



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΥΦΥΗΣ ΔΙΑΧΕΙΡΙΣΗ ΡΟΩΝ ΔΕΔΟΜΕΝΩΝ ΠΟΥ ΠΡΟΕΡΧΟΝΤΑΙ
ΑΠΟ ΙΟΤ ΣΥΣΚΕΥΕΣ

ΙΩΑΝΝΗΣ ΤΕΡΠΟ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κωνσταντίνος Κολομβάτσος
Επίκουρος Καθηγητής
Τμήματος Πληροφορικής & Τηλεπικοινωνιών Πανεπιστημίου Θεσσαλίας

Λαμία έτος 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΥΦΥΗΣ ΔΙΑΧΕΙΡΙΣΗ ΡΟΩΝ ΔΕΔΟΜΕΝΩΝ ΠΟΥ ΠΡΟΕΡΧΟΝΤΑΙ
ΑΠΟ ΙΟΤ ΣΥΣΚΕΥΕΣ

ΙΩΑΝΝΗΣ ΤΕΡΠΙΟ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κωνσταντίνος Κολομβάτσος
Επίκουρος Καθηγητής
Τμήματος Πληροφορικής & Τηλεπικοινωνιών Πανεπιστημίου Θεσσαλίας

Λαμία έτος 2022



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

INTELLIGENT MANAGEMENT OF DATA FLOWS COMING FROM
IOT DEVICES

IOANNIS TERPO

FINAL THESIS

ADVISOR

Konstantinos Kolomvatsos
Assistant Professor
Lamia year 2022

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 26/052022

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η μεγάλη ανάπτυξη στον τομέα του υλικού, του λογισμικού και των τεχνολογιών επικοινωνίας έχουν τροφοδοτήσει την δημιουργία και την εμφάνιση συσκευών στο διαδίκτυο που παρέχουν παρατηρήσεις και μετρήσεις δεδομένων από τον φυσικό κόσμο στον ψηφιακό. Με την αύξηση του αριθμού αυτών των συσκευών και την ανάπτυξη των τεχνολογιών, ο όγκος των δεδομένων που εξάγονται θα αυξάνονται. Η τεχνολογία των συσκευών που είναι συνδεδεμένες στο διαδίκτυο αποκαλούμενη ως Διαδίκτυο των Πραγμάτων (IoT), δεν έχει σταματήσει να επεκτείνεται το σημερινό διαδίκτυο παρέχοντας συνδεσιμότητα και αλληλεπιδράσεις μεταξύ του φυσικού και του ψηφιακού κόσμου. Το IoT παράγει μεγάλα δεδομένα τα οποία χαρακτηρίζονται από την ταχύτητα τους όσον αφορά την εξάρτησή τους από τον χρόνο και την θέση, έχοντας πολλαπλές μορφές και διαφορετικές ποιότητες δεδομένων. Η ευφυής επεξεργασία και ανάλυση αυτών των μεγάλων δεδομένων είναι το κλειδί για την ανάπτυξη έξυπνων εφαρμογών που θα βελτιώσουν την ζωή του ανθρώπου. Η παρούσα πτυχιακή εργασία αναλαμβάνει την προσομοίωση αυτών των ροών δεδομένων από IoT συσκευές και την προ επεξεργασία τους και για την σωστή αξιοποίηση τους σε μοντέλα μηχανικής μάθησης για την πρόβλεψη διάγνωσης από ιατρικά δεδομένα.

ABSTRACT

Major developments in hardware, software and communication technologies have contributed to the creation and emergence of devices on the internet that provide observations and measurements of data from the physical world to the digital world. As the number of these conferences increase and technologies develop, the volume of data being extracted will increase. The technology of internet-connected devices called the Internet of Things (IoT) has not stopped at extending the present internet by providing connectivity and interactions among the physical and digital worlds. The IoT generates big data which is characterized by its speed in terms of its dependence on time and location, having multiple formats and different data qualities. Intelligent processing and analysis of this big data is the key to developing intelligent applications that will improve human life. This thesis undertakes the simulation of these data streams from IoT devices and their preprocessing for proper utilization in machine learning models for predictive diagnosis in the healthcare domain.

Table of Contents

ΠΕΡΙΛΗΨΗ.....	I
ABSTRACT.....	III
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	2
1.1 ΣΚΟΠΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	2
1.2 ΔΟΜΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	2
ΚΕΦΑΛΑΙΟ 2 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	4
2.1 ΕΠΙΒΛΕΠΟΜΕΝΗ ΜΑΘΗΣΗ	6
2.1.1 ΠΑΛΙΝΔΡΟΜΗΣΗ.....	7
2.1.2 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ.....	8
2.2 ΜΗ ΕΠΙΒΛΕΠΟΜΕΝΗ ΜΑΘΗΣΗ.....	9
2.2.1 ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ.....	9
2.2.2 ΣΥΣΤΑΛΟΠΟΙΗΣΗ	10
2.3 ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ	12
2.4 ΜΕΘΟΔΟΙ ΑΞΙΟΛΟΓΗΣΗΣ ΜΟΝΤΕΛΟΥ	13
2.4.1 ΟΡΘΟΤΗΤΑ ΜΟΝΤΕΛΟΥ.....	13
2.4.2 ΜΕΤΡΑ ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΤΗΤΑΣ ΤΗΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ.....	14
2.4.3 ΠΙΝΑΚΑΣ ΣΥΓΧΥΣΗΣ	14
2.4.4 ΔΙΑΧΩΡΙΣΜΟΣ.....	16
ΚΕΦΑΛΑΙΟ 3 INTERNET OF THINGS.....	18
3.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ	19
3.2 ΤΕΧΝΟΛΟΓΙΕΣ	21
3.3 ΣΥΣΚΕΥΕΣ.....	24
3.4 ΕΦΑΡΜΟΓΕΣ ΙΟΤ.....	27
3.4.1 ΣΥΣΤΗΜΑ ΥΓΕΙΑΣ.....	27
3.4.2 ΕΚΠΑΙΔΕΥΣΗ.....	28
3.4.3 ΈΞΥΠΝΕΣ ΠΟΛΕΙΣ.....	29
ΚΕΦΑΛΑΙΟ 4 ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ.....	31
4.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΓΕΝΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	32
4.2 ΔΕΔΟΜΕΝΑ	33
4.3 ΕΠΙΠΕΔΟ ΣΥΣΚΕΥΩΝ	36
4.3.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΚΕΥΩΝ.....	36
4.3.2 ΤΕΧΝΟΛΟΓΙΕΣ.....	38
4.3.3 ΥΛΟΠΟΙΗΣΗ	38
4.4 ΕΠΙΠΕΔΟ ΜΕΣΟΛΑΒΗΤΗ ΑΡΙ.....	42
4.4.1 ΤΕΧΝΟΛΟΓΙΕΣ.....	43

4.4.2 ΥΛΟΠΟΙΗΣΗ	44
4.5 ΕΞΥΠΗΡΕΤΗΤΗΣ ΚΑΙ ΜΟΝΤΕΛΟ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ	46
4.5.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ	46
4.5.2 ΤΕΧΝΟΛΟΓΙΕΣ.....	48
4.5.3 ΥΛΟΠΟΙΗΣΗ	48
<u>ΚΕΦΑΛΑΙΟ 5 ΠΕΡΙΓΡΑΦΗ ΕΚΤΕΛΕΣΗΣ</u>	<u>59</u>
<u>ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΕΚΤΑΣΕΙΣ</u>	<u>66</u>
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	66
6.2 ΒΕΛΤΙΩΣΕΙΣ	66
6.2.1 ΣΥΣΚΕΥΕΣ ΙΟΤ.....	66
6.2.2 ΜΟΝΤΕΛΟ ΠΡΟΒΛΕΨΗΣ	67
<u>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</u>	<u>68</u>

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

Στην σημερινή κοινωνία παρατηρείται εκθετική ανάπτυξη στον τεχνολογικό τομέα με αποτέλεσμα να κινείται στους ίδιους ρυθμούς η παραγωγή των δεδομένων σε πολύ μεγάλες ποσότητες. Καθημερινά χρησιμοποιούνται αλλά και ερευνώνται τεχνολογικοί μέθοδοι αξιοποίησης της μεγάλης αυτής πληροφορίας που παράγεται με σκοπό την επίλυση προβλημάτων σε τομείς που συνδέονται άρρηκτα ή και μη με την τεχνολογία.

Τα τελευταία χρόνια η αύξηση των έξυπνων συσκευών στο περιβάλλον που ονομάζεται Δίκτυο των Πραγμάτων (IoT), έχει συμβάλει και αυτό με την σειρά του, δίνοντας την δυνατότητα στην αυτόματη ή και μη συλλογή σημαντικών δεδομένων άμεσα από τις συσκευές από οποιαδήποτε χρονική στιγμή και από οποιονδήποτε χώρο.

Αν αναλογιστούμε την ποσότητα της παραγόμενης πληροφορίας που συλλέγεται, καθίσταται αδύνατη η διαδικασία της επεξεργασίας να γίνει από τον άνθρωπο. Επομένως κρίνεται απαραίτητη η δημιουργία ενός αυτόματου και αυτόνομου συστήματος, το οποίο θα είναι υπεύθυνο για την συλλογή, επεξεργασία και ανάλυση των δεδομένων.

1.1 Σκοπός της πτυχιακής εργασίας

Βασικός σκοπός της εργασίας είναι η αξιοποίηση των δεδομένων που προέρχονται από έξυπνες IoT συσκευές. Αναπτύσσεται μια προσομοίωση συσκευών IoT και πραγματοποιείται η μεταφορά των δεδομένων, η συλλογή αυτών των δεδομένων και η κατάλληλη προ επεξεργασία τους. Τα δεδομένα αυτά προέρχονται από τον τομέα της υγείας και με την σωστή ανάλυση αυτών εφαρμόζονται σε ένα μοντέλο μηχανικής μάθησης το οποίο θα είναι υπεύθυνο για την συσταδοποίηση των ασθενών σε ομάδες και την τελική εξαγωγή πρόβλεψης της μελλοντικής διάγνωσης.

1.2 Δομή της πτυχιακής εργασίας

Η πτυχιακή εργασία δομείται σε 6 κεφάλαια κατά τον ακόλουθο δομημένο τρόπο. Στο Κεφάλαιο 2 γίνεται ανάλυση της έννοιας της Μηχανικής Μάθησης και οι επιμέρους κατηγορίες της, αναφέροντας διάφορους δημοφιλείς αλγόριθμους που χρησιμοποιούνται στις συγκεκριμένες κατηγορίες.

Στο Κεφάλαιο 3 περιγράφεται η τεχνολογία του διαδικτύου των πραγμάτων (IoT). Σε αυτό το κεφάλαιο αναλύεται η αρχιτεκτονική του IoT και γίνεται λόγος για τις συσκευές που χρησιμοποιούνται σε ένα περιβάλλον IoT. Τέλος αναφέρονται μερικές εφαρμογές στον τομέα του Διαδικτύου των Πραγμάτων.

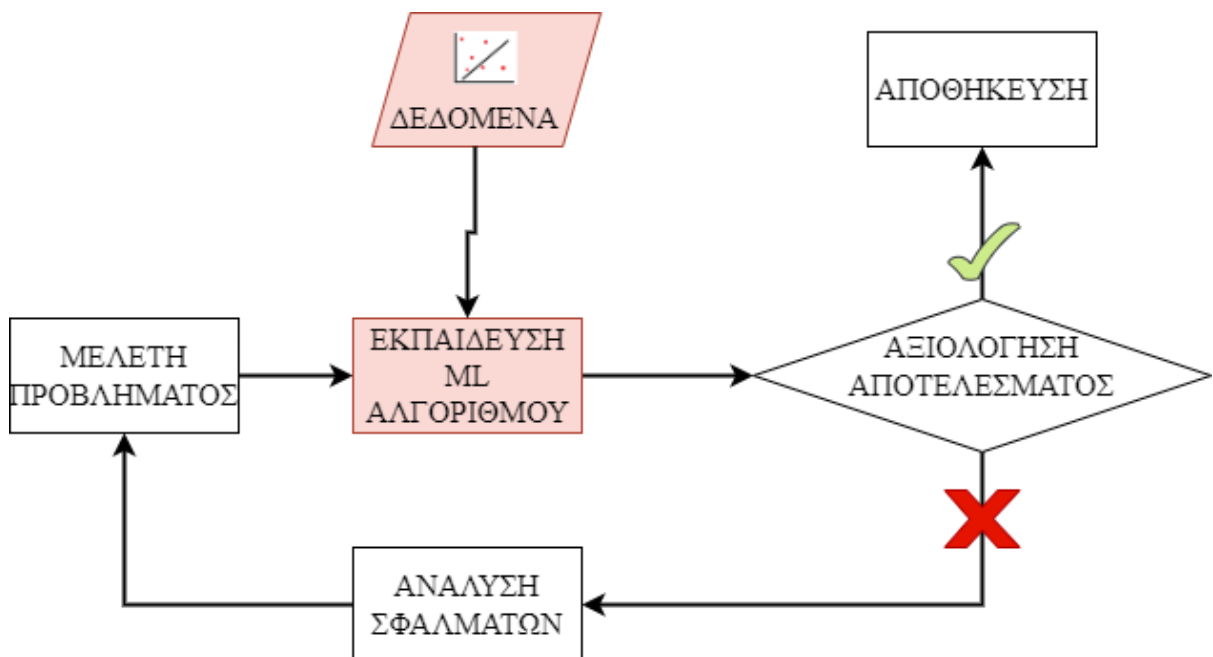
Στο Κεφάλαιο 4 περιγράφεται αναλυτικά η ανάπτυξη της προσομοίωσης των συσκευών IoT, η ευφυής διαχείριση των ροών δεδομένων που προέρχονται από τις αναφερόμενες συσκευές και η ενσωμάτωσή τους σε μοντέλα μηχανικής μάθησης για την εξαγωγή προβλέψεων.

Στο κεφάλαιο 5, γίνεται η παρουσίαση του τελικού σταδίου της υλοποίησης όλων των επιμέρους προγραμμάτων και η τελική δοκιμή του τελικού μοντέλου πρόβλεψης διάγνωσης.

Τέλος στο κεφάλαιο 6, αναφέρονται τα τελικά συμπεράσματα που προέκυψαν στην υλοποίηση της συνολικής εργασίας ακολουθούμενη από μελλοντικές προεκτάσεις και βελτιώσεις.

ΚΕΦΑΛΑΙΟ 2 Μηχανική Μάθηση

Η μηχανική μάθηση στον τομέα της πληροφορικής είναι ένα υποσύνολο της τεχνητής νοημοσύνης. Χρησιμοποιώντας τις υπολογιστικές δυνατότητες ενός υπολογιστή σχεδιάζουμε συστήματα τα οποία έχουν την δυνατότητα εκμάθησης από δεδομένα παραδειγμάτων ή παρελθοντικών εμπειριών σε περιβάλλον σχεδιασμένο κατάλληλα για την εκπαίδευσή τους. Με την συνεχή εκτέλεση αυτών των συστημάτων, το μοντέλο μηχανικής μάθησης θα μπορεί να εκπαιδευτεί και να βελτιώνεται με τον καιρό με τελικό στόχο να διαμορφωθεί ένα μοντέλο το οποίο θα μπορεί να χρησιμοποιηθεί για την πρόβλεψη αποτελεσμάτων αναπάντητων ερωτήσεων έχοντας ως βάση και σημείο αναφοράς την προηγούμενη μάθηση.

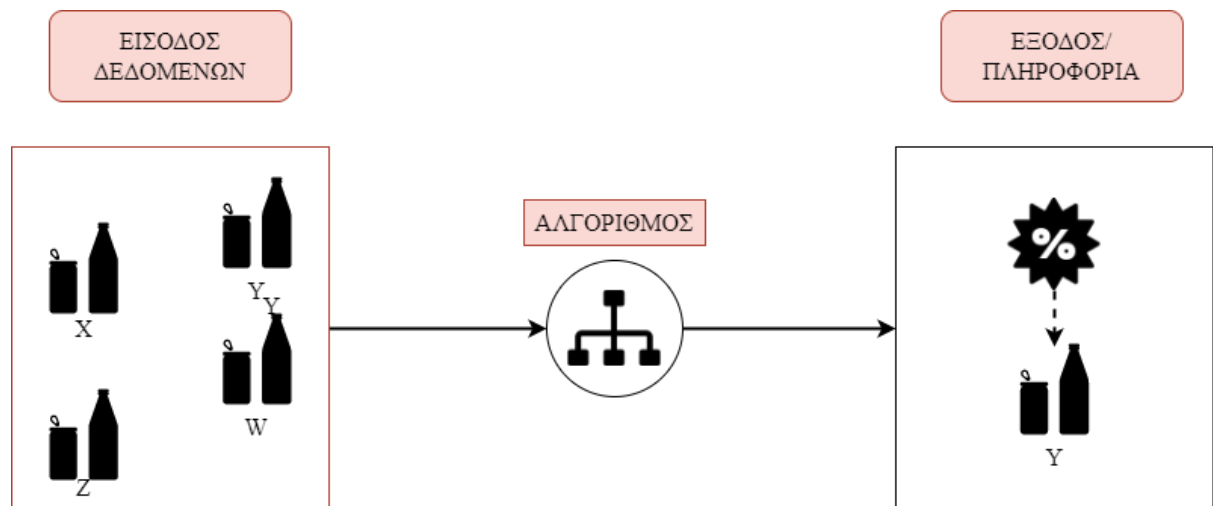


Σχήμα 1: Προσέγγιση Μηχανικής Μάθησης

Η προέλευση της σύγχρονης έννοιας της μηχανικής μάθησης όπως την ξέρουμε σήμερα συνδέεται με το όνομα του ψυχολόγου Frank Rosenblatt από το πανεπιστήμιο Cornell, ο οποίος βασιζόταν σε ιδέες για το πως λειτουργεί το νευρωνικό σύστημα του ανθρώπου, δημιούργησε μια ομάδα η οποία κατασκεύασε μια μηχανή η οποία θα αναγνώριζε γράμματα από το Rosenblatt αλφάβητο (1957,1959,1960). Η μηχανή ονομάστηκε “perceptron” από τον δημιουργό της. Χρησιμοποιούσε αναλογικά και ψηφιακά σήματα και περιλάμβανε μια τιμή κατωφλίου που μετέτρεπε τα αναλογικά σήματα σε διακριτά. Αποτελέσε πρωτότυπο των σύγχρονων τεχνητών νευρωνικών δικτύων (ANN) και το μοντέλο της μάθησης του έμοιαζε αρκετά στα μοντέλα των ζώων και της ανθρώπινης μάθησης που αναπτύχθηκαν στην ψυχολογία. Ο Rosenblatt πραγματοποίησε την πρώτη μαθηματική μελέτη του perceptron Rosenblatt (1959). Ωστόσο, το θεώρημα Novikoff (1962), το οποίο δίνει τις προϋποθέσεις για την σύγκλιση ενός perceptron

αλγόριθμοι μάθησης με πεπερασμένο αριθμό βημάτων έχει γίνει περισσότερο γνωστό.

Ένα παράδειγμα για την κατανόηση της λειτουργίας της μηχανικής μάθησης είναι μια αλυσίδα από Σούπερ μάρκετ, η οποία έχει εκατοντάδες καταστήματα σε όλη την χώρα παρέχοντας χιλιάδες προϊόντα σε εκατομμύρια καταναλωτές. Στα σημεία που πωλούνται τα προϊόντα καταγράφονται όλες οι λεπτομέρειες της κάθε συναλλαγής, όπως ημερομηνία, αναγνωριστικό καταναλωτή, όνομα προϊόντος και η ποσότητα του προϊόντος, τα χρήματα που δαπανήθηκαν και άλλες λεπτομέρειες. Η αποθήκευση αυτών των δεδομένων κρίνεται χρήσιμη μόνο όταν τα δεδομένα αυτά αναλύονται και μετατρέπονται σε πληροφορία από την οποία μπορούμε να επωφεληθούμε, όπως για παράδειγμα η πρόβλεψη της συμπεριφοράς των καταναλωτών σε σχέση με ένα σύνολο προϊόντων για την μεγιστοποίηση του κέρδους αφού γίνει αλλαγή της αρχικής του τιμής.



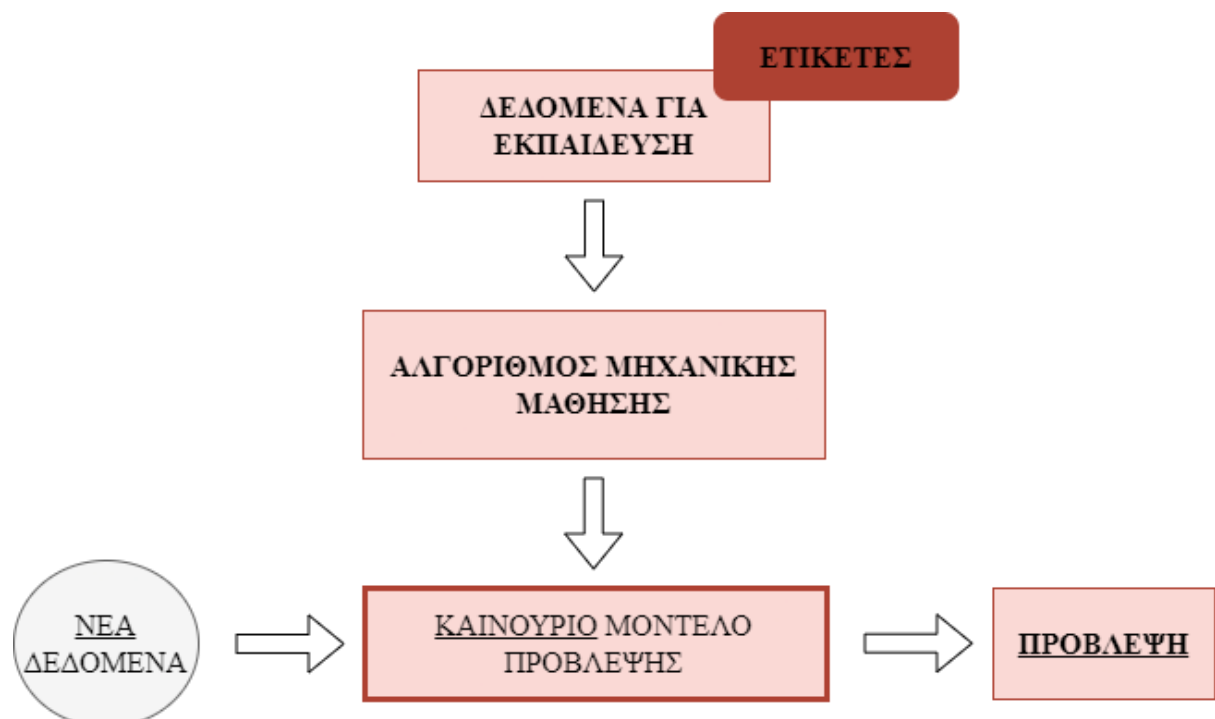
Σχήμα 2: Παράδειγμα μηχανικής μάθησης σε Super Market

Υπάρχουν πολλοί διαφορετικοί τύποι αλγορίθμων που μπορεί κάποιος να αναπτύξει στην μηχανική μάθηση. Αυτό που κάνει τον κάθε τύπο να ξεχωρίζει είναι το πόσο και τι είδους επίβλεψη θα γίνει κατά την διάρκεια της εκπαίδευσης και την προ απαιτούμενη έξοδος που απαιτείται από τον προγραμματιστή. Στα επόμενα υποκεφάλαια θα αναφερθούν τα είδη της μηχανικής μάθησης και οι διαφορετικοί τύποι αλγορίθμων που υπάρχουν.

