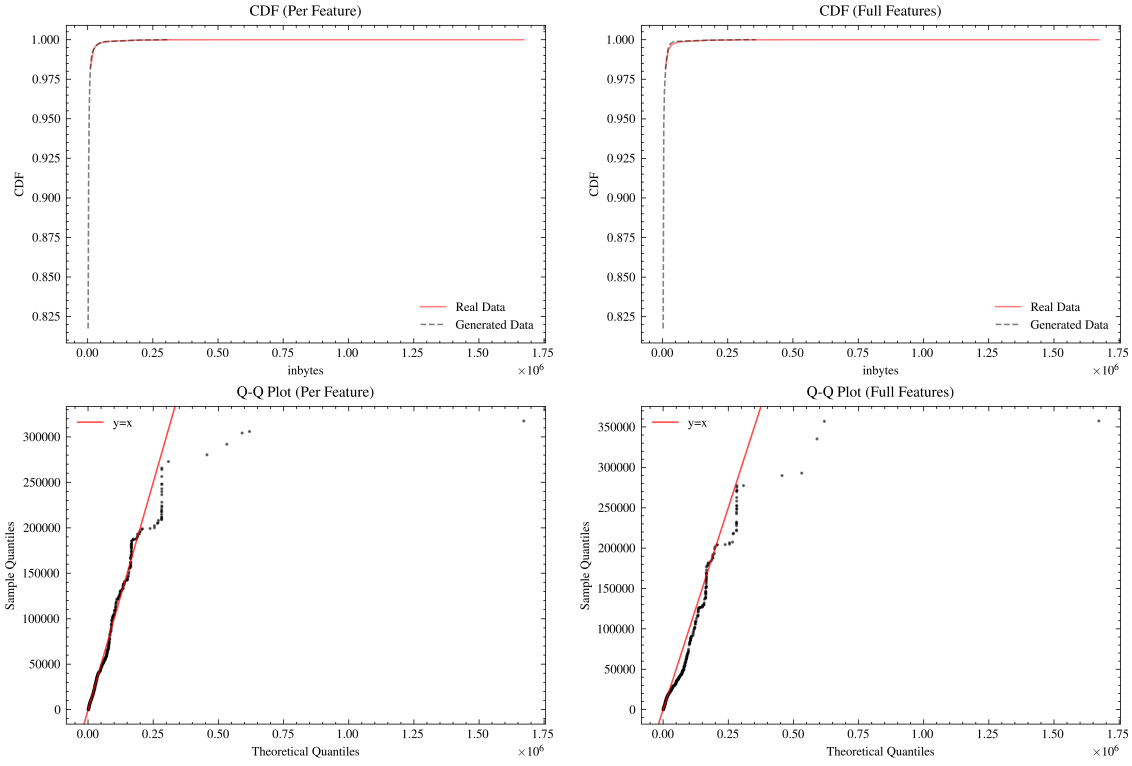
Προφίλ 7 Διαγράμματα



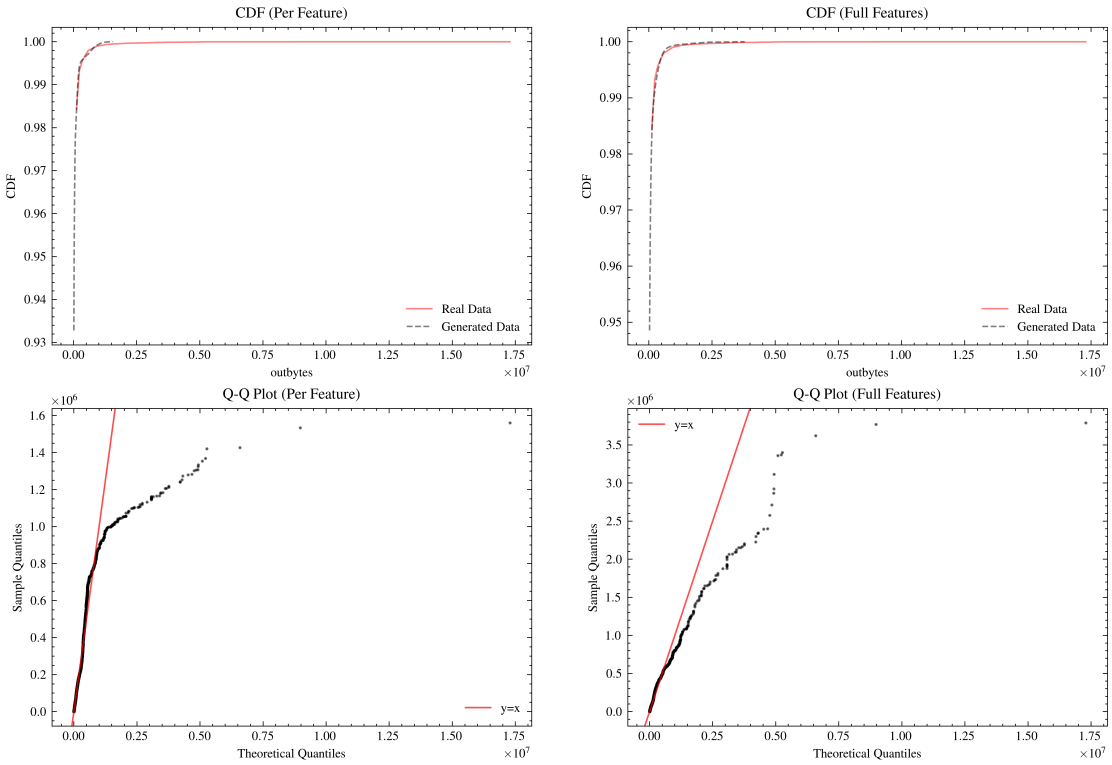
Σχήμα 6.27: Προφίλ 7 Duration



Σχήμα 6.28: Προφίλ 7 Inter-Arrival



Σχήμα 6.29: Προφίλ 7 In-Bytes



Σχήμα 6.30: Προφίλ 7 Out-Bytes

Προφίλ 7 Kullback-Leibler Divergence

```
Duration Per Feature KL(Real || Generated): 7.3882e-06 nats
Duration Per Feature KL(Generated || Real): 7.3772e-06 nats
Duration Full Features KL(Real || Generated): 0.0003602736 nats
Duration Full Features KL(Generated || Real): 0.0003429403 nats

Interval Per Feature KL(Real || Generated): 6.1223e-06 nats
Interval Per Feature KL(Generated || Real): 6.1271e-06 nats
Interval Full Features KL(Real || Generated): 3.786e-07 nats
Interval Full Features KL(Generated || Real): 3.787e-07 nats

Inbytes Per Feature KL(Real || Generated): 0.000241741 nats
Inbytes Per Feature KL(Generated || Real): 0.0002422038 nats
Inbytes Full Features KL(Real || Generated): 0.0002880072 nats
Inbytes Full Features KL(Generated || Real): 0.000293859 nats

Outbytes Per Feature KL(Real || Generated): 1.46786e-05 nats
Outbytes Per Feature KL(Generated || Real): 1.45481e-05 nats
Outbytes Full Features KL(Real || Generated): 7.7949e-06 nats
Outbytes Full Features KL(Generated || Real): 7.7568e-06 nats
```

Προφίλ 7 Maximum Mean Discrepancy

```
Profile 7 Per Feature Bootstrapping MMD Mean:      1.0323303730096265e+22
Profile 7 Per Feature Bootstrapping MMD Variance:  2.7360170806906667e+44

Profile 7 Full Features Bootstrapping MMD Mean:      4.6134112660832675e+21
Profile 7 Full Features Bootstrapping MMD Variance:  6.9993058708281995e+43
```

Σύνοψη - Συμπεράσματα

Αρχικά, αξίζει να σημειωθεί ότι λόγω του μεγάλου αριθμού δειγμάτων που υπήρχαν για τις παραπάνω κατανομές, η μετρική MMD με polynomial kernel χρησιμοποιήθηκε με bootstrapping. Δηλαδή, η μετρική χρησιμοποίησε τυχαία δειγματοληψία με αντικατάσταση (μίμηση της διαδικασίας δειγματοληψίας). Το bootstrapping έχει αποδειχθεί ότι εκχωρεί μέτρα ακρίβειας σε εκτιμήσεις δειγμάτων και επιτρέπει την εκτίμηση της δειγματοληπτικής κατανομής σχεδόν κάθε στατιστικού με τη χρήση μεθόδων τυχαίας δειγματοληψίας.

Με βάση τα παραπάνω διαγράμματα και τη Kullback-Leibler Divergence προκύπτει ότι, όπως ήταν προφανές, η πρώτη μέθοδος που εκπαιδεύτηκε ανά χαρακτηριστικό (θεωρεί ανεξάρτητα χαρακτηριστικά), έχει καλύτερα αποτελέσματα στις επιμέρους κατανομές συγκριτικά με τη δεύτερη μέθοδο. Δηλαδή, παρατηρείται καλύτερο φιτ στις CDF κατανομές και τα σημεία στα Q-Q plots είναι πιο κοντά στη ευθεία $y = x$. Η Kullback-Leibler Divergence, σχεδόν σε όλα τα προφίλ, παρουσιάζει μικρότερη τιμή στην πρώτη μέθοδο (δηλαδή μικρότερη απόκλιση των real και generated distributions) από ότι στην δεύτερη.

Εν αντιθέση, με βάση τη μετρική MMD, η δεύτερη μέθοδος (που θεωρεί τα χαρακτηριστικά ροής εξαρτημένα το ένα από το άλλο) παρουσιάζει στην πλειονότητα των προφίλς καλύτερα αποτελέσματα στο σύνολο των τεσσάρων γνωρισμάτων. Δηλαδή, η μέση τιμή και η διασπορά των bootstrapping MMD είναι συνήθως μικρότερες συγκριτικά με αυτές της πρώτης προσέγγισης. Τα συμπεράσματα αυτά είναι απολύτως λογικά, καθώς οπως αναφέρθηκε και προηγουμένως, η πρώτη μέθοδος εξιδεικεύτηκε στο να μάθει καλά μια συγκεκριμένη κατανομή (χαρακτηριστικό ροής), ενώ η δεύτερη εκπαιδεύτηκε στο σύνολο και των τεσσάρων κατανομών θεωρώντας τες εξαρτημένες.

Τέλος, όσο αναφορά την ποιότητα και την ακρίβεια των παραγόμενων δειγμάτων συγκριτικά με τα πραγματικά, παρατηρούνται σχετικά αρκετά ικανοποιητικά αποτελέσματα και με τις δύο προσεγγίσεις. Υπάρχουν πορφίλ, όπως το πρώτο, (Σχήμα 6.3 - 6.6) που παρουσιάζουν σχεδόν ταύτιση, ενώ μερικά άλλα, όπως το τέταρτο, (Σχήμα 6.15 - 6.18) που παρουσιάζουν χειρότερες συμπεριφορές εκμάθησης. Τα αποτελέσματα αυτά οφείλονται από πληθώρα παραγόντων, όπως το πλήθος και η κατανομή των πραγματικών δειγμάτων, το πλήθος των outliers, κτλ. Αξίζει να σημειωθεί ότι, όπως αναφέρθηκε και στο Κεφάλαιο της διερευνητικής ανάλυσης δεδομένων, η συντρηπτική πλειοψηφία των τιμών των χαρακτηριστικών είναι αρκετά μικρές, έχοντας μόνο ελάχιστες μεγάλες (outliers), τις οποίες μερικές φορές αδυνατεί να μοντελοποιήσει η εκάστοτε προσέγγιση.

Κεφάλαιο 7

Ταξινόμηση Ροών σε Κλάσεις

Στο συγκεκριμένο κεφάλαιο χρησιμοποιούνται αλγόριθμοι **επιβλεπόμενης μάθησης** και **νευρωνικά δίκτυα** για την ταξινόμηση της κίνησης στις κλάσεις - κατηγορίες χρηστών που έχουν αναφερθεί στις προηγούμενες ενότητες. Σκοπός της παρούσας μελέτης είναι η δημιουργία και η εκπαίδευση ενός μοντέλου **ταξινομητή**, ο οποίος να μπορεί να διαχωρίσει μελλοντικές κινήσεις χρηστών στις αντίστοιχες κλάσεις, χωρίς να απαιτείται εξωτερικός χειρισμός με κανόνες, όπως έγινε στο Κεφάλαιο 6.1. Δηλαδή, στην ουσία τους κανόνες που ορίστηκαν προηγουμένως, να μπορεί το μοντέλο να τους μάθει μόνο του μέσω εκπαίδευσης και έτσι να διαχωρίζει μελλοντικές άγνωστες κινήσεις.

Η δημιουργία ενός τέτοιου έμπιστου μοντέλου έχει και αρκετές πιθανές επεκτάσεις στο κομμάτι της **ανίχνευσης επιθέσων - ανωμαλιών**. Για παράδειγμα, κίνηση η οποία δεν ταξινομείται σε κάποια από τις ήδη υπάρχουσες κλάσεις μπορεί να θεωρηθεί πιθανή απειλή. Αν δηλαδή το ανώτερο ποσοστό που θα προβλέψει το μοντέλο για ένα δείγμα να ανήκει σε μια κλάση είναι χαμηλότερο από ένα κατώφλι (threshold), τότε θα θεωρείται πιθανή επίθεση και μπορεί έτσι να υφισταθεί από μια παραπάνω επεξεργασία η συγκεκριμένη κίνηση. Η σκέψη αυτή θα μπορούσε να έχει προοπτικές σκεπτόμενοι ότι δεν έχουμε labeled δεδομένα επίθεσης, οπότε είναι μια μορφή μη επιβλεπόμενης μάθησης για την ανίχνευση ανωμαλιών.

7.1 Περιγραφή Συνόλου Δεδομένων

Λόγω του ότι τα συνολικά δεδομένα ροής που έχουν καταγραφεί για τις 33 ημέρες είναι υπερβολικά μεγάλα σε μέγεθος για μια τέτοιου είδους ανάλυση, υπήρξε η ανάγκη να απομονωθεί ένα συγκεκριμένος είδος κίνησης. Με βάση την ανάλυση και των προηγούμενων κεφαλαίων, η πιο αξιοσημείωτη είναι αυτή που έχει ως στόχο τις νοσοκομειακές υπηρεσίες που έχουν ήδη αναφερθεί. Επομένως, το σύνολο δεδομένων αποτελείται από καθαρά **νοσοκομειακές υπηρεσίες**.

Το χαρακτηριστικό στόχος της ταξινόμησης (label) είναι οι κλάσεις που είχαν δημιουργηθεί στο Κεφαλαίο 6.1. Από τις συνολικά επτά που είχαν προκύψει, αυτές με αρκετά ικανοποιητικό αριθμό ροών είναι οι Ιατροί, η Διοίκηση Κλινικών, η Κεντρική Διοίκηση και οι Νοσηλευτές.

Τα **δείγματα** είναι συνολικά 4922720 και τα **χαρακτηριστικά** 13 (μαζί με την ετικέτα).

Πιο ειδικά, για τα 13 γνωρίσματα έχουμε τις εξής τιμές:

1. SRC_MACHINE: 65 διαφορετικές κατηγορίες συσκευής (παθολογική, γραμματεία, κτλ.)
2. DST_ID: 10 διαφορετικές νοσοκομειακές υπηρεσίες στόχοι
    - 284: his
    - 3891: eservices.yeka
    - 3982: galinos
    - 4501: apografi
    - 9546: eopyy
    - 11646: promitheus
    - 12926: idika/EfkaServices
    - 14810: idika
    - 43649: ebaby.ypes
    - 47780: vpn university
3. L4_DST_PORT: 0, 80, 443, 1723, 7778, 51001, 51002, 51003, 51004
4. PROTOCOL: 6 (TCP), 47 (UDP)
5. L7_PROTO_NAME: HTTP, TLS, TLS.Microsoft, GRE, Whois-DAS, HTTP.Microsoft, Unknown
6. IN_BYTES: continuous
7. IN_PKTS: continuous
8. OUT_BYTES: continuous
9. OUT_PKTS: continuous
10. DURATION: continuous
11. SHIFT: 1 (night), 2 (morning), 3 (evening)
12. IS_WORKDAY: binary
13. CLASS - CATEGORY: Doctor, Clinic Administration, Central Administration, Nurse

7.2 Προεπεξεργασία Συνόλου Δεδομένων

Έπειτα από μια πρώτη απομόνωση της νοσοκομειακής κίνησης και φιλτράρισμα των σημαντικών τεσσάρων κλάσεων, το επόμενο βήμα ήταν η προεπεξεργασία (**data preprocessing**). Σε πολλά προβλήματα ταξινόμησης και μηχανικής μάθησης γενικότερα, το preprocessing είναι ένα πολύ σημαντικό κομμάτι και συχνά πολύ χρονοβόρο ολόκληρης της διαδικασίας, το οποίο όμως έχει συνήθως πολύ μεγάλο αντίκτυπο στην επίδοση του συστήματος. Τα βήματα προεπεξεργασίας που ακολουθήθηκαν ήταν τα εξής:

- Αφαιρέθηκαν οι **απουσιάζουσες τιμές** από το dataset. Συνολικά βρέθηκαν μόλις τέσσερις απουσιάζουσες στο πεδίο SRC_MACHINE, για τις οποίες λογικά δεν βρέθηκε η αντίστοιχη πληροφορία στα DHCP logs.

- Μετατράπηκαν οι **κατηγορικές μεταβλητές** σε binary, ώστε να είναι διαχειρίσιμες από τους αλγορίθμους μηχανικής μάθησης με τη μέθοδο «get_dummies» της βιβλιοθήκης Pandas.

- **Κανονικοποιήθηκαν τα χαρακτηριστικά** με το z-score (Standard Scaler) σύμφωνα με τον τύπο $z = \frac{X-\mu}{\sigma}$, που μετατρέπει το χαρακτηριστικό ώστε να έχει μέση τιμή μηδέν και διακύμανση μονάδα, σαν την κανονική κατανομή. Αυτή η διαδικασία είναι ιδιαίτερα σημαντική καθώς χαρακτηριστικά με πολύ μεγάλες διαφορές στις απόλυτες τιμές τους μπορούν να προκαλέσουν προβλήματα στην εκπαίδευση και να δώσουν ταξινομητές με μη βέλτιστη απόδοση. Για παράδειγμα, ένα χαρακτηριστικό με πολύ μεγάλες τιμές θα έχει μεγαλύτερη επίδραση στον υπολογισμό της απόστασης στον αλγόριθμο kNN από ότι ένα με μικρές τιμές, χωρίς αυτό να σημαίνει απαραίτητα ότι είναι περισσότερο καθοριστικό για το διαχωρισμό των κλάσεων. Η κανονικοποίηση μετασχηματίζει τις τιμές των χαρακτηριστικών ώστε να αμβλυνθούν αυτές οι διαφορές.

- **Κωδικοποιήθηκαν οι ετικέτες** των κλάσεων - κατηγοριών ως ακέραιοι αριθμοί. Αν και οι περισσότεροι εκτιμητές για ταξινόμηση στη scikit-learn μετατρέπουν εσωτερικά τις ετικέτες σε ακέραιους αριθμούς, θεωρείται καλή πρακτική η παροχή ετικετών κλάσης ως ακέραιους για να αποφευχθούν τυχόν δυσλειτουργίες. Για την κωδικοποίηση των ετικετών κλάσης, χρησιμοποιήθηκε μια προσέγγιση παρόμοια με την αντιστοίχιση των διατεταγμένων χαρακτηριστικών που αναφέρθηκε προηγουμένως. Δεν έχει σημασία ο ακέραιος αριθμός που εκχωρείται σε μια συγκεκριμένη ετικέτα, αρκεί να είναι μοναδικός. Έτσι, απλά απαριθμήθηκαν οι ετικέτες από το 0 μέχρι το 4.

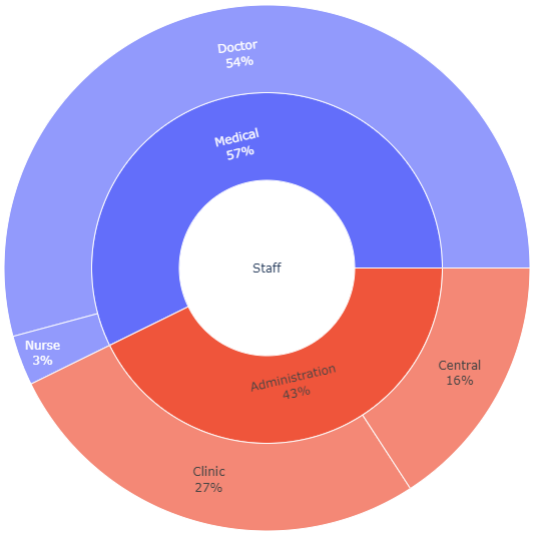
- Μειώθηκε η διαστατικότητα με τεχνική feature extraction και συγκεκριμένα με την **ανάλυση σε κύριες συνιστώσες (principal components analysis - PCA**). Η ανάλυση σε κύριες συνιστώσες (PCA) είναι η ευρέως πιο διαδεδομένη μέθοδος μείωσης της διαστατικότητας. Αρχικά υπολογίζεται ο πίνακας συσχέτισης (covariance matrix) των μεταβλητών των δεδομένων. Από αυτόν τον πίνακα βρίσκονται οι γραμμικώς συσχετισμένες μεταβλητές και εξάγωντας τα ιδιοδιανύσματα του πίνακα, είναι εφικτή η μετατροπή του με έναν ορθογώνιο μετασχηματισμό και η εύρεση της βάσης του νέου

πίνακα. Αυτή η βάση του χώρου αποτελεί ένα νέο σύνολο μεταβλητών που είναι γραμμικά ασυσχέτιστες και ονομάζονται κύριες συνιστώσες. Η παράμετρος «n_components» (αριθμός συνιστωσών που θα κρατηθούν) βελτιστοποιήθηκε με Grid-Search, όπως θα αναλυθεί περεταίρω στις επόμενες ενότητες.

- Χωρίστηκε το σύνολο δεδομένων σε τυχαία train και test set με αναλογίες 70% - 30% αντίστοιχα.

7.3 Binary Classification

Αρχικά, η ταξινόμηση πραγματοποιήθηκε στις δύο γενικότερες κλάσεις, του ιατρικού προσωπικού και του διοικητικού προσωπικού. Πιο συγκεκριμένα, οι ιατροί και οι νοσηλευτές αποτελούν το ιατρικό προσωπικό, ενώ η κεντρική διαχείριση και η διαχείριση κλινικών αποτελούν το διοικητικό προσωπικό. Η κατανομή των δύο γενικών κλάσεων και των τεσσάρων υποκλάσεων στα συνολικά δεδομένα ροής παρουσιάζεται στο Σχήμα 7.1.



Σχήμα 7.1: Κατανομή των κλάσεων στο σύνολο δεδομένων

Οι δύο αλγόριθμοι ταξινόμησης που χρησιμοποιήθηκαν είναι οι παρακάτω:

- Gaussian Naive Bayes (Baseline)
- kNN (Optimized)

Ο Gaussian Naive Bayes χρησιμοποιήθηκε ως μια γραμμή βάσης (baseline) για μια πρώτη εικόνα της ταξινόμησης, ενώ ο kNN βελτιστοποιήθηκε όσο αναφορά τον αριθμό των γειτόνων (best n_neighbors=15) και το πλήθος των κύριων συνιστωσών PCA (best n_components=7). Η απόδοση των ταξινομητών, δηλαδή τα αποτελέσματα των μετρικών και οι χρόνοι εκτέλεσης παρουσιάζονται στον Πίνακα 7.1, ενώ οι πίνακες σύγχυσης στο Σχήμα 7.2.