2.1 Επιβλεπόμενη Μάθηση

Η επιβλεπόμενη μάθηση (supervised learning) εκπροσωπεί ένα μεγάλο κομμάτι της ερευνητικής δραστηριότητας στην μηχανική μάθηση. Έχει μεγάλη επιτυχία σε πραγματικές εφαρμογές. Η επιβλεπόμενη μάθηση αποκαλείται επίσης κατηγοριοποιημένη μάθηση (classification learning) ή επαγωγική μάθηση (inductive learning) στην μηχανική μάθηση. Πιο συγκεκριμένα το μοντέλο εκπαιδεύεται από ένα σύνολο δεδομένων με αποτέλεσμα το καινούριο παραγόμενο μοντέλο να μπορεί να προβλέψει αποτελέσματα με την βοήθεια ενός νέου συνόλου δεδομένων.

Αυτό το είδος μάθησης είναι ανάλογη με την ανθρώπινη μάθηση από προηγούμενες εμπειρίες για την απόκτηση κάποιας νέας γνώσης ώστε να βελτιωθεί κάποια διαδικασία επίλυσης ορισμένων πραγματικών εφαρμογών. Υπάρχουν δύο ειδών επιβλεπόμενης μάθησης, η κατηγοριοποίηση (classification) και η παλινδρόμηση (regression).



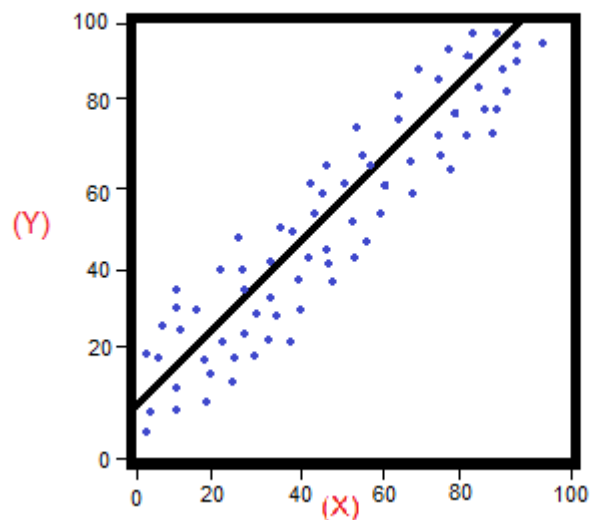
Σχήμα 3: Παράδειγμα Ροής Μηχανικής Μάθησης

Στην πάνω εικόνα (Σχήμα 3) παρουσιάζεται η λειτουργία της επιβλεπόμενης μάθησης. Πιο αναλυτικά, δίνεται από το χρήστη ένα ή περισσότερα σύνολα εκπαίδευσης (training data), τα σύνολα αυτά περιλαμβάνουν υποχρεωτικά ετικέτα κατηγοριοποίησης (label) και στην συνέχεια γίνεται η επιλογή και η χρήση του αλγορίθμου μηχανικής μάθησης για την εκπαίδευση του μοντέλου. Έχοντας γίνει η εκπαίδευση και το νέο μοντέλο πρόβλεψης έχει παραχθεί, εισάγονται νέα σύνολα δεδομένων χωρίς ετικέτα και εφαρμόζονται στο μοντέλο πρόβλεψης το οποίο θεμελιώθηκε από τον προηγούμενο αλγόριθμο μηχανικής μάθησης και παράγεται το αποτέλεσμα της πρόβλεψης.

2.1.1 Παλινδρόμηση

Η μέθοδος της παλινδρόμησης είναι ένα είδος επιβλεπόμενης μηχανικής μάθησης που μαθαίνει από σύνολα δεδομένων με δεδομένες ετικέτες και μετά την διαδικασία της εκπαίδευσης είναι σε θέση να προβλέψει μια σειρά συνεχόμενων αποτελεσμάτων αφού δοθούν τα δεδομένα στον αλγόριθμο. Τα δεδομένα αυτά είναι ένας αριθμός από τον παράγοντα της μεταβλητής και την μεταβλητή αποτελέσματος. Με βάση την πιο πάνω λογική ο αλγόριθμος προσπαθεί να βρει μια σχέση που έχουν οι μεταβλητές μεταξύ τους, η οποία θα αποτελέσει σημαντικός παράγοντας για την πρόβλεψη του αποτελέσματος.

Για την καλύτερη κατανόηση της παλινδρόμησης υπάρχει το παράδειγμα της αγοράς ενός αυτοκινήτου, η κατανάλωση καυσίμου είναι ένας σημαντικός παράγοντας για την επιλογή του αυτοκινήτου. Έχοντας τα χαρακτηριστικά του αυτοκινήτου (βάρος, ίπποι, κυβικά, κ.α.) μια πιθανή μέθοδος για την πρόβλεψη χιλιομέτρων ανά λίτρο καυσίμου είναι η παλινδρόμηση. Με την απεικόνιση του μέσου όρου χιλιομέτρων ανά λίτρο του κάθε αυτοκινήτου, δίνεται η δυνατότητα χρήσης τεχνικών παλινδρόμησης για την αναζήτηση συσχετίσεων μεταξύ τιμής χιλιομέτρων ανά λίτρο και τα εισαγόμενα χαρακτηριστικά των αυτοκινήτων. Σε αυτήν την περίπτωση η συνάρτηση παλινδρόμησης μπορεί να αναπαρασταθεί με, $Y = f(X)$, όπου το Y θα είναι τιμή χιλιομέτρων ανά λίτρο και το X θα είναι τα χαρακτηριστικά εισόδου (βάρος, ίπποι, κυβικά, κ.α.). Η συνάρτηση f και η κλίση του βοηθάει στην πρόβλεψη για την τελική αγορά ή όχι του αυτοκινήτου. Αυτός ο μηχανισμός ονομάζεται παλινδρομικός.



Σχήμα 4: Παράδειγμα Δεδομένων Παλινδρόμησης

Στην εικόνα (Σχήμα 4) απεικονίζεται μια ευθεία γραμμικής παλινδρόμησης, στην οποία αφού υπάρχουν οι τιμές X και Y , εφαρμόζεται η ευθεία γραμμή έχοντας ως σκοπό την μείωση της απόστασης μεταξύ αυτών.

Οι Αλγόριθμοι που μπορούν να εφαρμόσουν την μέθοδο της παλινδρόμησης είναι η γραμμική παλινδρόμηση (linear regression), τα νευρωνικά δίκτυα (neural networks), τα δέντρα αποφάσεων (decision trees) και τα τυχαία δάση (random forests).

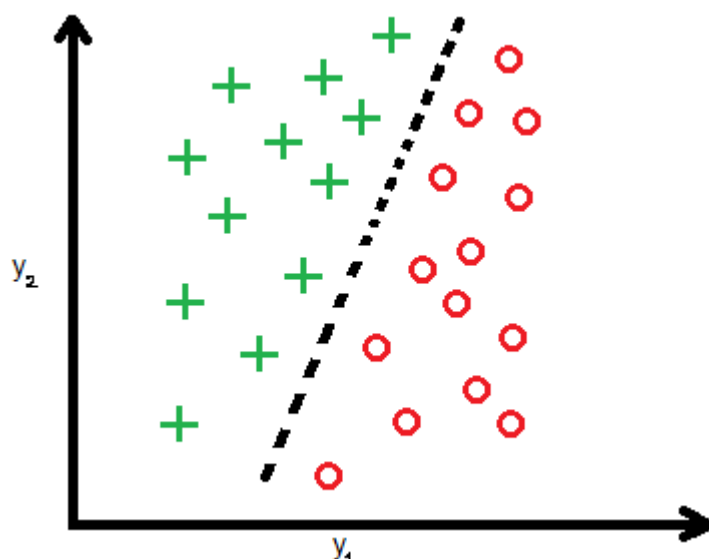
2.1.2 Κατηγοριοποίηση

Μια από τις κύριες εργασίες της μηχανικής μάθησης είναι η κατηγοριοποίηση (classification). Η κατηγοριοποίηση έχει ως στόχο την πρόβλεψη κατηγοριών στην μορφή ετικετών με την βοήθεια προηγούμενων παρατηρήσεων. Το μεγαλύτερο ποσοστό των κατηγοριοποιήσεων διακρίνεται σε

- Δυαδική κατηγοριοποίηση (binary classification) στην οποία πραγματοποιείται η εκμάθηση διάφορων κανόνων από τον αλγόριθμο ώστε να μπορεί να γίνει διάκριση των ετικετών ανάμεσα σε δύο πιθανές κατηγορίες.
- Πολλαπλή κατηγοριοποίηση (multiclass classification) η οποία κατηγοριοποιεί και αυτήν τα δεδομένα αλλά σε περισσότερες από δυο κατηγορίες.

Η διαδικασία της κατηγοριοποιήσεως χωρίζεται σε δυο στάδια, την μάθηση και την κατηγοριοποίηση-εκπαίδευση (training).

Αλγόριθμοι που χρησιμοποιούν την κατηγοριοποίηση είναι οι k κοντινότεροι γείτονες (k nearest neighbors), απλοϊκός Bayes (Naïve Bayes), λογιστική παλινδρόμηση (logistic regression), τα δέντρα αποφάσεων (decision trees), μηχανές διανυσμάτων υποστήριξης (gradient boosting machines) και τα νευρωνικά δίκτυα (neural networks).



Σχήμα 5: Παράδειγμα αναπαράστασης κατηγοριοποίησης

Στην παραπάνω αναπαράσταση (Σχήμα 5), φαίνεται ότι η κατηγοριοποίηση είναι δυαδική και το συγκεκριμένο σύνολο δεδομένων αποτελείται από Y δείγματα. Η διαγώνια γραμμή που εμφανίζεται ανάμεσα στα δείγματα αποτελεί σημείο διαχωρισμού για την εφαρμογή της μεθόδου κατηγοριοποίησης. Στην περίπτωση της πολλαπλής κατηγοριοποίησης θα υπήρχαν περισσότερες από δύο κατηγοριοποιήσεις.

2.2 Μη Επιβλεπόμενη Μάθηση

Σε αντίθεση με την επιβλεπόμενη μηχανική μάθηση, στην μη επιβλεπόμενη μάθηση η διαδικασία της μάθησης πραγματοποιείται χωρίς την εποπτεία του προγραμματιστή, όπου ο αλγόριθμος προσπαθεί να βρει κάποια προηγούμενα άγνωστα μοτίβα σε σύνολα δεδομένων. Η διαδικασία αυτήν βοηθάει στην μοντελοποίηση συναρτήσεων πυκνότητας πιθανότητας, στην εύρεση σφαλμάτων στα δεδομένα και πολλά ακόμα.

Μεγάλη πρόκληση στην μη επιβλεπόμενη μάθηση αποτελεί η αξιολόγηση της εξόδου του αλγόριθμου και η χρησιμότητα της παραγόμενης πληροφορίας. Οι αλγόριθμοι εφαρμόζονται σε δεδομένα χωρίς ετικέτες δημιουργώντας το πρόβλημα να μην γνωρίζουμε ποιο θα πρέπει να είναι το σωστό αποτέλεσμα.

Όπως στην επιβλεπόμενη μάθηση, αντίστοιχα και στην μη επιβλεπόμενη μάθηση υπάρχουν δυο υποκατηγορίες, η ομαδοποίηση δεδομένων (clustering) και ο μετασχηματισμός δεδομένων (transformation). Οι αλγόριθμοι που χρησιμοποιούν μη επιβλεπόμενη μάθηση είναι:

- Ανάλυση κύριων συνιστώσεων (Principal Components Analysis – PCA)
- Γραμμική διαχωριστική ανάλυση (Linear Discriminant Analysis - LDA)
- Τυποποίηση (Standardization)
- Κανονικοποίηση (Normalization)
- K- Means
- Fuzzy C-Means
- PAM (Partitioning Around Medoids)
- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
- DBSCAN (Density-based spatial clustering of applications with noise)

2.2.1 Μετασχηματισμός

Για την δημιουργία καινούριων αναπαραστάσεων ενός συνόλου δεδομένων, γίνεται η χρήση αλγορίθμων που έχουν ως σκοπό να μετασχηματίσουν (Transformation) τα δεδομένα. Οι νέοι αυτοί μετασχηματισμοί κατανοούνται ευκολότερα από τους ανθρώπους ή τους αλγορίθμους που θα χρησιμοποιηθούν για την μηχανική μάθηση, ώστε να γίνει καλύτερη η σύγκριση των δεδομένων που είχαμε στην αρχή.

Η πιο συνηθισμένη εφαρμογή αλγορίθμων για τον μετασχηματισμό των δεδομένων είναι η μείωση των διαστάσεων (Dimensionality Reduction). Ο λόγος που χρησιμοποιείται αυτός ο αλγόριθμος είναι επειδή τα αρχικά δεδομένα που χρησιμοποιούμε και επεξεργαζόμαστε (raw data) έχουν πολλές διαστάσεις, αυξάνοντας τις πιθανότητες να παρουσιαστεί κάποιο πρόβλημα, όπως περιορισμένος χώρος στη μνήμη και χαμηλή υπολογιστική απόδοση των αλγορίθμων της μηχανικής μάθησης. Η διαδικασία της μείωσης των διαστάσεων σε ένα σύνολο δεδομένων πραγματοποιείται κατά την διαδικασία της προ επεξεργασίας στην οποία εφαρμόζεται μείωση του θορύβου των δεδομένων αλλά και συμπίεση των δεδομένων με μικρότερο διάστημα τιμών, χωρίς να αλλοιωθεί η πραγματική σημασία της πληροφορίας. Αλγόριθμοι που χρησιμοποιούν την μείωση διαστάσεων είναι η τυποποίηση (Standardization), η κανονικοποίηση (Normalization), γραμμική διαχωριστική ανάλυση (Linear Discriminant Analysis - LDA) και η ανάλυση κυρίων συνιστωσών (Principal Components Analysis-PCA).

2.2.2 Συσταδοποίηση

Η συσταδοποίηση (clustering) είναι μια από τις πιο διαδεδομένες τεχνικές για διερευνητική ανάλυση δεδομένων. Σε πολλούς επιστημονικούς κλάδους γίνεται η προσπάθεια να αποκτηθεί μια πρώτη διαίσθηση για δεδομένα με την ανάλυση για σημασιολογικά απαραίτητες ομαδοποιήσεις μεταξύ των σημείων των δεδομένων. Πιο αναλυτικά, η συσταδοποίηση είναι η διεργασία της ομαδοποίησης κάποιων σετ αντικειμένων σε ομάδες με αντικείμενα που έχουν παρόμοια χαρακτηριστικά, έχοντας ως αποτέλεσμα τα αντικείμενα με ίδια χαρακτηριστικά να βρίσκονται στην ίδια ομάδα ενώ αντικείμενα με διαφορετικά χαρακτηριστικά να διαφοροποιούνται σε άλλες ομάδες.

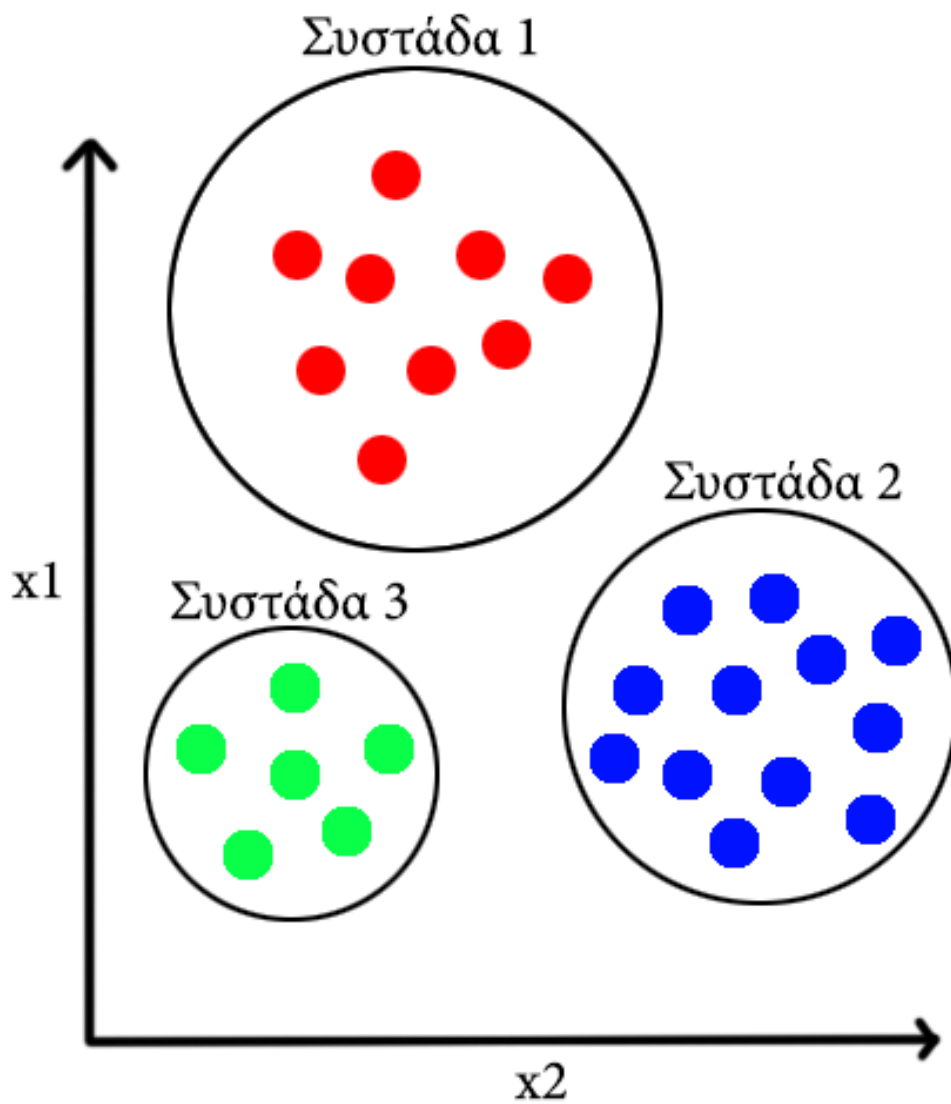
Ο τρόπος με τον οποίο θα γίνει η δημιουργία των συστάδων καθορίζεται από την επιλογή του αλγόριθμου που θα χρησιμοποιηθεί. Οι διαθέσιμοι αλγόριθμοι είναι οι εξής:

- Ασαφής συσταδοποίηση (Fuzzy Clustering)
- Διαιρετική συσταδοποίηση (Partitional Clustering)
- Ιεραρχική συσταδοποίηση (Hierarchical Clustering)
- Συσταδοποίηση με βάση την πυκνότητα (Density-based Clustering)
- Συσταδοποίηση με βάση το πλέγμα (Grid-based Clustering)
- Συσταδοποίηση υποχώρων (Subspace Clustering)

Η βασικότερη διαφορά της συσταδοποίησης με την κατηγοριοποίηση είναι ότι η συσταδοποίηση δεν βασίζεται σε προκαθορισμένες κατηγορίες από τον χρήστη. Παραδείγματα παρόμοιων αλγορίθμων αποτελούν ο SVD (Singular Value Decomposition), ο K-MEANS, Fuzzy C-MEANS, ο PAM (partition around medoids), ο BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) και ο DBSCAN (Density-based spatial clustering of applications with noise).

Στην παρακάτω εικόνα (Σχήμα 6) αναπαριστάτε το παράδειγμα την συσταδοποίησης σε ένα σύνολο δεδομένων. Τα δεδομένα οργανώνονται σε τρεις μικρότερες ομάδες,

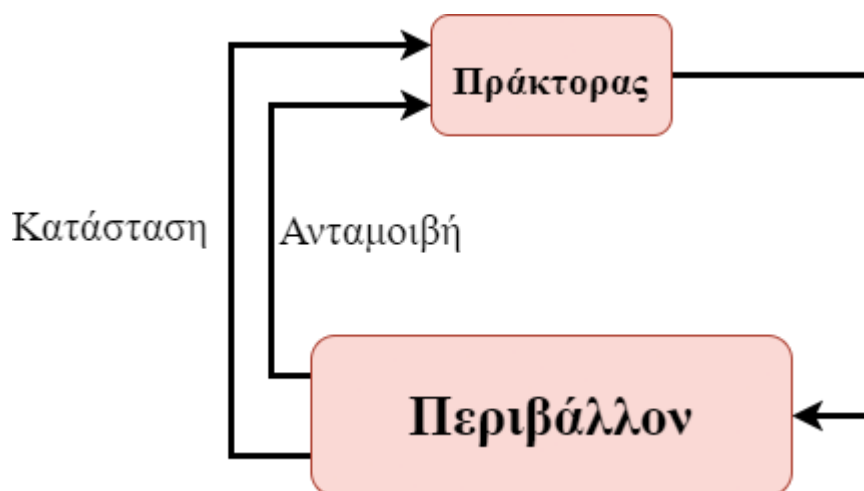
έχοντας ως κοινό χαρακτηριστικό μεταξύ των αντικειμένων τα χαρακτηριστικά x_1 και x_2 .



Σχήμα 6: Παράδειγμα - Αναπαράσταση της συσταδοποίησης

2.3 Ενισχυτική Μάθηση

Πέρα από την επιβλεπόμενη και την μη επιβλεπόμενη μάθηση, οι οποίες θεωρούνται από τα πιο σημαντικά είδη μάθησης, ένα ακόμα είδος μηχανικής μάθησης είναι η ενισχυτική μάθηση (Reinforcement Learning). Η μέθοδος της ενισχυτικής μάθησης περιλαμβάνει την εκμάθηση του τρόπου με τον οποίο το αριθμητικό σήμα ανταμοιβής θα μεγιστοποιηθεί αφού έχει γίνει η χαρτογράφηση καταστάσεων σε ενέργειες. Η μέθοδος αυτήν αποτελείται από τρία συστατικά στοιχεία (Components), τον ευφυή πράκτορα (agent), το περιβάλλον (environment) και τις ενέργειες (actions). Ο μαθητευόμενος και ο υπεύθυνος λήψης αποφάσεων ονομάζεται πράκτορας (agent). Το μέρος στο οποίο ο πράκτορας αλληλοεπιδρά και οτιδήποτε πέρα από αυτόν είναι το περιβάλλον. Αυτά τα δύο στοιχεία αλληλοεπιδρούν συνεχώς, ο πράκτορας επιλέγει ενέργειες και το περιβάλλον ανταποκρίνεται σε αυτές τις ενέργειες και παρουσιάζει νέες καταστάσεις στον πράκτορα.



Σχήμα 7: Αλληλεπίδραση Πράκτορα-Περιβάλλον στην Ενισχυτική μάθηση

Στην παραπάνω εικόνα του σχήματος παρουσιάζονται όλες οι αλληλεπίδρασης του πράκτορα με το περιβάλλον. Αναλυτικότερα, ο πράκτορας τοποθετείτε σε ένα σχεδιασμένο περιβάλλον με πιθανότητες καταστάσεις (states), με την κάθε ενέργεια του πράκτορα να του αποδίδετε μια ανταμοιβή (reward). Για την καλύτερη κατανόηση της ενισχυτικής μάθησης, ένα παράδειγμα αποτελούν τα παιχνίδια όπως το σκάκι, όπου ο πράκτορας θα πρέπει να διαλέξει ποιες ενέργειες θα πρέπει να

ακολουθήσει, καθώς και οι ανταμοιβές που θα του δίνονται ανάλογα αν κερδίσει ή αν χάσει.

2.4 Μέθοδοι Αξιολόγησης Μοντέλου

Μετά την δημιουργία του μοντέλου δεν μπορεί να γίνει η αξιολόγηση του με τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου της μηχανικής μάθησης. Ο λόγος που αυτό καθίσταται αδύνατο είναι επειδή το μοντέλο το οποίο έχει δημιουργηθεί είναι σε θέση να θυμάται μαζικά το σύνολο δεδομένων που έχει χρησιμοποιήσει και κάθε φορά που θα γίνεται η εισαγωγή ενός δεδομένου που έχει θα βγάζει σωστά αποτελέσματα καταργώντας τον σκοπό που θέλουμε να εφαρμόσουμε το μοντέλο μας. Η εισαγωγή ως είσοδο δεδομένα που δεν έχουν ξαναεμφανιστεί ή δεν έχει ξαναεκπαιδευτεί το μοντέλο μας με αυτά, χωρίς ετικέτα (label) κατηγοριοποίησης, γίνεται πιο σωστή ανάδειξη της αποδοτικότητας του μοντέλου και έτσι γίνεται η πραγματική αξιολόγηση του μοντέλου.

Αναλυτικότερα, με αυτόν τον τρόπο διαχωρισμού, τα δεδομένα χωρίζονται σε δύο τύπους, τα δεδομένα τα οποία υπάρχουν για να χρησιμοποιηθούν ώστε να εκπαιδευτεί και ονομάζονται σύνολα εκπαίδευσης, και τα δεδομένα που θα χρησιμοποιηθούν στην αξιολόγηση του μοντέλου, εκτιμώντας το ποσοστό της αποτελεσματικότητας του, αυτά τα δεδομένα ονομάζονται σύνολα ελέγχου.

Τα μέτρα με τα οποία αξιολογούν την αποτελεσματικότητα σε ένα σύστημα είναι η ορθότητα (accuracy), η ακρίβεια (precision), η ανάκληση (recall) και το αρμονικό μέσο (F measure). Στον υπολογισμό αυτών των μέτρων, βασικό ρόλο παίζει ο υπολογισμός του πίνακα σύγχυσης (confusion matrix), αφού με αυτόν τον τρόπο γίνεται πιο εύκολος ο υπολογισμός των μέτρων.

2.4.1 Ορθότητα Μοντέλου

Ένα από τα σημαντικότερα, μέτρα αξιολόγησης ενός μοντέλου κατηγοριοποίησης είναι ο υπολογισμός της ορθότητας του μοντέλου, δίνοντας στον προγραμματιστή που έφτιαξε το μοντέλο μηχανικής μάθησης την δυνατότητα της αποτίμησης του ποσοστού ακριβείας που θα έχει το συγκεκριμένο μοντέλο στην κατηγοριοποίηση σε μελλοντικά δεδομένα.

Συγκεκριμένα η ορθότητα ενός μοντέλου ως προς ένα σύνολο δεδομένων είναι το ποσοστό εκ των οποίων η κατηγοριοποίηση έγινε σωστά από το μοντέλο εκπαίδευσης, χωρίς να έχει σημασία αν η προβλέψει που έγινε ήταν θετική ή αρνητική. Όταν το μοντέλο που έχει κατασκευαστεί έχει αποδεκτή ακρίβεια, τότε θα μπορεί να χρησιμοποιηθεί με σκοπό την κατηγοριοποίηση καινούριων άγνωστων δεδομένων

$$\text{Ακρίβεια} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{TrueNegative} + \text{FalsePositive} + \text{FalseNegative}}$$

Εξίσωση 1: Πρώτος τύπος ορθότητας

$$\text{Ακρίβεια} = \frac{TP + TN}{TP + TN + FP + FN}$$

Εξίσωση 2: Δεύτερος τύπος ορθότητας

Πιο συγκεκριμένα, όπως βλέπουμε στον αναλυτικό τύπο της (εξίσωσης 1 και 2), η ορθότητα μπορεί να οριστεί ως μια τιμή σωστών προβλέψεων, θετικών και αρνητικών δεδομένων-δειγμάτων, ως προς τον συνολικό αριθμό όλων των δειγμάτων που χρησιμοποιήθηκαν.

2.4.2 Μέτρα αποτελεσματικότητας της κατηγοριοποίησης

Ένα από τα πιο σημαντικά μέτρα της αποτελεσματικότητας κάποιου αλγόριθμου με στόχο την κατηγοριοποίηση, για την καλύτερη ακρίβεια των προβλέψεων, βασικό ρόλο διαθέτουν η ακρίβεια, ο αρμονικός μέσος και η ανάκληση.

Η ακρίβεια (accuracy) ενός μοντέλου θεωρείται ένα μέτρο υπολογισμού της αποτελεσματικότητας του, έχοντας ως στόχο τον περιορισμό των ψευδών θετικά (false positive) κατηγοριοποιημένων στοιχείων. Σε ένα μοντέλο απαιτείται να υπάρχει υψηλή ακρίβεια διότι αυτό θα σημαίνει ότι δεν παράγονται πολλά ψεύτικα θετικά κατηγοριοποιημένα στοιχεία. Η ακρίβεια είναι ευρέως γνωστή και ως θετική πρόβλεψη τιμής (Positive Predictive Value – PPV).

Η ανάκληση (recall) από την άλλη πλευρά, υπολογίζει πόσα από τα θετικά στοιχεία της δειγματοληψίας έχουν εντοπιστεί από τις θετικές προβλέψεις. Γίνεται χρήση της ανάκλησης όταν πρέπει να εντοπιστούν όλα τα θετικά δείγματα, αυτό συμβαίνει όταν είναι σημαντικό να αποφύγουμε ψευδής αρνητικά στοιχεία.

$$\text{Ανάκληση} = \frac{TP}{TP + FP}$$

Εξίσωση 3: Τύπος ανάκλησης

Αν και η ακρίβεια και η ανάκληση είναι πολύ σημαντικά μέτρα, εστιάζοντας μόνο σε ένα από αυτά δεν παρέχεται η πλήρης εικόνα. Ένας τρόπος με τον οποίο μπορεί να γίνει η σύνοψη και των δύο είναι το f-score ή f-measure, το οποίο είναι το αρμονικό μέσο (harmonic mean) της ανάκλησης και της ακρίβειας.

$$F (f - score) = 2 \cdot \frac{\text{ακρίβεια} \cdot \text{ανάκληση}}{\text{ακρίβεια} + \text{ανάκληση}}$$

Εξίσωση 4: Τύπος f-score

2.4.3 Πίνακας Σύγχυσης

Ένας καλύτερος τρόπος για να γίνει η αξιολόγηση της απόδοσης ενός κατηγοριοποιητή είναι η δημιουργία πίνακα σύγχυσης (confusion matrix). Η γενική ιδέα είναι, η καταμέτρηση του αριθμού των περιπτώσεων όπου για παράδειγμα η κλάση A έχει κατηγοριοποιηθεί ως κλάση B. Αναλυτικότερα, ο πίνακας σύγχυσης δίνει πληροφορίες για τις αληθινές και προβλεπόμενες κατηγοριοποιήσεις που παρέχονται από ένα μοντέλο κατηγοριοποίησης.

Η αξιολόγηση της απόδοσης γίνεται με την χρήση των δεδομένων στον πίνακα, τα οποία συμβολίζονται με FP (False Positive), FN (False Negative), TP (True Positive) και TN (True Negative).

Η σημασία του κάθε συμβολισμού για την περίπτωση δυαδικής κλάσης είναι η εξής:

- FP: είναι το πλήθος των αρνητικών δειγμάτων των οποίων η πρόβλεψη έγινε λάθος ως θετικό από το μοντέλο.
- FN: είναι το πλήθος των θετικών δειγμάτων των οποίων η πρόβλεψη έγινε λάθος ως αρνητικό από το μοντέλο.
- TP: είναι το πλήθος των θετικών δειγμάτων των οποίων η πρόβλεψη έχει γίνει σωστά από το μοντέλο.
- TN: είναι το πλήθος των αρνητικών δειγμάτων των οποίων η πρόβλεψη έχει γίνει σωστά από το μοντέλο.

Η κάθε μέτρηση του πίνακα μπορεί να υπολογιστεί ως ποσοστό. Τα ποσοστά αυτά διατυπώνονται ως εξής:

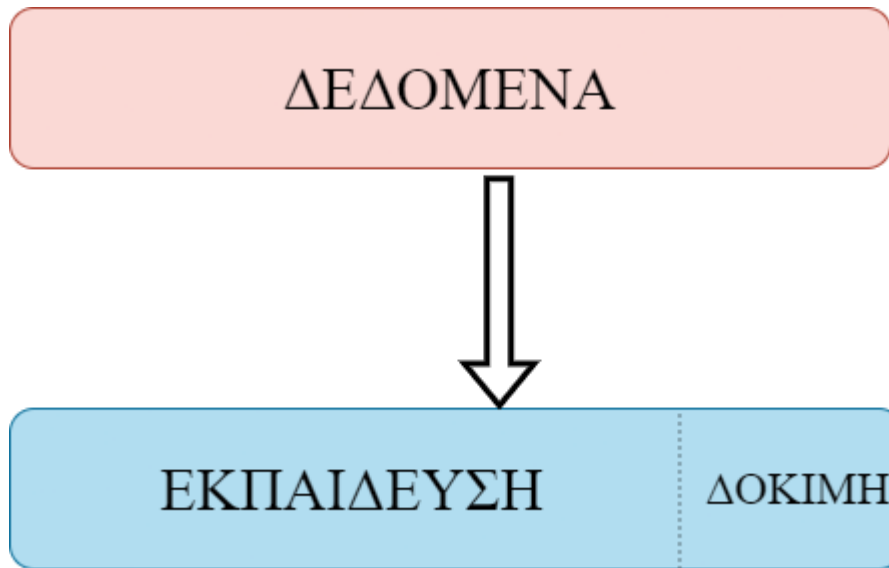
- FPR: $\frac{FP}{TN+FP}$ = το ποσοστό των δειγμάτων που είναι θετικά και η πρόβλεψη του έχει γίνει λάθος ως θετικά.
- FNR: $\frac{FN}{TP+FN}$ = το ποσοστό των δειγμάτων που είναι θετικά και η πρόβλεψη του έχει γίνει λάθος ως αρνητικά.
- TPR: $\frac{TP}{TP+FN}$ = το ποσοστό των δειγμάτων που είναι θετικά και η πρόβλεψη τους έχει γίνει σωστά
- TNR: $\frac{TN}{TN+FP}$ = το ποσοστό των δειγμάτων που είναι αρνητικά και η πρόβλεψη τους έχει γίνει σωστά

	Πρόβλεψη Αρνητικής Κλάσης	Πρόβλεψη Θετικής Κλάσης
Πραγματική Αρνητική Κλάση	TN	FP
Πραγματική Θετική Κλάση	FN	TP

Σχήμα 8: Πίνακας αποτίμησης

2.4.4 Διαχωρισμός

Η μέθοδος του διαχωρισμού (split) είναι μια από τις πιο δημοφιλείς προσεγγίσεις για τον χωρισμό ενός συνόλου δεδομένων σε ένα σύνολο εκπαίδευσης του μοντέλου και σε ένα σύνολο δοκιμής του μοντέλου. Επιπλέον, αυτήν η μέθοδος ονομάζεται και Hold-out. Η λειτουργία του καθορίζεται από το ποσοστό που θα ορίσει ο προγραμματιστής, για παράδειγμα αν δοθεί το ποσοστό 80% με την μέθοδο Split, αυτό θα σημαίνει ότι από το σύνολο δεδομένων, το τυχαίο ποσοστό του 80% των δεδομένων θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου ενώ το υπόλοιπο 20% θα χρησιμοποιηθεί ως σύνολο δοκιμής για την αξιολόγηση του μοντέλου.



Σχήμα 9: Παράδειγμα διαχωρισμού ενός συνόλου δεδομένων με την μέθοδο Split

ΚΕΦΑΛΑΙΟ 3 Internet of Things

Το διαδίκτυο των πραγμάτων αποτελείται από έναν κόσμο άπειρων διασυνδεδεμένων συσκευών, οι συσκευές αυτές έχουν την δυνατότητα να ανιχνεύουν, να ενεργοποιούν και να επικοινωνούν και μεταξύ τους και με το περιβάλλον (για παράδειγμα έξυπνα αντικείμενα - έξυπνα πράγματα) καθώς παρέχουν την δυνατότητα να μοιράζονται την πληροφορία και να επιδρούν αυτόνομα σε καταστάσεις του πραγματικού κόσμου ξεκινώντας διαδικασίες και δημιουργώντας υπηρεσίες με ή χωρίς την άμεση βοήθεια κάποιου συντονιστή.

Στα μέσα του 1926, ο γνωστός επιστήμονας Nikola Tesla οραματιζόταν έναν συνδεδεμένο κόσμο. Σε μια συνέντευξη του στο αμερικανικό περιοδικό Colliers (Kennedy,1926):

“Όταν η ασύρματη σύνδεση εφαρμοστεί τέλεια, ολόκληρη η Γη θα μετατραπεί σε έναν τεράστιο εγκέφαλο, που στην πραγματικότητα, όλα τα πράγματα αποτελούν σωματίδια ενός πραγματικού και ρυθμικού συνόλου [...] και τα εργαλεία μέσα από τα οποία θα μπορούμε να κάνουμε κάτι τέτοιο θα είναι απίστευτα απλά εάν συγκριθούν με τα σημερινά τηλέφωνα. Ο άνθρωπος θα μπορεί να τα κουβαλάει στην τσέπη του”

- Nikola Tesla

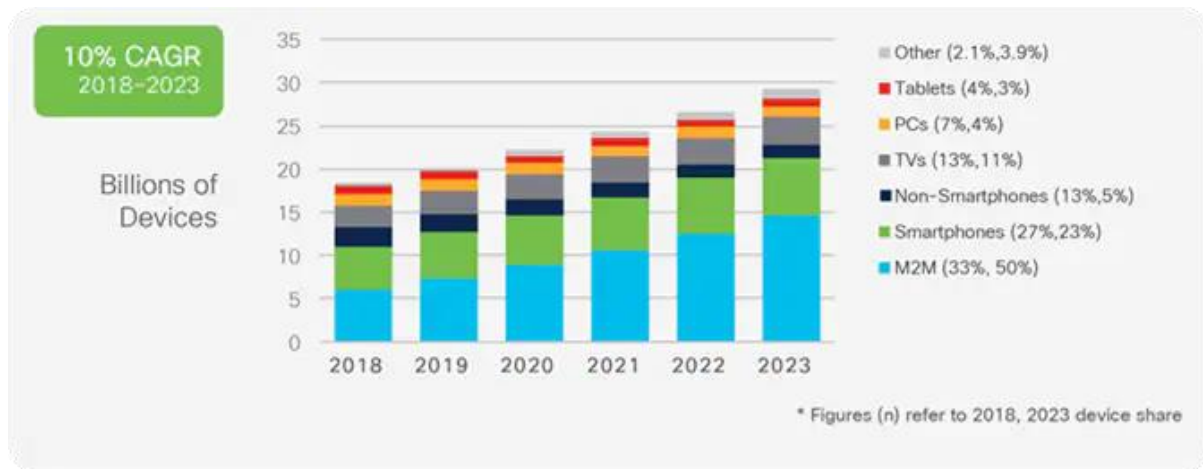
Ο πρώτος που χρησιμοποίησε τον όρο Internet of Things (IoT) ήταν ο Kevin Ashton το 1999 (Ashton ,2009), στο πλαίσιο εφοδιασμού της αλυσιδωτής διαχείρισης ραδιοσυχνότητων μέσω ταυτοποίησης (RFID) με ετικέτα ή αντικείμενα με γραμμωτό κώδικα (πράγματα) που προσφέρουν μεγαλύτερη αποτελεσματικότητα και υπευθυνότητα στις επιχειρήσεις. Όπως έγραψε ο Kevin Ashton στο RFID Journal (22 Ιουνίου 2009):

“ Αν υπήρχαν υπολογιστές οι οποίοι γνώριζαν τα πάντα για τα πράγματα – χρησιμοποιώντας δεδομένα τα οποία συγκεντρώθηκαν χωρίς καμία βοήθεια από εμάς – θα μπορούσαμε να παρακολουθούμε και να μετράμε τα πάντα, να μειώσουμε την σπατάλη, την απώλεια και το κόστος. Θα γνωρίζαμε το πότε τα αντικείμενα χρειάζονται αντικατάσταση, επιδιόρθωση ή ανάκληση, και αν ήταν φρέσκα ή είχαν περάσει τα καλύτερα τους.”

- Kevin Ashton

Ένα δίκτυο συνδέει πολλές συσκευές μεταξύ τους όπως υπολογιστές, εκτυπωτές και πολλές άλλες. Η σύνδεση αυτή γίνεται είτε με την χρήση καλωδίων είτε με την χρήση ραδιοκυμάτων ή υπέρυθρων σημάτων για την μεταφορά πληροφοριών από το ένα σημείο στο άλλο. Ένα δίκτυο δεν χρειάζεται να είναι μόνο για τους υπολογιστές, μπορεί επίσης να είναι του αναλογικού τηλεφωνικού συστήματος και η διασύνδεση αυτών των δικτύων είναι γνωστή ως διαδίκτυο. Ωστόσο, η τεχνολογική πρόοδος στο πρόσφατο παρελθόν έχει μετατρέψει το διαδίκτυο στο δίκτυο που συνδέονται τα πάντα, ακόμα και καθημερινά αντικείμενα μπορούν να αναγνωρίζονται και να ελέγχονται μέσω ετικετών RFID, ενσωματωμένων αισθητήρων και έξυπνων τηλεφώνων. Το διαδίκτυο των πραγμάτων προσφέρει συνδεσιμότητα των συσκευών, των συστημάτων και των υπηρεσιών που ξεπερνούν την σχέση Μηχανή προς

Μηχανή (Machine-to-Machine/M2M) επικοινωνεί και εξυπηρετεί πολλές διαφορετικές εφαρμογές σε διάφορους τομείς.



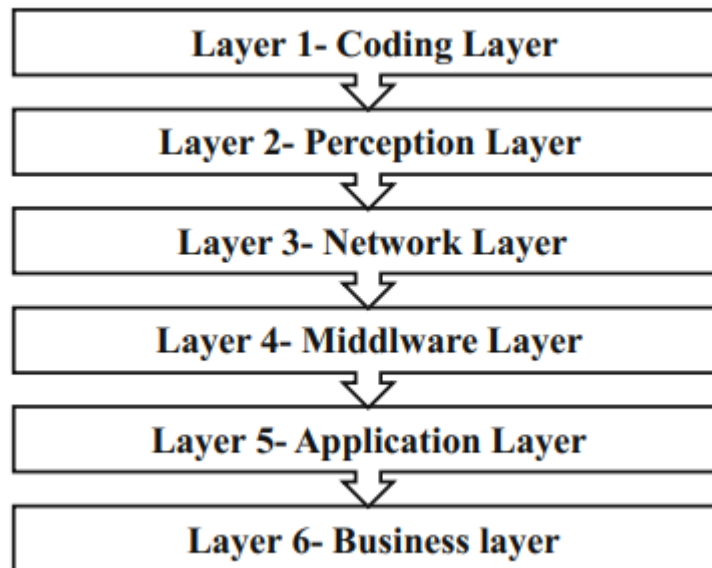
Σχήμα 10: Αύξηση των συνδεδεμένων συσκευών όπως υποδεικνύεται από την CISCO

3.1 Αρχιτεκτονική

Η αρχιτεκτονική που υπάρχει μέχρι και σήμερα για το διαδίκτυο εφαρμόστηκε πριν από περίπου τέσσερις δεκαετίες με την γνωστή μορφή των TCP/IP πρωτοκόλλων, όμως την σημερινή εποχή δεν είναι δυνατή η υποστήριξη και εξυπηρέτηση του τεράστιου δικτύου που διαθέτουν τα συστήματα IoT. Ως εκ τούτου, υπάρχει ακόμα η ανάγκη για την ανάπτυξη μιας νέας και σταθερής αρχιτεκτονικής η οποία θα είναι ικανή να διαχειριστεί στο δίκτυο της πάνω από 30 δισεκατομμύρια συνδεδεμένα αντικείμενα.