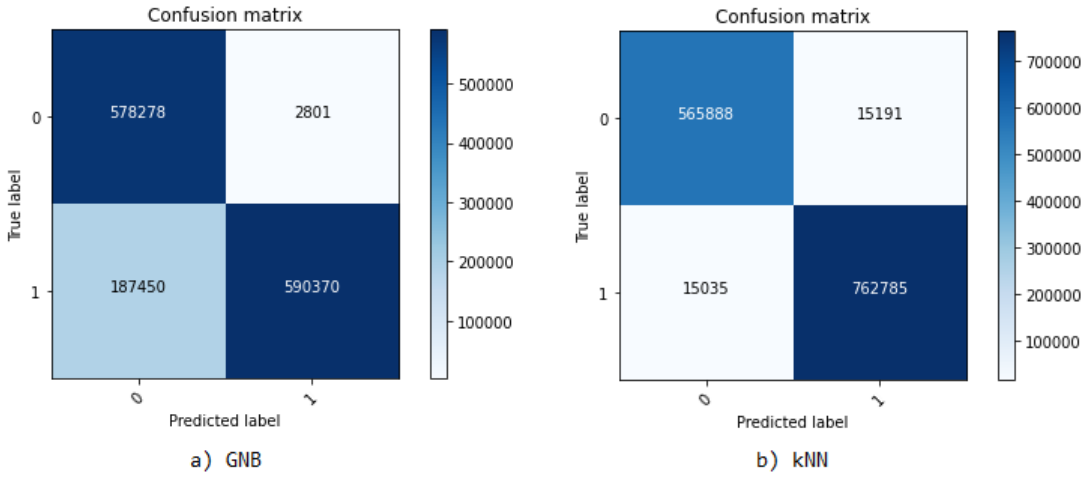
Αξίζει να σημειωθεί ότι η μετρική Precision στην ουσία υποδήλωνε το εξής: Αναφορικά με τις προβλέψεις για τη κλάση 0 (administration), πόσες ανταποκρινόντουσαν στη πραγματικότητα (ήταν «0» και όχι «1»). Αντίστοιχα, η τιμή Recall υποδήλωνε το εξής: Αναφορικά

με τη κλάση 0 (administration), πόσες προβλέψεις ήταν σύμφωνες με αυτό (ήταν «0» και όχι «1»). Επομένως, η μελέτη του precision, recall παρατηρείται ότι μπορεί να οδηγεί σε κάποια κοινά συμπεράσματα, ωστόσο τα επιμέρους αποτελέσματα δεν παύουν να είναι διαφορετικά. Άλλωστε υπάρχει και το εν λόγο trade off μεταξύ των δύο αυτών εννοιών. Για το λόγο αυτό μελετήθηκε και η απόδοση ως προς μετρική f1 (micro και macro). Να τονιστεί ότι χρησιμοποιώντας στρατηγική macro, οι μετρικές υπολογίζονται ανεξάρτητα και στη συνέχεια υπολογίζεται μια μέση τιμή, ενώ χρησιμοποιώντας τη τεχνική micro, λαμβάνονται υπόψιν η κατανομή των κλάσεων.

Classifier	**F1 Score (%)**		**Precision (%)**		**Recall (%)**		**Time (s)**	**Accuracy (%)**
	micro	macro	micro	macro	micro	macro		
Gaussian Naive Bayes	85.99	85.99	42.76	21.38	42.76	50.0	6.2	86.0
kNN	97.78	97.73	97.78	97.73	97.78	97.73	2010.5	97.78

Πίνακας 7.1: Μετρικές απόδοσης binary ταξινομητών

Οι παραπάνω μετρικές που σχολιάστηκαν εξάγονται από τους πίνακες σύγχυσης που φαίνονται στο Σχήμα 7.2. Αναφορικά με το χρόνο εκτέλεσης, είναι λογικό λόγω του GridSearch ο χρόνος για έυρευση βέλτιστων υπερπαραμέτρων να είναι μεγάλος. Στην ουσία το πρόβλημα είχε 2 παραμέτρους για βελτιστοποίηση (γείτονες και PCA συνιστώσες) και δοκιμάστηκαν N τιμές για το καθένα. Λόγω των 2 εμφωλευμένων for loops του GridSearch, η πολυπλοκότητα ήταν $O(N^2)$, γεγονός που δικαιολογεί το μεγάλο χρόνο των 2010.5 δευτερολέπτων.



Σχήμα 7.2: Πίνακες σύγχυσης για τον a) GNB και b) kNN ταξινομητή

Συμπερασματικά, παρατηρείται ότι ο βελτιστοποιημένος αλγόριθμος kNN κατάφερε να επιτύχει ένα ποσοστό F1 Score και Accuracy στο **97.78%**. Ένα υψηλό ποσοστό, που φανερώνει ότι το μοντέλο μπορεί να ταξινομεί σε πολύ καλό βαθμό την δικτυακή κίνηση ροών σε ιατρική (medical) και διοικητική (administration).

7.4 Multiclass Classification

Στη συνέχεια, λόγω του καλού αποτελέσματος της δυαδικής ταξινόμησης, αποφασίστηκε να μελετηθεί το ίδιο πρόβλημα και για τις τέσσερις υποκλάσεις, doctor (κλάση 1), clinic administration (κλάση 2), central administration (κλάση 3), nurse (κλάση 4». Το βασικό μειονέκτημα της συγκεκριμένης μεθόδου, όσο αναφορά το πλήθος των ροών ανά κλάση, είναι ότι το dataset δεν είναι ισορροπημένο (μικρό ποσοστό της κλάσης των νοσοκόμων). Οι αλγόριθμοι ταξινόμησης που χρησιμοποιήθηκαν είναι οι παρακάτω:

- Gaussian Naive Bayes
- Random Forest
- LightGBM
- SVC Linear
- kNN

Αξίζει να σημιεωθεί ότι το χαρακτηριστικό SRC_MACHINE, δηλαδή οι πληροφορίες για τις συσκευές από τα DHCP logs, δίνουν μια προκατάληψη (bias) στον ταξινομητή, επομένως αποφασίστηκε να χωριστεί το πρόβλημα στη χρήση του ή όχι. Πιο συγκεκριμένα, εφαρμόστηκαν αριχκά οι αλγόριθμοι ταξινόμησης σε ολόκληρα τα δεδομένα και στη συνέχεια ο βέλτιστος ταξινομητής εκπαιδεύτηκε και στα δεδομένα χωρίς το SRC_MACHINE. Στην τελευταία μάλιστα περίπτωση εκπαιδεύτηκε και ένα μοντέλο νευρωνικού δικτύου (MLP).

7.4.1 Ταξινόμηση με χρήση DHCP πληροφοριών

Ο Gaussian Naive Bayes χρησιμοποιήθηκε και πάλι ως μια γραμμή βάσης (baseline) για μια πρώτη εικόνα της ταξινόμησης. Οι υπόλοιποι, αρχικά, εκπαιδεύτηκαν με τις default τιμές των υπερπαραμέτρων τους και στη συνέχεια αυτοί που είχαν την καλύτερη απόδοση βελτιστοποιήθηκαν περαιτέρω. Από τις πρώτες εκπαιδεύσεις φάνηκε ότι ο αλγόριθμος kNN υπερτερεί συγκρητικά με τους υπόλοιπους. Συγκεκριμένα, οι βέλτιστες υπερπαράμετροι που προέκυψαν ήταν οι εξής:

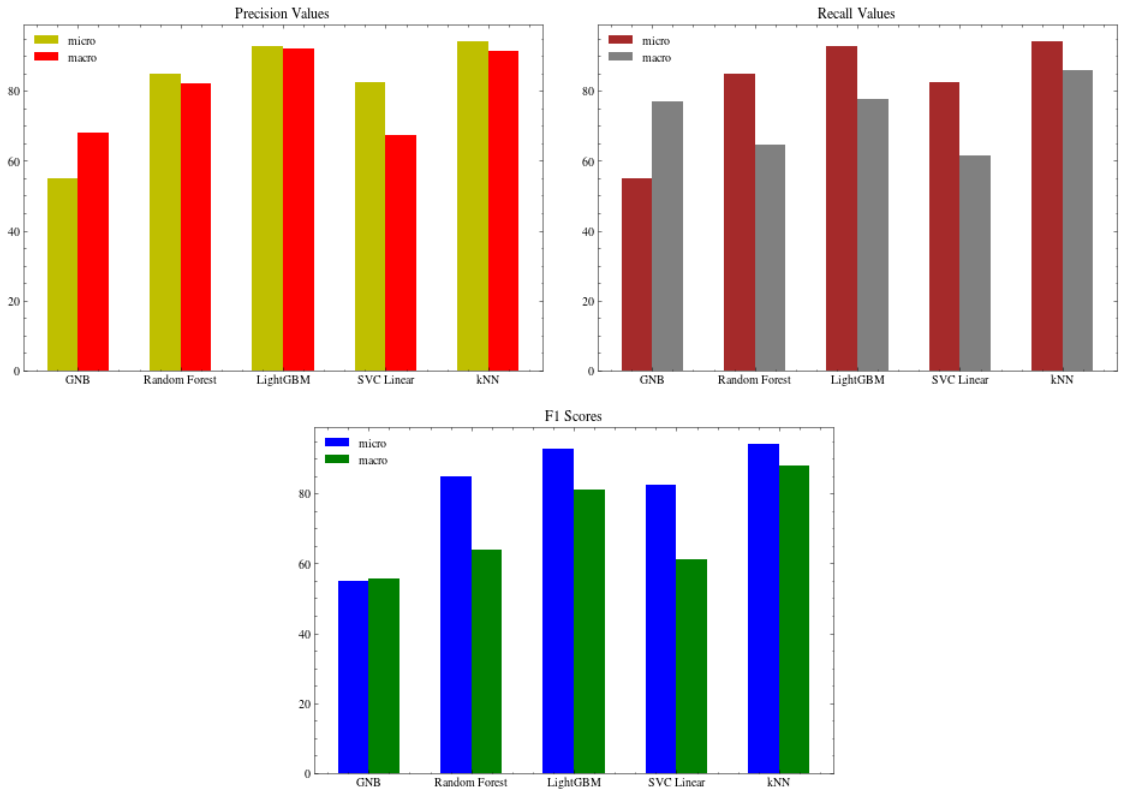
- n_neighbors: 15 (Αριθμός γειτόνων)
- weights: distance (Η συνάρτηση βάρους που χρησιμοποιείται στην πρόβλεψη)
- metric: minkowski (Η μετρική απόστασης που χρησιμοποιείται για το δέντρο. Αφού επίσης p=2 είναι ισοδύναμη με την τυπική ευκλείδεια απόσταση)
- pca_n_components: 5 (Πλήθος των κύριων συνιστωσών PCA)

Η απόδοση των ταξινομητών, δηλαδή τα αποτελέσματα των μετρικών παρουσιάζονται στον Πίνακα 7.2 καθώς και σε ραβδογράμματα στο Σχήμα 7.3. Με βάση αυτά τα αποτελέσματα συμπεραίνεται ότι την καλύτερη απόδοση την είχε ο αλγόριθμος kNN με **94.25%** Accuracy

και F1-micro. Ακολουθεί ο LightGBM με 92.87%. Αξίζει επίσης να σημειωθεί ότι στην πλειοψηφία των περιπτώσεων, όπως φαίνεται και από τα διαγράμματα, το macro average είναι μικρότερο από το micro average. Αυτό συμβαίνει διότι υπάρχουν πιο χαμηλά ποσοστά επιτυχίας σε μια συγκεκριμένη κλάση, αυτή των νοσοκόμων (nurse).

Classifier	**F1 Score (%)**		**Precision (%)**		**Recall (%)**		**Accuracy (%)**
	micro	macro	micro	macro	micro	macro	
Gaussian Naive Bayes	54.97	55.82	54.97	68.05	54.97	76.92	54.97
Random Forest	84.78	63.93	84.78	82.01	84.78	64.6	85.02
LightGBM	92.87	81.07	92.87	92.13	92.87	77.59	92.87
SVC Linear	82.45	61.03	82.45	67.49	82.45	61.56	82.45
kNN	94.25	88.07	94.25	91.26	94.25	85.79	94.25

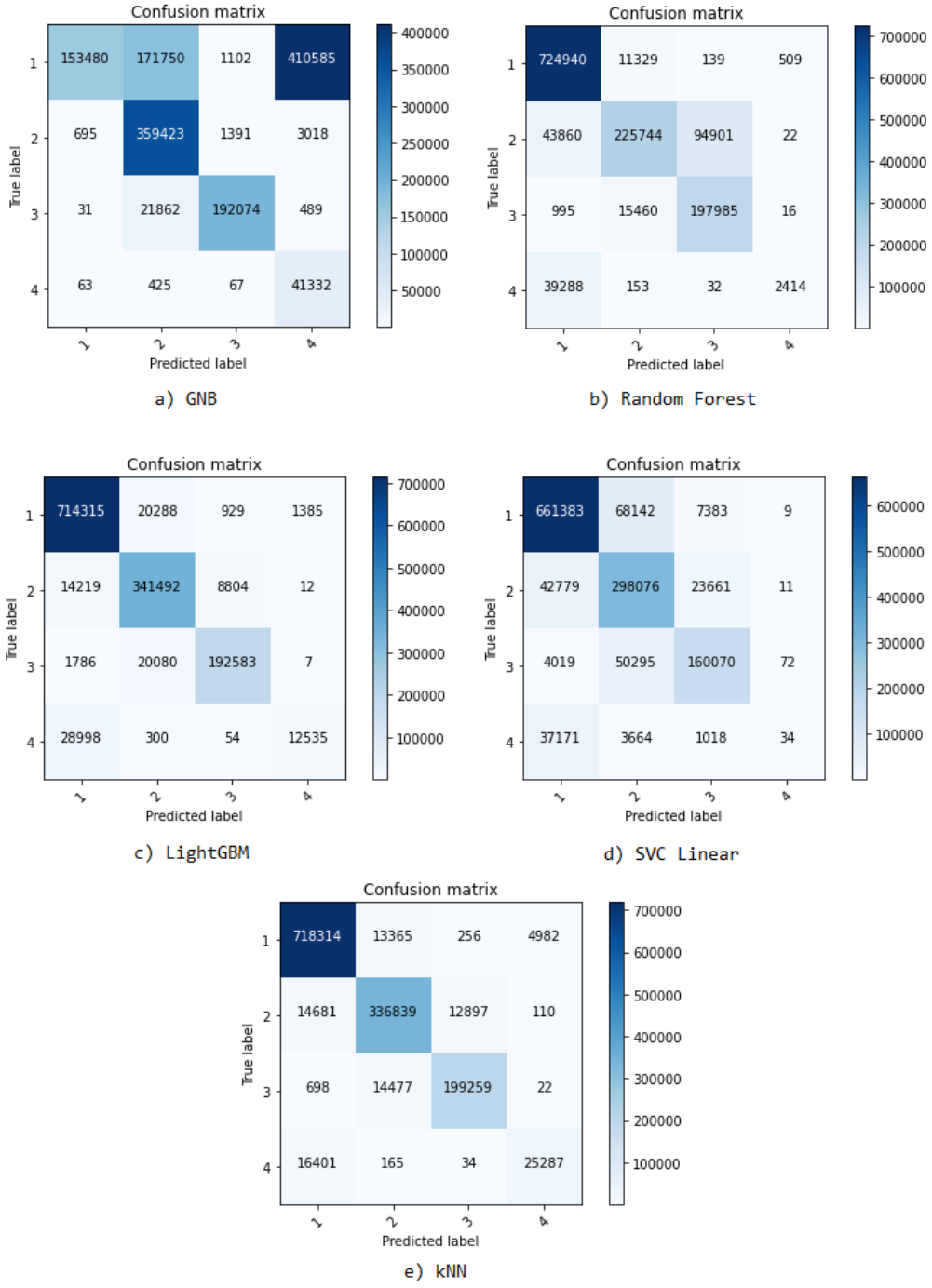
Πίνακας 7.2: Μετρικές απόδοσης multiclass ταξινομητών



Σχήμα 7.3: Ραβδογράμματα μετρικών απόδοσης ταξινομητών

Το πρόβλημα αυτό γίνεται ιδιαίτερα εμφανές στους πίνακες σύγχυσης που παρουσιάζονται στο Σχήμα 7.4. Τα περισσότερα μοντέλα αδυνατούν να ξεχωρίσουν την κίνηση των νοσοκόμων (κλάση 4) από αυτή των ιατρών (κλάση 1), λόγω των χαμηλών δειγμάτων της. Επίσης, όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, η κλάση nurse προκύπτει από κάποιους

κανόνες οι οποίοι λάμβαναν υπόψη συνολικά τις ροές του χρήστη και όχι μία μία μεμωνομένη ροή, όπως γίνεται σε αυτή τη περίπτωση. Επίσης από τους παρακάτω πίνακες διαφαίνονται οι σχέσεις - αποστάσεις μεταξύ των κλάσεων (εκεί που δηλαδή έχουμε τις λάθος προβλέψεις). Συγκεκριμένα, τα περισσότερα «λάθη» παρατηρούνται ανάμεσα στο ζευγάρι ιατροί-νοσοκόμοι και λίγο λιγότερα στο ζευγάρι ιατροί-διαχείριση κλινικών. Αυτό φανερώνει επίσης τη παρόμοια δικτυακή κίνηση που έχουν οι χρήστες αυτών των κλάσεων.



Σχήμα 7.4: Πίνακες σύγχυσης για multicast ταξινομητές

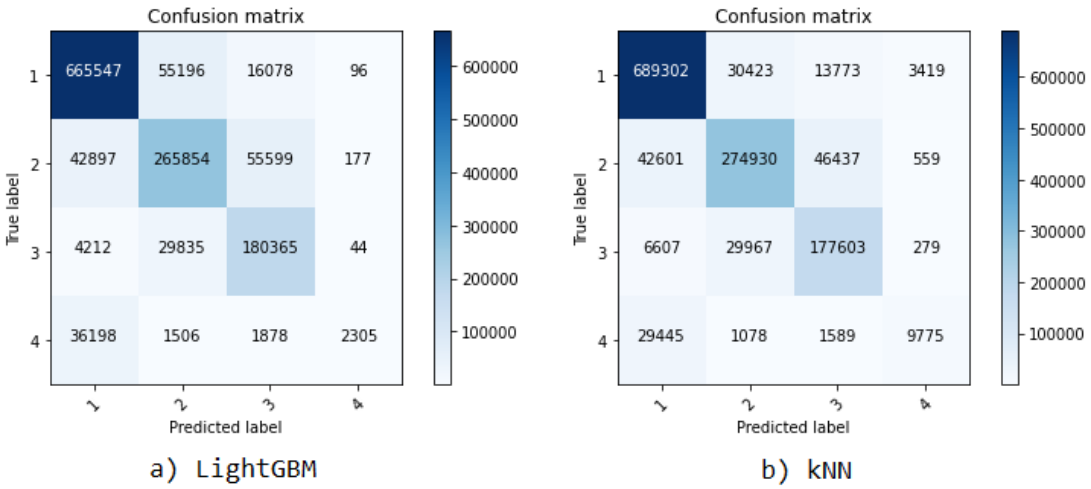
Καταλυκτικά λοιπόν, συμπεραίνουμε ότι με χρήση των πληροφοριών του DHCP για τις συσκευές (γνώρισμα SRC_MACHINE), τα καλύτερα αποτελέσματα τα πετυχαίνει ο αλγόριθμος kNN με F1 micro και Accuracy 94.25%, ενώ F1 macro 88.07%.

7.4.2 Ταξινόμηση χωρίς χρήση DHCP πληροφοριών

Στη συνέχεια αυτής της ανάλυσης, εκπαιδεύτηκαν οι καλύτεροι ταξινομητές (LightGBM και kNN) στα δεδομένα χωρίς τη χρήση των πληροφοριών από τα DHCP logs (χαρακτηριστικό SRC_MACHINE), το οποίο μπορεί να δίνει μια προκατάληψη (bias) στους ταξινομητές. Η απόδοση των ταξινομητών, δηλαδή τα αποτελέσματα των μετρικών παρουσιάζονται στον Πίνακα 7.3 και οι πίνακες σύγχυσης στο Σχήμα 7.5.

Classifier	**F1 Score (%)**		**Precision (%)**		**Recall (%)**		**Accuracy (%)**
	micro	macro	micro	macro	micro	macro	
LightGBM	82.05	62.78	82.05	80.82	82.05	63.21	82.05
kNN	84.82	70.82	84.82	78.83	84.82	68.78	84.82

Πίνακας 7.3: Μετρικές απόδοσης ταξινομητών χωρίς πληροφορίες DHCP



Σχήμα 7.5: Πίνακες σύγχυσης για ταξινομητές χωρίς πληροφορίες DHCP

Με βάση τα παραπάνω αποτελέσματα και τους πίνακες σύγχυσης παρατηρείται ότι τις καλύτερες αποδόσεις τις έχει επιτύχει και πάλι ο αλγόριθμος kNN (βέλτιστος με n_neighbors:15, pca_n_components: 15) με ποσοστά **F1 micro** και **Accuracy 84.82%**, ενώ **F1 macro 70.82%**.

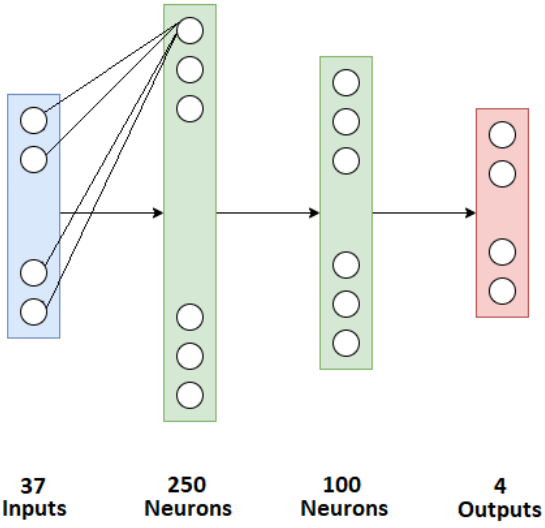
Συμπεραίνεται επομένως, ότι οι πληροφορίες από τα DHCP logs για τις συσκευές (χαρακτηριστικό SRC_MACHINE), παίζει καθοριστικό ρόλο στη πρόβλεψη των κλάσεων, καθώς

έχει μειώσει το ποσοστό επιτυχίας του kNN κατά περίπου 9% τόσο στο micro όσο και στο macro average.

Νευρωνικό Δίκτυο MLP

Εν συνεχεία με τους παραπάνω δύο αλγορίθμους ταξινόμησης, εκπαιδεύτηκε και ένα νευρωνικό δίκτυο Multilayer Perceptron με χρήση της βιβλιοθήκης PyTorch της Python.