Η νέα αυτή αρχιτεκτονική θα πρέπει να χρησιμοποιεί πρωτόκολλο ανοιχτού κώδικα ώστε να μπορεί να υποστηρίξει εφαρμογές που προϋπάρχουν στο δίκτυο και να παρέχεται ασφάλεια και ποιότητα των υπηρεσιών (QoS – Quality of Service). Βασικές προκλήσεις για την εφαρμογή του διαδικτύου των πραγμάτων είναι προστασία των δεδομένων και η ιδιωτικότητα των δεδομένων. Συνεπώς, διάφορες αρχιτεκτονικές ασφάλειας πολλαπλών επιπέδων προτείνονται για την περαιτέρω βελτίωση του.

Ο Xu Cheng και οι συνεργάτες του πρότειναν μια αρχιτεκτονική έξι επιπέδων βασισμένο σε μια ιεραρχική δομή όπως παρουσιάζεται στο παρακάτω σχήμα 11



Σχήμα 11: Αρχιτεκτονική έξι επιπέδων προτεινόμενη από τον Xu Cheng

Τα έξι επίπεδα της αρχιτεκτονικής αναλύονται από το σχήμα 11 ως εξής:

- **Πρώτο επίπεδο – Coding Layer – Επίπεδο Κωδικοποίησης**

Το επίπεδο της κωδικοποίησης (Coding) αποτελεί το βασικό επίπεδο της IoT αρχιτεκτονικής, στην οποία τα αντικείμενα (πράγματα) διαθέτουν ταυτοποίηση. Κάθε αντικείμενο ξεχωριστά διαθέτει δικό του μοναδικό αναγνωριστικό, με αποτέλεσμα η διάκριση των αντικειμένων να γίνεται ευκολότερη.

- **Δεύτερο επίπεδο – Perception Layer – Επίπεδο Αντίληψης**

Στο επίπεδο της αντίληψης (Perception), θέτεται μια φυσική σημασία για το κάθε αντικείμενο. Οι αισθητήρες που παρέχουν τα δεδομένα περιορίζονται σε διαφορετικές μορφές όπως RFID ετικέτες, IR αισθητήρες και άλλα δίκτυα αισθητήρων με δυνατότητες όπως η ανίχνευση της υγρασίας, της θέσης, της θερμοκρασίας, της ταχύτητας κ.α. Το επίπεδο αυτό είναι υπεύθυνο για την συλλογή των σημαντικών πληροφοριών κάθε αντικειμένου από τις συσκευές των αισθητήρων που συνδέονται με αυτές, καθώς γίνεται μετατροπή των λαμβανόμενων πληροφοριών σε ψηφιακά σήματα, ώστε να μπορέσει να γίνει η μετάβαση στο επίπεδο δικτύου για περαιτέρω ενέργειες.

- **Τρίτο επίπεδο – Network Layer – Επίπεδο Δικτύου**

Αυτό το επίπεδο έχει ως στόχο την λήψη σημαντικών πληροφοριών με την μορφή ψηφιακών σημάτων από το δεύτερο επίπεδο (Perception Layer – Επίπεδο Αντίληψης), επιπλέον έχει ως στόχο την μετάδοση αυτών στο σύστημα επεξεργασίας που διαθέτετε στο τέταρτο επίπεδο, ονομαζόμενο Middleware (Ενδιάμεσο Λογισμικό), με την χρήση μέσων μετάδοσης όπως το

Bluetooth, Wi-Fi, WiMAX, Zigbee, 3G, 4G, 5G και άλλα, με τα πρωτόκολλα όπως IPv6, IPv4, DDS, MQTT και άλλα.

- **Τέταρτο Επίπεδο – Middleware Layer – Επίπεδο Ενδιάμεσου Λογισμικού**

Αυτό το επίπεδο είναι υπεύθυνο για την λήψη πληροφοριών από τις συσκευές αισθητήρων. Περιλαμβάνονται τεχνολογίες όπως η πανταχού παρούσα υπολογιστική (Ubiquitous Computing) η οποία εξασφαλίζει την άμεση πρόσβαση στην βάση δεδομένων της για την λήψη όλων των δεδομένων που χρειάζονται. Με την χρήση της Ευφυής Εξοπλισμένης Επεξεργασίας (IPE – Intelligent Processing Equipment), πραγματοποιείτε η λήψη των επεξεργασμένων δεδομένων καθώς και μία πλήρη αυτοματοποιημένη ενέργεια, η οποία βασίζεται στα δεδομένα της για τα επεξεργασμένα αποτελέσματα της.

- **Πέμπτο Επίπεδο – Application Layer – Επίπεδο Εφαρμογής**

Σε αυτό το επίπεδο παρέχεται μια εξατομικευμένη υπηρεσία με βάση τις ανάγκες του χρήστη, γίνεται η χρήση του αποτελέσματος των επεξεργασμένων δεδομένων. Διάφορες υψηλού επιπέδου ευφυής εφαρμογές υπάρχουν για όλων των ειδών τις βιομηχανίες. Οι σχετιζόμενες IoT εφαρμογές θα μπορούσαν να είναι για την παρακολούθηση φυσικών καταστροφών, παρακολούθηση υγείας, έξυπνα σπίτια, έξυπνες μεταφορές, έξυπνος πλανήτης και άλλα. Αυτές οι εφαρμογές ενθαρρύνουν την επέκταση του διαδικτύου των πραγμάτων, και αυτό το επίπεδο αποτελεί ύψιστη σημασία για την ανάπτυξη και την μετατροπή IoT σε μεγάλης κλίμακας δικτύου.

- **Έκτο Επίπεδο – Business Layer – Επίπεδο Επιχείρησης**

Το επίπεδο επιχείρησης είναι το ανώτερο επίπεδο της αρχιτεκτονικής στο διαδίκτυο των πραγμάτων (IoT), στο συγκεκριμένο επίπεδο παράγονται διάφορα μοντέλα επιχείρησης για τις αποτελεσματικές επιχειρησιακές στρατηγικές. Στο επίπεδο αυτό πραγματοποιείται η διαχείριση των εφαρμογών και των υπηρεσιών που παρέχονται από το IoT.

3.2 Τεχνολογίες

Με την ανάπτυξη ενός IoT δικτύου, δίνεται η δυνατότητα να μετατρέψει τα αντικείμενα σε μοναδικά αναγνωριζόμενα και τους δίνεται η δυνατότητα να συνδέονται και να επικοινωνούν με άλλα αντικείμενα οπουδήποτε και οποιαδήποτε στιγμή.

Η αδιάλειπτη επικοινωνία επιτυγχάνεται με τρία στοιχεία του IoT που είναι τα εξής:

- **Hardware:** Αποτελείται από Αισθητήρες (Sensors), Ενεργοποιητές (Actuators), Ενσωματωμένο υλικό επικοινωνίας (Embedded Communication Hardware).
- **Middleware:** Αποτελείται από εργαλεία αποθήκευσης (Storage tools) και Υπολογιστικά εργαλεία (Computing tools)
- **Παρουσίαση:** Αποτελείται από εργαλεία οπτικοποίησης και εργαλεία διερμηνείας.

Παρακάτω αναλύονται μερικές από τις τεχνολογίες που συνθέτουν τα παραπάνω εξαρτήματα. Αυτές οι τεχνολογίες συμβάλλουν στην υλοποίηση ολόκληρου του ΙοΤ οικοσυστήματος.

Radio Frequency Identification – RFID

Η αναγνώριση των ραδιοσυχνοτήτων (Radio Frequency Identification) είναι μια σημαντική τεχνολογία στον κλάδο της ενσωματωμένης επικοινωνίας (embedded communication), η οποία υποβοηθά στον σχεδιασμό μικροτσιπ πομποδέκτη (transceiver microchip) για την ασύρματη επικοινωνία. Η τεχνολογία RFID υποστηρίζει την αναγνώριση και την παρακολούθηση των αντικειμένων, το οποίο είναι εμφυτευμένο ως μια ετικέτα (tag) η οποία είναι μικροσκοπική και πολύ φθηνή, έχοντας ως αποτέλεσμα να καθίσταται εύκολη η ενσωμάτωσή τους σε οποιοδήποτε αντικείμενο. Υπάρχουν δύο κύριοι τύποι ετικετών, διακρίνονται σε ενεργές ετικέτες και παθητικές ετικέτες. Ο τύπος των ενεργών ετικετών διαθέτουν μια δική τους μπαταρία η οποία τις τροφοδοτεί ώστε να είναι πάντα ενεργές και να εκπέμπονται τα σήματα δεδομένων χωρίς περιορισμούς, ενώ οι παθητικές ετικέτες δεν τροφοδοτούνται με κάποια μπαταρία και πρέπει να ενεργοποιηθούν ώστε να αντληθούν τα δεδομένα της. Παρ' όλα αυτά, και οι δύο ετικέτες έχουν μεγάλο εύρος εφαρμογής και μπορούν να ενσωματωθούν σε ζωντανά ή και μη αντικείμενα. Ένα σύστημα RFID αποτελείται από ετικέτες που περιέχουν τα στοιχεία του αντικείμενου που είναι ενσωματωμένο και τα στοιχεία από αυτών που τα διαβάζουν (Readers). Οι ετικέτες στέλνουν τα σήματα δεδομένων που περιέχουν τις συλλεγόμενες πληροφορίες όπως η ταυτοποίηση, η θέση και άλλες σχετικές με το αντικείμενο και στην συνέχεια πραγματοποιείται η μετάδοση αυτών με την χρήση ραδιοσυχνοτήτων στους αναγνώστες, οι οποίες έπειτα κατανέμονται στους επεξεργαστές για την ανάλυση τους.

Wireless Sensor Networks – WSN

Τα ασύρματα δίκτυα αισθητήρων είναι μια πρωτοποριακή τεχνολογία για τις εφαρμογές, είναι σχεδιασμένες να λειτουργούν εξ αποστάσεως. Η συγκεκριμένη τεχνολογία χρησιμοποιεί ενεργειακά αποδοτικές, αξιόπιστες, χαμηλού κόστους και μικρού μεγέθους συσκευές οι οποίες βασίζονται σε ολοκληρωμένα κυκλώματα και υποστηρίζουν ασύρματες επικοινωνίες. Συνεπώς, το WSN είναι ένα δίκτυο αισθητήρων που αποτελείται από πολλούς έξυπνους αισθητήρες, οι οποίοι συλλέγουν, επεξεργάζονται, αναλύουν και διαδίδουν πολύτιμα δεδομένα που συλλέγονται μέσα από ένα δίκτυο. Κάθε αισθητήρας ξεχωριστά αποτελεί μια μονάδα επικοινωνίας, ενεργοποίησης και ανίχνευσης, δηλαδή είναι κατά βάση ένας πομποδέκτης (transceiver) που διαθέτει μικροελεγκτή, κεραία, ένα κύκλωμα που δρα ως μια διεπαφή και μια πηγή ισχύος που μπορεί να είναι είτε μπαταρία είτε

κάποια συσκευή συλλογής ενέργειας. Το ολοκληρωμένο δίκτυο αναμένεται να έχει κατάλληλη τοπολογία, επίπεδο MAC και δρομολόγηση ώστε να εξασφαλίζεται η επεκτασιμότητα και η σταθερότητα του δικτύου, και να υπάρχει η δυνατότητα σύνδεσης με τον σταθμό για την μετάδοση δεδομένων είτε με ένα είτε με πολλά άλματα (hop). Το ενδιαμέσο λογισμικό του WSN επιτρέπει την πρόσβαση στους πόρους των αισθητήρων που είναι ετερογενείς, ενσωματώνοντας την υποδομή του κυβερνοχώρου με δίκτυα αισθητήρων και αρχιτεκτονική SOA (Service Oriented Architecture). Αξίζει να σημειωθεί ότι το WSN προβλέπεται να παρέχει ασφαλή και αποτελεσματική μέθοδο συγκέντρωσης δεδομένων, διασφαλίζοντας έτσι την αξιοπιστία των δεδομένων που συλλέχθηκαν από τους αισθητήρες και ενισχύεται η διάρκεια ζωής του δικτύου.

Cloud Computing

Το Cloud Computing πρόκειται για μια ευφυής τεχνολογία η οποία περιλαμβάνει την σύγκλιση πολλών διακομιστών (Servers) σε μια πλατφόρμα τύπου cloud έχοντας ως στόχο τον διαμοιρασμό των πόρων και την πρόσβαση σε αυτούς από οπουδήποτε και οποιαδήποτε στιγμή. Ο τομέας του cloud αποτελεί αναπόσπαστο μέρος του IoT, αφού αποθηκεύει τις πληροφορίες που έχουν συγκεντρωθεί από τις διάφορες συσκευές IoT, πραγματοποιείται επεξεργασία πάνω σε αυτές και παρουσιάζεται μια ανάλυση για κάποιο μελλοντικό σκοπό.

Αποθήκευση και ανάλυση δεδομένων

Τα συστήματα IoT παράγουν άφθονα και διαφορετικού τύπου δεδομένα, κρίσιμος παράγοντας που επηρεάζει την ανάπτυξη αυτών των δεδομένων είναι η αποθήκευση τους, η κατοχή αυτών και η λήξη τους. Από την συνολική ενέργεια που παράγεται, το 5% καταναλώνεται από το διαδίκτυο, αυτό σημαίνει πως τα κέντρα δεδομένα θα πρέπει να είναι αποδοτικά στην ενέργεια και αξιόπιστα. Επομένως, βασική ανάγκη είναι η αποθήκευση των δεδομένων και η ανάλυση τους. Θα πρέπει να σχεδιαστούν αλγόριθμοι που κάνουν έξυπνη χρήση των δεδομένων και θα προσαρμόζεται ανάλογα στις ανάγκες ως κεντρικός ή κατανεμημένος αλγόριθμος. Για την αποτελεσματική λήψη των αποφάσεων απαιτείται ένα νέο είδος αλγόριθμων όπως οι εξελικτικοί, γενετικοί αλγόριθμοι και τα νευρωνικά δίκτυα. Τα συστήματα που δημιουργούνται θα πρέπει να έχουν χαρακτηριστικά όπως διαλειτουργικές, ενσωματωμένες και προσαρμοστικές επικοινωνίες. Το σύστημα βασίζεται σε αρθρωτή (modular) αρχιτεκτονική για την ανάπτυξη υλικού και λογισμικού.

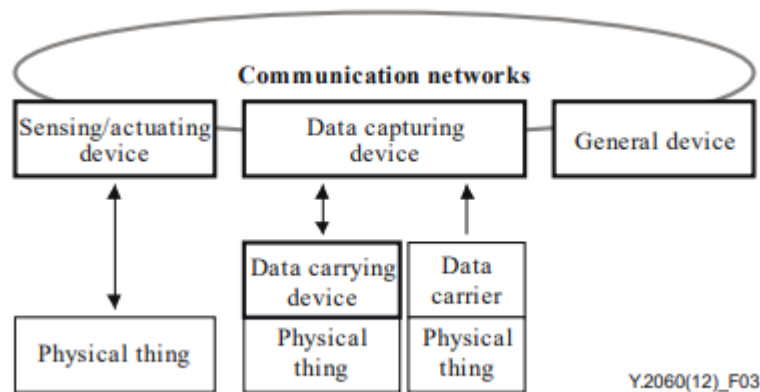
Οπτικοποίηση

Η οπτικοποίηση είναι μια από τους πιο σημαντικούς τομείς κατά την ενασχόληση με το IoT, καθώς δίνει την δυνατότητα στον χρήστη να αλληλεπιδρά με το εικονικό περιβάλλον. Για την οπτικοποίηση της υλοποίησης του IoT θα πρέπει να αναπτυχθεί από την μεριά του απλού ανθρώπου (layman perspective), δηλαδή θα πρέπει να είναι εύκολη, απλή και φιλική προς τον χρήστη. Η υλοποίηση αυτήν μπορεί εύκολα να επιτευχθεί μέσω των εξελίξεων όπως την οθόνη αφής και τις τεχνολογίες αναγνώρισης ομιλίας. Η μετατροπή των παραγόμενων δεδομένων από πληροφορία σε γνώση θα έχει ως αποτέλεσμα την ταχύτερη διαδικασία λήψης

αποφάσεων. Η αναπαράσταση μπορεί να αναπροσαρμοστεί ανάλογα με τις απαιτήσεις του τελικού χρήστη και αντίστοιχα να οπτικοποιηθούν.

3.3 Συσκευές

Μια συσκευή αποτελεί ένα κομμάτι εξοπλισμού με υποχρεωτικές δυνατότητες επικοινωνίας και προαιρετικές δυνατότητες που αφορούν την ανίχνευση, την ενεργοποίηση, την συλλογή, την αποθήκευση και την επεξεργασία των δεδομένων. Οι συσκευές συλλέγουν ποικίλα είδη πληροφοριών και τα παρέχουν στα δίκτυα πληροφοριών και επικοινωνιών για την περαιτέρω επεξεργασία τους. Κάποιες συσκευές εκτελούν επίσης λειτουργίες ανάλογα με τις πληροφορίες που λαμβάνουν από τα δίκτυα παροχής πληροφοριών και επικοινωνιών.



Σχήμα 12: Τα είδη και οι σχέσεις των συσκευών

Στο παραπάνω σχήμα γίνεται αναπαράσταση των διάφορων τύπων συσκευών (devices) και οι σχέσεις αυτών των συσκευών με τα φυσικά αντικείμενα (Physical things)

Οι συσκευές κατηγοριοποιούνται σε συσκευές που μεταφέρουν τα δεδομένα, σε συσκευές που πραγματοποιούν την συλλογή των δεδομένων, σε συσκευές που υποστηρίζουν υπηρεσίες ανίχνευσης και ενεργοποίησης και σε γενικές συσκευές οι οποίες αναλύονται παρακάτω:

- **Data – carrying devices (Συσκευές μεταφοράς δεδομένων):**

Η συσκευή μεταφοράς δεδομένων είναι μια συσκευή η οποία είναι συνδεδεμένη σε ένα φυσικό αντικείμενο – πράγμα (physical thing) με στόχο την έμμεση σύνδεση του φυσικού αντικειμένου με το δίκτυο επικοινωνίας.

- **Data – capturing devices (Συσκευές καταγραφής δεδομένων):**

Η συσκευή καταγραφής δεδομένων μπορεί να αναφερθεί ως μια συσκευή ανάγνωσης και εγγραφής δεδομένων, έχοντας επίσης την δυνατότητα να αλληλοεπιδράσει με τις υπόλοιπες συσκευές μεταφοράς δεδομένων.

- **Sensing and actuating devices (Συσκευές ανίχνευσης και ενεργοποίησης):**

Οι συσκευές ανίχνευσης ενεργούν ως μετρητές πληροφοριών σχετικά με το περιβάλλον που είναι τοποθετημένες και έχουν την δυνατότητα να μετατρέπουν την πληροφορία σε ψηφιακό σήμα. Οι συσκευές ενεργοποίησης μπορούν και μεταφράζουν τα σήματα που λαμβάνουν από το δίκτυο, μετατρέποντας τα σήματα σε πραγματικές λειτουργίες.

- **General devices (Γενικές συσκευές):**

Οι γενικές συσκευές πληρούν τις δυνατότητες της επεξεργασίας και της επικοινωνίας, μπορεί να επικοινωνεί με τα δίκτυα με την βοήθεια ενσύρματων και ασύρματων τεχνολογιών. Περιλαμβάνουν εξοπλισμό και εφαρμόζονται σε διάφορα IoT συστήματα, όπως βιομηχανικός εξοπλισμός, οικιακές συσκευές, συσκευές υγείας και έξυπνα τηλέφωνα.

Αφού γνωστοποιήθηκαν οι κατηγορίες των συσκευών (devices) ακολουθούν μερικά παραδείγματα από συσκευές που χρησιμοποιούνται για την ανίχνευση και την παραγωγή δεδομένων χρήσιμων για τα συστήματα IoT.

- **Αισθητήρες (Sensors):**

- Ανίχνευση φυσικών ιδιοτήτων:
 - ✓ Αισθητήρες θερμοκρασίας
 - ✓ Αισθητήρες πίεσης
 - ✓ Αισθητήρες κίνησης
 - ✓ Αισθητήρες στάθμης
 - ✓ Αισθητήρες εικόνας
 - ✓ Αισθητήρες εγγύτητας
 - ✓ Αισθητήρες ποιότητας νερού
 - ✓ Χημικοί αισθητήρες
 - ✓ Αισθητήρες αερίων
 - ✓ Αισθητήρες καπνού
 - ✓ Αισθητήρες υπερύθρων (IR)
 - ✓ Αισθητήρες επιτάχυνσης
 - ✓ Γυροσκοπικοί αισθητήρες
 - ✓ Αισθητήρες υγρασίας
 - ✓ Οπτικοί αισθητήρες
- Ανίχνευση ανθρώπων αισθήσεων:
 - ✓ Θερμίστορ (Thermistor)

- ✓ Αισθητήρες ηχητικής πίεσης
- ✓ Αισθητήρες οσμών
- ✓ Αισθητήρες απεικόνισης

Wearable devices:

Οι wearable συσκευές είναι αυτές που μπορούν να "φορεθούν" από τον άνθρωπο αλλά και τα ζώα. Διακρίνονται σε μικρές συσκευές και τοποθετούνται στα μέρη του σώματος με σκοπό την καταγραφή ζητούμενων πληροφοριών. Οι wearable συσκευές διαδραματίζουν ιδιαίτερα σημαντικό ρόλο στον κλάδο της υγειονομικής περίθαλψης, των κατασκευών και των logistics/μεταφορών.

Μερικά παραδείγματα wearables είναι τα εξής:

- ✓ Παρακολουθητές γλυκόζης (Glucose trackers)
- ✓ Παρακολουθητής άσθματος (Asthma trackers)
- ✓ Έξυπνοι φακοί επαφής (Smart contact lenses)
- ✓ Ακουστικά βαρηκοΐας (Hearing aid)
- ✓ Παρακολουθητής ύπνου (Sleep trackers)
- ✓ Έξυπνα κράνη (Smart helmet)

3.4 Εφαρμογές ΙοΤ

Υπάρχουν δυνατότητες εξέλιξης στο δίκτυο του ΙοΤ, αυτό φαίνεται στην καθημερινή ανάπτυξη των πολυάριθμων εφαρμογών. Πολλές εφαρμογές μπορούν να αναπτυχθούν και να χρησιμοποιηθούν με κύρια βάση το ΙοΤ. Σε αυτό το κεφάλαιο θα αναφερθούν μερικές από τις εφαρμογές που βασίζονται στην τεχνολογία του ΙοΤ σε μερικούς τομείς για την καλύτερη κατανόηση του τρόπου με τον οποίο λειτουργεί αυτή η τεχνολογία και κάποιες που δεν υπάρχουν αλλά θα μπορούσαν να αναπτυχθούν.

3.4.1 Σύστημα Υγείας

Το Διαδίκτυο των πραγμάτων στον τομέα της υγείας αναφέρεται ως Internet of Medical Things (IoMT), το οποίο έχει παρουσιάσει μεγάλη ανάπτυξη στην εφαρμογή τους σε συστήματα αλλά και στην ανάπτυξη τους. Η ανάπτυξη αυτήν οφείλεται στην πρόοδο των τεχνολογιών ανίχνευσης, όπως αυτές που λαμβάνουν φυσικά δεδομένα όπως του σφυγμού, του αναπνευστικού ρυθμού της αρτηριακής πίεσης και άλλων.

Οι εφαρμογές που προσφέρουν οι ΙοΤ συσκευές είναι οι εξής:

- **Έξυπνη παρακολούθηση ασθενών μέσω συσκευών ΙοΤ**

Πρόκειται για την φροντίδα των ασθενών που νοσηλεύονται στο νοσοκομείο χωρίς εμπλοκή νοσηλευτικού προσωπικού. Μια εφαρμογή τέτοιου τύπου είναι η SMI, η οποία παρέχει ειδικούς αισθητήρες για την πρόληψη των πληγών που οφείλονται στην ακινησία. Οι αισθητήρες είναι σε θέση να μετράνε και να αλλάζουν την πίεση του στρώματος για την αποφυγή επιπλοκών όπως είναι τα έλκη.

- **Διάγνωση εξοπλισμού**

Ένα μεγάλο ποσοστό των δαπανών υγειονομικής περίθαλψης προέρχεται από την κακή διαχείριση του εξοπλισμού. Παράδειγμα αποτελούν οι απρόβλεπτες βλάβες των συσκευών όπου οδηγούν σε απρόβλεπτες δαπάνες. Μια εφαρμογή που διορθώνει αυτό το πρόβλημα είναι το e-ALERT από την Philips, η οποία ελέγχει την κατάσταση των συσκευών και ενημερώνει το προσωπικό όταν προβλέπει κάποια μελλοντική βλάβη.

- **Παρακολούθηση της προσωπικής υγείας**

Υπάρχουν όλο και περισσότερες καινοτόμες συσκευές και εφαρμογές που επικεντρώνονται στον κάθε χρήστη και τον βοηθούν να παραμείνει υγιής. Πέρα από την παρακολούθηση την σωματικής υγείας, οι υπηρεσίες IoMT μπορούν να είναι χρήσιμες για όσους πάσχουν από κατάθλιψη και άλλες παρόμοιες καταστάσεις. Η αρχή της λειτουργίας τους βασίζεται στην ανάλυση των απαντήσεων των χρηστών σε κρίσιμες ερωτήσεις και στην παρακολούθηση ορισμένων δεικτών της κατάστασής τους, είτε πρόκειται για σφυγμό, είτε για πίεση, είτε για κίνηση της κόρης του ματιού και ούτω καθεξής. Εάν υπάρχει λόγος ανησυχίας, το σύστημα στέλνει μια ειδοποίηση.

- **Τηλεϊατρική**

Όπου συνδυάζει όλα τα παραπάνω με βασικό γνώρισμα ότι αντί για τους εργαζόμενους στον τομέα της υγείας, οι βασικές ενέργειες εκτελούνται από διάφορες έξυπνες συσκευές σε συνδυασμό με εφαρμογές IoT για την υγειονομική περίθαλψη. Η τηλεϊατρική καθιστά δυνατή τη συλλογή και επεξεργασία δεδομένων για κάθε ασθενή, διαδικτυακές διαβουλεύσεις για επείγοντα ζητήματα, ανταλλαγή επαγγελματικών εμπειριών, δημιουργία χάρτη με την δραστηριότητα του ασθενούς, παροχή επείγουσας φροντίδας σε κρίσιμες περιπτώσεις.

3.4.2 Εκπαίδευση

Ο τομέας της εκπαίδευσης είναι ένας από τους πιο προσαρμοστικούς και αποτελεσματικούς όσον αφορά την ανάπτυξη συσκευών IoT στη χρήση του, προκειμένου να καταστεί η εκπαίδευση πιο συνεργατική, διαδραστική και προσβάσιμη σε όλους. Οι συσκευές IoT παρέχουν στους μαθητές αξιόπιστη πρόσβαση σε όλα, από το μαθησιακό υλικό μέχρι τα κανάλια επικοινωνίας για την καλή κατανόηση, και δίνουν στους εκπαιδευτικούς τη δυνατότητα να μετρούν την πρόοδο της μάθησης των μαθητών σε πραγματικό χρόνο.

Μερικές από τις εφαρμογές που προσφέρουν οι IoT συσκευές είναι οι εξής:

- **Έξυπνες Αίθουσες**

Οι έξυπνοι πίνακες έκαναν τη μάθηση διαδραστική με την προβολή εικόνων. Εκτός από το να γράφουν στον πίνακα, μπορούν να μετακινούν ψηφιακά αντικείμενα σε αυτόν. Οι λέξεις και οι αριθμοί στα σχολικά βιβλία και στους πίνακες δεν εξηγούν τις έννοιες τόσο εύκολα όσο οι έξυπνοι πίνακες. Κατά συνέπεια, η εκπαίδευση στο IoT γίνεται επίσης μέρος των δεξιοτήτων των εκπαιδευτικών.

Το AR θα επιτρέψει την ανατομία χωρίς την ανάγκη να υπάρχουν ζωντανά ζώα. Επιπλέον, τα ηλεκτρονικά βιβλία με δυνατότητα λήψης μπορούν να ελέγχουν και να παρακολουθούν τη διδακτέα ύλη. Περαιτέρω, οι μαθητές μπορούν να αφιερώσουν επιπλέον χρόνο στην ανάπτυξη ιδεών και την εκτέλεση πρακτικών ασκήσεων λόγω των λειτουργιών φωνητικής εντολής και των gadgets για την καταγραφή σημειώσεων από ομιλία σε κείμενο.

- **Ασφάλεια στις εγκαταστάσεις**

Τα περισσότερα σχολεία και ιδρύματα στερούνται οποιασδήποτε τεχνολογίας για την ανίχνευση κακοποιήσεων, επιθέσεων, κλοπών ή άλλων εγκλημάτων. Δεν διαθέτουν επίσης σχέδιο περιορισμού σε περίπτωση έκτακτης ανάγκης. Το IoT μπορεί να τα βοηθήσει. Ένα παράδειγμα μπορεί να είναι οι έξυπνες κάμερες ώστε να παρακολουθούν τυχόν μη αποδεκτές δραστηριότητες και να αρχίσουν να τις καταγράφουν. Αυτό είναι εφικτό χάρη στο IoT και το σύστημα δικτύου του που συνδέει κάθε υπολογιστή στις εγκαταστάσεις. Επιπλέον, εάν κάποιος προσπαθήσει να παραβιάσει το σχολείο, οι έξυπνες πόρτες θα μπορούν να κλειδώσουν αυτόματα και να ενεργοποιήσουν τους αισθητήρες για να καλέσουν βοήθεια. Επίσης, σε περίπτωση εκδήλωσης πυρκαγιάς, οι ανιχνευτές που βασίζονται στο IoT μπορούν να

ενεργοποιήσουν τους συναγερμούς και να αρχίσουν να επιλύουν το ζήτημα πριν οι άνθρωποι φτάσουν στη ζώνη πυρκαγιάς.

- **Παρακολούθηση των μαθητών**

Είναι δυνατή η παρακολούθηση των δραστηριοτήτων των φοιτητών μέσω της πύλης του ινστιτούτου. Οι αισθητήρες IoT μπορούν να συλλέγουν δεδομένα και να προσφέρουν ένα εξατομικευμένο σχέδιο μάθησης. Επιπλέον, μπορούν επίσης να προτείνουν τα θέματα ή τους πόρους που μπορεί να βοηθήσουν τους μαθητές. Το IoT θα ωφελήσει επίσης τους καθηγητές παρακολουθώντας αν οι μαθητές τους έχουν ολοκληρώσει τις αξιολογήσεις. Επίσης, το IoT μπορεί να παρακολουθεί την πρόοδο των μαθητών και να ειδοποιεί τις αρμόδιες αρχές εάν αντιμετωπίζουν οποιοδήποτε πρόβλημα σε ένα συγκεκριμένο θέμα. Για την αποτροπή της κατάχρησης των εγκαταστάσεων διαδικτύου, τα σχολεία μπορούν να χρησιμοποιούν συστήματα Wi-Fi που επιτρέπουν τη χρήση του δικτύου μόνο για συγκεκριμένους σκοπούς

3.4.3 Έξυπνες Πόλεις

Όταν οι πόλεις αντιμετωπίζουν ζητήματα που προκαλούνται από την πυκνότητα του πληθυσμού, μπορούν να προκύψουν πολλά προβλήματα, όπως η ατμοσφαιρική ρύπανση, η έλλειψη γλυκού νερού, τα βουνά σκουπιδιών και η αύξηση της κυκλοφορίας.

Οι έξυπνες πόλεις μπορούν να αξιοποιήσουν το IoT και τις έξυπνες τεχνολογίες με τους ακόλουθους τρόπους:

- **Έξυπνη υποδομή**

Οι ψηφιακές τεχνολογίες αποκτούν ολοένα και μεγαλύτερη σημασία για να έχουν οι πόλεις τις προϋποθέσεις για συνεχή ανάπτυξη, τα κτίρια και οι αστικές υποδομές πρέπει να σχεδιάζονται πιο αποτελεσματικά και βιώσιμα. Οι πόλεις θα πρέπει επίσης να επενδύσουν σε ηλεκτρικά αυτοκίνητα και αυτοκινούμενα οχήματα για να διατηρήσουν τις εκπομπές CO₂ σε χαμηλά επίπεδα. Για παράδειγμα, για να μειωθεί η ανάγκη για ηλεκτρική ενέργεια, ο έξυπνος φωτισμός δίνει φως μόνο όταν κάποιος πραγματικά περνάει από τα έξυπνα φώτα, η ρύθμιση των επιπέδων φωτεινότητας και η παρακολούθηση της καθημερινής χρήσης είναι και τα δύο σημαντικά στοιχεία των έξυπνων φώτων.

- **Διαχείριση της κυκλοφορίας**

Μία από τις μεγαλύτερες προκλήσεις που αντιμετωπίζουν οι μεγάλες πόλεις είναι η εξερεύνηση τρόπων βελτιστοποίησης της κυκλοφορίας. Για παράδειγμα, το Λος Άντζελες είναι μια από τις πιο πολυσύχναστες πόλεις στον κόσμο και έχει εφαρμόσει μια έξυπνη λύση μεταφορών για τον έλεγχο

της κυκλοφοριακής ροής. Ενσωματωμένοι στο οδόστρωμα αισθητήρες στέλνουν σε πραγματικό χρόνο ενημερώσεις για τη ροή της κυκλοφορίας σε μια κεντρική πλατφόρμα διαχείρισης της κυκλοφορίας, η οποία αναλύει τα δεδομένα και προσαρμόζει αυτόματα τα φανάρια στην κατάσταση της κυκλοφορίας μέσα σε δευτερόλεπτα. Ταυτόχρονα, χρησιμοποιούνται ιστορικά δεδομένα για να προβλεφθεί πού μπορεί να κατευθυνθεί η κυκλοφορία - και καμία από αυτές τις διαδικασίες δεν απαιτεί ανθρώπινη συμμετοχή.

- **Έξυπνη διαχείριση αποβλήτων**

Οι εφαρμογές διαχείρισης αποβλήτων συμβάλλουν στη βελτίωση της αποτελεσματικότητας της συλλογής αποβλήτων και στη μείωση του λειτουργικού κόστους, ενώ παράλληλα αντιμετωπίζουν καλύτερα όλα τα περιβαλλοντικά ζητήματα που σχετίζονται με την μη αποτελεσματική συλλογή αποβλήτων. Σε αυτές τις λύσεις, το δοχείο απορριμμάτων λαμβάνει έναν αισθητήρα στάθμης, όταν επιτευχθεί ένα συγκεκριμένο όριο, η πλατφόρμα διαχείρισης ενός οδηγού φορτηγού λαμβάνει ειδοποίηση μέσω του smartphone του. Το μήνυμά τους βοηθά να αποφύγουν τους μισοάδειους αποχετευτικούς αγωγούς, εμφανίζοντας να αδειάζουν ένα γεμάτο δοχείο

ΚΕΦΑΛΑΙΟ 4 Προτεινόμενο σύστημα

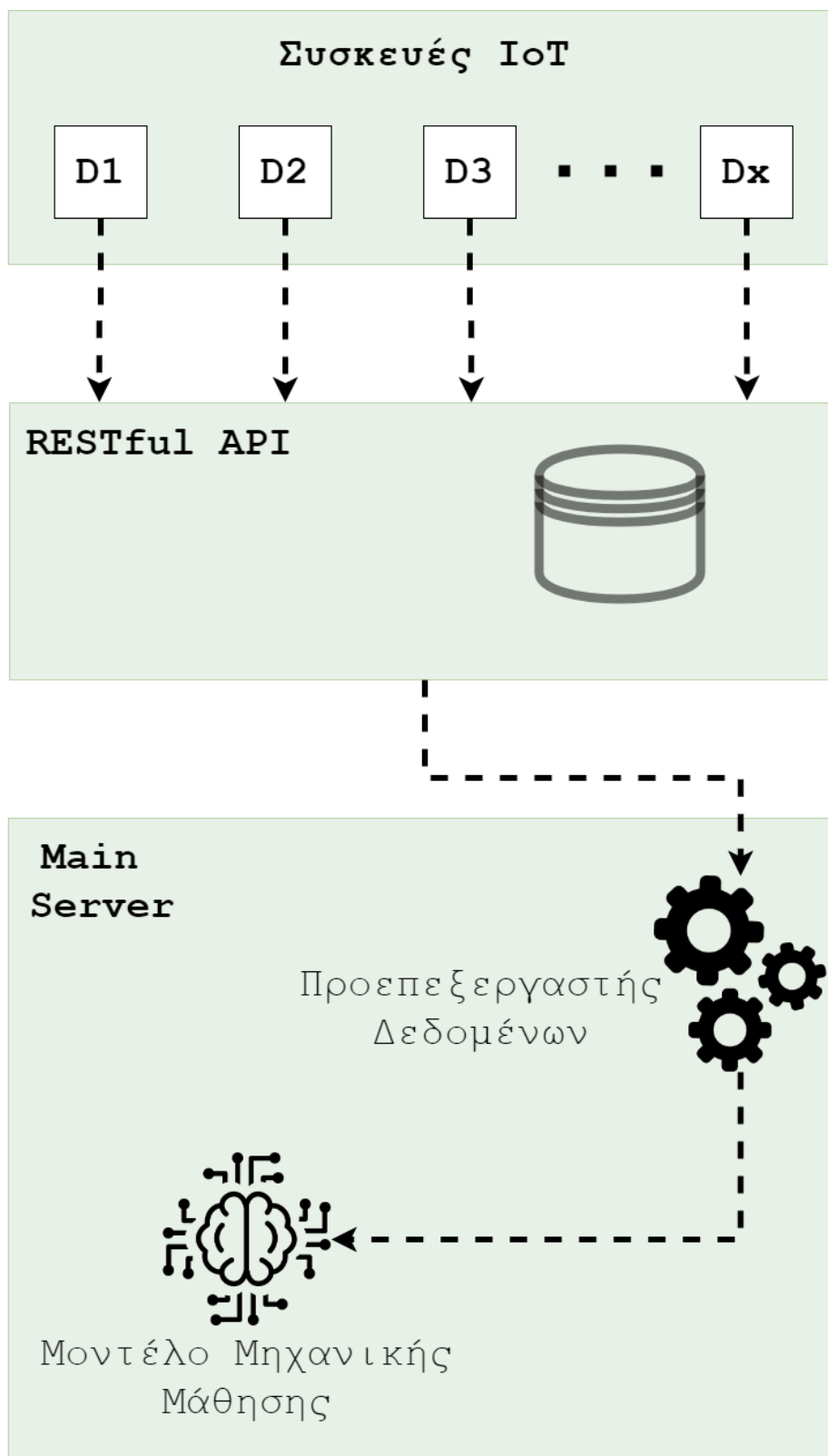
Ένα σύστημα το οποίο χρησιμοποιεί συσκευές IoT που παράγουν δεδομένα μεγάλου μεγέθους, θα πρέπει να είναι σε θέση να αναλάβει την διαχείριση τους και την κατάλληλη προ επεξεργασία τους με αποτέλεσμα τα δεδομένα αυτά να χρησιμοποιηθούν για την παραγωγή πρόσθετης πληροφορίας η οποία θα μπορεί να αποφέρει λύσεις σε άλλα προβλήματα.

Το σύστημα που υλοποιήθηκε στα πλαίσια της πτυχιακής εργασίας αποτελεί μια προσομοίωση Internet of Things συσκευών, των οποίων τα δεδομένα συλλέγονται, προ επεξεργάζονται σε κατάλληλη μορφή και έπειτα προωθούνται σε μοντέλα μηχανικής μάθησης με στόχο την πρόβλεψη κατηγοριοποίησης.

Στο συγκεκριμένο κεφάλαιο ακολουθεί αναλυτική περιγραφή για την υλοποίηση και την ανάπτυξη του συνολικού συστήματος της πτυχιακής εργασίας. Αρχικά το σύστημα διαχωρίστηκε σε 3 διαφορετικές και ανεξάρτητες μονάδες. Η πρώτη μονάδα αναφέρετε στην δημιουργία και την προσομοίωση των συσκευών, η δεύτερη μονάδα αποτελεί τον μεσάζοντα χώρο αποθήκευσης του συστήματος και τον μεσολαβητή επικοινωνίας των υπόλοιπων υπομονάδων και τέλος η τρίτη υπομονάδα αναλαμβάνει την συλλογή των δεδομένων, την τελική εκπαίδευση του μοντέλου και την εξαγωγή προβλέψεων.

Στις παρακάτω υποενότητες της κάθε μονάδας αναλύεται και περιγράφεται η αρχιτεκτονική της κάθε μιας ξεχωριστά, περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της και οι λειτουργίες που εκτελούν.

4.1 Αρχιτεκτονική Γενικού Συστήματος



Σχήμα 13: Γενική Αρχιτεκτονική Συστήματος

Στην αρχιτεκτονική όπως φαίνεται και στην εικόνα 13 το σύστημα χωρίζεται σε 3 επίπεδα. Αρχικά στο πρώτο επίπεδο αναπαρίστανται οι καθορισμένες IoT συσκευές οι οποίες είναι ανεξάρτητες με όλα τα άλλα επίπεδα του συστήματος και έχουν μεταβλητό αριθμό. Οι IoT συσκευές του πρώτου επιπέδου επικοινωνούν με το δεύτερο επίπεδο, ώστε να στείλουν τις καταχωρήσεις που έχουν στην μνήμη τους.

Το δεύτερο επίπεδο με την σειρά του πρόκειται για ένα RESTful API (Representational state transfer – application program interface) το οποίο λειτουργεί σαν μεσολαβητής επικοινωνίας και κοινής χρήσης δεδομένων για το πρώτο και το τρίτο επίπεδο, αποθηκεύοντας τα δεδομένα που λαμβάνει από τις συσκευές IoT στην βάση δεδομένων του.

Τέλος, το τελευταίο επίπεδο είναι υπεύθυνο για την συλλογή των δεδομένων με την βοήθεια του δεύτερου επιπέδου, τα ληφθέντα αυτά δεδομένα περνάνε μέσα από έναν μηχανισμό όπου προ επεξεργάζονται. Στην συνέχεια τα δεδομένα τροφοδοτούνται στο μοντέλο μηχανικής μάθησης για να ακολουθήσει η εκπαίδευση του και η πρόβλεψη της τελικής διάγνωσης.

Η σειρά με την οποία η κάθε μονάδα ξεκινάει να εκτελεί τις λειτουργίες τις είναι με αυτήν που φαίνεται στο σχήμα, ξεκινώντας από το πρώτο και καταλήγοντας στο τελευταίο.

4.2 Δεδομένα

Για την ανάπτυξη του συστήματος το οποίο θα είναι σε θέση να συνεργάζεται με τις συσκευές IoT, για να εκπαιδευτεί το μοντέλο μηχανικής μάθησης θα πρέπει οι συσκευές αυτές να παράγουν δεδομένα. Στην συγκεκριμένη υλοποίηση οι συσκευές IoT δεν έχουν πραγματική μορφή και δεν συνεργάζονται σε πραγματικό περιβάλλον με ασθενείς. Το πρόβλημα αυτό αντιμετωπίστηκε με την βοήθεια ενός συνόλου δεδομένων (dataset) που χρησιμοποιήθηκε για να συμπεριληφθούν δεδομένα στην μνήμη των συσκευών, ώστε να γίνει σωστά η προσομοίωση τους και να λειτουργούν σαν αυτές που είναι τοποθετημένες σε κάποιο παρόμοιο περιβάλλον.