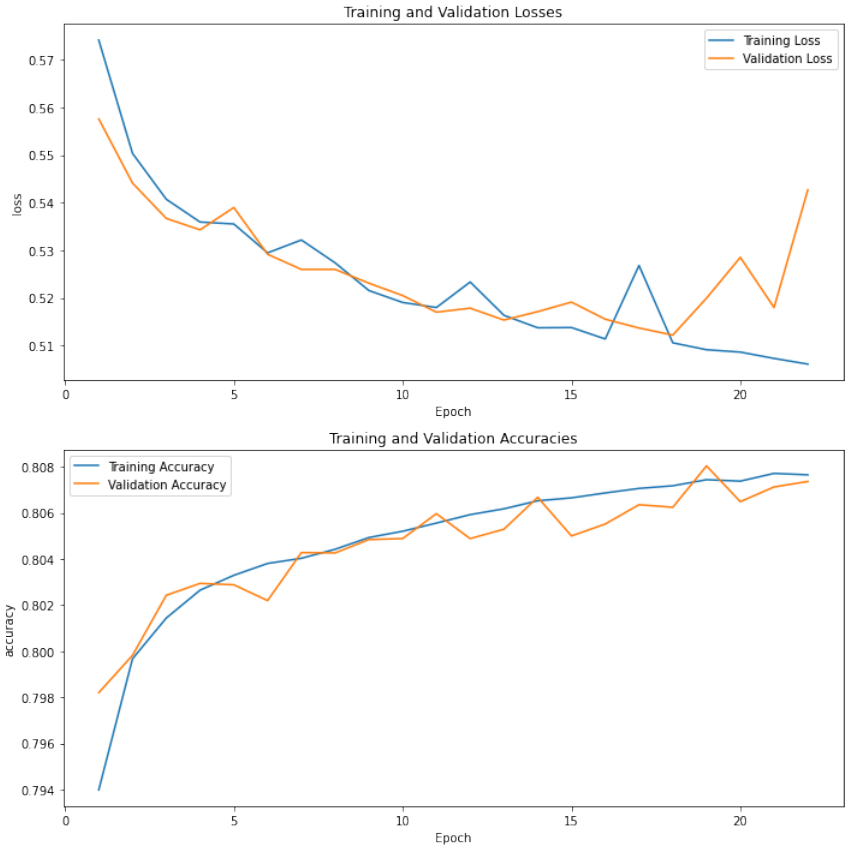
Το σύνολο δεδομένων, έπειτα από την προεπεξεργασία που αναφέρθηκε προηγουμένως (απουσιάζουσες τιμές, κατηγορικά χαρακτηριστικά, κανονικοποίηση), αποτελείται από **37 γνωρίσματα**. Χωρίστηκε και εδώ σε train - test με ποσοστά 70-30%, ενώ το train set χωρίστηκε εκ νέου και σε validation set σε ποσοστό 10%. Και τα τρία σύνολα (train, test, validation) διασπάστηκαν σε batches με μέγεθος 64.



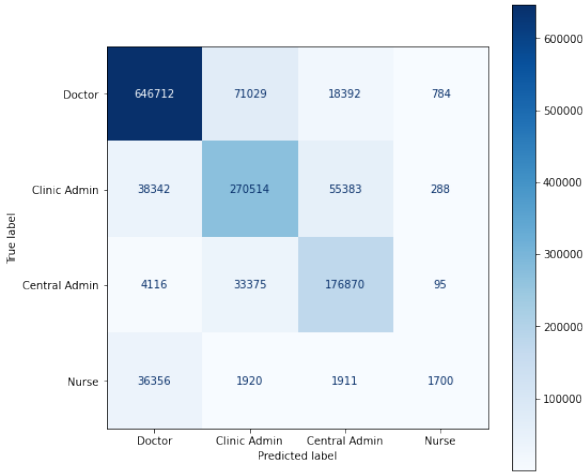
Σχήμα 7.6: Τεχνητό Νευρωνικό Δίκτυο για Ταξινόμηση

Το μοντέλο, όπως φαίνεται και στο Σχήμα 7.6, αποτελείται από δύο κρυφά επίπεδα των 250 και 100 νευρώνων αντίστοιχα και ένα επίπεδο εξόδου με 4 νευρώνες, όσες και οι κλάσεις προς ταξινόμηση. Στα κρυφά επίπεδα εφαρμόστηκε η συνάρτηση ενεργοποίησης **ReLU**, ενώ στο επίπεδο εξόδου η **Softmax** που δηλώνει την πιθανότητα να ανήκει το δείγμα στην κλάση αυτή και έτσι το αποδίδει στον κόμβο - κλάση με την μεγαλύτερη πιθανότητα. Υπολογίστηκε ότι το μοντέλο έχει συνολικά 35.004 εκπαιδεύσιμες παραμέτρους. Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε η συνάρτηση κόστους **cross entropy loss** με τη μέθοδο οπισθοδιάδοσης λάθους (**backpropagation**) και τον αλγόριθμο βελτιστοποίησης **Adam**. Χρησιμοποιήθηκε επίσης η τεχνική **early stopping** για την πρόωρη διακοπή της εκπαίδευσης όταν το κόστος δεν βελτιωθεί μετά από 5 συνεχόμενες εποχές.

Η εκπαίδευση πραγματοποιήθηκε στο περιβάλλον Google Colaboratory με χρήση GPU για αρχικά 30 εποχές, αλλά λόγω της πρόωρης διακοπής σταμάτησε στις **22 εποχές**. Οι γραφικές παραστάσεις του κόστους (loss) και της ορθότητας (accuracy) ανά εποχή για το train και validation set φαίνονται στο Σχήμα 7.7.



Σχήμα 7.7: Γραφικές παταστάσεις κόστους και ορθότητας ανά εποχή



Σχήμα 7.8: Πίνακας σύγχυσης για ταξινομητή MLP

Οι μετρικές απόδοσης του νερωνικού δικτύου φαίνονται στον Πίνακα 7.4, ενώ ο πίνακας σύγχυσης στο Σχήμα 7.8.

Συμπερασματικά, παρατηρείται ότι η απόδοση του MLP ταξινομητή είναι λίγο χειρότερη από αυτή του kNN. Όπως και προηγουμένως, το «πρόβλημα» παρατηρείται στο ζευγάρι κλάσεων ιατρών - νοσοκόμων, όπου το μοντέλο αδυνατεί να προλέψει ορθά αρκετά δείγματα. Αξίζει

Classifier	**F1 Score (%)**		**Precision (%)**		**Recall (%)**		**Accuracy (%)**
	micro	macro	micro	macro	micro	macro	
MLP	80.70	61.19	80.70	72.56	80.70	62.13	80.70

Πίνακας 7.4: Μετρικές απόδοσης MLP ταξινομητή

επίσης να σημειωθεί ότι το παραπάνω μοντέλο είναι ένα αρχικό στάδιο για τη μέθοδο ταξινόμησης με MLP, καθώς δεν έχει βελτιστοποιηθεί. Μια παραπάνω βελτιστοποίηση ως προς π.χ. τη δομή του νευρωνικού δικτύου (κρυφά επίπεδα, νευρώνες), του αλγορίθμου βελτιστοποίησης, του learning rate, του early stopping, του batch size κτλ, θα μπορούσε να είχε οδηγήσει σε ένα μοντέλο ισάξιο ή και καλύτερο από τον kNN. Παρόλο αυτά, η βελτιστοποίηση ενός νευρωνικού δικτύου ξεφεύγει από τα όρια της συγκεκριμένης εργασίας, που δεν έχει σκοπό την εμβάθυνση σε τέτοιο βαθμό.

Κεφάλαιο 8

Επίλογος

Στο συγκεκριμένο κεφάλαιο συνοψίζεται η μελέτη και τα συμπεράσματα που εκπονήθηκαν στο πλαίσιο της παρούσας διπλωματικής εργασίας, καθώς και προτίνονται πιθανές μελλοντικές επεκτάσεις της.

8.1 Σύνοψη και συμπεράσματα

Ο κύριος σκοπός της εργασίας ήταν η παραγωγή ρεαλιστικών δεδομένων νοσοκομειακής και υγειονομικής περίθαλψης μέσω συγκεκριμένων προφίλ συμπεριφοράς χρηστών. Ο σκοπός αυτός επιτεύχθει σε ένα μεγάλο βαθμό μέσω των μοντέλων μείξης που εκπαιδεύτηκαν και προσομοίωσαν ικανοποιητικά τα προφίλ των χρηστών που αποτελούνταν από NetFlow ροές. Προκειμένου όμως να καταλήξει στο τελικό αποτέλεσμα, ιδιαίτερα σημαντική ήταν η φάση της διερευνητικής ανάλυσης των δεδομένων τα οποία καταγράφηκαν από τη νοσοκομειακή υποδομή. Οι τεχνικές εξόρυξης δεδομένων και τα στατιστικά στοιχεία βοήθησαν σε μια πρώτη κατανόηση των συμπεριφορών και των προτύπων δικτυακής κίνησης για την μετέπειτα δημιουργία των προφίλ. Επιπλέον, ιδιαίτερα εποφελής στο profiling ήταν και η κατηγοριοποίηση (labeling) των χρηστών σε διάφορες νοσοκομειακές «ομάδες» μέσω των κανόνων συμπεριφοράς. Μάλιστα, εκτός από τους χειροκίνητους κανόνες παρατηρήθηκε ότι η κατηγοριοποίηση μπορεί να επιτευχθεί και μέσω αλγορίθμων ταξινόμησης ή ακόμα και νευρωνικών δικτύων. Ένα βασικό μειονέκτημα το οποίο αξίζει να σημιεωθεί και θα μπορούσε να βελτιωθεί μελλοντικά είναι η διαχείριση των ακραίων τιμών, είτε μικρών, είτε μεγάλων. Τα δεδομένα αποτελούνταν στην πλειοψηφία τους από μικρές τιμές και ορισμένες ακραίες μεγάλες (outliers), οι οποίες δεν μπορούσαν να μοντελοποιηθούν εύκολα. Θα μπορούσε να μελετηθεί περισσότερο σε βάθος η διαχείριση αυτών των ακραίων τιμών (πιθανή απομάκρυνση τους από τα δεδομένα) ή αντίστοιχα των υπερβολικά μικρών τιμών, καθώς δεν προσφέρουν ιδιαίτερη γνώση για την δικτυακή συμπεριφορά ενός χρήστη. Καταλυκτικά, πρέπει να επισημανθεί ότι, παρόλα αυτά τα αρχικά προβλήματα που υπήρχαν στα δεδομένα λόγω των πραγματικών καταγραφών - μετρήσεων, τα μοντέλα μείξης κατάφεραν εν τέλη να προσομοιώσουν τα χαρακτηριστικά των NetFlows και να παράξουν αρκετά ρεαλιστικά τεχνητά δεδομένα ροής.

8.2 Μελλοντικές επεκτάσεις

Η συνεισφορά της εργασίας, όπως έχει αναφερθεί και προηγουμένως, είναι σε δάφορους τομείς. Αρχικά, η καταγραφή και η ανάλυση δικτυακής κίνησης με προσεγγίσεις βασιζμένες στις ροές (πχ. NetFlows) έχει κερδίσει τα τελευταία χρόνια την προσοχή της επιστυμονικής κοινότητας. Η εξόρυξη δεδομένων και η ανάλυση στατιστικών των ροών που πραγματοποιήθηκαν στην παρούσα εργασία, θα μπορούσαν να φανούν ιδιαίτερα χρήσιμες σε θέματα Software Defined Networks (SDN), και γενικά παρακολούθησης δικτύων. Επιπλέον, η δημιουργία ενός έμπιστου μοντέλου κατηγοριοποίησης χρήστη (profiling) ανάλογα με την συμπεριφορά της κίνησή του, προσφέρει αρκετές πιθανές επεκτάσεις στο κομμάτι της ανίχνευσης επιθέσων - ανωμαλιών. Εν συνεχεία, ιδιαίτερα χρήσιμη για πιθανές μελλοντικές έρευνες είναι η μέθοδος μοντελοποίησης και προσομοίωσης της δικτυακής κίνησης μέσω παραγωγικών μοντέλων. Η παραγωγή τεχνητών ρεαλιστικών δεδομένων μπορεί να διασφαλίσει την ορθή και αξιόπιστη αξιολόγηση όλων των παραπάνω ιδεών και τεχνικών, καθώς τέτοια δεδομένα δεν είναι εύκολα διαθέσιμα στη βιβλιογραφία. Τέλος, η παραπάνω παραγωγή ρεαλιστικών χαρακτηριστικών ροής θα μπορούσε να φανεί επωφελής στην πραγματική προσομοίωση ανρθώπινης κίνησης (επισκεψιμότητα του χρήστη σε ιστοσελίδες κτλ.) μέσω πιθανών εφαρμογών - εργαλείων.

Βιβλιογραφία

[1] D. Kreutz, F.M.V. Ramos, P.E. Veríssimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey, Proc. IEEE 103 (1) (2015) 14–76, http://dx.doi.org/10.1109/JPROC.2014.2371999.

[2] R. Wójcik, A. Jajszczyk, Flow oriented approaches to QoS assurance, ACM Comput. Surv. 44 (1) (2012) http://dx.doi.org/10.1145/2071389.2071394.

[3] S. Shin, L. Xu, S. Hong, G. Gu, Enhancing network security through software defined networking (SDN), in: 2016 25th International Conference on Computer Communication and Networks, ICCCN, 2016, pp. 1–9, http://dx.doi.org/10. 1109/ICCCN.2016.7568520.

[4] Ring, Markus & Schlör, Daniel & Landes, Dieter & Hotho, Andreas. (2018). Flow-based Network Traffic Generation using Generative Adversarial Networks. Computers & Security. 82. 10.1016/j.cose.2018.12.012.

[5] Jurkiewicz, Piotr & Rzym, Grzegorz & Borylo, Piotr. (2018). Flow length and size distributions in campus Internet traffic.

[6] M. Pustisek, I. Humar, J. Bester, Empirical analysis and modeling of peer-to-peer traffic flows, in: The 14th IEEE Mediterranean Electrotechnical Conference, MELECON 2008, 2008, pp. 169–175, http://dx.doi.org/10.1109/MELCON.2008.4618429.

[7] Zhang, Yin & Breslau, Lee & Paxson, Vern & Shenker, Scott. (2002). On the Characteristics and Origins of Internet Flow Rates. ACM SIGCOMM Computer Communication Review. 32. 10.1145/964725.633055.

[8] Qian, Feng & Gerber, Alexandre & Mao, Zhuoqing & Sen, Subhabrata & Spatscheck, Oliver & Willinger, Walter. (2009). TCP revisited: a fresh look at TCP in the wild. 76-89. 10.1145/1644893.1644903.

[9] Lee, DongJin & Brownlee, Nevil. (2007). Passive measurement of one-way and two-way flow lifetimes. ACM SIGCOMM Computer Communication Review. 37. 17-28. 10.1145/1273445.1273448.

[10] Kim, Myung-Sup & Won, Young & Hong, James. (2006). Characteristic analysis of Internet traffic from the perspective of flows. Computer Communications. 29. 1639-1652. 10.1016/j.comcom.2005.07.015.

[11] Jakalan, Ahmad & Gong, Jian & Zhang, Weiwei & Su, Qi. (2015). Clustering and Profiling IP Hosts Based on Traffic Behavior. Journal of Networks. 10. 99-107. 10.4304/jnw.10.2.99-107.

[12] Xu, Kuai & Zhang, Zhi-Li & Bhattacharyya, Supratik. (2005). Profiling Internet backbone traffic: Behavior models and applications. Computer Communication Review. 35. 169-180. 10.1145/1080091.1080112.

[13] Cai, Jun & Liu, Waixi. (2013). A New Method of Detecting Network Traffic Anomalies. Applied Mechanics and Materials. 347-350. 10.2991/iccsee.2013.699.

[14] K. Xu, Z. Zhang and S. Bhattacharyya, "Internet Traffic Behavior Profiling for Network Security Monitoring," in IEEE/ACM Transactions on Networking, vol. 16, no. 6, pp. 1241-1252, Dec. 2008, doi: 10.1109/TNET.2007.911438.

[15] V. Frias-Martinez, S. J. Stolfo and A. D. Keromytis, "Behavior-Profile Clustering for False Alert Reduction in Anomaly Detection Sensors," 2008 Annual Computer Security Applications Conference (ACSAC), 2008, pp. 367-376, doi: 10.1109/ACSAC.2008.30.

[16] Ortiz, Jorge & Crawford, Catherine & Le, Franck. (2019). DeviceMien: network device behavior modeling for identifying unknown IoT devices. 106-117. 10.1145/3302505.3310073.

[17] Mori, Tatsuya & Uchida, Masato & Goto, Shigeki. (2005). Flow analysis of internet traffic: World Wide Web versus peer-to-peer. Systems and Computers in Japan. 36. 70-81. 10.1002/scj.v36:11.

[18] Andrew S. Tanenbaum & David J. Wetherall, "Computer Networks", Fifth Edition (2011), ISBN 978-960-461-447-9

[19] RFC 1122, Requirements for Internet Hosts – Communication Layers, R. Braden (ed.), October 1989.

[20] RFC 1123, Requirements for Internet Hosts – Application and Support, R. Braden (ed.), October 1989.

[21] Fielding, Roy T.; Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk; Masinter, Larry; Leach, Paul J.; Berners-Lee, Tim (June 1999). Hypertext Transfer Protocol – HTTP/1.1., IETF, RFC 2616

[22] Tim Berner-Lee (1991-01-01). "The Original HTTP as defined in 1991". www.w3.org. World Wide Web Consortium.

[23] Belshe, M.; Peon, R.; Thomson, M. "Hypertext Transfer Protocol Version 2, Use of TLS Features".

[24] "Secure your site with HTTPS". Google Support. Google Inc. Archived from the original on 2015-03-01.

[25] "Network Working Group (May 2000). "HTTP Over TLS". The Internet Engineering Task Force.

[26] "Usage Statistics of Default protocol https for Websites, July 2019". w3techs.com.

[27] T. Dierks· E. Rescorla (August 2008). "The Transport Layer Security (TLS) Protocol, Version 1.2"

[28] Gillis, Alexander S. "What is DHCP (Dynamic Host Configuration Protocol)?". TechTarget: SearchNetworking.

[29] RFC 1034, Domain Names - Concepts and Facilities, P. Mockapetris, The Internet Society (November 1987)

[30] RFC 781, Internet Protocol - DARPA Internet Program Protocol Specification, Information Sciences Institute, J. Postel (Ed.), The Internet Society (September 1981)

[31] RFC 1035, Domain Names - Implementation and Specification, P. Mockapetris, The Internet Society (November 1987)

[32] Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX / Hofstede, Rick; Čeleda, Pavel; Trammell, Brian; Drago, Idilio; Sadre, Ramin; Sperotto, Anna; Pras, Aiko. - In: IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. - ISSN 1553-877X. - ELETTRONICO. - 16:4(2014), pp. 2037-2064. [10.1109/COMST.2014.2321898].

[33] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011 (Internet Standard), Internet Engineering Task Force, September 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7011.txt.

[34] C. Mills, D. Hirsh, and G. Ruth, "Internet Accounting: Background, "RFC 1272, Internet Engineering Task Force, November 1991. [Online]. Available: http://www.ietf.org/rfc/rfc1272.txt.

[35] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A Parameterizable Methodology for Internet Traffic Flow Profiling," IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pp. 1481–1494, 1995.

[36] ——, "NetFlow Services Solutions Guide," 2007, accessed on 2 May 2014. [Online]. Available: http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html.

[37] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Internet Engineering Task Force, October 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3954.txt.

[38] L. Deri, E. Chou, Z. Cherian, K. Karmarkar, and M. Patterson, "Increasing Data Center Network Visibility with Cisco NetFlow-Lite," in Proceedings of the 7th International Conference on Network and Service Management, CNSM'11, 2011, pp. 1–6..

[39] IETF, "IP Flow Information Export (ipfix)," accessed on 2 May 2014. [Online]. Available: http://datatracker.ietf.org/wg/ipfix/charter/.

[40] N. Brownlee, "Flow-Based Measurement: IPFIX Development and Deployment," IEICE Transactions on Communications, vol. 94, no. 8, pp. 2190–2198, 2011.

[41] Γεωργούλη, Α., 2015. Τεχνητή νοημοσύνη. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: http://hdl.handle.net/11419/3381.

[42] Phil Simon (18 March 2013). Too Big to Ignore: The Business Case for Big Data. Wiley. ISBN 978-1-118-63817-0..

[43] Mitchell, T. (1997). Machine Learning, McGraw Hill, Machine Learning, McGraw Hill, p.2.

[44] Harnad, Stevan (2008), ≪The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence≫ : Epstein, Robert; Peters, Grace, The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer, Kluwer.

[45] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) Foundations of Machine Learning, The MIT Press ISBN 9780262018258.

[46] T. Hastie, R. Tibshirani, J. H. Friedman. The Elements of Statistical Learning. Springer, 2001..

[47] Boser, Guyon, Vapnik, A Training Algorithm for Optimal Margin Classifiers,1992.

[48] Βερύκιος, Β., Καγκλής, Β., Σταυρόπουλος, Η., 2015. Η επιστήμη των δεδομένων μέσα από τη γλώσσα R. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: ηττπ://ηδλ.ηανδλε.νετ/11419/2965.

[49] Rossi, Richard J. (2018). Mathematical Statistics : An Introduction to Likelihood Based Inference. New York: John Wiley & Sons. p. 227. ISBN 978-1-118-77104-4.

[50] Hendry, David F.; Nielsen, Bent (2007). Econometric Modeling: A Likelihood Approach. Princeton: Princeton University Press. ISBN 978-0-691-13128-3.

[51] Chambers, Raymond L.; Steel, David G.; Wang, Suojin; Welsh, Alan (2012). Maximum Likelihood Estimation for Sample Surveys. Boca Raton: CRC Press. ISBN 978-1-58488-632-7.

[52] Stoica, P.; Selen, Y. (2004), "Model-order selection: a review of information criterion rules", IEEE Signal Processing Magazine (July).

[53] Schwarz, Gideon E. (1978), "Estimating the dimension of a model", Annals of Statistics.

[54] Claeskens, G.; Hjort, N. L. (2008), Model Selection and Model Averaging, Cambridge University Press.

[55] Kullback, S., and R. A. Leibler. "On Information and Sufficiency." The Annals of Mathematical Statistics, vol. 22, no. 1, Institute of Mathematical Statistics, 1951, pp. 79–86, http://www.jstor.org/stable/2236703..

[56] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. Journal of Machine Learning Research, 13(Mar):723–773, 2012.

[57] Kantorovich, L. V. (1939). "Mathematical Methods of Organizing and Planning Production".

[58] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). "6.5 Back-Propagation and Other Differentiation Algorithms". Deep Learning. MIT Press. ISBN 9780262035613.

Παράρτημα Α΄

Μοντελοποίηση ανά χαρακτηριστικό

Παρακάτω παρατίθενται οι μείξεις των κατανομών, δηλαδή οι παράμετροι και το βάρος της κάθε συνιστώσας, για κάθε προφίλ και για κάθε χαρακτηριστικό, σύμφωνα με τη 1η μέθοδο της μοντελοποίησης ανά χαρακτηριστικό.