Το σύνολο δεδομένων (Dataset) που χρησιμοποιήθηκε για το σύστημα προέρχεται από το Kaggle στην έρευνα CYP-GUIDES (Cytochrome Psychotropic Genotyping

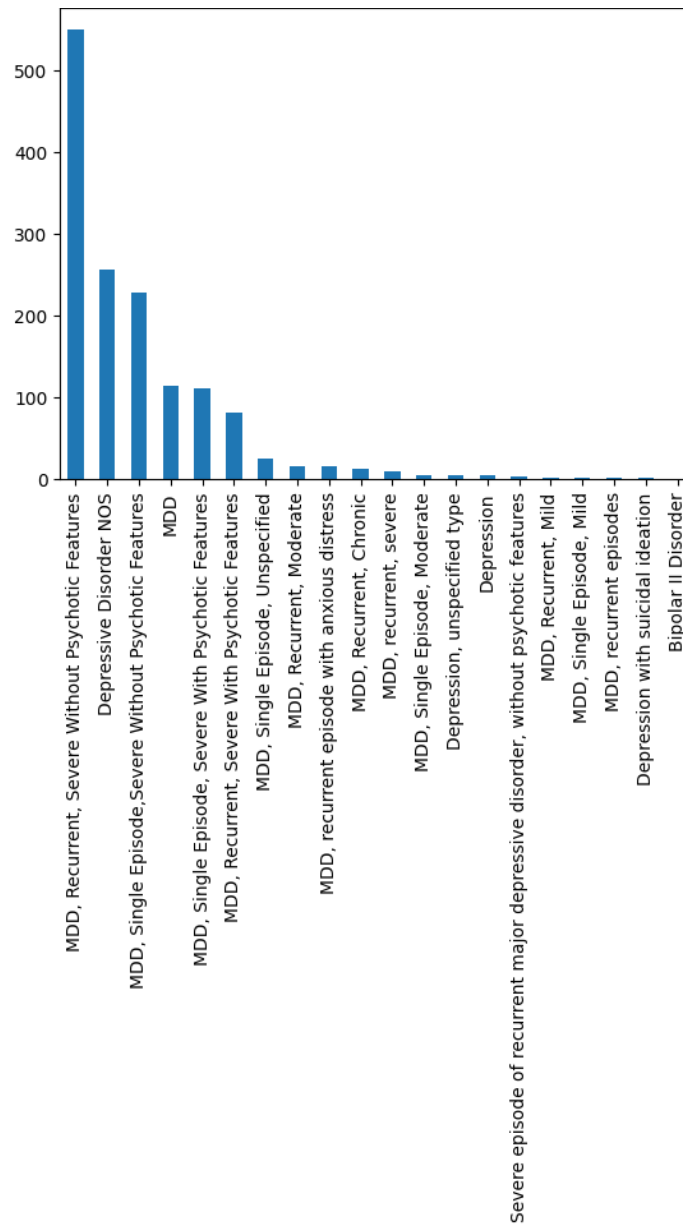
Under Investigation Decision Support) η οποία είναι μια τυχαιοποιημένη ελεγχόμενη δοκιμή (RCT) που συγκρίνει φαρμακευτικές αγωγές που χρησιμοποιήθηκαν σε νοσηλευόμενους ασθενείς με μείζον καταθλιπτική διαταραχή.

Τα δεδομένα αυτά αριθμούνται σε 1500 καταχωρήσεις από 1500 διαφορετικούς ασθενείς και περιλαμβάνει δημογραφικά χαρακτηριστικά όπως φύλο, ηλικία, εθνικότητα, χαρακτηριστικά τα οποία αναφέρονται στα φάρμακα που χρησιμοποιήθηκαν ως αγωγή για τους νοσηλευόμενους και τα στοιχεία νοσηλείας των ασθενών καταγεγραμμένα από το ίδιο το ίδρυμα περίθαλψης των ασθενών.

Πριν την τελική χρήση του συνόλου δεδομένων στην υλοποίηση του συστήματος, τα δεδομένα αναλύθηκαν και αριθμήθηκαν τα είδη των διαγνώσεων που εμφανίζονται κάθε φορά. Τα αποτελέσματα όπως φαίνονται στο διάγραμμα είναι τα εξής:

1. MDD, Recurrent, Severe Without Psychotic Features: 550
 2. Depressive Disorder NOS: 256
 3. MDD, Single Episode, Severe Without Psychotic Features : 228
 4. MDD: 114
 5. MDD, Single Episode, Severe With Psychotic Features: 111
 6. MDD, Recurrent, Severe With Psychotic Features : 81
- (τα υπόλοιπα έχουν πολύ μικρή συχνότητα εμφάνισης)

Σχήμα 14: Διάγραμμα Συχνότητας Εμφάνισης Διαγνώσεων.



Από τις διαγνώσεις με τις περισσότερες εμφανίσεις, παραλείφθηκε η διάγνωση Depressive Disorder NOS (Not Otherwise Specified) διότι η NOS θεωρείται ξεπερασμένη διαγνωστική κατηγορία και θεωρείται υπεραπλοποιημένη κατηγοριοποίηση.

Τέλος οι κοινές διαγνώσεις ενώθηκαν ώστε να αυξηθεί ο αριθμός των διαγνώσεων για το τελικό μοντέλο προβλέψεις. Πιο αναλυτικά η διάγνωση «MDD, Single Episode, Severe Without Psychotic Feature» και η «MDD, Single Episode, Severe With Psychotic Feature» μετατράπηκε στην διάγνωση «MDD, Single Episode», επίσης στην ίδια λογική η διάγνωση «MDD, Recurrent, Severe Without Psychotic Features» και η «MDD, Recurrent, Severe With Psychotic Features» μετατράπηκαν στην διάγνωση «MDD, Recurrent»

Οπότε οι τελικές διαγνώσεις οι οποίες επιλέχθηκαν για να χρησιμοποιηθούν στην τελική πρόβλεψη της κατηγοριοποίησης είναι η διάγνωση «MDD, Single Episode» και η διάγνωση «MDD, Recurrent».

4.3 Επίπεδο Συσκευών

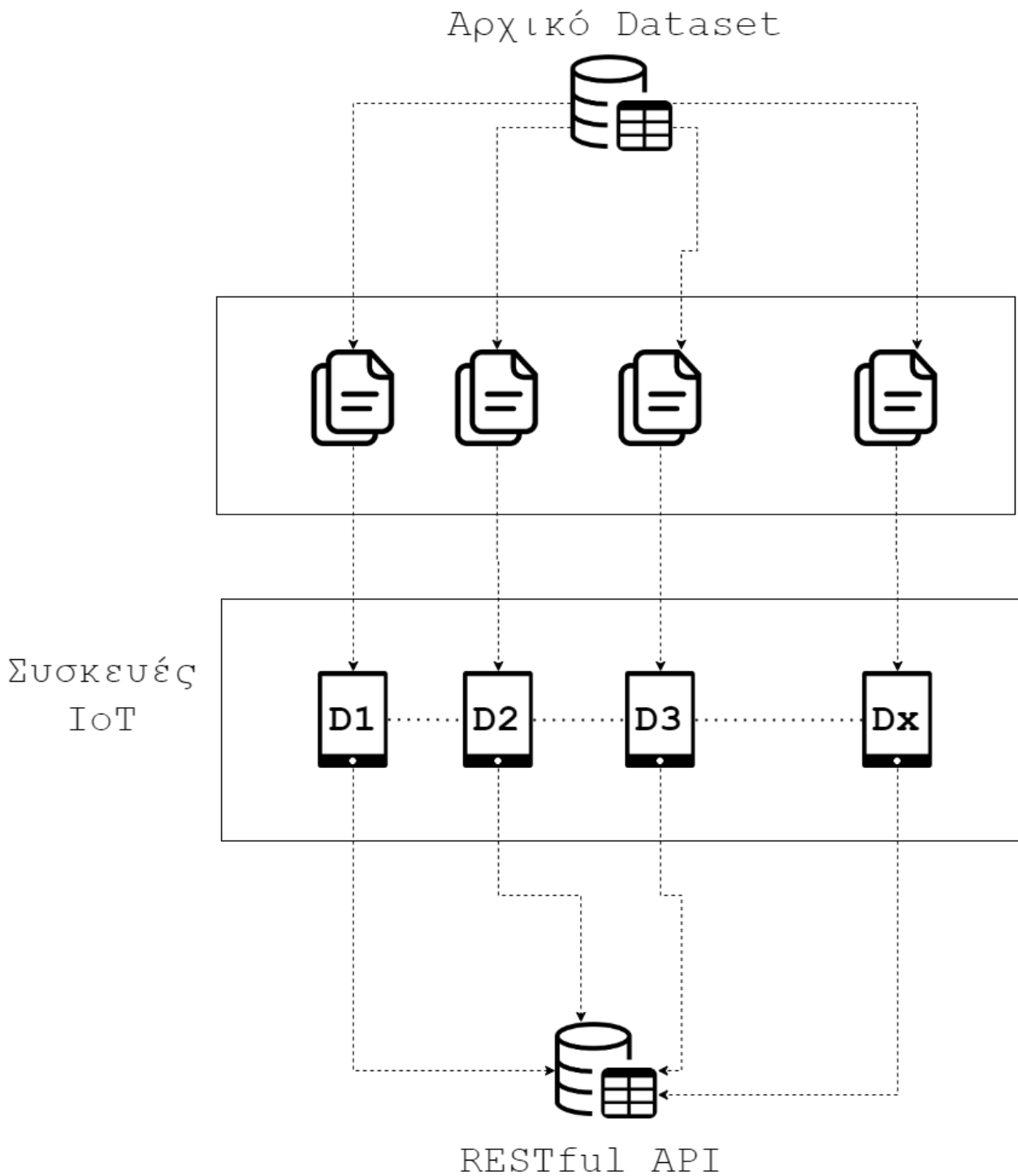
Όπως αναφέρθηκε και στην αρχιτεκτονική του συστήματος, στο πρώτο επίπεδο συμπεριλαμβάνονται οι συσκευές IoT. Στο επίπεδο αυτό γίνεται μια προσομοίωση των συσκευών IoT σε μορφή προγράμματος, έχοντας τα ίδια χαρακτηριστικά που θα είχε μια πραγματική συσκευή IoT.

Σε αυτήν την ενότητα θα περιγραφεί αναλυτικά η δομή μιας προσομοιωμένης συσκευής IoT, οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της, ο τρόπος με τον οποίο δημιουργείται η συσκευή μαζί με παραδειγματικά αποσπάσματα κώδικα που υλοποιούν τις λειτουργίες που εκτελούνται στην μονάδα των συσκευών.

4.3.1 Αρχιτεκτονική Συσκευών

Η Αρχιτεκτονική των συσκευών όπως αναπαρίσταται και στο παρακάτω διάγραμμα αποτελείται από το σύνολο δεδομένων που επιλέχθηκε και τις συσκευές IoT. Όπως φαίνεται από το πρώτο στάδιο αρχικά λαμβάνεται το σύνολο δεδομένων το οποίο διασπάται σε μικρότερα σύνολα δεδομένων, το πλήθος των συνόλων αυτών είναι όσο είναι και το πλήθος των συσκευών. Μόλις η τροφοδοσία των συσκευών με δεδομένα

ολοκληρωθεί οι συσκευές στέλνουν όλα τα δεδομένα στην κύρια βάση του συστήματος μέσω του RESTful API.



Σχήμα 15: Αρχιτεκτονική συσκευών IoT

4.3.2 Τεχνολογίες

Στο πρώτο επίπεδο του συστήματος, για την ανάπτυξη των συσκευών IoT και για την υλοποίηση του μηχανισμού επεξεργασίας του συνόλου δεδομένων που χρησιμοποιείτε για την προσθήκη καταχωρήσεων στην μνήμη τους, πραγματοποιήθηκε με την βοήθεια της γλώσσας προγραμματισμού Python της 3^{ης} έκδοσης. Επιλέχθηκε η Python για την υλοποίηση των συσκευών κατά κύριο λόγο επειδή είναι ειδικά σχεδιασμένη για την υλοποίηση επιχειρηματικής λογικής (business logic) απευθείας σε επίπεδο συσκευής.

Η γλώσσα προγραμματισμού Python υποστηρίζει την ενσωμάτωση βιβλιοθηκών, μια συλλογή δηλαδή από πακέτα κώδικα που μπορούν να χρησιμοποιηθούν επανειλημμένα σε διαφορετικά προγράμματα. Για την επεξεργασία του συνόλου δεδομένων ώστε να ενσωματωθούν κατάλληλα στις συσκευές, έγινε χρήση της βιβλιοθήκης Pandas. Επιπρόσθετα επειδή οι συσκευές θα πρέπει να επικοινωνούν με το RESTful API και να στέλνουν τα δεδομένα τους εκεί, χρησιμοποιήθηκε η βιβλιοθήκη Requests ώστε να τους δοθεί η δυνατότητα να εκτελούν HTTP αιτήματα. Τέλος για την καλύτερη προσομοίωση, χρησιμοποιήθηκε η ενσωματωμένη βιβλιοθήκη της Python Threading, για την δημιουργία νημάτων (threads) ώστε οι συσκευές να εκτελούν ταυτόχρονα τις λειτουργίες τους.

4.3.3 Υλοποίηση

Σε αυτήν την υπενότητα θα αναλυθεί ο τρόπος με τον οποίο υλοποιήθηκε ο αυτόματος διαχωρισμός του συνόλου δεδομένων σε επιμέρους περισσότερα μικρά σύνολα δεδομένων, ο τρόπος με τον οποίο κατασκευάστηκαν οι IoT συσκευές μαζί με την ενσωμάτωση των δεδομένων σε αυτές και την επικοινωνία των συσκευών με τον μεσολαβητή του δεύτερου επιπέδου API.

Αρχικά δημιουργούμε τις συσκευές IoT ώστε να εισαχθούν στην μνήμη τους τα επιμέρους δεδομένα. Για την δημιουργία των συσκευών πρώτα δηλώνουμε μια κλάση με ονομασία “Device” ώστε να μπορεί να γίνει δημιουργία και διαχείριση πολλών αντικειμένων, στην προκειμένη περίπτωση τα αντικείμενα είναι οι IoT συσκευές μας. Η κλάση αυτή διαθέτει αναγνωριστικό συσκευής, όνομα κατόχου συσκευής, ξεχωριστή μνήμη καταχωρήσεων που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου και ξεχωριστή που θα χρησιμοποιηθεί για την δοκιμή του μοντέλου.

Επιπλέον διαθέτει κάποια πρόσθετα χαρακτηριστικά και συναρτήσεις που χρησιμοποιούνται για περιπτώσεις debugging και εμφάνισης πληροφορίας στο τερματικό.

```
class Device:
    def __init__(self, id, data_train, data_test):
        self.id = id
        self.name = f'Patient{id}'
        self.data_train = data_train
        self.data_train_entries = data_train.shape[0]
        self.data_test = data_test
        self.data_test_entries = data_test.shape[0]

    def __str__(self):
        return f'Device:{self.id}' \
            f' | Name:{self.name}' \
            f' | Training Entries:{self.data_train_entries}' \
            f' | Test Entries: {self.data_test_entries}'

    def showTrainData(self):
        print(self.data_train)

    def showTestData(self):
        print(self.data_test)
```

Κώδικας 1: Device Class

Στην συνέχεια, αφού οριστεί η κλάση, δημιουργούνται τα αντικείμενα της κλάσης δηλαδή οι συσκευές. Η συνάρτηση η οποία είναι υπεύθυνη για την δημιουργία των αντικειμένων ονομάζεται createDevices και δέχεται ως παραμέτρους, τον αριθμό των συσκευών που θα αρχικοποιηθούν, το σύνολο δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου και το σύνολο δεδομένων που θα χρησιμοποιηθεί για την δοκιμή του μοντέλου.

```

devices = createDevices(numberDevices=numDevices,
                        dataframe_training=df_training,
                        dataframe_testing=df_testing)

```

Κώδικας 2: createDevices Συνάρτηση

Στην συνάρτηση createDevices δημιουργείται μια λίστα όπου αποθηκεύονται οι συσκευές που δημιουργήθηκαν. Στην συγκεκριμένη συνάρτηση δημιουργείται επίσης μια λίστα η οποία στην κάθε θέση της έχει πολλά μικρά σύνολα δεδομένων για την εκπαίδευση του και άλλη μια παρόμοια λίστα για την δοκιμή μοντέλου. Οι λίστες αυτές γεμίζουν με σύνολα δεδομένων χρησιμοποιώντας την συνάρτηση dataFeeding, η οποία σπάει το αρχικό σύνολο δεδομένων σε επιμέρους μικρότερα σύνολα διαστάσεων ίσο με τον αριθμό των συσκευών.

```

def dataFeeding(numDataframes, dataset, ):
    list_of_datasets = []
    numEntries = int(len(dataset.index) / numDataframes)
    start = 0
    end = numEntries

    df = dataset.iloc[start:end]
    list_of_datasets.append(df)
    if numDataframes > 1:
        for i in range(1, numDataframes + 1):
            start = i * numEntries
            df = dataset.iloc[start:start + numEntries]
            list_of_datasets.append(df)
    return list_of_datasets

def createDevices(numberDevices, dataframe_training, dataframe_testing):
    # creating devices
    listDevices = []
    listDatasets_training = dataFeeding(numberDevices, dataframe_training)
    listDatasets_testing = dataFeeding(numberDevices, dataframe_testing)
    for i in range(1, numberDevices + 1):
        listDevices.append(Device(i, listDatasets_training[i - 1], listDatasets_testing[i - 1]))
    return listDevices

```

Κώδικας 3: Συναρτήσεις createDevices και dataFeeding

Μετά την δημιουργία των συσκευών και την προσθήκη δεδομένων στην μνήμη τους ακολουθεί η ενσωμάτωση λειτουργιών για την κάθε συσκευή. Κάθε συσκευή θα έχει ως στόχο να στέλνει μια προς μια τις καταχωρήσεις που έχει αποθηκευμένες στην μνήμη του, θα ενεργεί δηλαδή σαν πραγματική συσκευή σε ένα απομακρυσμένο

περιβάλλον. Η λειτουργία της κάθε συσκευής ορίζεται από τις συναρτήσεις `deviceThread_training_data` και `deviceThread_testing_data`. Στην συνάρτηση `deviceThread_training_data` χρησιμοποιούνται τα δεδομένα για την εκπαίδευση που έχει η κάθε συσκευή μετατρέπονται σε μορφή JSON με τις εκχωρήσεις να στέλνονται μια προς μια στην βάση δεδομένων του εγκατεστημένου API (με διεύθυνση endpoint http://127.0.0.1:8000/data/patient_entries/ και http://127.0.0.1:8000/data/patient_entries_test/) που θα αναλυθεί στις επόμενες ενότητες. Το ίδιο γίνεται και στην συνάρτηση `deviceThread_testing_data` με διαφορά τα δεδομένα που χρησιμοποιούνται είναι εκείνα για την δοκιμή του τελικού μοντέλου.

```
def deviceThread_testing_data(device):
    """ """
    logging.info(f'[Thread {device.id}-test][Device{device.id}]: starting')
    r = requests.get(url=cc.URL_RAW_DATA_TEST, headers=cc.HEADERS_POST)
    json_ = r.json()
    if r.status_code == 200:
        for index, row in device.data_test.iterrows():
            data_json_post = {
                "Gender": row['GENDER'], "Age": row['AGE'],
                "Race": row['RACE_ETHNICITY'], "Diagnosis": row['Diagnosis'],
                "MD": row['MD'], "Assignment": row['Assignment'],
                "EMR": row['EMR'], "LOS": row['LOS'],
                "RAR": row['RAR'], "A": row['A'],
                "D": row['D'], "E": row['E'],
                "F": row['F'], "G": row['G'],
                "H": row['H'], "I": row['I'],
                "B": row['B'], "C": row['C'],
                "J": row['J'], "K": row['K'],
                "L": row['L'], "M": row['M'],
                "N": row['N'], "O": row['O'],
                "P": row['P'], "Q": row['Q'],
                "R": row['R'], "S": row['S'],
                "T": row['T'], "U": row['U'],
                "V": row['V'], "W": row['W'],
                "X": row['X'], "Y": row['Y'],
                "Z": row['Z'], "AA": row['AA'],
                "AB": row['AB'], "AC": row['AC'],
                "AD": row['AD'], "PsychotropicMedications": row['PsychotropicMedications'],
                "Administrations": row['Administrations'], "TherapeuticGuidances": row['TherapeuticGuidances']
            }
            r = requests.post(cc.URL_RAW_DATA_TEST, data=json.dumps(data_json_post), headers=cc.HEADERS_POST)
    logging.info(f'[Thread {device.id}-test][Device{device.id}]: finishing')
```

Κώδικας 4: Αποστολή καταχωρήσεων στο API

Επιπλέον για την καλύτερη προσομοίωση των συσκευών IoT, όλες οι συσκευές εκτελούν τις λειτουργίες τους ταυτόχρονα με την βοήθεια νημάτων. Δηλαδή κάθε νήμα περιλαμβάνει μια διαφορετική συσκευή. Για την υλοποίηση των νημάτων χρησιμοποιείται η συνάρτηση Thread της ενσωματωμένης βιβλιοθήκης threading όπου κάθε κατασκευασμένο νήμα αποθηκεύεται σε μια λίστα ενεργών νημάτων και αναφέρει την κατάσταση του στο τερματικό.

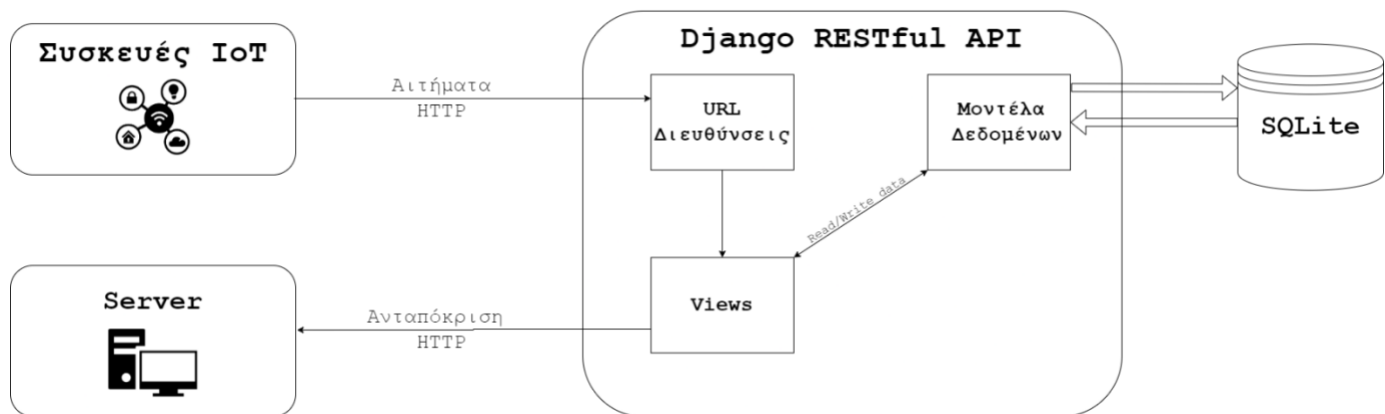
```
threads = list()
for device in devices:
    logging.info(f'[MAIN]-train: create and start thread {device.id}')
    x = threading.Thread(target=deviceThread_training_data, args=(device,))
    y = threading.Thread(target=deviceThread_testing_data, args=(device,))
    threads.append(x)
    threads.append(y)
    x.start()
    y.start()
for thread in threads:
    thread.join()
```

Κώδικας 5: Ταυτόχρονη Λειτουργία Συσκευών

4.4 Επίπεδο Μεσολαβητή API

Σε αυτήν την ενότητα θα αναλυθεί το δεύτερο επίπεδο του συστήματος, το οποίο είναι υπεύθυνο για την συλλογή των καταχωρήσεων από τις συσκευές IoT και την αποθήκευση αυτών στην βάση δεδομένων που θα χρησιμοποιήσει αργότερα ο κεντρικός Εξυπηρετητής (Server) στο τελευταίο επίπεδο του συστήματος.