Προφίλ 1

```
"duration_mix" : [
[0.2486271879145453, Log-Normal(9.211497647942224, 0.0006262009855719467)]
[0.24693882981545667, Log-Normal(10.329910891612128, 0.0063478421872172016)]
[0.15277140098886324, Log-Normal(8.891080923835565, 0.6341314741727269)]
[0.04640185644833524, Log-Normal(10.759109649082257, 0.4174590391037051)]
[0.1988912276799102, Normal(1012.6391270470348, 430.1928908337403)]
[0.10636949715288936, Normal(31380.29485183459, 609.2252666987779)]
]
"interval_mix" : [
[0.550916427137693, Log-Normal(4.008985061150544, 3.8041356031083677)]
[0.00019175671329159607, Log-Normal(17.00889498662671, 0.07676497297970421)]
[0.15474363588492526, Log-Normal(9.873304952918076, 0.49757574466858767)]
[1.4050646890525387e-20, Log-Normal(16.373797226675226, 0.20734659839271888)]
[4.7739452678247905e-05, Normal(13654016.990673719, 31928.82607554653)]
[0.21887001883866078, Normal(30639.641485964894, 343.271194050793)]
[3.1468234560608706e-05, Normal(18910194.93335262, 48651.697982958416)]
[4.492568976133088e-05, Normal(3582446.092071249, 5850.882484519429)]
[0.07515402804842936, Normal(300967.67560455727, 1539.6605840065795)]
]
"inbytes_mix" : [
[0.15977023777500066, Log-Normal(9.82224183156174, 0.22553885865889958)]
[0.17364909489845579, Log-Normal(7.1748959313950404, 1.4056002900264886)]
[0.08731849011448327, Log-Normal(10.571025194321756, 0.13341722701016484)]
[0.3324762684114312, Log-Normal(10.281170905316287, 0.15228703699271792)]
[0.2311165077573294, Normal(2070.061674916831, 27.981878798103295)]
[0.002906493510544991, Normal(79398.06414619244, 15232.199738455469)]
[2.6607486902883492e-05, Normal(247986.39864463592, 3761.43030000433692)]
[0.012131768762144, Normal(52434.49119551326, 11362.200513112293)]
[0.0006045312837080089, Normal(127321.61892919372, 23079.468683317293)]
]
"outbytes_mix" : [
[0.04063271153975397, Log-Normal(11.258260818433, 0.8413824618246649)]
[0.8577900250352108, Normal(5294.300320435706, 2513.4144715111215)]
[0.0012839491908301917, Normal(279839.3869217293, 101299.44210980584)]
[0.0004288699617316881, Normal(485698.5559165726, 111337.35154004244)]
[0.0682353050787846, Normal(29886.25018864911, 1833.6930247183973)]
[0.03162913919368888, Normal(23246.275693230913, 9946.309380232107)]
]
```

Προφίλ 2

```
"duration_mix" : [
[0.2137049146299292, Log-Normal(9.21203108250692, 0.001740963021711295)]
[0.2325050294055574, Log-Normal(9.790877800369685, 0.5699127832818185)]
[0.019128710165757977, Log-Normal(9.905071948218382, 0.0016579004375898395)]
[0.12424013779253133, Log-Normal(2.695968770286978, 2.0279050347593444)]
[0.03879971588293785, Log-Normal(9.617614774500986, 0.0018511322434774154)]
[0.1942482169828117, Normal(5020.633182692486, 14.219004902677)]
[0.17583331012883907, Normal(8011.405522859913, 2190.2312972857217)]
[0.0015399650116355778, Normal(103543.42787587985, 44618.5251122771)]
]
"interval_mix" : [
[0.5416656461023988, Log-Normal(10.036160868505636, 0.6024164984257107)]
[1.4788086717329689e-05, Log-Normal(16.482657365830384, 0.07163787949405764)]
[0.0003453300089753789, Log-Normal(12.38859651125328, 0.00034046632232027777)]
[0.09245100604171304, Log-Normal(11.695328252049482, 0.00018080823135292078)]
[0.2604616117384258, Normal(7284.491968766068, 4214.930525175315)]
[0.00037667366709855 09, Normal(1453901.0159923304, 1529731.8650185)]
[0.10468494435467111, Normal(94537.99230580471, 17070.00356415721)]
]
"inbytes_mix" : [
[0.19792143256182082, Log-Normal(7.7259451237023855, 0.6251934405398106)]
[0.0009219185593685285, Log-Normal(11.932954390225026, 0.36927224650980167)]
[0.3031998897294221, Log-Normal(9.22038570888629, 0.8428819281930202)]
[3.3453651356731884e-05, Normal(471376.28438507376, 22481.733296054692)]
[0.29550861463022304, Normal(664.2293881901109, 0.9913163199274915)]
[0.007730349003608174, Normal(26278.39578663049, 1757.2717810860674)]
[0.19468434186420078, Normal(505.0260701531485, 300.7777714193216)]
]
"outbytes_mix" : [
[0.8911046910233872, Log-Normal(7.558293175074017, 2.0104365192030342)]
[0.1088953089766128, Normal(64297.60074033466, 31812.42650188489)]
]
```

Προφίλ 3

```
"duration_mix" : [
[0.6152332400102555, Log-Normal(4.574406121406986, 0.7113436368327534)]
[0.0139232777209271, Log-Normal(10.456336159534393, 0.23483647028132545)]
[0.0007092374604953595, Log-Normal(11.132518175480138, 0.34379363594600965)]
[0.11753937452455002, Normal(3706.9194223245195, 2311.5004689167117)]
[0.15120383801502182, Normal(15565.45710112074, 4205.705280256492)]
[0.02933390927112214, Normal(25342.58780153954, 4241.113696151707)]
[0.001747723896407146, Normal(110340.52862327502, 347.2285059572063)]
[0.07030939910122118, Normal(14195.32164039371, 127.03087595735593)]
]
"interval_mix" : [
[0.8821084078506793, Log-Normal(4.151554394343059, 1.7475464449821396)]
[7.425657514694801e-05, Log-Normal(16.515625795805065, 0.12905090977217396)]
[0.064563439380081 59, Log-Normal(10.09545187751917, 0.945493297242052)]
[0.005416213097536837, Log-Normal(13.523292788006076, 1.2645452164920385)]
[0.005421506834612918, Normal(45614.506634551835, 15506.877526258244)]
[0.015399839054147765, Normal(17128.743589718037, 5864.187934507676)]
[3.1285084112938 87e-05, Normal(17949213.516518507, 1929453.2752552114)]
[0.001063705312265628, Normal(5964772.519330084, 2353151.5220401515)]
[0.025921346811415984, Normal(6015.05510538864, 1883.7804857103117)]
]
"inbytes_mix" : [
[0.47470320272740024, Log-Normal(6.911924967390147, 0.03680512926744208)]
[0.008537260387008607, Log-Normal(12.93980624402501, 0.5152975083453774)]
[0.027222537986244497, Log-Normal(10.480022377308815, 0.450469413180111)]
[0.14866154090675596, Normal(12974.876963670307, 4626.103606768351)]
[0.13635063166953038, Normal(5547.399515810908, 2583.7907877300668)]
[0.04416096923843871, Normal(23672.126666539738, 5835.063230282112)]
[0.16036385708462164, Normal(1186.5569788016235, 614.5294367526548)]
]
"outbytes_mix" : [
[0.691512298989163, Log-Normal(8.18025341058268, 1.1860460713582925)]
[0.0026030078398570 28, Log-Normal(13.422547329879439, 0.5160872466524526)]
[0.15279392296647756, Log-Normal(11.891702851186347, 0.3113992595720538)]
[0.08116337373743185, Log-Normal(10.848679222658 27, 0.3561052204479071)]
[0.06923536619975344, Log-Normal(12.524388281305363, 0.3108414099550445)]
[0.002692030267317 1916, Normal(657974.3316126554, 233306.79676200467)]
]
```

Προφίλ 4

```
"duration_mix" : [
[0.9279759041081306, Log-Normal(5.255781707384768, 0.6240941544488614)]
[0.0004328618339805268, Log-Normal(13.345160158732607, 0.769463008754886)]
[0.039992465785843874, Log-Normal(6.949976261933815, 2.607715455041629)]
[0.022547649181818598, Log-Normal(11.661602594670507, 0.02393583713073841)]
[0.009051119090226378, Normal(13202.104665007155, 5953.240537728167)]
]
"interval_mix" : [
[0.8943796432940084, Log-Normal(7.519278278377811, 0.9171330730276449)]
[2.2871444027093843e-05, Normal(20595206.36571514, 12268496.261591926)]
[1.4794682195631805e-05, Normal(243814.33397817996, 90250.66184475794)]
[0.10558269057976878, Normal(124.63582645813133, 82.69111532861297)]
]
"inbytes_mix" : [
[0.9681518696819149, Log-Normal(5.815454970714217, 0.6872158641268232)]
[0.0036632006939336472, Log-Normal(9.506251938128308, 2.2902429081095947)]
[1.6331380044652412e-05, Log-Normal(16.49726879951374, 0.0373937463838301)]
[0.021765256614208126, Log-Normal(8.736541154252452, 0.15892574023921358)]
[3.1899850289899824e-05, Log-Normal(16.71705297237083, 0.06730477029530364)]
[0.006371441779608665, Normal(2798.2278480797568, 787.7161771633525)]
]
"outbytes_mix" : [
[0.9548012348720477, Normal(472.8877016906775, 263.4835101159007)]
[0.00056240001800928, Normal(12123470.142553022, 15015674.770856496)]
[0.0027474635396387763, Normal(54535.22906247433, 81761.00706093066)]
[0.04188890157030425, Normal(2507.7327537943615, 1553.2681183838968)]
]
```

Προφίλ 5

```
"duration_mix" : [
[0.17367133303133306, Log-Normal(1.7381205325838154, 0.13286133898080332)]
[0.28942903608474957, Log-Normal(9.615913212109682, 0.0001704765781762754)]
[9.6233208424054 08e-05, Log-Normal(11.66427537068512, 0.06212073402906307)]
[0.07204040309045381, Log-Normal(6.724926969921522, 0.21687346348134814)]
[2.3092049524025846e-05, Normal(68874.13121346627, 1376.1193251657817)]
[0.2961648914040777, Normal(110.1400711270911, 19.294223203069425)]
[0.14407795096130427, Normal(197.69802204358925, 143.98892862715195)]
[0.0242811478124 50454, Normal(8420.891302568349, 4678.741622383191)]
[0.00021591235768300725, Normal(33566.974766516476, 13019.134158352477)]
]
"interval_mix" : [
[0.2827768232391855, Log-Normal(7.487661384176024, 1.121362995532371)]
[0.07034498621157602, Log-Normal(9.676926057776157, 0.21259224753979925)]
[0.007940267694661229, Log-Normal(10.526062636584522, 0.05183174561425 11)]
[0.11890359467186495, Log-Normal(2.7413776587923357, 1.6677435520636712)]
[0.2934771713238141, Log-Normal(8.962704274785429, 0.4995420158391068)]
[0.11632607321576942, Normal(506.2517687758342, 0.9489712074111835)]
[0.005463194501958 54, Normal(40000.63171635407, 1.1406576073571244)]
[0.10476788914117016, Normal(24603.282839016367, 5557.162546746535)]
]
"inbytes_mix" : [
[0.9992373867231336, Log-Normal(5.361295912145171, 1.1030805804643458)]
[0.0007626132768664847, Normal(566168.5066034413, 1945675.4803759356)]
]
"outbytes_mix" : [
[0.7676590630571668, Normal(98.23710234414484, 67.7645316585102)]
[0.002017022339110871, Normal(3216670.3427723604, 7596852.820715463)]
[0.012133633112676415, Normal(13011.449475639061, 8579.08771793834)]
[0.21819028149104597, Normal(4949.104797542689, 1583.0460213632255)]
]
```

Προφίλ 6

```
"duration_mix" : [
[0.5679325625275506, Log-Normal(1.0141789124183727, 0.3084885492045814)]
[0.05257874125405323, Log-Normal(9.800393211225627, 1.3502111191745778)]
[0.0029003885024539568, Log-Normal(12.38633988229951, 0.001358870869503029)]
[0.00012862845557810865, Log-Normal(13.080627671320114, 0.0008454537630982784)]
[0.251882237042194, Log-Normal(3.8145561102874317, 1.9379540295572997)]
[0.03258882814919209, Normal(118282.45594406931, 1857.446467081796)]
[0.00038968238127057124, Normal(359662.71353625826, 307.96898423122593)]
[0.03888208592250391, Normal(33422.72979803486, 4358.161014813424)]
[0.028849670262434266, Normal(58892.904319596426, 20568.188723786505)]
[0.023867175502769088, Normal(64.98702391474922, 2.5903514498538214)]
]
"interval_mix" : [
[0.2878475072396667, Log-Normal(8.823689307473954, 0.0672791024659978)]
[7.677483468805966e-05, Log-Normal(16.123369999258312, 0.002468097960720993)]
[8.376239211008059e-05, Log-Normal(16.640620583730914, 0.0017610355295504083)]
[0.3500838484448139, Log-Normal(9.644749769522162, 2.279350888144236)]
[0.034869543856710074, Normal(70813.21890446993, 4976.136958154863)]
[0.040028537384854415, Normal(176753.56296207363, 77369.79329354978)]
[0.28701002584715707, Normal(8641.568817310272, 1469.8450140806306)]
]
"inbytes_mix" : [
[0.22372761067953756, Log-Normal(9.948049173227018, 4.872006514605817)]
[0.018668800790424784, Log-Normal(19.72683202839113, 0.1574241607220529)]
[0.003183628601551328, Log-Normal(20.346342459957686, 0.16789864965861045)]
[0.032653708741122514, Normal(137890188.47846806, 2414727.464856438)]
[2.5621536667528096e-05, Normal(577206093.7243531, 200700645.30298716)]
[0.6991880416127357, Normal(682.0000000000462, 0.0003957001489224225)]
[0.02255258803796045, Normal(202816984.839549, 42106436.85861039)]
]
"outbytes_mix" : [
[0.7430491555916965, Normal(143.37614275182148, 6.344382054720168)]
[0.015999078212353808, Normal(6850344.107902865, 6476910.640238525)]
[0.2409517661959497, Normal(1161605.1159956171, 1674444.8784980646)]
]
```

Προφίλ 7

```
"duration_mix" : [
[0.35028226945027335, Log-Normal(3.843153615032494, 0.11825922391894067)]
[0.016825080461888662, Log-Normal(9.618041624800739, 0.0178393303702256)]
[0.0013441135037005059, Log-Normal(10.980034854360778, 0.35766670041013937)]
[0.29564294073000436, Log-Normal(5.339857983814471, 0.5568834544337632)]
[0.3076908903609679, Log-Normal(8.607579055223379, 0.7204017035053225)]
[0.0003070818353786328, Normal(118116.57334462884, 522.9444972961027)]
[0.02790762365778653, Normal(14784.461792554854, 36.624392567079326)]
]
"interval_mix" : [
[0.09643993323974186, Log-Normal(10.68042451257495, 0.07088386138716157)]
[0.004013699057152498, Log-Normal(14.257717563353772, 0.07162537581396392)]
[0.8993745085635639, Log-Normal(7.202726875531586, 2.7194275318797314)]
[0.00017185913954181068, Normal(17988674.13261719, 22295.6160937526)]
]
"inbytes_mix" : [
[0.5511492225220713, Log-Normal(4.143915957321891, 0.5518919838951083)]
[0.4435009637806921, Log-Normal(7.4081670710824445, 0.9984997250680289)]
[0.003860569292669935, Normal(32541.678940784812, 14096.272899102603)]
[0.0014892444045664956, Normal(133826.26248967726, 79335.6591883326)]
]
"outbytes_mix" : [
[0.9513115433117892, Log-Normal(5.586430333358825, 1.8182509926331916)]
[0.005957988309653997, Normal(523086.4689885836, 355323.26184020704)]
[0.02465657705146922, Normal(33869.51330543251, 17472.634393230914)]
[0.01807389132708764, Normal(124461.59416426907, 60801.00320867779)]
]
```

Παράρτημα Β΄

Μοντελοποίηση όλων των χαρακτηριστικών

Παρακάτω παρατίθενται οι μείξεις των κατανομών, δηλαδή οι παράμετροι και το βάρος της κάθε συνιστώσας, για κάθε προφίλ, σύμφωνα με τη 2η μέθοδο της μοντελοποίησης όλων των χαρακτηριστικών.

Προφίλ 1

```
Component 01 Weight: 0.14637307663064936
Component 01 Mean:   [30632.211753410433, 394.56653556693925, 25483.635806809696, 5246.41521811486 75]
Component 01 Cov1:   [37676.28007026112, 45737.40336857988, 699972.2924567403, 96441.65305804281]
Component 01 Cov2:   [45737.40336857988, 233299.33932737153, 1600104.12012451, 188492.75988179978]
Component 01 Cov3:   [699972.2924567403, 1600104.12012451, 54862422.76955043, 6332752.4475746555]
Component 01 Cov4:   [96441.65305804281, 188492.75988179978, 6332752.4475746555, 879298.6053705284]
Component 02 Weight: 4.4292864419542045e-05
Component 02 Mean:   [33338.33333333331, 14726041.666666655, 72379.33333333328, 364877.9999999997]
Component 02 Cov1:   [50285644.22222318, 3361898044.777775, 122363093.88888879, 824883051.3333328]
Component 02 Cov2:   [3361898044.777775, 260422410686.88867, 3670104136.7777753, 34156185539.33331]
Component 02 Cov3:   [122363093.88888879, 3670104136.7777753, 868307828.2222224, 4662579231.999996]
Component 02 Cov4:   [824883051.3333328, 34156185539.33331, 4662579231.999996, 25889212082.66665]
Component 03 Weight: 4.4292864419542045e-05
Component 03 Mean:   [29072.666666666646, 27581385.333333313, 39290.33333333331, 146181.66666666657]
Component 03 Cov1:   [406705250.8888896, -7655216044.222216, 329231585.7777775, 1323686129.8888881]
Component 03 Cov2:   [-7655216044.222216, 163028587387.55542, -6324476592.444439, -24784479818.222206]
Component 03 Cov3:   [329231585.7777775, -6324476592.444439, 267374470.22222307, 1070656434.4444436]
Component 03 Cov4:   [1323686129.8888881, -24784479818.222206, 1070656434.4444436, 4309045409.555553]
Component 04 Weight: 0.0001122796527891758 2
Component 04 Mean:   [37873.53942501542, 4511252.892930878, 35319.629319626845, 88386.79295237333]
Component 04 Cov1:   [189987293.56397048, -3255895638.8484764, 21196822.625809625, 303421664.97418976]
Component 04 Cov2:   [-3255895638.8484764, 134546289137.22052, 3491271290.545868, 2333442649.8001823]
Component 04 Cov3:   [21196822.625809625, 3491271290.545868, 212673119.0080054, 498812500.3492284]
Component 04 Cov4:   [303421664.97418976, 2333442649.8001823, 498812500.3492283, 1908515516.7606273]
Component 05 Weight: 5.9057152559389 38e-05
Component 05 Mean:   [29071.749999999985, 23837601.24999999, 47473.99999999998, 190672.4999999999]
Component 05 Cov1:   [464054703.68750083, -7817560793.187497, 556954486.2499998, 1969847432.874999]
Component 05 Cov2:   [-7817560793.187497, 161161144185.18744, -9426181950.249996, -31554468624.124985]
Component 05 Cov3:   [556954486.2499998, -9426181950.249996, 830273379.0000006, 3171120454.4999986]
Component 05 Cov4:   [1969847432.874999, -31554468624.124985, 3171120454.4999986, 12501369227.249996]
Component 06 Weight: 0.06294435848358171
Component 06 Mean:   [20786.546031841066, 300979.9286723258, 38218.546975634155, 29381.47802257754]
Component 06 Cov1:   [202398407.21048093, 6894035.036012322, 2845543.3635899527, 3211537.8121178597]
Component 06 Cov2:   [6894035.036012322, 2373172.5335814, 211928.30238434873, 958823.7845914481]
Component 06 Cov3:   [2845543.3635899527, 211928.30238434885, 46981754.113920555, 8715318.4589725]
Component 06 Cov4:   [3211537.8121178593, 958823.7845914481, 8715318.4589725, 6180881.933013705]
Component 07 Weight: 4.4292864419542045e-05
Component 07 Mean:   [28504.666666666646, 11266658.333333325, 47574.3333333333, 237443.33333333317]
Component 07 Cov1:   [224811172.22222304, 6082238134.44444, 166141294.44444433, 740668489.444444]
Component 07 Cov2:   [6082238134.44444, 183287499289.55545, 1237828214.8888876, 12672400051.888878]
Component 07 Cov3:   [166141294.44444433, 1237828214.8888876, 689085534.8888893, 1828128447.8888874]
Component 07 Cov4:   [740668489.444444, 12672400051.888878, 1828128447.8888874, 5336791626.8888855]
Component 08 Weight: 0.0024274424449572 38
Component 08 Mean:   [27042.661175090565, 1350629.8241857912, 25205.4858704786, 54285.88852034771]
Component 08 Cov1:   [372834808.4343802, -2687403214.5972486, 195626078.31444594, 466528348.9003104]
Component 08 Cov2:   [-2687403214.5972495, 1160470891775.987, -1612185009.091733, -1615584615.0985975]
Component 08 Cov3:   [195626078.31444594, -1612185009.091733, 247446946.82319754, 517149019.27202624]
Component 08 Cov4:   [466528348.9003104, -1615584615.0985966, 517149019.27202624, 1393527249.26105]
Component 09 Weight: 7.382077386572882e-05
Component 09 Mean:   [50727.9303107356, 9007741.129909884, 27214.81130521603, 109184.90031817651]
Component 09 Cov1:   [1784919758.2298493, -1601265866.0392663, 357023190.87895626, 3717018922.3163714]
Component 09 Cov2:   [-1601265866.0392668, 13211660709.678303, -391415009.34531134, -9180462207.2216]
Component 09 Cov3:   [357023190.87895626, -391415009.34531134, 136257890.28082868, 865040002.9508007]
Component 09 Cov4:   [3717018922.316372, -9180462207.2216, 865040002.9508007, 10799766884.97706]
Component 10 Weight: 4.4292864419542045e-05
Component 10 Mean:   [51176.66666666663, 19006848.999999985, 87264.99999999994, 321616.66666666645]
Component 10 Cov1:   [68267944.22222318, -803152002.9999994, 350839495.66666645, 954499164.8888881]
Component 10 Cov2:   [-803152002.9999994, 20265643432.666653, -4139308723.6666636, -263425742.33333313]
Component 10 Cov3:   [350839495.66666645, -4139308723.6666636, 1803031128.6666663, 4893369196.999996]
```

```
Component 10 Cov4:   [954499164.8888881, -263425742.33333313, 4893369196.999996, 24462684580.222206]
Component 11 Weight: 0.01628313041594579
Component 11 Mean:   [43196.43544462021, 47149.83399153723, 27811.049473561034, 74952.58398648405]
Component 11 Cov1:   [972245505.064602, 245090940.3328437, 336798805.2809398, 794620768.9901923]
Component 11 Cov2:   [245090940.3328437, 1132316026.2555432, 25658001.045347024, 79132982.97891635]
Component 11 Cov3:   [336798805.2809398, 25658001.045347027, 325253907.6439664, 752600033.8592464]
Component 11 Cov4:   [794620768.9901923, 79132982.97891635, 752600033.8592464, 2783823946.535874]
Component 12 Weight: 0.024405089715376926
Component 12 Mean:   [35201.742353015325, 85.97740891477851, 26748.894785400093, 62201.813853463354]
Component 12 Cov1:   [514251789.96290904, 25284.65385362598, 208180831.5758783, 231330744.0003218]
Component 12 Cov2:   [25284.653853625597, 28090.029286573834, -88320.65764016831, 3415919.309562903]
Component 12 Cov3:   [208180831.5758783, -88320.65764016833, 307357955.5885457, 513529576.5624242]
Component 12 Cov4:   [231330744.00032178, 3415919.3095629034, 513529576.5624242, 5434155805.860799]
Component 13 Weight: 4.4292864419542045e-05
Component 13 Mean:   [50312.99999999996, 8008448.333333327, 25762.99999999998, 150308.33333333323]
Component 13 Cov1:   [1410368341.9999998, 12669516873.333324, 669158545.9999995, 4347778871.33333]
Component 13 Cov2:   [12669516873.333324, 119169762798.22212, 6170804493.333329, 37115553691.55553]
Component 13 Cov3:   [669158545.9999995, 6170804493.333329, 322244914.66666746, 2004986450.666665]
Component 13 Cov4:   [4347778871.33333, 37115553691.55553, 2004986450.666665, 14106241963.555546]
Component 14 Weight: 0.2286624141996084
Component 14 Mean:   [10012.135928070307, 27762.052927923734, 2138.3988148429885, 6766.395230776745]
Component 14 Cov1:   [36.52625097925813, -1701.9161799683025, -38.492955335005824, 1672.1726716611856]
Component 14 Cov2:   [-1701.9161799683025, 43669926.925974645, 83291.72704879723, 682934.988965447]
Component 14 Cov3:   [-38.492955335005824, 83291.72704879723, 30459.35195727467, 118756.47638802645]
Component 14 Cov4:   [1672.1726716611856, 682934.9889654468, 118756.47638802645, 7373514.661110279]
Component 15 Weight: 0.00026797780353232485
Component 15 Mean:   [39819.47649007005, 6009106.23516529, 34707.92586831874, 82276.83855624574]
Component 15 Cov1:   [409838194.0664685, -3442681923.284122, 267386258.51526833, 605422588.4020195]
Component 15 Cov2:   [-3442681923.284123, 479641929116.7029, -3639374398.2570457, 3086394035.712399]
Component 15 Cov3:   [267386258.51526833, -3639374398.2570457, 288539159.66172296, 423696246.3336746]
Component 15 Cov4:   [605422588.4020194, 3086394035.7124004, 423696246.3336746, 1771309177.44593]
Component 16 Weight: 0.03657159257371108
Component 16 Mean:   [1060.5603515212968, 790.8247846340613, 28724.148355650075, 5347.506476557711]
Component 16 Cov1:   [37242.87927365778, 35995.925619304035, 423825.90563003696, 72226.08799101104]
Component 16 Cov2:   [35995.925619304035, 287626.985801219, 966951.4604355937, 138553.40144888771]
Component 16 Cov3:   [423825.90563003696, 966951.4604355937, 33573459.03794194, 4165394.4913272825]
Component 16 Cov4:   [72226.08799101104, 138553.40144888771, 4165394.4913272825, 631607.6537339617]
Component 17 Weight: 0.0026701765537714708
Component 17 Mean:   [68614.57056038505, 521828.7114595743, 61553.00232531121, 207410.73175988553]
Component 17 Cov1:   [1113063488.8490171, -2461196833.253399, 730209317.7697166, 1107739439.0960696]
Component 17 Cov2:   [-2461196833.2533994, 407199297159.5352, 1899593956.2840505, 7676769435.304276]
Component 17 Cov3:   [730209317.7697166, 1899593956.2840505, 1205448350.1529088, 2407253599.4484987]
Component 17 Cov4:   [1107739439.0960696, 7676769435.304276, 2407253599.4484987, 13367327235.804409]
Component 18 Weight: 0.04316236021786701
Component 18 Mean:   [18205.392925682052, 19713.06441254792, 16857.79951323856, 16444.60125711597]
Component 18 Cov1:   [155172272.1850924, -48140626.212113865, 147583769.3210289, 83682292.22574174]
Component 18 Cov2:   [-48140626.212113865, 133965988.41389255, -71450548.72740649, -37096886.5798168]
Component 18 Cov3:   [147583769.3210289, -71450548.72740649, 270911653.937897, 145330218.4185533]
Component 18 Cov4:   [83682292.22574173, -37096886.5798168, 145330218.4185533, 112823714.77059083]
Component 19 Weight: 5.905715255938938e-05
Component 19 Mean:   [13314.249999999995, 21989103.24999999, 18001.499999999993, 59058.49999999997]
Component 19 Cov1:   [182117201.18750092, -3484559964.0624995, 178443119.3749999, 433482836.62499976]
Component 19 Cov2:   [-3484559964.0624995, 276863762411.1874, -379640615.87499946, 9703691575.624996]
Component 19 Cov3:   [178443119.3749999, -379640615.87499946, 275413375.2500009, 1100054531.2499995]
Component 19 Cov4:   [433482836.62499976, 9703691575.624996, 1100054531.2499995, 5614189126.749998]
Component 20 Weight: 0.10409608604781778
Component 20 Mean:   [7393.98549887872095, 24796.187843337633, 1112.6048996372506, 1982.3997908643742]
Component 20 Cov1:   [10575695.425545977, -5293154.844187964, 990228.7944529973, 2218152.844620202]
Component 20 Cov2:   [-5293154.844187964, 710249390.8815563, 771643.7694763377, 864668.986384441]
Component 20 Cov3:   [990228.7944529973, 771643.7694763376, 607530.4054475462, 1637500.0466818796]
Component 20 Cov4:   [2218152.844620202, 864668.9863844407, 1637500.0466818796, 4633199.238941943]
Component 21 Weight: 4.4292864419542045e-05
Component 21 Mean:   [20579.999999999985, 17159051.33333332, 18752.666666666653, 110266.33333333326]
Component 21 Cov1:   [390952384.6666674, -5698715.999999678, 37326537.33333331, 1084197434.3333325]
Component 21 Cov2:   [-5698715.999999678, 71742604070.22218, -3368348822.2222195, -11544232494.444437]
Component 21 Cov3:   [37326537.33333331, -3368348822.2222195, 161658434.88888976, 644693060.1111106]
Component 21 Cov4:   [1084197434.3333325, -11544232494.444437, 644693060.1111106, 4859242261.555553]
Component 22 Weight: 0.015959217037393947
Component 22 Mean:   [20178.40986937034, 185111.75030746878, 23774.71278439353, 31209.591064672808]
Component 22 Cov1:   [250406200.2987916, 357452607.6367507, 45279258.250138626, 93091445.92230952]
Component 22 Cov2:   [357452607.6367507, 13050411829.937557, 646253184.8804895, 150230482.0947044]
Component 22 Cov3:   [45279258.25013862, 646253184.8804895, 253378443.8691497, 181761342.6226219]
Component 22 Cov4:   [93091445.9223095, 150230482.09470442, 181761342.6226219, 428240234.5395048]
Component 23 Weight: 0.00010335001697893138
Component 23 Mean:   [34712.14285714285, 13608568.14285714, 55372.57142857141, 210673.1428571428]
Component 23 Cov1:   [411860497.2653071, 4108562501.836734, 212411726.632653, 454858.4081632068]
Component 23 Cov2:   [4108562501.836734, 100516754600.97955, -917750709.5102032, -28900321869.73468]
Component 23 Cov3:   [212411726.632653, -917750709.5102032, 1183248337.3877556, 4897390214.9183655]
Component 23 Cov4:   [454858.4081632068, -28900321869.73468, 4897390214.9183655, 30550780260.122437]
Component 24 Weight: 0.032291959968849634
Component 24 Mean:   [18300.142174739925, 1115.7273659317632, 20775.713548877295, 5957.2485943041465]
Component 24 Cov1:   [241854242.09887618, 2824725.577421544, 3741440.9363465304, 747275.3588876391]
Component 24 Cov2:   [2824725.5774215437, 2081659.121025495, -1038391.4687322463, -893484.5996979834]
Component 24 Cov3:   [3741440.9363465314, -1038391.4687322463, 140775930.28048423, 28329752.461771797]
Component 24 Cov4:   [747275.3588876384, -893484.5996979834, 28329752.461771797, 8447106.111494038]
Component 25 Weight: 5.905715255938938e-05
Component 25 Mean:   [42266.49999999998, 10088122.749999996, 45902.24999999998, 165181.7499999999]
Component 25 Cov1:   [1008506396.7500005, 5885451941.874997, -82949536.87499999, 288349309.3749999]
Component 25 Cov2:   [5885451941.874997, 38313444561.687485, -829222211.1874995, 1322995583.437499]
Component 25 Cov3:   [-82949536.87499999, -829222211.1874995, 70817044.18750097, 454651386.8124998]
Component 25 Cov4:   [288349309.3749999, 1322995583.437499, 454651386.8124998, 5999482498.687497]
Component 26 Weight: 0.0048705075043246945
Component 26 Mean:   [60991.14497822379, 9927.002667585633, 63202.568302067826, 286064.0226189172]
Component 26 Cov1:   [1444615203.3722486, 117466029.67381762, 892736782.5939871, 617620417.4318831]
Component 26 Cov2:   [117466029.67381762, 187316132.34832084, 64409509.42960117, -186555459.86341265]
Component 26 Cov3:   [892736782.5939871, 64409509.42960117, 1248932617.9970875, 2094967028.234293]
Component 26 Cov4:   [617620417.4318831, -186555459.86341268, 2094967028.234293, 36916998148.774925]
Component 27 Weight: 0.27822317416222364
Component 27 Mean:   [16526.69354353561, 1.0332649420432132, 25292.061170202294, 4998.4746925454565]
Component 27 Cov1:   [220122328.77369082, -646.404855673355, 8774333.076942062, 3669567.556013161]
Component 27 Cov2:   [-646.404855673355, 1.1566684931630387, 801.6439809735258, 136.40858178420288]
Component 27 Cov3:   [8774333.076942062, 801.6439809735261, 79611747.34492914, 11784817.582355354]
Component 27 Cov4:   [3669567.556013161, 136.40858178420029, 11784817.582355354, 1997093.7032008464]
Component 28 Weight: 5.905715255938938e-05
Component 28 Mean:   [24035.99999999999, 25248307.24999999, 46661.24999999998, 209891.9999999999]
Component 28 Cov1:   [90282361.00000095, 2529113138.249999, -50795829.74999998, -405548933.99999976]
Component 28 Cov2:   [2529113138.249999, 205743605657.6874, 2962341475.1874986, 28772689859.49999]
Component 28 Cov3:   [-50795829.74999998, 2962341475.1874986, 171461903.18750092, 1558210881.499999]
Component 28 Cov4:   [-405548933.99999976, 28772689859.49999, 1558210881.499999, 15766327699.999992]
```

Προφίλ 2

```
Component 01 Weight: 0.054570107091064905
Component 01 Mean:   [7483.34749156946, 19940.0071768437, 1165.835035847162, 943.316569481082]
Component 01 Cov1:   [7422885.447269444, −434700.1078030143, 661931.6264811676, 477380.6914086701]
Component 01 Cov2:   [−434700.1078030145, 176584762.66188076, −154022.40444308886, −52766.17646875505]
Component 01 Cov3:   [661931.6264811676, −154022.40444308886, 340104.6040029391, 232408.09527916583]
Component 01 Cov4:   [477380.6914086701, −52766.17646875502, 232408.09527916586, 176871.7799828426]
Component 02 Weight: 7.301561804069907e−06
Component 02 Mean:   [40047.99999999991, 23180.99999999995, 456379.999999999, 22017667.99999995]
Component 02 Cov1:   [1.0000000000076e−06, 4.44692297308e−21, 8.63973606199e−20, 4.2283884726 93e−18]
Component 02 Cov2:   [4.4469229730e−21, 1.0000000000025e−06, 5.0398460361e−20, 2.466559942404e−18]
Component 02 Cov3:   [8.639736061993844e−20, 5.0398460361e−20, 1.00000009791e−06, 4.7921736023e−17]
Component 02 Cov4:   [4.2283884726 93e−18, 2.4665599424045e−18, 4.792173602385e−17, 1.0000000023 45e−06]
Component 03 Weight: 7.301561804069907e−06
Component 03 Mean:   [27.99999999999994, 13403406.99999997, 560.9999999999987, 8310.999999999982]
Component 03 Cov1:   [1e−06, 1.799945012915384e−21, 7.552869850948038e−26, 1.0985992510469875e−24]
Component 03 Cov2:   [1.799945012915384e−21, 1.000000088817 84e−06, 3.7269449679 18e−20, 5.4210108e−19]
Component 03 Cov3:   [7.552869850948038e−26, 3.7269449679 18913e−20, 1e−06, 2.2747466845 20821e−23]
Component 03 Cov4:   [1.098599251046e−24, 5.421010862427e−19, 2.274746684520e−23, 1.0000000000 04e−06]
Component 04 Weight: 0.0003222097468323574
Component 04 Mean:   [22904.066823992678, 194989.40402784338, 25321.442951644793, 121982.39316479156]
Component 04 Cov1:   [970445688.8124298, −1114178900.757089, 1246019241.736558, 9659069188.279032]
Component 04 Cov2:   [−1114178900.757089, 4176723413.1337724, −1482442904.1709332, −10274486500.198236]
Component 04 Cov3:   [1246019241.736558, −1482442904.1709332, 1668235064.6167715, 12198923587.330223]
Component 04 Cov4:   [9659069188.279032, −10274486500.198238, 12198923587.330223, 1019319255 37.90605]
Component 05 Weight: 0.015550214955570003
Component 05 Mean:   [38497.86603107771, 32661.291511053732, 38164.072687201755, 138244.72450288368]
Component 05 Cov1:   [185437312.10098416, 3569601.3870495227, 150056768.76337928, 436439072.2689628]
Component 05 Cov2:   [3569601.3870495236, 298467787.2044071, −6381320.673555985, −114238747.59322664]
Component 05 Cov3:   [150056768.76337925, −6381320.673555985, 273698788.9236957, 1007562585.3177757]
Component 05 Cov4:   [436439072.2689628, −114238747.59322669, 1007562585.3177756, 4617210617.945277]
Component 06 Weight: 3.650780759484221e−05
Component 06 Mean:   [26.200000695155655, 4403690.818621847, 633.3999965670494, 8593.400010735446]
Component 06 Cov1:   [340.1600014040714, 2471665.4009858994, −630.2799611612218, −3085.8803117638163]
Component 06 Cov2:   [2471665.4009858994, 159042225373.4029, −2666603.9435215495, 105669780.27324004]
Component 06 Cov3:   [−630.2799611612218, −2666603.9435215495, 3087.0397953503893, 3441.4410853235204]
Component 06 Cov4:   [−3085.8803117638154, 105669780.27324004, 3441.4410853235204, 250989.04684659102]
Component 07 Weight: 0.2868600381479744
Component 07 Mean:   [7099.971665076198, 65853.02721530196, 664.2226208508105, 430.99999999999415]
Component 07 Cov1:   [5965526.666325286, 6385962.732164029, 989.6639436456899, 5.126396666707645e−22]
Component 07 Cov2:   [6385962.732164026, 2069413747.357959, 555.2676718299585, 4.5286063171006 51e−21]
Component 07 Cov3:   [989.6639436456896, 555.2676718299588, 1.0274825702610024, 5.138526064081866e−23]
Component 07 Cov4:   [5.1261810680 67472e−22, 4.528606317100651e−21, 5.138532380448277e−23, 1e−06]
Component 08 Weight: 0.07039887179776126
Component 08 Mean:   [7948.7438804871645, 9077.675756257055, 1780.69548845162 22, 5296.1634428372845]
Component 08 Cov1:   [14995197.36665105, 4267615.81722993, 2190797.4259063425, 218456.62002554312]
Component 08 Cov2:   [4267615.817229929, 1445740 7.003943235, 621892.0355436017, −193840.15467826274]
Component 08 Cov3:   [2190797.4259063425, 621892.0355436017, 707244.398485107, 157694.61685210478]
Component 08 Cov4:   [218456.6200255432, −193840.15467826274, 157694.61685210484, 5123494.1764723025]
Component 09 Weight: 1.4603123608139797e−05
Component 09 Mean:   [20840.999999999978, 5011.9999999999945, 143771.49999999985, 5180210.999999994]
Component 09 Cov1:   [33535681.00000096, 19782055.999999978, −165596540.49999982, 706189285.9999992]
Component 09 Cov2:   [19782055.999999978, 11669056.000000987, −97682227.9999999, 416567535.9999995]
Component 09 Cov3:   [−165596540.49999982, −97682227.9999999, 817702620.25, −3487106842.999996]
Component 09 Cov4:   [706189285.9999992, 416567535.9999995, −3487106842.999996, 14870826915.999985]
Component 10 Weight: 0.025230038550161003
Component 10 Mean:   [20376.5913758542, 53579.95230911007, 16486.846338631 17, 52297.984507284586]
Component 10 Cov1:   [64075144.95002375, 45749420.877922356, 40432459.46936431, 125068701.35637291]
Component 10 Cov2:   [45749420.877922356, 775588670.9574221, 3330220.8967522606, 96458774.21368448]
Component 10 Cov3:   [40432459.46936431, 3330220.896752262, 65272222.52740759, 219155416.67581332]
Component 10 Cov4:   [125068701.3563729, 96458774.21368447, 219155416.67581332, 1069339766.9549141]
Component 11 Weight: 0.0022420686542149212
Component 11 Mean:   [79679.79241454393, 53109.815412930184, 107729.30422313347, 458818.3174036348]
Component 11 Cov1:   [3260256385.5442696, 103551658.65238543, 4575634880.454824, 13914667288.9282]
Component 11 Cov2:   [103551658.65238541, 1546533497.1960056, 85909319.68035018, −2270079755.7511907]
Component 11 Cov3:   [4575634880.454824, 85909319.68035012, 7596627690.057135, 23549930465.549385]
Component 11 Cov4:   [13914667288.9282, −2270079755.7511907, 23549930465.549385, 99628919268.83766]
Component 12 Weight: 0.0001693845114714508
Component 12 Mean:   [730.7524923370236, 1199213.4168064715, 1054.264837629689, 9089.044206099392]
Component 12 Cov1:   [9813517.158438647, 804240045.1478729, 6171927.657391141, 9199526.480582861]
Component 12 Cov2:   [804240045.1478729, 427481302677.9228, 519822038.2163469, 608659125.4633207]
Component 12 Cov3:   [6171927.657391141, 519822038.2163469, 3889663.0621555173, 5784607.880493567]
Component 12 Cov4:   [9199526.480582861, 608659125.4633207, 5784607.880493567, 9077417.536651492]
Component 13 Weight: 0.0005549211359551122
Component 13 Mean:   [766.2424451245188, 291.87113963590366, 22802.068483678562, 1599777.1298682324]
Component 13 Cov1:   [68631.47680774826, 6745.587179352889, 571045.5175800456, 2943933.5421507843]
Component 13 Cov2:   [6745.587179352889, 159857.1022488004, 544405.3476602602, 10454456.786810266]
Component 13 Cov3:   [571045.5175800456, 544405.3476602602, 37718638.093716286, 690522439.4422845]
Component 13 Cov4:   [2943933.5421507843, 10454456.786810266, 690522439.4422845, 27171007872.175755]
Component 14 Weight: 0.14558842481015744
Component 14 Mean:   [10974.515740434292, 18981.121474241227, 3960.6753767937953, 8446.572030244255]
Component 14 Cov1:   [13357184.433077395, −1602917.2783738058, 4099675.180943098, 6101969.306719737]
Component 14 Cov2:   [−1602917.2783738058, 111242163.0033696, −3290690.7314869678, −8711744.13401573]
Component 14 Cov3:   [4099675.180943098, −3290690.7314869678, 3676636.733641966, 3978335.5648145694]
Component 14 Cov4:   [6101969.306719737, −8711744.134015728, 3978335.5648145694, 15381524.416910544]
Component 15 Weight: 0.05107435335579649
Component 15 Mean:   [3473.2597355963235, 66211.42320930904, 627.3253603408748, 2536.0672510241097]
Component 15 Cov1:   [12672046.456933104, 147290177.50788534, 174471.3348946692, −6427180.41271594]
Component 15 Cov2:   [147290177.50788534, 2396178660.0118375, 2606804.5547601087, −97067041.8374978]
Component 15 Cov3:   [174471.3348946692, 2606804.5547601087, 3912.318985144329, −117154.28385786327]
Component 15 Cov4:   [−6427180.41271594, −97067041.8374978, −117154.28385786327, 4399261.699220196]
Component 16 Weight: 0.05052564544930622
Component 16 Mean:   [10621.096605933268, 51971.930448427025, 6016.397468487334, 23816.64072254969]
Component 16 Cov1:   [8299559.289812186, 599334.5928077651, 3426726.8501522015, 11032708.687485585]
Component 16 Cov2:   [599334.5928077647, 801091734.4945929, −29301577.847260114, −184283428.26452246]
Component 16 Cov3:   [3426726.8501522015, −29301577.847260114, 10591584.154974617, 48239680.796890035]
Component 16 Cov4:   [11032708.687485585, −184283428.2645225, 48239680.79689004, 269584585.5400356]
Component 17 Weight: 0.08175428489810524
Component 17 Mean:   [17972.454243379718, 17635.987026407944, 10658.648598997343, 30869.978233660222]
Component 17 Cov1:   [51973371.20975045, 759217.5498932478, 20610482.336583592, −5210913.250894684]
Component 17 Cov2:   [759217.5498932478, 86833211.88425049, 4040185.723250711, 20204058.156970043]
Component 17 Cov3:   [20610482.336583592, 4040185.723250711, 23439027.00835489, 33103475.603492986]
Component 17 Cov4:   [−5210913.250894683, 20204058.156970046, 33103475.60349298, 250643961.1718706]
Component 18 Weight: 0.005593323801496954
Component 18 Mean:   [56394.885518175965, 13086.613319717473, 59533.740619663535, 204594.80530569147]
Component 18 Cov1:   [875354003.1410893, 68375492.41910866, 781795737.457787, 3019191231.6440587]
Component 18 Cov2:   [68375492.41910864, 82368474.59765278, 107582078.22093798, 429769618.45076346]
Component 18 Cov3:   [781795737.4577868, 107582078.22093795, 1280522811.1618152, 4763838538.288699]
Component 18 Cov4:   [3019191231.6440587, 429769618.45076346, 4763838538.288699, 20389274599.191166]
```

```
Component 19 Weight: 0.027371167366612437
Component 19 Mean:   [8948.59088732972, 11801.613060898515, 7096.106574320947, 36642.99433541089]
Component 19 Cov1:   [29779653.675972655, 11438732.103090871, 19888568.803087674, 19666775.35996356]
Component 19 Cov2:   [11438732.103090871, 33262160.462978162, 8191572.26448632, 9875349.239755876]
Component 19 Cov3:   [19888568.803087678, 8191572.264486319, 15225682.110558765, 19329783.03672553]
Component 19 Cov4:   [19666775.35996356, 9875349.239755876, 19329783.03672553, 65436667.18927537]
Component 20 Weight: 0.02358505588798331
Component 20 Mean:   [26160.398706015993, 19848.803442970882, 25953.521320006774, 94303.18332974041]
Component 20 Cov1:   [73660313.77370468, −18407009.61697137, 44314990.64611982, 58914267.23533581]
Component 20 Cov2:   [−18407009.61697137, 78163076.05805974, −13357467.107481917, −8838562.766219078]
Component 20 Cov3:   [44314990.64611982, −13357467.107481917, 45916693.93204095, 85233612.06435744]
Component 20 Cov4:   [58914267.2353358, −8838562.76621908, 85233612.06435744, 277220455.63105685]
Component 21 Weight: 0.0014557854065361303
Component 21 Mean:   [45.4033299933183085, 95166.3021333091, 585.8081838137757, 8333.676384072223]
Component 21 Cov1:   [3664.31941392969 93, 2185058.2652419107, −1431.3871699014549, −519.1405841938365]
Component 21 Cov2:   [2185058.2652419107, 28830613141.89058, −632480.1892841213, −592665.6333971032]
Component 21 Cov3:   [−1431.3871699014549, −632480.1892841209, 4886.21091214436, 1160.030725458989]
Component 21 Cov4:   [−519.1405841938365, −592665.6333971029, 1160.030725458989, 1016.9021623941675]
Component 22 Weight: 1.4603123608139797e−05
Component 22 Mean:   [111.49999999999987, 6288270.4999999935, 543.9999999999994, 8309.49999999999]
Component 22 Cov1:   [4422.250000999996, 10948659.749999987, −1130.4999999999986, 99.74999999999989]
Component 22 Cov2:   [10948659.749999987, 27106823522.24997, −2798905.4999999967, 246962.24999999974]
Component 22 Cov3:   [−1130.4999999999986, −2798905.4999999967, 289.00000099999966, −25.499999999999997]
Component 22 Cov4:   [99.74999999999989, 246962.24999999974, −25.49999999999997, 2.2500009999999975]
Component 23 Weight: 0.015250988797211757
Component 23 Mean:   [43.014851802006355, 0.36425793254503275, 339.8495598914771, 2698.66431436349 88]
Component 23 Cov1:   [4466.595814183842, 14.5145731247468 25, 5642.206645973008, 44869.11644645517]
Component 23 Cov2:   [14.5145731247468 25, 0.2898559400506719, 64.96362227894268, 594.8085715172289]
Component 23 Cov3:   [5642.206645973008, 64.96362227894267, 60451.85509809958, 537872.4332388927]
Component 23 Cov4:   [44869.11644645516, 594.8085715172288, 537872.4332388927, 4905096.957980463]
Component 24 Weight: 0.007477485105975698
Component 24 Mean:   [142.13787893798215, 277.15975309843606, 2241.0581567822014, 106276.64471200046]
Component 24 Cov1:   [31982.099065820734, −156.25718404488555, 363393.29503165226, 22592784.870839518]
Component 24 Cov2:   [−156.25718404488526, 158038.00988706088, −24282.21357232164, −1829826.7235054148]
Component 24 Cov3:   [363393.29503165226, −24282.21357232164, 4963290.779417933, 316064653.850686]
Component 24 Cov4:   [22592784.870839518, −1829826.7235054148, 316064653.850686, 22353048106.459736]
Component 25 Weight: 2.1904685412209688e−05
Component 25 Mean:   [11789.333333333325, 11000.666666666659, 182188.3333333332, 6236544.999999995]
Component 25 Cov1:   [56270566.88888985, 60710536.11111106, 431115445.8888886, −2100646701.666665]
Component 25 Cov2:   [60710536.11111106, 65810581.55555651, 453368565.1111108, −2205341775.666665]
Component 25 Cov3:   [431115445.8888886, 453368565.1111108, 3749740162.888887, −18412782367.666653]
Component 25 Cov4:   [−2100646701.666665, −2205341775.666665, −18412782367.666653, 90454087472.6666]
Component 26 Weight: 7.3015618040699 07e−06
Component 26 Mean:   [46.99999999999989, 15468174.999999966, 560.9999999999987, 8310.999999999982]
Component 26 Cov1:   [1e−06, 3.5734202462290 71e−21, 1.3328593854614186e−25, 1.9387045606711542e−24]
Component 26 Cov2:   [3.5734202462290 71e−21, 1.000000001 12e−06, 4.19281308890e−20, 6.0986372202e−19]
Component 26 Cov3:   [1.3328593854614186e−25, 4.1928130889087777e−20, 1e−06, 2.274746684520821e−23]
Component 26 Cov4:   [1.9387045606711542e−24, 6.09863722023e−19, 2.2747466845e−23, 1.000000000004e−06]
Component 27 Weight: 0.018694874445301293
Component 27 Mean:   [32170.99044110582, 9862.003605780557, 27290.662773314416, 99641.43366789645]
Component 27 Cov1:   [182877214.0600697, 25187323.110111427, 123397339.86370866, 111929362.31893976]
Component 27 Cov2:   [25187323.110111434, 27526602.707360454, 17374308.782441773, 9220727.778896475]
Component 27 Cov3:   [123397339.86370866, 17374308.782441773, 182046187.0860299, 421278313.41004974]
Component 27 Cov4:   [111929362.31893975, 9220727.778896471, 421278313.41004974, 2084193077.8605838]
Component 28 Weight: 0.09306860085150657
Component 28 Mean:   [3138.370611959742, 33907.76980166439, 370.1729331385163, 242.59228694225803]
Component 28 Cov1:   [15012941.620190475, −14648507.748486973, 957973.490288892, 602170.8874347282]
Component 28 Cov2:   [−14648507.748486975, 1230444034.4878237, 443582.06228452845, 203781.8263073704]
Component 28 Cov3:   [957973.490288892, 443582.0622845287, 95249.58192413943, 60791.811405928136]
Component 28 Cov4:   [602170.8874347282, 203781.82630737027, 60791.811405928136, 38941.038517649504]
Component 29 Weight: 0.02255263222869587
Component 29 Mean:   [17869.27425123143, 9305.109338298138, 19015.354730565832, 75937.18633375021]
Component 29 Cov1:   [28283537.856533244, 4309710.595278009, 15133362.528088138, 48192000.59696298]
Component 29 Cov2:   [4309710.595278009, 17569407.599556778, −5899543.096304451, −5866166.0526504]
Component 29 Cov3:   [15133362.528088138, −5899543.096304451, 40046869.43168314, 111205547.70856833]
Component 29 Cov4:   [48192000.59696299, −5866166.0526504, 111205547.70856833, 414980544.4218972]
```