Στις παρακάτω υποενότητες θα ακολουθήσει αναλυτική περιγραφή της αρχιτεκτονικής του RESTful API, στην συνέχεια θα αναφερθούν οι τεχνολογίες που χρησιμοποιήθηκαν για την δημιουργία του API και στην τελευταία υποενότητα θα αναλυθεί η βάση δεδομένων και τα μοντέλα δεδομένων που χρησιμοποιούνται από τον μεσολαβητή.



Κώδικας 6: Αρχιτεκτονική του μεσολαβητή RESTful API

4.4.1 Τεχνολογίες

Για την ανάπτυξη του μεσολαβητή API χρησιμοποιείται το Django REST (DRF) το οποίο είναι μια ολοκληρωμένη και καλά υποστηριζόμενη βιβλιοθήκη Python/Django ανοικτού κώδικα που στοχεύει στην κατασκευή εξελιγμένων διαδικτυακών API. Πρόκειται για μια ευέλικτη και πλήρως εξοπλισμένη εργαλειοθήκη με αρθρωτή και προσαρμόσιμη αρχιτεκτονική που καθιστά δυνατή την ανάπτυξη τόσο απλών, ολοκληρωμένων σημείων API όσο και περίπλοκων δομών REST. Ο λόγος που επιλέχθηκε να υλοποιηθεί με REST API αντί για υπηρεσίας SOAP είναι επειδή με αυτόν το τρόπο χρησιμοποιούμε απλές καθορισμένες διεύθυνσης (endpoints) ως πόρους δεδομένων και οι τροποποιήσεις των πόρων γίνεται μέσω απλών αιτημάτων HTTP με λειτουργίες (GET, POST, PUT, DELETE).

Οι λειτουργίες διακρίνονται ως εξής, POST αίτημα για την δημιουργία ενός ή περισσότερων νέων πόρων, GET αίτημα για την ανάκτηση ενός ή περισσότερων πόρων, PUT αίτημα για την ενημέρωση ενός ή περισσότερων πόρων και τέλος DELETE αίτημα για την διαγραφή ενός ή περισσότερων πόρων.

4.4.2 Υλοποίηση

Αρχικά πριν ξεκινήσει η ανάπτυξη του μοντέλου δεδομένων και η δημιουργία των URL endpoint διευθύνσεων, δηλώνεται στις ρυθμίσεις του Django αρχείου ο τύπος της βάσης δεδομένων που θα χρησιμοποιηθεί, στην προκειμένη περίπτωση θα είναι η SQLite. Επιλέχθηκε η SQLite ως σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων επειδή δεν είναι απαιτητική όσον αφορά την εγκατάσταση, τη διαχείριση της βάσης δεδομένων και τους απαιτούμενους πόρους.

Στην συνέχεια δημιουργήθηκε το μοντέλο δεδομένων όπου η βάση θα διαχειρίζεται. Το μοντέλο στην Django δηλώνεται ως κλάση (class) με συμπεριλαμβανόμενα τα χαρακτηριστικά και τον τύπο του κάθε χαρακτηριστικού. Τα χαρακτηριστικά που προστέθηκαν είναι εκείνα που διαθέτει και το αρχικό σύνολο δεδομένων. Δημιουργήθηκε ένα μοντέλο για δεδομένα που θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου και ένα μοντέλο για τα δεδομένα που θα χρησιμοποιηθούν για την δοκιμή του μοντέλου.

Ακολουθεί στιγμιότυπο κώδικα με το μοντέλο και τον τύπο όλων των χαρακτηριστικών. Το αναγνωριστικό του ασθενή χρησιμοποιήθηκε ως πρωτεύων κλειδί ενώ τα υπόλοιπα χαρακτηριστικά δηλώθηκαν με πεδία κατά κύριο λόγο τύπου Character και Integer.

```

class PatientEntryTest(models.Model):
    Gender = models.CharField(max_length=20, blank=True, default='')
    Age = models.IntegerField()
    Race = models.CharField(max_length=20, blank=True, default='')
    Diagnosis = models.CharField(max_length=256, blank=True, default='')
    MD = models.CharField(max_length=20, blank=True, default='')
    Assignment = models.CharField(max_length=20, blank=True, default='')
    EMR = models.CharField(max_length=20, blank=True, default='')
    LOS = models.IntegerField()
    RAR = models.IntegerField()
    A = models.IntegerField()
    B = models.IntegerField()
    C = models.IntegerField()
    D = models.IntegerField()
    E = models.IntegerField()
    F = models.IntegerField()
    G = models.IntegerField()
    H = models.IntegerField()
    I = models.IntegerField()
    J = models.IntegerField()
    K = models.IntegerField()
    L = models.IntegerField()
    M = models.IntegerField()
    N = models.IntegerField()
    O = models.IntegerField()
    P = models.IntegerField()
    Q = models.IntegerField()
    R = models.IntegerField()
    S = models.IntegerField()
    T = models.IntegerField()
    U = models.IntegerField()
    V = models.IntegerField()
    W = models.IntegerField()
    X = models.IntegerField()
    Y = models.IntegerField()
    Z = models.IntegerField()
    AA = models.IntegerField()
    AB = models.IntegerField()
    AC = models.IntegerField()
    AD = models.IntegerField()
    PsychotropicMedications = models.IntegerField()
    Administrations = models.IntegerField()
    TherapeuticGuidances = models.CharField(max_length=20, blank=True, default='')

```

Κώδικας 7: Μοντέλο δεδομένων API

Αφού ολοκληρωθεί η δημιουργία των μοντέλων που θα χρησιμοποιηθούν, η Django δίνει την δυνατότητα προσαρμογής της προβολής των πόρων μέσα από το αρχείο “views.py”. Στο αρχείο views γίνεται ο ορισμός των HTTP λειτουργιών (GET, POST, PUT, DELETE) που μπορούν οι εξωτερικές μονάδες να χρησιμοποιήσουν. Τέλος, αφού έχουν ολοκληρωθεί τα μοντέλα που θα χρησιμοποιηθούν και το αρχείο views με τις επιτρεπτές λειτουργίες, θέτονται οι τελικές URL endpoint διευθύνσεις με αποτέλεσμα το API να είναι έτοιμο να χρησιμοποιηθεί.

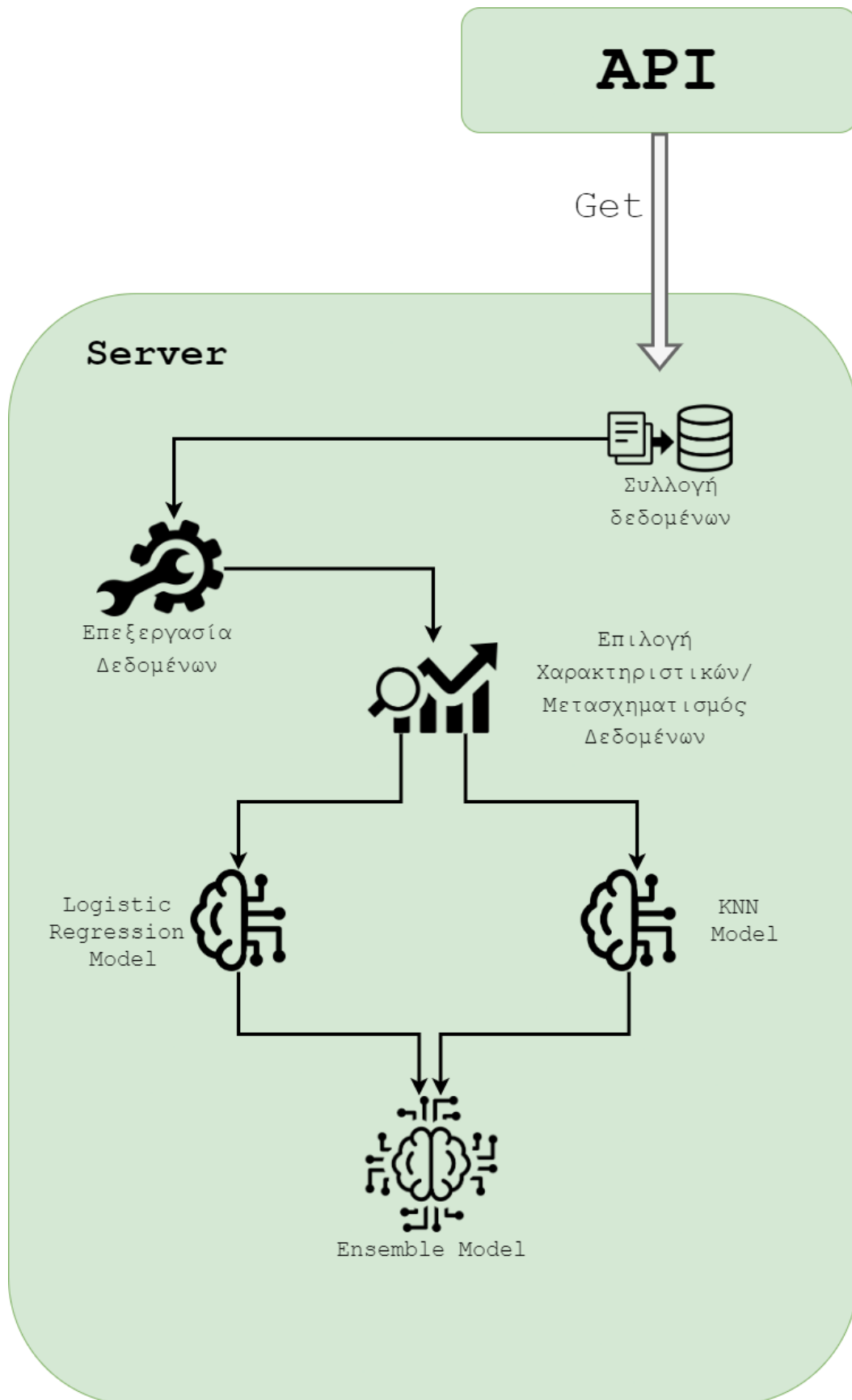
4.5 Εξυπηρετητής και Μοντέλο Μηχανικής Μάθησης

Σε αυτήν την ενότητα θα γίνει ανάλυση του τρίτου και τελευταίου επιπέδου του συστήματος. Το επίπεδο αυτό έχει την ιδιότητα ενός απομακρυσμένου εξυπηρετητή (Server) με βασικές λειτουργίες την συλλογή των συσκευών από τον μεσολαβητή API, την προεπεξεργασία αυτών σε κατάλληλη μορφή και χρήση των επεξεργασμένων τελικών δεδομένων στο μοντέλο μηχανικής μάθησης για την πρόβλεψη της τελικής διάγνωσης του ασθενή.

Στις επόμενες υποενότητες θα ακολουθήσει περιγραφή και αναπαράσταση της αρχιτεκτονικής του Server, μια περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν για την ανάπτυξη του Server και του μοντέλου και τέλος ο τρόπος με τον οποίο έγινε η προεπεξεργασία των δεδομένων η υλοποίηση του μοντέλου μηχανικής μάθησης.

4.5.1 Αρχιτεκτονική

Όπως φαίνεται και στο παρακάτω σχήμα της αρχιτεκτονικής του τρίτου επιπέδου, λαμβάνονται και συλλέγονται τα δεδομένα από τον μεσολαβητή API, τα δεδομένα αυτά στέλνονται για επεξεργασία και έπειτα περνάνε για ανάλυση ώστε να επιλεγθούν τα πιο αποδοτικά χαρακτηριστικά και να γίνει μετασχηματισμός των επιλεγμένων δεδομένων. Όταν τα δεδομένα πάρουν μια τελική μορφή, εισάγονται στο μοντέλο λογιστικής παλινδρόμησης (Logistic Regression model) και στο μοντέλο πλησιέστερων γειτόνων (KNN model). Τέλος τα δύο αυτά μοντέλα ενσωματώνονται στο Ensemble μοντέλο και με την βοήθεια αυτού γίνεται η τελική πρόβλεψη της διάγνωσης.



Σχήμα 16: Αρχιτεκτονική τρίτου επιπέδου Server

4.5.2 Τεχνολογίες

Για την ανάπτυξη του εξυπηρετητή (Server), το οποίο είναι υπεύθυνο για την επεξεργασία και ανάλυση των δεδομένων και για την ανάπτυξη των μοντέλων μηχανικής μάθησης, χρησιμοποιήθηκαν κάποιες ενσωματωμένες και μη βιβλιοθήκες της Python. Για την συλλογή των δεδομένων από τον μεσολαβητή API χρησιμοποιήθηκε η βιβλιοθήκη `request`. Στην επεξεργασία των δεδομένων σε κατάλληλη μορφή ώστε να μπορούν να χρησιμοποιηθούν για τις επόμενες λειτουργίες του εξυπηρετητή χρησιμοποιήθηκε η βιβλιοθήκη `Pandas`. Με την βοήθεια της βιβλιοθήκης `scikit-learn` υλοποιήθηκαν οι διεργασίες επιλογής αποδοτικότερων χαρακτηριστικών και οι μετασχηματισμοί των δεδομένων, και ενσωματώθηκαν οι αλγόριθμοι μηχανικής μάθησης που προσφέρονται από αυτήν. Μια ακόμη σημαντική τεχνολογία που χρησιμοποιήθηκε από τον εξυπηρετητή είναι η βιβλιοθήκη `pickle`, η οποία αποθηκεύει το τελικό και ολοκληρωμένο μοντέλο για να μπορεί να χρησιμοποιηθεί χωρίς να χρειάζεται να ξανά εκπαιδευτεί το μοντέλο από την αρχή.

4.5.3 Υλοποίηση

4.5.3.1 Στάδιο 1^ο Συλλογή Δεδομένων

Αρχικά το πρώτο βήμα του εξυπηρετητή είναι να συλλέξει όλα τα δεδομένα που ο μεσολαβητής έχει αποθηκεύσει στην βάση του από τις συσκευές IoT. Για να ξεκινήσει την συλλογή ο εξυπηρετητής στέλνει αίτημα στον μεσολαβητή για να λάβει τα δεδομένα. Με την βοήθεια της συνάρτησης `get_data` που υλοποιήθηκε, ο εξυπηρετητής λαμβάνει τα δεδομένα μετά το αίτημα που στάλθηκε στην URL endpoint διεύθυνση και τα δεδομένα που λαμβάνει έχουν JSON μορφή. Το επόμενο στάδιο του εξυπηρετητή είναι να προσπελάσει τα JSON δεδομένα και να τα αποθηκεύσει τοπικά σε αρχείο CSV ως σύνολο δεδομένων `dataset`, αυτό γίνεται με την υλοποιημένη συνάρτηση `json_to_csv`. Όλοι αυτήν η διαδικασία γίνεται ώστε τα δεδομένα να μπορέσουν να αναλυθούν, να μετασχηματιστούν και στο τέλος να ενσωματωθούν στα μοντέλα συσταδοποίησης.

```

def get_data():
    """
    Getting data from raw api in JSON format.
    """
    links = [cc.URL_RAW_DATA]
    request = requests.get(cc.URL_RAW_DATA,
                           headers=cc.HEADERS)
    json_ = request.json()
    while json_['next'] is not None:
        links.append(json_['next'])
        json_ = requests.get(json_['next']).json()

    bigJson = {'results': []}
    for link in links:
        request = requests.get(link,
                               headers=cc.HEADERS)
        json_data = request.json()
        for data_line in json_data['results']:
            # print(data_line)
            # bigJson.update(data_line)
            bigJson['results'].append(data_line)
    return bigJson

def json_to_csv(json_data):
    link = "api_dataset.csv"
    columns = list(json_data['results'][0].keys())

    f = open(link, "w")
    # 1st line columns
    i = 1
    for column in columns:
        if i != len(columns):
            f.write(column)
            f.write(',')
            i += 1
        else:
            f.write(column)
            f.write('\n')

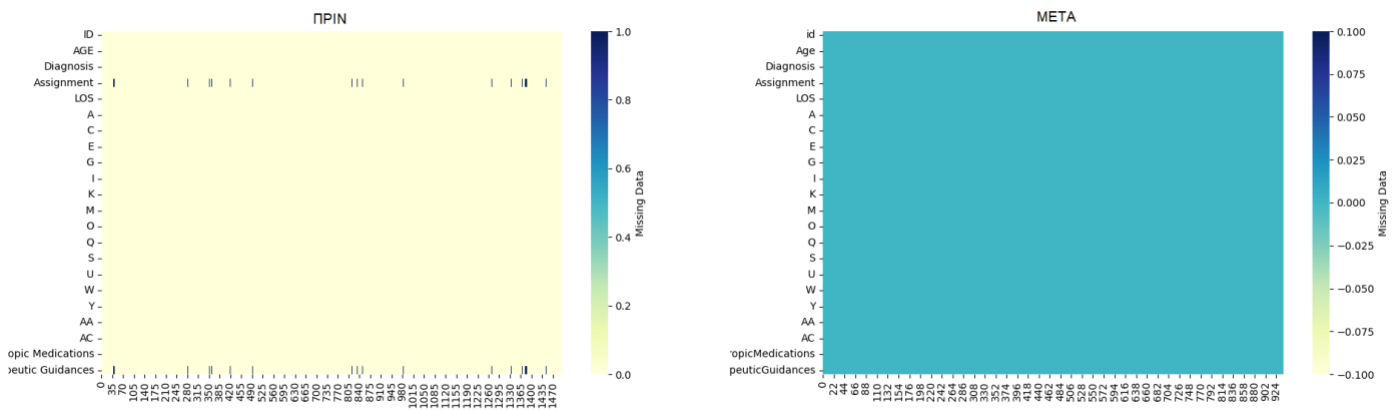
    for line in json_data['results']:
        for index_, col in enumerate(columns):
            if col == 'Diagnosis':
                f.write('\n"{}"\n".format(str(line[f'{col}'])))
            else:
                f.write(str(line[f'{col}']))
            if index_ != len(columns) - 1:
                f.write(',')
            f.write('\n')
    return link

```

Κώδικας 8: Συναρτήσεις get_data και json_to_csv

Το δεύτερο βήμα του εξυπηρετητή είναι να διαβάσει το αρχείο csv που δημιουργήθηκε στο προηγούμενο βήμα (“api_dataset.csv”), με την βοήθεια της βιβλιοθήκης Pandas να το μετατρέψει σε μορφή DataFrame ώστε να μπορέσουν να χρησιμοποιηθούν τα περαιτέρω εργαλεία για την αρχική προεπεξεργασία των δεδομένων.

Με την ολοκλήρωση του DataFrame, το επόμενο στάδιο που θα ακολουθήσει ο εξυπηρετητής είναι να αφαίρεση όλες τις εκχωρήσεις που έχουν κενές τιμές και τις εκχωρήσεις που έχουν τιμές NaN. Ένας τρόπος επίλυσης αυτού του ζητήματος ήταν να δημιουργηθεί ένα μοντέλο πρόβλεψης των κενών τιμών αλλά παραλείφθηκε λόγω μικρού πλήθους των συνολικών κενών τιμών.



Σχήμα 17: Σύνολο δεδομένων πριν και μετά την αφαίρεση των κενών τιμών

4.5.3.2 Στάδιο 2^ο Μετατροπή σε αριθμητικά δεδομένα

Επειδή το σύνολο δεδομένων περιέχει γνωρίσματα (attributes) που διαθέτουν κατηγορικές τιμές και επειδή τα μοντέλα της μηχανικής μάθησης μπορούν να λειτουργήσουν μόνο με αριθμητικές τιμές, δημιουργήθηκε η συνάρτηση `label_encoding` η οποία μετατρέπει τα κατηγορικά γνωρίσματα σε αριθμητικά ώστε να μπορεί να εκπαιδευτεί το μοντέλο και η συνάρτηση `label_decoding` ώστε να μετατρέψει τα αριθμητικά ξανά σε κατηγορικά ώστε να μπορεί να γίνει η πρόβλεψη της διάγνωσης του ασθενή.

Ο τρόπος με τον οποίο λειτουργεί η συνάρτηση μετατροπής `data_to_encode` είναι να επιλέγει τις μη αριθμητικές στήλες από ένα σύνολο δεδομένων, να τις αποθηκεύει σε μια λίστα και έπειτα με την βοήθεια της συνάρτησης `transform` της βιβλιοθήκης `scikit-learn` μετατρέπει τις μη αριθμητικές τιμές σε αριθμητικές. Η αριθμητική τιμή που εισάγεται κάθε φορά υπολογίζεται με το πλήθος των μοναδικών τιμών που υπάρχουν σε κάθε γνώρισμα, όπου η κάθε μοναδική τιμή έχει ένα δικό της αριθμητικό αναγνωριστικό.