Προφίλ 3

```
Component 01 Weight: 0.1590604355592868
Component 01 Mean:   [13161.455947296245, 1.1821552102623119, 12192.544441676657, 128936.06544178321]
Component 01 Cov1:   [57288402.778789714, −232.97271797408297, 22208475.072318826, 38140106.20104483]
Component 01 Cov2:   [−232.97271797408285, 2.1910880804838 69, −536.5603781601687, −6385.928038900652]
Component 01 Cov3:   [22208475.072318826, −536.5603781601687, 59926815.77150128, 517470502.2555219]
Component 01 Cov4:   [38140106.20104483, −6385.928038900652, 517470502.2555219, 7816544098.838078]
Component 02 Weight: 0.003693859071308501
Component 02 Mean:   [5511.4145581716 35, 1006130.4434961327, 1144.8181863637044, 1353.0250496133358]
Component 02 Cov1:   [23927604.945647452, 10916021 22.7308612, 246290.90654197274, 1555852.311174577]
Component 02 Cov2:   [1091602122.7308612, 1984516254905.2952, 517367070.49767476, 652037098.6598253]
Component 02 Cov3:   [246290.90654197274, 517367070.4976748, 1367271.5875552741, 1667555.0842691786]
Component 02 Cov4:   [1555852.311174577, 652037098.6598253, 1667555.0842691786, 2322124.1281406786]
Component 03 Weight: 0.42175504439080 43
Component 03 Mean:   [100.79318537383766, 104.89176525733103, 1001.815346066642, 4350.805876330045]
Component 03 Cov1:   [2760.2510190750386, −2109.402514504692, 713.1394633091893, −125973.6351207147]
Component 03 Cov2:   [−2109.402514504692, 2119.1956513841587, −570.5479202517292, 108557.73961579367]
Component 03 Cov3:   [713.1394633091893, −570.5479202517292, 1414.2104409810877, −37042.698524444146]
Component 03 Cov4:   [−125973.6351207147, 108557.73961579367, −37042.69852444415, 6321491.658550104]
Component 04 Weight: 6.827569726366806e−05
Component 04 Mean:   [1522.2500000334244, 15060291.750023292, 2103.7500000211035, 6173.500000061692]
Component 04 Cov1:   [1591600.6875045807, 963969157.0614287, 939790.5625006498, 631456.6249479656]
Component 04 Cov2:   [963969157.0614287, 631243967040.4945, 586558666.685664, 575522670.074852]
Component 04 Cov3:   [939790.5625006498, 586558666.685664, 561613.1875003687, 492208.8749624064]
Component 04 Cov4:   [631456.6249479658, 575522670.074852, 492208.8749624065, 8088273.24989814]
Component 05 Weight: 0.017700358981440 46
Component 05 Mean:   [5284.8341739571715, 15946.109703963773, 160.88071413683863, 125.99013301287863]
Component 05 Cov1:   [1549232.5792278699, −3670347.068436245, 30481.718959977323, 2692.1865269834325]
Component 05 Cov2:   [−3670347.068436245, 828239350.7721674, 49271.66206297131, 16335.548562671403]
Component 05 Cov3:   [30481.718959977323, 49271.66206297129, 729.3470063952523, 80.15492652561497]
Component 05 Cov4:   [2692.186526983433, 16335.548562671403, 80.15492652561497, 12.039371591050166]
Component 06 Weight: 6.827569312524274e−05
Component 06 Mean:   [3500.750207801953, 10143472.49128256, 2076.2500836999, 3969.499822412452]
Component 06 Cov1:   [6669910.379886375, −369470803.15765554, 1325855.6060826466, −3800621.2469258783]
Component 06 Cov2:   [−369470803.15765554, 39437537109.116356, 142254357.28370598, 401942485.5952394]
Component 06 Cov3:   [1325855.6060826466, 142254357.283706, 2743257.2383283884, 1196829.4426922905]
Component 06 Cov4:   [−3800621.2469258783, 401942485.5952394, 1196829.4426922905, 5937576.58999725]
Component 07 Weight: 0.06285891599015157
```

```
Component 07 Mean:   [11457.16226470996, 77.13073442049202, 9441.18705355799, 91511.5833676234]
Component 07 Cov1:   [62382606.72910003, 48337.91484922204, 24766438.544448927, 162621984.14993197]
Component 07 Cov2:   [48337.91484922203, 6633.592209852613, -8688.63393815359, -352671.42324326857]
Component 07 Cov3:   [24766438.544448927, -8688.633938153589, 47916823.39393824, 468075827.42360026]
Component 07 Cov4:   [162621984.14993197, -352671.4232432686, 468075827.42360026, 6262672032.914927]
Component 08 Weight: 5.120677294916878e-05
Component 08 Mean:   [9728.666666666659, 7564896.333333327, 5524.666666666662, 17487.666666666653]
Component 08 Cov1:   [21573254.222223204, 626513401.1111107, 8365469.555555549, 58741936.22222217]
Component 08 Cov2:   [626513401.1111107, 43811174608.22218, 611573368.1111107, 3793763432.111108]
Component 08 Cov3:   [8365469.555555549, 611573368.1111107, 8548597.55555655, 52822945.555555515]
Component 08 Cov4:   [58741936.22222217, 3793763432.111108, 52822945.555555515, 330113496.22222304]
Component 09 Weight: 0.013499703580122206
Component 09 Mean:   [19810.505331208285, 3.0151893762981854, 196719.3884946002, 359968.799700497]
Component 09 Cov1:   [165981447.70897746, -3090.847189715279, -566283313.620671, 858761874.9778243]
Component 09 Cov2:   [-3090.847189715279, 13.816093561629598, 85476.92510299035, -224044.2292508366]
Component 09 Cov3:   [-566283313.6206709, 85476.92510299035, 63833072142.610146, -16207367292.649174]
Component 09 Cov4:   [858761874.9778243, -224044.2292508366, -16207367292.649174, 36593629332.67296]
Component 10 Weight: 0.027193714636341765
Component 10 Mean:   [6985.368949764416, 12388.095128560175, 1454.9462512200203, 1217.5943507314341]
Component 10 Cov1:   [49684697.85006959, -9106125.083347794, 1575760.7569185493, 125738.97386129209]
Component 10 Cov2:   [-9106125.083347792, 235294534.42581776, 847199.7224430989, -698151.8346710299]
Component 10 Cov3:   [1575760.7569185495, 847199.7224430987, 388099.79167411424, 159783.29721898516]
Component 10 Cov4:   [125738.9738612921, -698151.8346710299, 159783.29721898516, 215894.86967819824]
Component 11 Weight: 0.00205206596888834236
Component 11 Mean:   [23639.4667285177, 260684.87422755538, 32671.78488820697, 200807.58408452204]
Component 11 Cov1:   [858815588.9335232, -2301340290.5562963, 337051725.3436263, -289900648.3297071]
Component 11 Cov2:   [-2301340290.5562963, 106017148258.26732, -2727545614.6373315, -9480727359.437317]
Component 11 Cov3:   [337051725.3436263, -2727545614.637332, 3184938980.595078, 1278683951.2503345]
Component 11 Cov4:   [-289900648.32970697, -9480727359.437317, 1278683951.2503345, 59485978055.14624]
Component 12 Weight: 0.021248524683819 25
Component 12 Mean:   [237.34695857481887, 17396.510650368913, 3767.816219903007, 48456.588908038066]
Component 12 Cov1:   [24343.68073996629, 711997.0874027371, 358081.2289946917, 3753791.4181761034]
Component 12 Cov2:   [711997.0874027371, 409027855.0048117, 8443987.493602175, 80082210.99663338]
Component 12 Cov3:   [358081.2289946917, 8443987.493602175, 8057868.50844744, 47628457.04412235]
Component 12 Cov4:   [3753791.4181761034, 80082210.99663338, 47628457.04412235, 1993584611.827669]
Component 13 Weight: 1.7068924316389 62e-05
Component 13 Mean:   [8.999999999999998, 20571202.999999955, 39.999999999999915, 39.999999999999915]
Component 13 Cov1:   [1e-06, 8.735027268559953e-22, 1.6660742318267733e-27, 1.6660742318267733e-27]
Component 13 Cov2:   [8.735027268559953e-22, 1.000000000199e-06, 3.811648262644e-21, 3.811648262e-21]
Component 13 Cov3:   [1.6660742318267733e-27, 3.811648262644343e-21, 1e-06, 7.270142102516829e-27]
Component 13 Cov4:   [1.6660742318267733e-27, 3.811648262644343e-21, 7.270142102516829e-27, 1e-06]
Component 14 Weight: 5.12067729491686e-05
Component 14 Mean:   [1518.000000000003, 12289203.333333323, 1917.0000000000023, 4571.333333333327]
Component 14 Cov1:   [2788794.0000010035, 392129608.0000009, 2152611.0000000023, -71848.99999999718]
Component 14 Cov2:   [392129608.0000009, 1250238783 00.22221, 228573339.66666746, 480404042.5555547]
Component 14 Cov3:   [2152611.0000000023, 228573339.66666746, 1740128.6666676675, -575557.6666666633]
Component 14 Cov4:   [-71848.99999999718, 480404042.5555547, -575557.6666666633, 3444509.555556547]
Component 15 Weight: 0.020702301814017745
Component 15 Mean:   [17441.174747172998, 21427.70195969053, 31734.880463914415, 152665.68084895922]
Component 15 Cov1:   [156590213.0717783, -35052540.36335438, 91572917.0186855, 189197020.29566622]
Component 15 Cov2:   [-35052540.36335438, 312329458.2739887, -86420997.13727124, 114089826.34746523]
Component 15 Cov3:   [91572917.0186855, -86420997.13727124, 380178790.6420771, -54336275.82317377]
Component 15 Cov4:   [189197020.29566622, 114089826.34746522, -54336275.823173754, 7570595144.3292265]
Component 16 Weight: 0.008003467567051331
Component 16 Mean:   [34495.95074592178, 13503.358130697956, 813.6840327386434, 512.2344352264377]
Component 16 Cov1:   [1647291394.252843, 820754068.6843029, -6205817.2905725995, -313659.35816180566]
Component 16 Cov2:   [820754068.6843028, 520567649.3099436, -3464474.0197747126, -387588.06252754905]
Component 16 Cov3:   [-6205817.290572601, -3464474.0197747126, 288595.26621225727, 173971.80952432388]
Component 16 Cov4:   [-313659.35816180566, -387588.06252754905, 173971.80952432385, 112946.36362954634]
Component 17 Weight: 0.04005184798705129
Component 17 Mean:   [13919.819057962499, 5282.760740296054, 11208.535312255777, 107151.84999503798]
Component 17 Cov1:   [48462478.22673244, -2139427.7954699807, 15429547.231019361, 97616745.7254832]
Component 17 Cov2:   [-2139427.7954699807, 13672790.527131528, -377868.6900572393, -2243324.677690736]
Component 17 Cov3:   [15429547.231019361, -377868.6900572393, 36546978.255558275, 366600909.9661475]
Component 17 Cov4:   [97616745.72548321, -2243324.6776907356, 366600909.96614754, 5797284727.494544]
Component 18 Weight: 1.7068924316389 62e-05
Component 18 Mean:   [65033.999999999854, 0.0, 39871.99999999991, 4939070.999999989]
Component 18 Cov1:   [1.0000000000000211e-06, 0.0, 1.2705494208814476e-20, 1.626303258728253e-18]
Component 18 Cov2:   [0.0, 1e-06, 0.0, 0.0]
Component 18 Cov3:   [1.2705494208814476e-20, 0.0, 1.0000000000000076e-06, 9.757819552369519e-19]
Component 18 Cov4:   [1.6263032587282 53e-18, 0.0, 9.757819552369519e-19, 1.0000000001249e-06]
Component 19 Weight: 0.07358997773588898
Component 19 Mean:   [257.3912691794383, 156.44476463506115, 1021.827786337 98, 3846.87107431413 14]
Component 19 Cov1:   [88200.17096461594, -8963.659455222421, 5223.553139312908, -113370.05684276519]
Component 19 Cov2:   [-8963.65945522242, 10149.574008358511, -1581.7491343131621, 157956.767505866]
Component 19 Cov3:   [5223.553139312908, -1581.7491343131624, 7144.560020426273, -62732.94818667756]
Component 19 Cov4:   [-113370.05684276519, 157956.767505866, -62732.94818667756, 6499501.226221686]
Component 20 Weight: 0.0001023815472 6223985
Component 20 Mean:   [3206.522671429899, 6239224.79549444, 4100.908688758954, 6470.4766736922375]
Component 20 Cov1:   [6764153.327245828, -1291321502.5310113, 2893600.8097719587, -4803967.992938524]
Component 20 Cov2:   [-1291321502.5310113, 320248846850.589, -691080615.0094941, 545899827.8575156]
Component 20 Cov3:   [2893600.8097719587, -691080615.0094941, 1979897.1848501682, -3038931.648882679]
Component 20 Cov4:   [-4803967.992938524, 545899827.8575156, -3038931.648882679, 11207639.734399464]
Component 21 Weight: 6.8275629268886 95e-05
Component 21 Mean:   [12691.003376853321, 3859264.4066775404, 3164.749960299076, 3826.499956857278]
Component 21 Cov1:   [29352703.174917936, 1435976407.0189164, -1350102.9910640463, -1320070.1562854866]
Component 21 Cov2:   [1435976407.0189164, 90879147166.85033, -184353726.63561115, -180879705.44986433]
Component 21 Cov3:   [-1350102.9910640456, -184353726.6356111, 1160349.2371475713, 1109300.9191861867]
Component 21 Cov4:   [-1320070.1562854864, -180879705.44986427, 1109300.9191861867, 1061948.7887031923]
Component 22 Weight: 0.032665439049915375
Component 22 Mean:   [4516.952380731844, 386.6351157212849, 10831.37497029656, 221394.85141845394]
Component 22 Cov1:   [37455749.57957031, 238496.4519852965, 41559679.05011406, -137591834.2063299]
Component 22 Cov2:   [238496.4519852965, 122401.52119998664, 337628.07942711795, -13784358.477606747]
Component 22 Cov3:   [41559679.05011406, 337628.07942711795, 65009925.28803322, -74236064.67422588]
Component 22 Cov4:   [-137591834.2063299, -13784358.477606747, -74236064.67422591, 15046623090.053171]
Component 23 Weight: 3.413784863277 92e-05
Component 23 Mean:   [90.49999999999999, 17731337.99999998, 740.9999999999992, 6857.999999999993]
Component 23 Cov1:   [552.2500000999994, -10753435.499999989, -469.9999999999995, 7.2378303598389 84e-25]
Component 23 Cov2:   [-10753435.499999989, 209391353648.99976, 9151859.99999999, 1.355252715606879e-19]
Component 23 Cov3:   [-469.9999999999995, 9151859.99999999, 400.000000999999954, 5.790264287871187e-24]
Component 23 Cov4:   [7.2378303598389 84e-25, 1.355252715606879e-19, 5.790264287871187e-24, 1e-06]
Component 24 Weight: 0.009188148154303803
Component 24 Mean:   [19286.34852695327, 6699.365509803425, 23749.839425809663, 411499.5852965981]
Component 24 Cov1:   [98538554.18119901, -19015747.15618243, 41599426.793968074, 1374334136.922617]
Component 24 Cov2:   [-19015747.15618243, 39079056.33027988, -44336597.20279433, -179193149.4799831]
Component 24 Cov3:   [41599426.793968074, -44336597.202794336, 150522091.0385243, -267073635.5282835]
Component 24 Cov4:   [1374334136.9226167, -179193149.4799831, -267073635.52828348, 110282832824.65675]
Component 25 Weight: 0.004143041843338847
Component 25 Mean:   [24034.41141772398, 16805.82181870657, 410127.75887025567, 285462.9967114539]
Component 25 Cov1:   [299609346.7741721, -40734095.893220894, -1135923976.07294, 1022381066.9169035]
Component 25 Cov2:   [-40734095.89322089, 353523005.4299726, 66348272.17205739, -495507039.98536414]
Component 25 Cov3:   [-1135923976.0729399, 66348272.1720574, 111658993543.51859, 2196686729.2677603]
```

```
Component 25 Cov4:   [1022381066.9169035, −495507039.9853642, 2196686729.2677603, 15649757650.67428]
Component 26 Weight: 0.004009250678741844
Component 26 Mean:   [410.66685242328504, 1710339.070321298, 1092.5275153252683, 4486.33886814412]
Component 26 Cov1:   [154159.24350072833, 59390052.44940085, 62858.90548492347, −578716.0315174812]
Component 26 Cov2:   [59390052.44940085, 6103254348801.904, −11765914.557565158, 1206935094.430485]
Component 26 Cov3:   [62858.90548492347, −11765914.557565155, 77512.62108584403, −395623.86487346253]
Component 26 Cov4:   [−578716.0315174812, 1206935094.4304848, −395623.86487346253, 7918000.616159765]
Component 27 Weight: 0.06402985274986085
Component 27 Mean:   [78.73728626636137, 189.30821376878308, 1713.602501566743, 24597.90831278044]
Component 27 Cov1:   [1567.0892885835804, 1352.8417148789695, 14478.392806075935, 247667.4245125921]
Component 27 Cov2:   [1352.8417148789695, 93955.84915348947, −78972.1469343896, −1968291.1116717833]
Component 27 Cov3:   [14478.392806075934, −78972.1469343896, 738865.3138766518, 10671791.126473498]
Component 27 Cov4:   [247667.4245125921, −1968291.1116717833, 10671791.126473498, 367003420.8426727]
Component 28 Weight: 0.0140761517475775 99
Component 28 Mean:   [15750.968903204794, 38197.6660587722, 11079.099728558363, 21522.429060717408]
Component 28 Cov1:   [93636527.83974454, −72302638.55481464, 45828771.31826315, 48307168.54003292]
Component 28 Cov2:   [−72302638.55481464, 1241005665.6145694, −59070359.09533611, −43582211.81675515]
Component 28 Cov3:   [45828771.31826315, −59070359.09533611, 71223218.48545356, 111982667.01432827]
Component 28 Cov4:   [48307168.54003292, −43582211.81675515, 111982667.01432827, 256799734.04142946]
```

Προφίλ 4

```
Component 01 Weight: 0.7633210937879734
Component 01 Mean:   [218.5701006675849, 2516.271143595025, 422.2100285767406, 541.0738046492296]
Component 01 Cov1:   [23998.453855698448, 8718.099335412213, 27164.39815430766, 27352.665589869004]
Component 01 Cov2:   [8718.099335412213, 5496053.6306246435, 15167.41183482289, 18482.43116256705]
Component 01 Cov3:   [27164.39815430766, 15167.41183482289, 39763.62901100573, 40871.46656953967]
Component 01 Cov4:   [27352.665589869008, 18482.43116256704, 40871.46656953967, 42810.45600045125]
Component 02 Weight: 0.00010871515542239047
Component 02 Mean:   [110829.85966018795, 1266.02099006856 32, 221625.22432624162, 26655248.833209205]
Component 02 Cov1:   [79041.67517993419, −20896.32366922698, 5836990.524782315, 412192179.1536764]
Component 02 Cov2:   [−20896.323669226967, 1121775.3324347767, 3152965.505666281, 271060677.74760616]
Component 02 Cov3:   [5836990.524782315, 3152965.505666281, 1138960018.3614056, 82477057164.47487]
Component 02 Cov4:   [412192179.1536765, 271060677.74760616, 82477057164.47488, 9502536436045.031]
Component 03 Weight: 0.0035059205449178813
Component 03 Mean:   [207993.4411001562, 2282.08100485381, 440835.40358453384, 9049.5474173582]
Component 03 Cov1:   [59251671149.17566, −5154453.718715455, −40430142868.195015, −171978676.21542156]
Component 03 Cov2:   [−5154453.71871546, 6380884.422557493, −627732605.9495906, −9657732.114122137]
Component 03 Cov3:   [−40430142868.195015, −627732605.9495906, 5261187853699.314, 68562138556.06927]
Component 03 Cov4:   [−171978676.21542156, −9657732.114122136, 68562138556.06927, 902486799.5285889]
Component 04 Weight: 0.0007316098665062523
Component 04 Mean:   [695109.7863240927, 40480.47752567513, 68018.0447959367, 4803076.008148844]
Component 04 Cov1:   [1534530187200.2263, −24692910086.001694, 3211928547.56335, −2913069569524.6123]
Component 04 Cov2:   [−24692910086.001694, 180975660217.34793, −2592848448.700818, −185473691678.8566]
Component 04 Cov3:   [3211928547.5633507, −2592848448.700818, 12303293890.304445, 1126330605798.479]
Component 04 Cov4:   [−2913069569524.6123, −185473691678.8566, 1126330605798.479, 119405184167128.7]
Component 05 Weight: 0.019788628059506937
Component 05 Mean:   [115960.33651280424, 24852.527705909168, 6320.652818683761, 0.0]
Component 05 Cov1:   [7133261.781782486, 17139200.07343772, −521430.8108951758, 0.0]
Component 05 Cov2:   [17139200.07343772, 655772637072.055, −11268606.079030298, 0.0]
Component 05 Cov3:   [−521430.8108951758, −11268606.079030298, 1165372.2399703914, 0.0]
Component 05 Cov4:   [0.0, 0.0, 0.0, 1e−06]
Component 06 Weight: 2.52825962192406 17e−06
Component 06 Mean:   [46172.9999999999, 2436.9999999999945, 1377615.999999997, 93932324.99999979]
Component 06 Cov1:   [1.0000000000103e−06, 5.55865371635e−22, 3.0831999280056e−19, 2.125036258071e−17]
Component 06 Cov2:   [5.558653716356334e−22, 1e−06, 1.651714247145882e−20, 1.1384122811097772e−18]
Component 06 Cov3:   [3.0831999280056e−19, 1.651714247145e−20, 1.00000000091e−06, 6.314393452555e−16]
Component 06 Cov4:   [2.12503625807e−17, 1.1384122811097e−18, 6.3143934525 55e−16, 1.000000043520e−06]
Component 07 Weight: 0.0590960842440827
Component 07 Mean:   [2798.4570952116 79, 2307.4517762561436, 1374.1158024316371, 1829.0858557918539]
Component 07 Cov1:   [22453434.168758307, 68030.9005291864, 1424681.756891316, 594994.9160477705]
Component 07 Cov2:   [68030.90052918634, 5309135.101881502, −139389.2499151003, −572023.2599593316]
Component 07 Cov3:   [1424681.7568913158, −139389.24991510028, 666327.3929180721, 767734.4862350948]
Component 07 Cov4:   [594994.9160477705, −572023.2599593316, 767734.4862350948, 2087844.5334925593]
Component 08 Weight: 0.026975381018572567
Component 08 Mean:   [13497.634894652505, 702.7401271678576, 230.325137431745 65, 140.71224417288752]
Component 08 Cov1:   [1034891589.8556689, 15090555.68835193, 2985643.2894021785, 4279454.476457498]
Component 08 Cov2:   [15090555.68835193, 956354.1504399508, 34340.59186450392, 62287.71984377006]
Component 08 Cov3:   [2985643.2894021785, 34340.59186450392, 13838.575017127758, 13728.034047491967]
Component 08 Cov4:   [4279454.476457498, 62287.719843770064, 13728.034047491967, 19716.491417957157]
Component 09 Weight: 0.12328499616895906
Component 09 Mean:   [262.78309677768243, 2216.1880054598914, 103.99999999999658, 63.999999999998]
Component 09 Cov1:   [336.2005439201969, −932.3883407539969, 2.884205399294 43e−23, 1.6816193564404e−23]
Component 09 Cov2:   [−932.3883407539965, 4673369.6020362, 2.5433426745104e−22, 1.4838576094271e−22]
Component 09 Cov3:   [2.8841108285819 31e−23, 2.5432681205585604e−22, 1e−06, 6.838073518396222e−24]
Component 09 Cov4:   [1.6815630062904998e−23, 1.4836023533200 08e−22, 6.838073518396222e−24, 1e−06]
Component 10 Weight: 0.00317998637518165
Component 10 Mean:   [4096.530612679467, 1161.0116007824925, 24774.114766304436, 45044.299191613434]
Component 10 Cov1:   [37809907.96703373, −455291.80811818445, 72748819.30258505, 80043029.24271165]
Component 10 Cov2:   [−455291.8081181846, 2014218.4698294546, 9780422.149236523, −9197172.875411067]
Component 10 Cov3:   [72748819.30258508, 9780422.149236519, 5075210301.780398, 663428309.969472]
Component 10 Cov4:   [80043029.24271165, −9197172.87541107, 663428309.969472, 7954791559.831301]
Component 11 Weight: 5.0565192438481 18e−06
Component 11 Mean:   [21348724.999999978, 941.499999999999, 493273.9999999995, 199038.99999999977]
Component 11 Cov1:   [281963230204 40.97, 2275343998.4999976, 183612.2500009998, 52036182.99999994, 21093769.499999978]
Component 11 Cov2:   [2275343998.4999976, 183612.2500009998, 52036182.99999994, 21093769.499999978]
Component 11 Cov3:   [644838330197.9993, 52036182.99999994, 14747187843.999985, 5978028425.999993]
Component 11 Cov4:   [261396403766.9997, 21093769.499999978, 5978028425.999993, 2423297528.999998]
```

Προφίλ 5

```
Component 01 Weight: 0.04473595201936859
Component 01 Mean:   [2808.9686511839623, 8709.384687717857, 154.03508619257926, 81.28508291790497]
Component 01 Cov1:   [18111751.953450754, −3823906.807048817, 32923.23708774933, −149069.13527352954]
Component 01 Cov2:   [−3823906.807048817, 79861118.02759437, 134735.06991684486, 69906.129539819]
Component 01 Cov3:   [32923.23708774934, 134735.06991684486, 6564.688341105514, 1932.097884835044]
Component 01 Cov4:   [−149069.13527352954, 69906.12953981898, 1932.0978848350435, 4978.491079661051]
Component 02 Weight: 1.6042737853642 12e−05
Component 02 Mean:   [24931.99999999997, 16.999999999999982, 697052.4999999992, 39007123.499999955]
Component 02 Cov1:   [9025.00000099999, −1234.9999999999986, −394297.49999999953, 118227.49999999987]
```

```
Component 02 Cov2:   [-1234.9999999999986, 169.0000009999998, 53956.49999999994, -16178.499999999982]
Component 02 Cov3:   [-394297.49999999953, 53956.49999999994, 17226650.25000098, -5165297.249999994]
Component 02 Cov4:   [118227.49999999987, -16178.499999999982, -5165297.249999994, 1548780.2500009984]
Component 03 Weight: 4.0106844634105 27e-05
Component 03 Mean:   [23256.999999999993, 0.5999999999999998, 263599.5999999999, 14895004.199999996]
Component 03 Cov1:   [295358162.8000009, -8014.199999999997, -1497774.3999999983, 2865761040.7999988]
Component 03 Cov2:   [-8014.199999999997, 1.4400009999999994, -5872.559999999998, -516681.7199999999]
Component 03 Cov3:   [-1497774.3999999983, -5872.559999999998, 29040690.240000986, 2107749313.2799995]
Component 03 Cov4:   [2865761040.7999988, -516681.7199999999, 2107749313.2799995, 185391121649.75995]
Component 04 Weight: 4.010684463410527e-05
Component 04 Mean:   [45494.999999999985, 1952.3999999999994, 160189.99999999994, 7879015.799999997]
Component 04 Cov1:   [140655921.60000098, -5066674.799999997, 136224407.19999996, 4396241883.999998]
Component 04 Cov2:   [-5066674.799999997, 11366375.440000996, -21642693.19999999, -633788592.3199998]
Component 04 Cov3:   [136224407.19999996, -21642693.19999999, 209513476.00000095, 5034017939.799998]
Component 04 Cov4:   [4396241883.999998, -633788592.3199998, 5034017939.799998, 187682246789.35995]
Component 05 Weight: 8.021368926821069e-06
Component 05 Mean:   [119020.99999999974, 3385.9999999999923, 18114749.99999996, 328554.99999999924]
Component 05 Cov1:   [1.0000000000686e-06, 2.024938139529e-21, 1.0733601507606e-17, 1.9820570965e-19]
Component 05 Cov2:   [2.0249381395298072e-21, 1e-06, 3.167903222731076e-19, 5.849821291974999e-21]
Component 05 Cov3:   [1.073360150760e-17, 3.16790322273e-19, 1.0000000167921e-06, 3.10081821e-17]
Component 05 Cov4:   [1.9820570965750e-19, 5.8498212919e-21, 3.100818213308e-17, 1.00000005725e-06]
Component 06 Weight: 0.49439144471131985
Component 06 Mean:   [70.77288295803204, 8209.266127935229, 115.67857030194942, 97.544249733488992]
Component 06 Cov1:   [2724.70528177013, 259947.90052617513, 2340.14240787082, 1720.3737554672075]
Component 06 Cov2:   [259947.90052617513, 98235171.18834883, 226679.58726465976, 170818.21675320304]
Component 06 Cov3:   [2340.14240787082, 226679.58726465976, 2835.2617385110275, 3186.4088947205582]
Component 06 Cov4:   [1720.3737554672077, 170818.21675320307, 3186.4088947205582, 4640.520699903811]
Component 07 Weight: 7.21914089419815e-05
Component 07 Mean:   [11693.238990147314, 93.22322180760827, 43790.621741978124, 2164829.0952934283]
Component 07 Cov1:   [153138116.54914123, 2844268.7543404223, -11231835.41966165, -44913074.11192713]
Component 07 Cov2:   [2844268.7543404223, 61652.869146665034, -95206.34527934325, -392358.35632927465]
Component 07 Cov3:   [-11231835.419661647, -95206.34527934325, 61007829.03824146, 38179472.398383275]
Component 07 Cov4:   [-44913074.11192712, -392358.35632927465, 38179472.398383275, 109006234.84376979]
Component 08 Weight: 0.0014881221591681831
Component 08 Mean:   [6374.520366142313, 1498.1139163970256, 16583.288807562, 436077.97972314456]
Component 08 Cov1:   [43380485.13433466, -1658111.8676837876, -10180825.552674111, -1054635733.853675]
Component 08 Cov2:   [-1658111.8676837876, 15446515.540169744, -2844277.9269244964, -382360480.3792143]
Component 08 Cov3:   [-10180825.552674111, -2844277.9269244964, 752972483.7406813, 2258395495.407976]
Component 08 Cov4:   [-1054635733.853675, -382360480.3792143, 2258395495.407976, 214111029358.46936]
Component 09 Weight: 8.021368926821069e-06
Component 09 Mean:   [18901.999999999956, 15.999999999999964, 754095.9999999984, 46043193.999999896]
Component 09 Cov1:   [1.0000000000019e-06, 1.550963648536e-24, 7.1150767569361e-20, 4.5536491244e-18]
Component 09 Cov2:   [1.5509636485369234e-24, 1e-06, 5.790264287871181e-23, 3.705769144237556e-21]
Component 09 Cov3:   [7.115076756936e-20, 5.79026428787e-23, 1.0000000026563e-06, 1.700029006457e-16]
Component 09 Cov4:   [4.55364912443 91e-18, 3.705769144237e-21, 1.7000290064572e-16, 1.000000010880e-06]
Component 10 Weight: 8.021368926821069e-06
Component 10 Mean:   [5575.999999999987, 0.0, 369379.9999999992, 21653047.99999995]
Component 10 Cov1:   [1.0000000000000002e-06, 0.0, 1.0376153603865157e-20, 6.166399856011293e-19]
Component 10 Cov2:   [0.0, 1e-06, 0.0, 0.0]
Component 10 Cov3:   [1.0376153603865157e-20, 0.0, 1.0000000000000664e-06, 3.9464959078472275e-17]
Component 10 Cov4:   [6.166399856011293e-19, 0.0, 3.9464959078472275e-17, 1.000000002345346e-06]
Component 11 Weight: 2.4064106780463173e-05
Component 11 Mean:   [102254.33333333326, 2712.6666666666647, 2870824.6666666646, 66161.99999999996]
Component 11 Cov1:   [576565790.8888893, 43453144.77777775, 9596854318.444439, 234530845.33333316]
Component 11 Cov2:   [43453144.77777775, 12841837.555556549, 1981195077.8888876, 37859528.66666664]
Component 11 Cov3:   [9596854318.444439, 1981195077.8888876, 325137743204.222, 6557648034.666662]
Component 11 Cov4:   [234530845.33333316, 37859528.66666664, 6557648034.666662, 137983976.00000092]
Component 12 Weight: 1.604273785364212e-05
Component 12 Mean:   [6923.499999999993, 23.999999999999975, 298343.49999999965, 17240606.49999998]
Component 12 Cov1:   [10461990.250000989, -16172.499999999982, 43243647.749999955, 128495364.24999985]
Component 12 Cov2:   [-16172.499999999982, 25.000000999999973, -66847.49999999993, -198632.49999999977]
Component 12 Cov3:   [43243647.749999955, -66847.49999999993, 178743530.2500008, 531123441.7499994]
Component 12 Cov4:   [128495364.24999985, -198632.49999999977, 531123441.7499994, 1578194802.2499993]
Component 13 Weight: 8.021368926821069e-06
Component 13 Mean:   [41924.999999999905, 0.0, 613554.9999999986, 33995656.999999925]
Component 13 Cov1:   [1.0000000000000088e-06, 0.0, 1.3213713977167056e-19, 7.047314121155763e-18]
Component 13 Cov2:   [0.0, 1e-06, 0.0, 0.0]
Component 13 Cov3:   [1.3213713977167056e-19, 0.0, 1.0000000000019515e-06, 1.0408340855860819e-16]
Component 13 Cov4:   [7.047314121155763e-18, 0.0, 1.0408340855860819e-16, 1.000000005551115e-06]
Component 14 Weight: 1.6042737853642 12e-05
Component 14 Mean:   [4655.4999999999945, 12.999999999999986, 152355.99999999983, 9741070.999999989]
Component 14 Cov1:   [20.250000999999998, 17.999999999999998, 1061.9999999999989, 2969.999999999997]
Component 14 Cov2:   [17.999999999999998, 16.000000999999983, 943.999999999999, 2639.999999999973]
Component 14 Cov3:   [1061.9999999999989, 943.999999999999, 55696.000000099994, 155759.99999999983]
Component 14 Cov4:   [2969.999999999997, 2639.9999999999973, 155759.99999999983, 435600.000000099954]
Component 15 Weight: 8.021368926821069e-06
Component 15 Mean:   [18917.999999999956, 0.0, 704580.9999999985, 42807802.9999999]
Component 15 Cov1:   [1.0000000000000019e-06, 0.0, 6.606856988583528e-20, 4.228388472693458e-18]
Component 15 Cov2:   [0.0, 1e-06, 0.0, 0.0]
Component 15 Cov3:   [6.606856988583528e-20, 0.0, 1.0000000000022903e-06, 1.4658413372003988e-16]
Component 15 Cov4:   [4.228388472693458e-18, 0.0, 1.4658413372003988e-16, 1.0000000093813845e-06]
Component 16 Weight: 2.4064106780463173e-05
Component 16 Mean:   [6445.3333333333285, 29.999999999999998, 115266.33333333324, 6499545.333333328]
Component 16 Cov1:   [12566755.555556549, 72693.33333333328, 9298062.222222215, 5625795.555555552]
Component 16 Cov2:   [72693.33333333328, 1740.6666676666657, 35362.333333333314, 33180.66666666664]
Component 16 Cov3:   [9298062.222222215, 35362.333333333314, 7136670.888889885, 4153589.2222222188]
Component 16 Cov4:   [5625795.555555552, 33180.66666666664, 4153589.2222222188, 2518824.2222232204]
Component 17 Weight: 8.021368926821069e-06
Component 17 Mean:   [119767.99999999974, 2200.999999999995, 4657070.99999999, 96265.99999999978]
Component 17 Cov1:   [1.000000000686e-06, 1.31025409028e-21, 2.6834003769016e-18, 5.7174723939 66e-20]
Component 17 Cov2:   [1.3102540902 83993e-21, 1e-06, 5.12454933088850 56e-20, 1.0918784085699 94e-21]
Component 17 Cov3:   [2.683400376901e-18, 5.1245493308885e-20, 1.000000001049e-06, 2.236166980e-18]
Component 17 Cov4:   [5.717472393966e-20, 1.0918784085 69e-21, 2.236166980751e-18, 1.000000000476e-06]
Component 18 Weight: 8.021368926821069e-06
Component 18 Mean:   [5435.999999999988, 32.99999999999993, 85655.99999999981, 4337341.999999991]
Component 18 Cov1:   [1.0000000000002e-06, 8.4010530962 41e-25, 2.2366963763 43e-21, 1.1011428314 30e-19]
Component 18 Cov2:   [8.4010530962 41669e-25, 1e-06, 1.344168495398667e-23, 6.617444900424206e-22]
Component 18 Cov3:   [2.2366963763 43e-21, 1.344168495 39e-23, 1.0000000000357e-06, 1.761828530288e-18]
Component 18 Cov4:   [1.1011428314 30e-19, 6.617444900424e-22, 1.761828530288e-18, 1.0000000000867e-06]
Component 19 Weight: 2.4064106780463173e-05
Component 19 Mean:   [118876.66666666658, 13444.99999999999, 1385496.6666666656, 33136.33333333331]
Component 19 Cov1:   [32008.2222232222, 143094.9999999999, 13107028.88888888, 387868.7777777775]
Component 19 Cov2:   [143094.9999999999, 13131438.00000099, -108754406.99999993, -4522695.999999996]
Component 19 Cov3:   [13107028.88888888, -108754406.99999993, 7609164420.222217, 242648400.11111096]
Component 19 Cov4:   [387868.7777777775, -4522695.999999996, 242648400.11111096, 7833880.222223216]
Component 20 Weight: 0.09299771487894597
Component 20 Mean:   [659.4699203195646, 7317.173707828257, 2240.7400764874733, 4677.662330742224]
Component 20 Cov1:   [90309.93205746888, -66382.18725717824, -19182.08230773013, 19534.876213149884]
Component 20 Cov2:   [-66382.18725717826, 63159090.959627874, -270297.3953350978, 84166.32675461694]
Component 20 Cov3:   [-19182.08230773013, -270297.395335098, 399109.62427640805, 225269.42834108483]
Component 20 Cov4:   [19534.876213149888, 84166.3267546169, 225269.42834108483, 366884.54940786934]
Component 21 Weight: 1.604273785364212e-05
```

```
Component 21 Mean:   [5592.999999999994, 12.999999999999986, 307213.49999999965, 18440476.99999998]
Component 21 Cov1:   [3400336.000000996, −1843.999999999998, 18859509.999999978, −68325731.99999993]
Component 21 Cov2:   [−1843.999999999998, 1.0000009999999988, −10227.499999999989, 37052.999999999956]
Component 21 Cov3:   [18859509.999999978, −10227.499999999989, 104601756.25000088, −378959557.4999996]
Component 21 Cov4:   [−68325731.99999993, 37052.999999999956, −378959557.4999996, 1372924808.9999995]
Component 22 Weight: 8.021368926821069e−06
Component 22 Mean:   [44356.9999999999, 0.0, 207016.99999999953, 11872587.999999974]
Component 22 Cov1:   [1.0000000000000103e−06, 0.0, 4.7433845046240717e−20, 2.65629532258948e−18]
Component 22 Cov2:   [0.0, 1e−06, 0.0, 0.0]
Component 22 Cov3:   [4.7433845046240717e−20, 0.0, 1.0000000000002168e−06, 1.2143064331837623e−17]
Component 22 Cov4:   [2.65629532258948e−18, 0.0, 1.2143064331837623e−17, 1.0000000006800116e−06]
Component 23 Weight: 0.05008004204094478
Component 23 Mean:   [296.0288275604249, 41.98771704321936, 963.9724578650226, 5005.115288553891]
Component 23 Cov1:   [56765.11342461317, 606.4688950378638, 87717.07569545058, 464132.0489916309]
Component 23 Cov2:   [606.4688950378637, 5074.3972203318535, 4543.662857412282, 15039.522395926457]
Component 23 Cov3:   [87717.07569545058, 4543.662857412282, 317803.2734894374, 1142402.8062802271]
Component 23 Cov4:   [464132.0489916309, 15039.522395926457, 1142402.8062802271, 8948464.062907536]
Component 24 Weight: 0.28000129931662937
Component 24 Mean:   [15001.663558904353, 7900.168807636375, 260.00000000001006, 0.0]
Component 24 Cov1:   [6.832218879728087, 366.8685185203963, 5.769867880327141e−21, 0.0]
Component 24 Cov2:   [366.8685185203964, 69711414.4161446, 3.0203536648283227e−21, 0.0]
Component 24 Cov3:   [5.769868637142102e−21, 3.020711580544251e−21, 1e−06, 0.0]
Component 24 Cov4:   [0.0, 0.0, 0.0, 1e−06]
Component 25 Weight: 0.012586018344916211
Component 25 Mean:   [14840.194852163935, 8144.908648717046, 202.80119231345708, 0.0]
Component 25 Cov1:   [138231.94372506897, −58473.87202511934, −483.025825955949, 0.0]
Component 25 Cov2:   [−58473.87202511935, 57040754.27741454, 10121.48393503685, 0.0]
Component 25 Cov3:   [−483.025825955949, 10121.483935036853, 488.33400928372527, 0.0]
Component 25 Cov4:   [0.0, 0.0, 0.0, 1e−06]
Component 26 Weight: 0.021790097606408750
Component 26 Mean:   [843.0917973455157, 7086.438182502471, 2642.6687473979164, 7419.1828711406615]
Component 26 Cov1:   [866014.9173767285, −488017.85154612025, 240252.7071985685, −1108830.0076013464]
Component 26 Cov2:   [−488017.85154612025, 60506703.85325283, 26215.977906050062, −1692972.227083002]
Component 26 Cov3:   [240252.7071985685, 26215.977906050106, 4182013.578428948, 1068646.3364210045]
Component 26 Cov4:   [−1108830.0076013461, −1692972.2270830015, 1068646.3364210448, 37837684.03029782]
Component 27 Weight: 0.0015763696009113044
Component 27 Mean:   [16088.008122579939, 10900.631781516691, 46966.569026461606, 3436.92329610648827]
Component 27 Cov1:   [350672665.501231, −35143114.566419885, 2023046652.7487373, 29510685.506378762]
Component 27 Cov2:   [−35143114.566419885, 189498060.88661715, −215749668.64196184, −5427224.374409454]
Component 27 Cov3:   [2023046652.7487373, −215749668.64196184, 26122701713.73827, 550091855.754679]
Component 27 Cov4:   [29510685.506378766, −5427224.374409455, 550091855.754679, 20958819.414926022]
```

Προφίλ 6

```
Component 01 Weight: 0.722207211014313
Component 01 Mean:   [8.1309917016755574, 19443.043715125415, 655.6227487167221, 143.33797651436458]
Component 01 Cov1:   [228.37947357592543, 453867.5262678884, −1551.415754097186, 78.695002019422246]
Component 01 Cov2:   [453867.5262678884, 1885335541.7727265, −3960132.7892463445, 200876.30090380003]
Component 01 Cov3:   [−1551.415754097186, −3960132.789246344, 13864.483324099412, −703.2708932006948]
Component 01 Cov4:   [78.695002019422246, 200876.30090380003, −703.2708932006948, 35.67316224931061]
Component 02 Weight: 0.01922819420881489
Component 02 Mean:   [119436.3440256573, 61204.51318034907, 291024175.80021393, 4081431.0401233486]
Component 02 Cov1:   [129995.88421804475, 3654452.6827081298, 5480617313.422525, 75889382.09092684]
Component 02 Cov2:   [3654452.6827081298, 2160552300.5877676, 961864726659.4421, 13377606315.155397]
Component 02 Cov3:   [5480617313.422525, 961864726659.4423, 1.3633088081952752e+16, 194250694073452.03]
Component 02 Cov4:   [75889382.09092684, 13377606315.155397, 194250694073452.03, 2770675470671.467]
Component 03 Weight: 0.046898607579531626
Component 03 Mean:   [48644.73138063332, 48954.91706828937, 36619221.674857646, 515919.10971058434]
Component 03 Cov1:   [1364152124.2845073, −281738328.9814447, 307101683022.4722, 4348946876.813432]
Component 03 Cov2:   [−281738328.9814447, 2158319516.0591927, −201021622708.73566, −2840766325.1527224]
Component 03 Cov3:   [307101683022.4722, −201021622708.7357, 5152755278801136.0, 72391035964246.17]
Component 03 Cov4:   [4348946876.813431, −2840766325.152723, 72391035964246.17, 1017029179314.1492]
Component 04 Weight: 0.0029213318947809612
Component 04 Mean:   [239455.80429450906, 79086.98411003123, 625115135.3896099, 8742654.317664899]
Component 04 Cov1:   [192455.16046251095, 6596635.142674559, 25992848642.49234, 353200341.6122933]
Component 04 Cov2:   [6596635.142674559, 6538482067.509358, 5822309621762.211, 81288947273.05353]
Component 04 Cov3:   [25992848642.49234, 5822309621762.211, 4.028685606527775e+16, 575058000401581.4]
Component 04 Cov4:   [353200341.6122933, 81288947273.05353, 575058000401581.5, 8221048371592.619]
Component 05 Weight: 0.0738495669769657
Component 05 Mean:   [750.3343675763815, 317923.35818020033, 270457.4159985445, 4493.508469360608]
Component 05 Cov1:   [1758344.0698277142, −73336389.3156737, 2821143.1267383415, 227328.71813833437]
Component 05 Cov2:   [−73336389.31567366, 378759875355.3536, 28400982067.512238, 461321516.7354271]
Component 05 Cov3:   [2821143.126738353, 28400982067.512238, 138223619244.1582, 2028681462.2595823]
Component 05 Cov4:   [227328.71813833434, 461321516.7354271, 2028681462.2595823, 30094647.73559061]
Component 06 Weight: 0.0004645384231817174
Component 06 Mean:   [353796.30137011127, 140843.86275072378, 1249599622.7177768, 17562583.40348222]
Component 06 Cov1:   [8942636712.333408, 5039583005.257629, 39626609658276.46, 554933345853.7559]
Component 06 Cov2:   [5039583005.257629, 10031890790.15395, 36671458829164.94, 5034483711166.50305]
Component 06 Cov3:   [39626609658276.46, 36671458829164.94, 2.084980340417391e+17, 2897869526153918.5]
Component 06 Cov4:   [554933345853.7559, 5034483711166.5031, 2897869526153918.5, 402907238386111.914]
Component 07 Weight: 0.018843958805940954
Component 07 Mean:   [42753.82595345578, 1043887.3062154263, 63400432.51893354, 2244327.6747248285]
Component 07 Cov1:   [1493005759.6523576, −34851595650.112465, 2361968479395.386, 84722941412.09422]
Component 07 Cov2:   [−34851595650.11245, 7544949445847.638, −51161907746364.31, −2072311819951.1829]
Component 07 Cov3:   [2361968479395.386, −51161907746364.305, 6047714465293936.0, 149910527974795.78]
Component 07 Cov4:   [84722941412.09422, −2072311819951.1829, 149910527974795.78, 29112210757623.906]
Component 08 Weight: 0.03232360261934304
Component 08 Mean:   [74062.42143628879, 139461.05048814826, 203474945.35736004, 2859533.7728786105]
Component 08 Cov1:   [2990539680.65626, −649336998.9326534, 9291717479661.424, 1304001115520.45796]
Component 08 Cov2:   [−649336998.9326534, 22248928589.3561, −1761948121087.299, −25355789123.524265]
Component 08 Cov3:   [9291717479661.424, −1761948121087.299, 3.5470924757722224e+16, 498853947389266.4]
Component 08 Cov4:   [1304001115520.45796, −25355789123.524265, 498853947389266.4, 7017515950076.762]
Component 09 Weight: 0.051178561750803674
Component 09 Mean:   [813.8364555319549, 3248.1469735241517, 2107724.022586561, 73367.79051286147]
Component 09 Cov1:   [2408572.791817905, 1350482.9841135752, 8930590370.20184, 299654027.9468322]
Component 09 Cov2:   [1350482.9841135752, 17670037.450247776, 5602014061.406542, 188692289.80294126]
Component 09 Cov3:   [8930590370.20184, 5602014061.406542, 34294486811651.05, 1150840781998.398]
Component 09 Cov4:   [299654027.9468322, 188692289.80294126, 1150840781998.3977, 38635383930.544464]
Component 10 Weight: 0.031768169954565 3
Component 10 Mean:   [34160.3953728124, 57961.291945868135, 137721277.29973957, 3466846.046541579]
Component 10 Cov1:   [29799525.206151813, 11872064.03159877, −2886100398.668959, 200303540.9084394]
Component 10 Cov2:   [11872064.031598765, 748799626.85582, −2741412485.419196, 124583047.03884505]
Component 10 Cov3:   [−2886100398.668959, −2741412485.419196, 6466400037124.513, 113861011354.04099]
Component 10 Cov4:   [200303540.9084394, 124583047.03884503, 113861011354.04099, 57024797474.89404]
Component 11 Weight: 0.0003162567717647816
Component 11 Mean:   [353153.2655193846, 56816.37039864112, 943917722.106331, 13202491.555394921]
```

```
Component 11 Cov1:   [161778919.81223336, -146582230.6251131, -1307660766724.5757, -18012210787.64918]
Component 11 Cov2:   [-146582230.62511307, 755489176.4300733, -290720655605.0145, -7013792429.043581]
Component 11 Cov3:   [-1307660766724.5754, -290720655605.0145, 5.951943591963482e+16, 858883129800336.1]
Component 11 Cov4:   [-18012210787.64918, -7013792429.04359, 858883129800336.1, 12414501875679.562]
```

Προφίλ 7

```
Component 01 Weight: 0.17697938316468298
Component 01 Mean:   [4151.036121108826, 14746.176689588081, 123.598523606264, 132.35791164492778]
Component 01 Cov1:   [40048576.31814515, 3317134.8534794156, -10253.922499737337, 282136.9079924099]
Component 01 Cov2:   [3317134.8534794142, 342240268.2751088, 3729.585455857077, 42620.66882850512]
Component 01 Cov3:   [-10253.922499737339, 3729.585455857077, 8.09780496671122, -67.74914649546928]
Component 01 Cov4:   [282136.9079924099, 42620.66882850511, -67.74914649546928, 2448.981184887025]
Component 02 Weight: 0.00016966030503692822
Component 02 Mean:   [387.7391309786062, 17950423.549006052, 843.69565349672 81, 11249.043495881258]
Component 02 Cov1:   [1651.75807956487, 399669.60571120004, 307.13798405711543, -1346.7281268349761]
Component 02 Cov2:   [399669.60571120004, 32465699151.28374, 1646104.8417842134, 22927911.681913488]
Component 02 Cov3:   [307.13798405711543, 1646104.8417842134, 3011.690026164569, 11907.969424326304]
Component 02 Cov4:   [-1346.7281268349761, 22927911.681913488, 11907.969424326304, 620839.9652632211]
Component 03 Weight: 0.3236723523384425
Component 03 Mean:   [47.48272045636573, 15889.892416396835, 40.99999999999812, 51.99999999999758]
Component 03 Cov1:   [42.37087229000133, 10724.702911406412, 4.10041100843e-24, 5.27809352841e-24]
Component 03 Cov2:   [10724.702911406412, 383229187.1265775, 1.37337222966e-21, 1.7687879553e-21]
Component 03 Cov3:   [4.100434572874719e-24, 1.373710652765695e-21, 1e-06, 4.562266604818779e-24]
Component 03 Cov4:   [5.2774105159566191e-24, 1.776369598041968e-21, 4.562266604818779e-24, 1e-06]
Component 04 Weight: 0.013092662697315387
Component 04 Mean:   [8745.8262141726, 3398.419979216185, 9738.957362754669, 222174.63782956428]
Component 04 Cov1:   [57288466.32532656, -2550826.413964506, 27453772.512061547, 435091624.9540416]
Component 04 Cov2:   [-2550826.4139645062, 9335978.101130703, -6043540.912730583, -276354624.822415]
Component 04 Cov3:   [27453772.512061547, -6043540.912730581, 54006415.504742734, 1274183975.1228569]
Component 04 Cov4:   [435091624.9540416, -276354624.822415, 1274183975.1228569, 58898537078.32878]
Component 05 Weight: 0.0024726582594816926
Component 05 Mean:   [5410.325677341228, 1262273.1620183461, 2362.607380113713, 10899.861230147466]
Component 05 Cov1:   [25507864.55061705, -1007728558.8110398, 1057548.9231345614, -10582624.878487866]
Component 05 Cov2:   [-1007728558.8110398, 1669874975560.7192, -260923516.19932413, 84342346.6297954]
Component 05 Cov3:   [1057548.9231345614, -260923516.19932398, 3422332.978920293, 12126153.6026061]
Component 05 Cov4:   [-10582624.878487863, 84342346.62979579, 12126153.602606097, 270500976.59615284]
Component 06 Weight: 0.010208224258086 73
Component 06 Mean:   [18922.177307757527, 11314.70664869971, 13779.317351052507, 27681.912675904445]
Component 06 Cov1:   [280210462.21476054, -50739546.35751395, 49205877.088451646, -21399722.519618887]
Component 06 Cov2:   [-50739546.35751395, 142833112.13793314, -55470159.903136365, -4344136.60175705]
Component 06 Cov3:   [49205877.088451646, -55470159.903136365, 214672662.33174035, 113795702.62159197]
Component 06 Cov4:   [-21399722.519618887, -4344136.601757052, 113795702.62159197, 1114360516.9344451]
Component 07 Weight: 7.376535241397132e-06
Component 07 Mean:   [13513.99999999997, 10053.999999999978, 265869.9999999994, 17287793.999999963]
Component 07 Cov1:   [1.0000000008e-06, 6.352747104407e-22, 1.694065894e-20, 1.084202172485e-18]
Component 07 Cov2:   [6.352747104407e-22, 1.000000000004e-06, 1.2705494208814e-20, 8.13151629364e-19]
Component 07 Cov3:   [1.6940658945085 97e-20, 1.27054942088e-20, 1.0000000338e-06, 2.16840434497e-17]
Component 07 Cov4:   [1.0842021724855 02e-18, 8.131516293641-19, 2.16840434497e-17, 1.0000001e-06]
Component 08 Weight: 3.6882757216073 41e-05
Component 08 Mean:   [7788.7843939041895, 14236890.933238905, 10704.980616625055, 134149.3895481316]
Component 08 Cov1:   [214960323.32424903, -5888710525.852478, 278441176.2129341, 3442818847.6342397]
Component 08 Cov2:   [-5888710525.852478, 848246803896.1973, -7802758587.39009, -100912629374.38963]
Component 08 Cov3:   [278441176.2129341, -7802758587.39009, 360741095.318045, 4461315807.809403]
Component 08 Cov4:   [3442818847.6342397, -100912629374.38963, 4461315807.809403, 55244928175.26433]
Component 09 Weight: 0.09745616500992937
Component 09 Mean:   [3534.6910872710346, 174.319690857637 06, 1116.39830064463, 981.6439098378561]
Component 09 Cov1:   [17594224.4194142, 157817.83888095556, 1008399.3670716931, 999121.5927289098]
Component 09 Cov2:   [157817.83888095556, 48923.665844972194, 107538.58822885802, 91124.8633912378]
Component 09 Cov3:   [1008399.3670716931, 107538.58822885802, 947064.3872114461, 783558.7893671305]
Component 09 Cov4:   [999121.5927289098, 91124.8633912378, 783558.7893671305, 708916.9362737124]
Component 10 Weight: 0.014240191226599166
Component 10 Mean:   [9260.348363025456, 58.858496586391 93, 9542.654233097774, 76866.977193913]
Component 10 Cov1:   [57032127.20471018, 17238.550614576197, 6246885.258315269, -68590703.41838416]
Component 10 Cov2:   [17238.550614576194, 4308.935760976065, 147401.92872797037, -61125.912096867796]
Component 10 Cov3:   [6246885.258315267, 147401.92872797034, 63443206.96238955, 101017364.15756738]
Component 10 Cov4:   [-68590703.41838416, -61125.91209686778, 101017364.15756738, 5431987634.770732]
Component 11 Weight: 0.06434734896560 39
Component 11 Mean:   [2795.861143413759, 18594.193147814203, 2248.728112585363, 1570.914279871272]
Component 11 Cov1:   [12999067.881595468, 3199510.6336428258, 71770.96225965641, 320419.02493012784]
Component 11 Cov2:   [3199510.633642827, 291711325.7978851, -1281559.4931380942, -1770394.0593545395]
Component 11 Cov3:   [71770.96225965646, -1281559.4931380937, 1317161.9116419412, 518677.37264435564]
Component 11 Cov4:   [320419.02493012784, -1770394.0593545402, 518677.37264435564, 373136.94951501035]
Component 12 Weight: 0.014452957808736443
Component 12 Mean:   [3750.194639974347, 83119.8525868498, 3486.772948402143, 5093.33749817264]
Component 12 Cov1:   [12967056.260599012, 41625074.20591106, 1597305.6135713116, -3247099.0316516105]
Component 12 Cov2:   [41625074.20591106, 10568182724.063202, -11193974.296009453, -147386630.6169421]
Component 12 Cov3:   [1597305.6135713116, -11193974.296009455, 3580739.5480363406, -871115.4493109959]
Component 12 Cov4:   [-3247099.0316516105, -147386630.6169421, -871115.4493109959, 13020851.484666102]
Component 13 Weight: 3.6882676206985 59e-05
Component 13 Mean:   [404.99999999999983, 21648888.999999993, 916.5999999999997, 9837.999999999996]
Component 13 Cov1:   [6171.200000999998, -7318648.399999998, -3254.399999999999, 166829.39999999994]
Component 13 Cov2:   [-7318648.399999998, 22971024229.199993, 18130591.999999993, -400962395.79999983]
Component 13 Cov3:   [-3254.399999999999, 18130591.999999993, 16567.040000999994, -296583.19999999984]
Component 13 Cov4:   [166829.39999999994, -400962395.79999983, -296583.19999999984, 7544478.400000998]
Component 14 Weight: 0.038653647781769 35
Component 14 Mean:   [3170.5655675849466, 955.0867931206915, 1998.3241199164913, 12220.865671235886]
Component 14 Cov1:   [10759472.62252164, -138116.57058745626, 831007.6528236055, 5385417.872541957]
Component 14 Cov2:   [-138116.57058745626, 740706.8666579265, 26495.30636656965, -1320903.0320714446]
Component 14 Cov3:   [831007.6528236055, 26495.30636656966, 424161.5827328364, 1840884.0925352192]
Component 14 Cov4:   [5385417.872541957, -1320903.9320714446, 1840884.0925352192, 110564453.49884804]
Component 15 Weight: 3.6882600695108 04e-05
Component 15 Mean:   [1455.2015333188983, 10596688.045961257, 4466.405133872815, 127667.5787852297]
Component 15 Cov1:   [1490020.8613155854, -112214261.40205364, 3694033.436352846, 216602822.2780526]
Component 15 Cov2:   [-112214261.40205364, 435613640661.8324, -851544959.271757, -66380709876.91346]
Component 15 Cov3:   [3694033.436352846, -851544959.2717571, 9993155.021959709, 615337266.5454571]
Component 15 Cov4:   [216602822.2780526, -66380709876.91346, 615337266.5454571, 41703667732.97232]
Component 16 Weight: 7.376535241397132e-06
Component 16 Mean:   [37591.99999999992, 10236.999999999978, 307713.9999999993, 8976298.99999998]
Component 16 Cov1:   [1.000000000063e-06, 1.7470054537119e-21, 5.5904174518e-20, 1.6398557858 84e-18]
Component 16 Cov2:   [1.7470054537119e-21, 1.0000000004e-06, 1.524659305057e-20, 4.472333961502e-19]
Component 16 Cov3:   [5.59041745187e-20, 1.524659305057e-20, 1.000000004878e-06, 1.431146867680e-17]
Component 16 Cov4:   [1.6398557858 84e-18, 4.47233396150e-19, 1.4311468676808e-17, 1.000000004e-06]
Component 17 Weight: 0.0034216289688 33473
Component 17 Mean:   [36413.6404096608, 180252.04578154706, 80378.68959894788, 699107.8354819418]
Component 17 Cov1:   [1437251574.6609042, -3658916322.466674, -255073437.33633828, -5695016294.413421]
```

```
Component 17 Cov2:   [-3658916322.466674, 292617139820.78516, -2579070508.8939614, -77246511143.74728]
Component 17 Cov3:   [-255073437.3363383, -2579070508.8939614, 12204829258.020863, -9634386635.002033]
Component 17 Cov4:   [-5695016294.413421, -77246511143.74728, -9634386635.002033, 1241654757253.3057]
Component 18 Weight: 0.01876912655207044
Component 18 Mean:   [158.98095650422582, 595223.9046738549, 878.366988672351, 1989.1536632303603]
Component 18 Cov1:   [11036.371783204306, -28598886.691683438, 71470.25233624804, 177072.04978214687]
Component 18 Cov2:   [-28598886.691683438, 441752100376.48926, -338310605.12612087, -848908665.6660432]
Component 18 Cov3:   [71470.25233624804, -338310605.12612087, 751351.7200629538, 1604533.221526752]
Component 18 Cov4:   [177072.04978214687, -848908665.6660432, 1604533.221526752, 5846728.743548622]
Component 19 Weight: 2.9506140957966885e-05
Component 19 Mean:   [13234.50000133628, 8008901.250112321, 12761.50000310488, 350335.5000896318]
Component 19 Cov1:   [190547320.29049274, 1887907328.190607, 262403047.05157286, 6984192295.589611]
Component 19 Cov2:   [1887907328.190607, 113630163334.75264, 3861309460.534889, 104527402791.21335]
Component 19 Cov3:   [262403047.05157286, 3861309460.534889, 378191037.31035656, 10082251586.526836]
Component 19 Cov4:   [6984192295.589611, 104527402791.21335, 10082251586.526836, 269518649514.76532]
Component 20 Weight: 0.07066963348551962
Component 20 Mean:   [2289.0418144235105, 15556.949471649072, 313.8436655771862, 269.51276619671523]
Component 20 Cov1:   [3846619.07528507, 1494195.655472811, 150068.3070202319, 37531.04283325003]
Component 20 Cov2:   [1494195.655472811, 288744565.57882816, 77684.6499363615, 19808.002878360996]
Component 20 Cov3:   [150068.30702023194, 77684.64993636149, 15380.421322172218, 18545.106575389804]
Component 20 Cov4:   [37531.04283325003, 19808.002878360963, 18545.106575389804, 42247.40965997398]
Component 21 Weight: 0.06901936821036839
Component 21 Mean:   [3946.259177518759, 71.82710912457613, 2125.370009461032, 3853.5854558875403]
Component 21 Cov1:   [15900297.598039791, -47565.66419907699, 839990.7045013789, -567340.2482033194]
Component 21 Cov2:   [-47565.664199077, 7259.685588933659, 28482.273553360723, 17717.481777281908]
Component 21 Cov3:   [839990.7045013789, 28482.273553360723, 1362881.8041042923, 58428.584883425545]
Component 21 Cov4:   [-567340.2482033193, 17717.48177728191, 58428.584883425545, 5862019.241771183]
Component 22 Weight: 0.02261013786393543
Component 22 Mean:   [6170.976575845918, 581.7462655636941, 5682.291265395138, 45226.20266167189]
Component 22 Cov1:   [19976816.15701097, 273203.3198821011, 2768412.9336796296, -31088542.95146286]
Component 22 Cov2:   [273203.3198821011, 272487.58959059085, 226716.61820990895, 1382316.4707613399]
Component 22 Cov3:   [2768412.9336796296, 226716.61820990886, 11208709.27718031, -41518341.4606498]
Component 22 Cov4:   [-31088542.951462865, 1382316.4707613399, -41518341.4606498, 1497526617.6766648]
Component 23 Weight: 0.059609945658038684
Component 23 Mean:   [7151.880776324767, 2116.508948563644, 2537.463912020829, 2492.4319398107955]
Component 23 Cov1:   [36448359.40432205, -1836553.0742765781, -367474.89692192926, -875708.8321573217]
Component 23 Cov2:   [-1836553.0742765781, 2991932.9052694226, -212286.58306506457, -7613.241288977587]
Component 23 Cov3:   [-367474.89692192926, -212286.58306506457, 2809499.4412285076, 1340011.3985144545]
Component 23 Cov4:   [-875708.832157322, -7613.241288977583, 1340011.3985144545, 2479432.466091874]
```