Στο ακόλουθο στιγμιότυπο κώδικα αναπαριστανται οι συναρτήσεις κωδικοποιήσεις σε αριθμητικές τιμές (`label_encoding` και `label_decoding`).

```

def label_encoding(df, dictionary_of_encoders, categorical_columns, save=False):
    """
        Dataset non numerics classes to numerics
    """
    for column in categorical_columns:
        df[column] = dictionary_of_encoders[column].transform(df[column])
    if save:
        save_url = 'api_dataset_encoded.csv'
        saving = df.to_csv(save_url, index=False)
    return df

def label_decoding(df, dictionary_of_encoders, categorical_columns):
    """
        Dataset's column class numeric back to non numeric
    """
    for col in categorical_columns:
        df[col] = dictionary_of_encoders[col].inverse_transform(df[col])

    return df

```

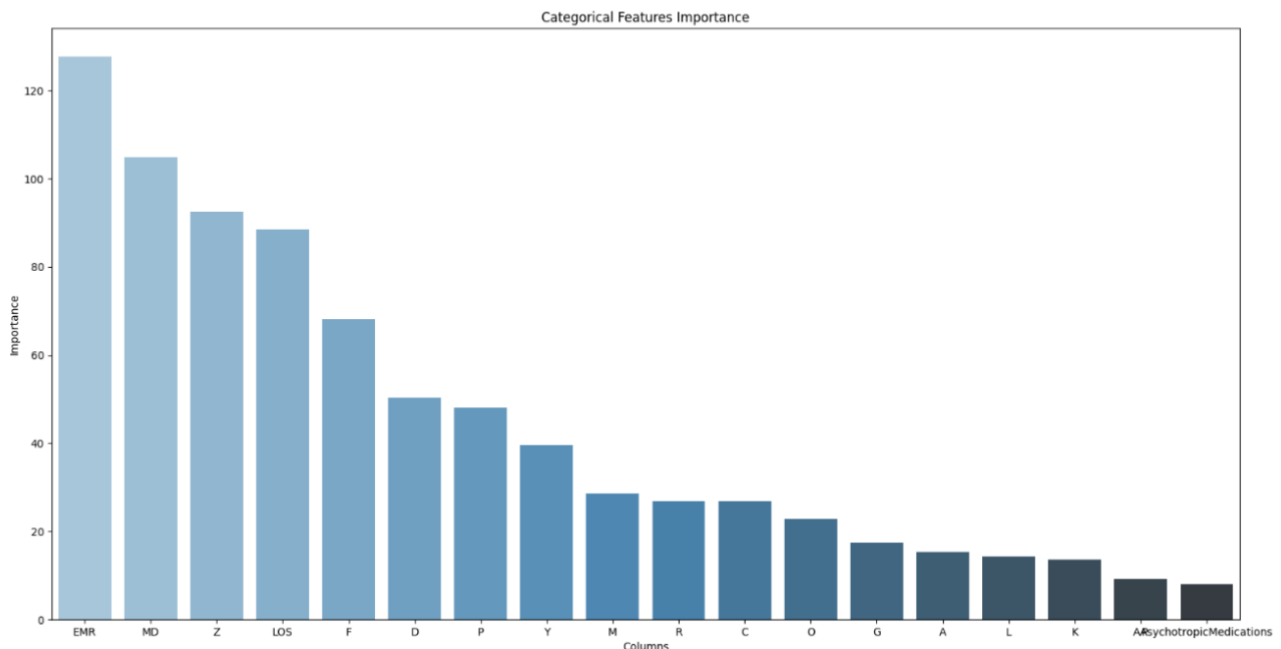
Κώδικας 9: Συναρτήσεις Label Encoding και Decoding

4.5.3.3 Στάδιο 3^ο Μετασχηματισμός Δεδομένων

Αφού ολοκληρωθεί η μετατροπή των τιμών σε αριθμητικές μπορεί να συνεχίσει η προεπεξεργασία των δεδομένων. Για την μείωση των διαστάσεων του μοντέλου, γίνεται μετασχηματισμός των χαρακτηριστικών δημιουργώντας ένα νέο σύνολο που διαθέτει μόνο τα περισσότερο σημαντικά χαρακτηριστικά.

Για την επιλογή των χαρακτηριστικών που θα χρησιμοποιηθούν στην εκπαίδευση του μοντέλου υλοποιήθηκε η συνάρτηση `feature_selection`. Η συνάρτηση αυτή δημιουργεί μια λίστα η οποία περιέχει όλα τα χαρακτηριστικά που έχουν αριθμητικές τιμές. Με την βοήθεια της συνάρτησης `SelectKBest` της `scikit` βιβλιοθήκης θα γίνει μείωση του αριθμού των συνολικών τελικών χαρακτηριστικών που θα χρησιμοποιηθούν που σημαίνει μείωση των διαστάσεων του συνόλου δεδομένων, κρατώντας μόνο τα περισσότερο πληροφοριακά χαρακτηριστικά.

Ο τρόπος με τον οποίο λειτουργεί η συνάρτηση `SelectKBest` είναι λαμβάνοντας ως παράμετρο μια συνάρτηση βαθμολογίας, η οποία πρέπει να είναι εφαρμόσιμη σε ένα ζεύγος (X, y) . Η συνάρτηση βαθμολογίας πρέπει να επιστρέφει έναν πίνακα βαθμολογιών, μια για κάθε χαρακτηριστικό $X[:, i]$ του X . Στην συνέχεια κρατάμε τα πρώτα K χαρακτηριστικά του X με τις υψηλότερες βαθμολογίες. Στην συγκεκριμένη υλοποίηση χρησιμοποιείται η χ^2 ως συνάρτηση βαθμολογίας, όπου η `SelectKBest` θα υπολογίσει το στατιστικό χ^2 μεταξύ κάθε χαρακτηριστικού του X και του y . Μια μικρή τιμή θα σημαίνει την μικρή ανεξαρτησία του από το y , ενώ μια μεγάλη τιμή σημαίνει ότι σχετίζεται με το y .



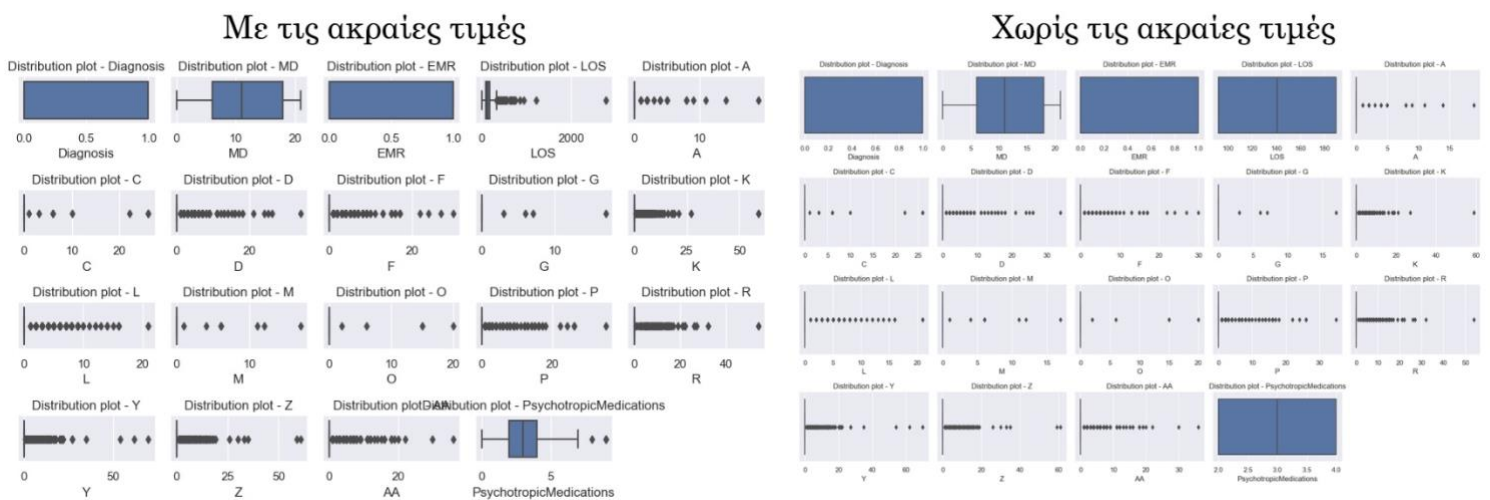
Σχήμα 18: Τα 18 χαρακτηριστικά με την μεγαλύτερη πληροφορία

Με την ολοκλήρωση της επιλογής των βέλτιστων χαρακτηριστικών πραγματοποιείται αναπαράσταση των όλων των δεδομένων με την μορφή boxplot διαγραμμάτων για την εύρεση πιθανών ακραίων τιμών (outliers). Όπως φαίνεται και στην παρακάτω αναπαράσταση των δεδομένων στα χαρακτηριστικά 'LOS' και 'PsycotropicMedications' φαίνεται ότι έχουμε δεδομένα που βρίσκονται σε μη φυσιολογική από τις άλλες τιμές που έχουμε στο σύνολο δεδομένων. Τα δεδομένα αυτά που έχουν ακραίες τιμές ανιχνεύτηκαν με την μέθοδο του ενδοτεταρτημοριακού εύρους (InterQuartile Range -IQR), όπου είναι το μέτρο διασποράς που ισούται με την διαφορά μεταξύ του ανώτερου και του κατώτερου τεταρτημόριου. Στην περίπτωση μας το ανώτερο είναι το 90^ο τεταρτημόριο και το κατώτερο το 10^ο τεταρτημόριο. Οι τιμές που είναι έξω από αυτό το πεδίο

ορισμού θεωρούνται ακραίες τιμές (outliers) και δεν συμπεριλαμβάνονται στο σύνολο δεδομένων.

```
#-----OUTLIERS-----
cols_todo = ['LOS', 'PsychotropicMedications']
for column in cols_todo:
    qua10 = df_encoded[column].quantile(0.10)
    qua90 = df_encoded[column].quantile(0.90)
    df_encoded[column] = np.where(df_encoded[column] < qua10, qua10, df_encoded[column])
    df_encoded[column] = np.where(df_encoded[column] > qua90, qua90, df_encoded[column])
    print(df_encoded[column].skew())
```

Κώδικας 10: Αποκλεισμός ακραίων τιμών



Σχήμα 19: Διαγράμματα κατανομής τιμών πριν και μετά την αφαίρεση των ακραίων τιμών.

Μόλις ολοκληρωθεί η επιλογή των χαρακτηριστικών και η αφαίρεση των ακραίων τιμών, το επόμενο βήμα του μετασχηματισμού των δεδομένων (data transformation) είναι εφαρμογή μεθόδων κανονικοποίησης των δεδομένων. Η κανονικοποίηση πραγματοποιείται ώστε να αυξηθεί η απόδοση των αλγορίθμων που θα χρησιμοποιηθούν στα μοντέλα πρόβλεψης. Ο τρόπος με τον οποίο λειτουργεί είναι με αντικατάσταση αριθμητικών τιμών με αντίστοιχων σε κάποιο άλλο εύρος τιμών. Για τα δεδομένα του εξυπηρετητή ο αλγόριθμος κλιμάκωσης των δεδομένων που χρησιμοποιήθηκε για τον μετασχηματισμό των δεδομένων είναι ο MinMaxScaler από την βιβλιοθήκη του scikit-learn.

Στην επιλογή MinMaxScaler όπου το ελάχιστο του χαρακτηριστικού γίνεται ίσο με μηδέν και το μέγιστο του χαρακτηριστικού ίσο με ένα. Ο MinMaxScaler συρρικνώνει τα δεδομένα εντός του δεδομένου εύρους. Μετασχηματίζει τα δεδομένα με κλιμάκωση των χαρακτηριστικών σε ένα δεδομένο εύρος. Κλιμακώνει τις τιμές σε ένα συγκεκριμένο εύρος τιμών χωρίς να αλλάζει το σχήμα της αρχικής κατανομής.

Ο τρόπος με τον οποίο γίνεται η MinMax κλιμάκωση είναι:

$$x_{before} = \frac{x - x_{\min(row)}}{x_{\max(row)} - x_{\min(row)}}$$

$$x_{after} = x_{before} * (max - min) + min$$

Εξίσωση 5: MinMax κλιμάκωση

Όπου:

- min, max = Διάστημα των χαρακτηριστικών
- $x_{\min(row)}$ = Ελάχιστη τιμή χαρακτηριστικών
- $x_{\max(row)}$ = Μέγιστη τιμή χαρακτηριστικών

4.5.3.4 Στάδιο 4^ο Εκπαίδευση Μοντέλου

Με την ολοκλήρωση του μετασχηματισμού των δεδομένων ο εξυπηρετητής είναι έτοιμος να προχωρήσει στην ενσωμάτωσή τους στα μοντέλα μηχανικής μάθησης ώστε να ακολουθήσει η εκπαίδευση τους. Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν είναι η Λογιστική Παλινδρόμηση (Logistic Regression) και Πλησιέστεροι Γείτονες (KNN).

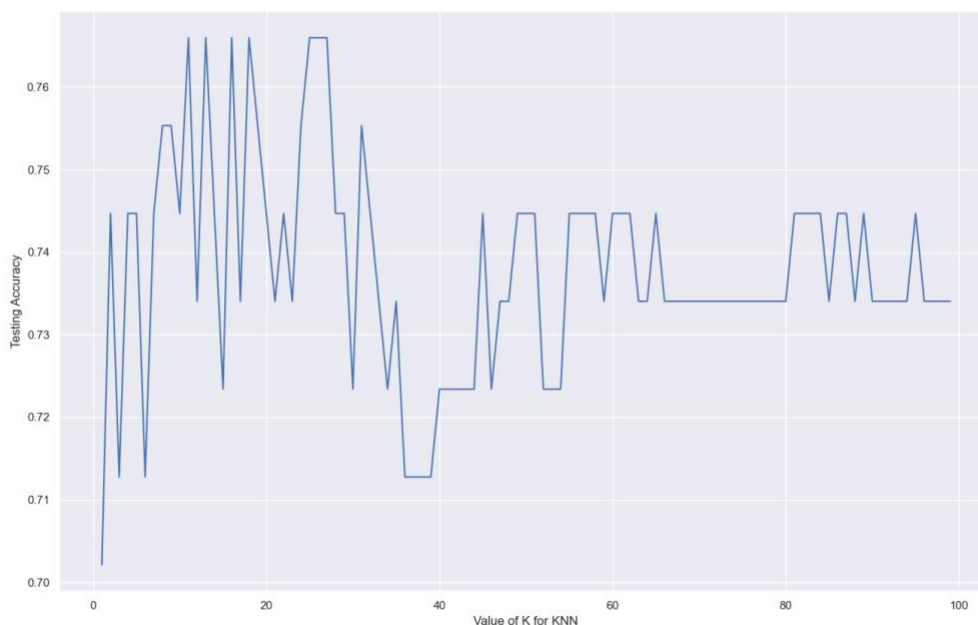
Αρχικά το πρώτο μοντέλο που υλοποιήθηκε για την συσταδοποίηση των δεδομένων και την πρόβλεψη του ποσοστού διάγνωσης είναι οι Πλησιέστεροι γείτονες (KNN), το οποίο αναπτύχθηκε με την βοήθεια της συνάρτησης KNeighborsClassifier της scikit-learn βιβλιοθήκης. Η υλοποίηση του συγκεκριμένου μοντέλου χωρίστηκε σε δύο μέρη, την βασική δημιουργία και εκπαίδευση του μοντέλου και μιας συνάρτησης που υπολογίζει το βέλτιστο πλήθος των γειτόνων που θα χρησιμοποιηθούν στο μοντέλο.

Η συνάρτηση για την εύρεση της βέλτιστης επιλογής του αριθμού των γειτόνων πραγματοποιείται με την δοκιμή του KNeighborsClassifier για κάθε πιθανό

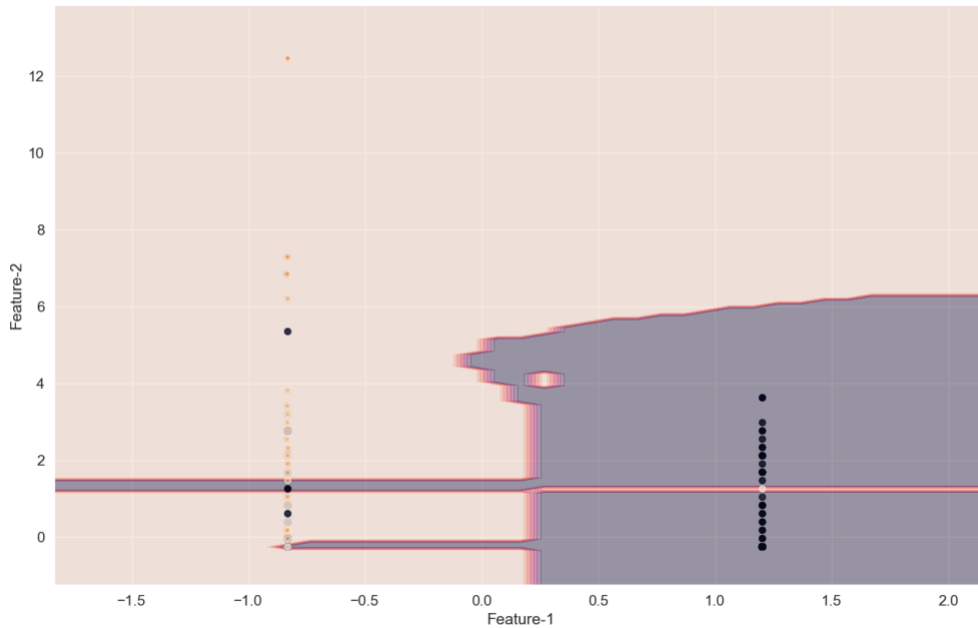
συνδυασμό της παραμέτρου που αφορά τον αριθμό γειτόνων που θα εφαρμοστούν αργότερα στην τελική εκπαίδευση. Μόλις η δοκιμή όλων των συνδυασμών ολοκληρωθεί, επιστρέφεται στην κύρια εκπαίδευση ο αριθμός των γειτόνων με τη βέλτιστη βαθμολογία που είχε ως αποτέλεσμα στην δοκιμή του.

Στα παρακάτω διαγράμματα αναπαρίστανται, η ακρίβεια του μοντέλου των κοντινότερων γειτόνων ανάλογα τον αριθμό γειτόνων που δόθηκε ως παράδειγμα και την ακρίβεια του μοντέλου στην πρόβλεψη του και στο άλλο τα όρια της συσταδοποίησης του KNN μοντέλου.

Αφού ο βέλτιστος αριθμός των γειτόνων επιστραφεί στην κύρια δημιουργία του KNN μοντέλου, χρησιμοποιείται ως παράμετρος για τον αριθμό των γειτόνων που θα χρησιμοποιηθούν στην ανάπτυξη του μοντέλου.



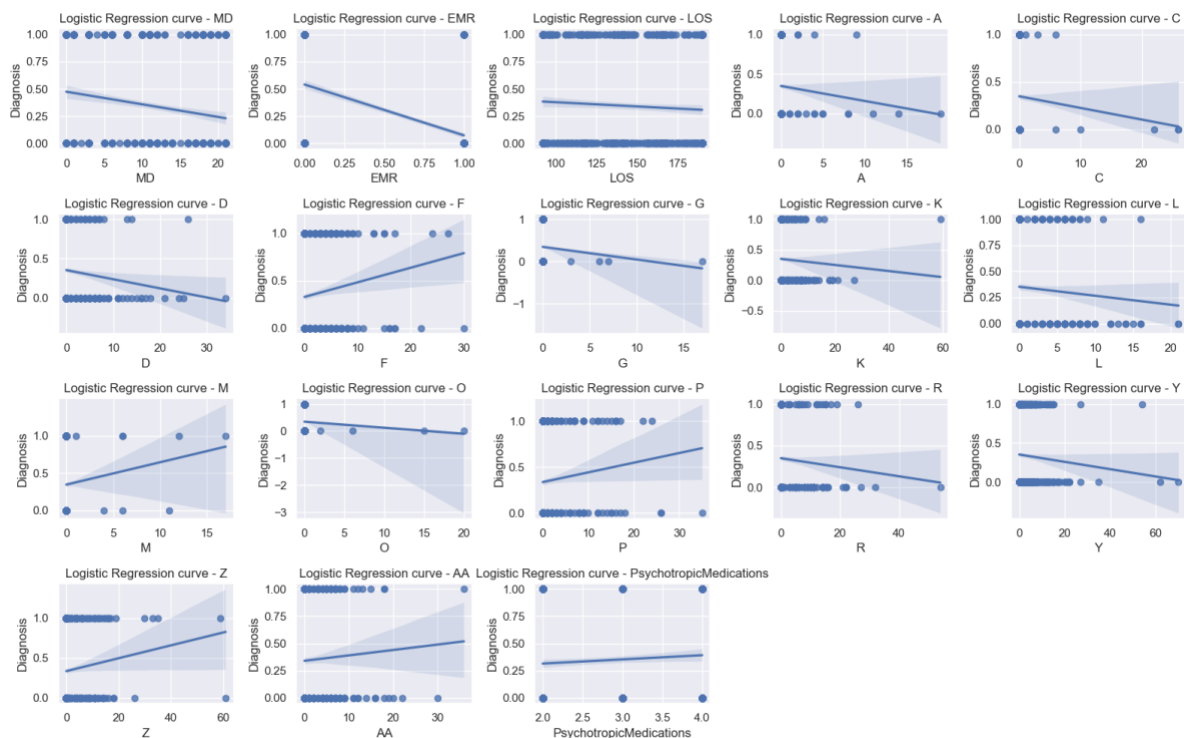
Σχήμα 20: Διάγραμμα ακρίβειας ανά πλήθος γειτόνων



Σχήμα 21: Όρια συσταδοποίησης του KNN μοντέλου

Έπειτα το επόμενο μοντέλο που δημιουργείται είναι το μοντέλο της Λογιστικής Παλινδρόμησης (Logistic Regression). Το μοντέλο αυτό υλοποιήθηκε με την βοήθεια της συνάρτησης `LogisticRegression` της `scikit-learn` βιβλιοθήκης. Οι παράμετροι που χρησιμοποιήθηκαν για την χρήση της συνάρτησης `LogisticRegression` και την εκπαίδευση του συγκεκριμένου μοντέλου είναι ως solver χρησιμοποιείται η παράμετρος `liblinear` επειδή είναι πιο αποδοτική για τα μικρότερα σύνολα δεδομένων.

Στην παρακάτω εικόνα εμφανίζονται οι συνδυασμοί διαγραμμάτων για το κάθε ένα χαρακτηριστικό του τελικού συνόλου δεδομένων σε σχέση με το χαρακτηριστικό της διάγνωσης. Στο κάθε ένα διάγραμμα αναπαρίσταται η καμπύλη της λογιστικής παλινδρόμησης.



Σχήμα 22: Καμπύλες Λογιστικής Παλινδρόμησης

Τέλος αφού ολοκληρωθεί η εκπαίδευση και των δύο μοντέλων, της Λογιστικής Παλινδρόμησης και των Πλησιέστερων γειτόνων (KNN), συμπεριλαμβάνονται και τα δύο σε έναν ensemble αλγόριθμο με στόχο την δημιουργία ενός ταξινομητή ψηφοφορίας (voting classifier). Το τελικό μοντέλο αυτό του voting classifier θα λαμβάνει υπόψιν τα αποτελέσματα πιθανοτήτων που τα άλλα μοντέλα είχαν ως έξοδο και με την βοήθεια αυτών θα υπολογίζεται-εκτιμάται η τελική πιθανότητα της διάγνωσης του ασθενή. Ως παράμετρο της ψήφους για την δημιουργία του μοντέλου χρησιμοποιήθηκε η μέθοδος soft, στην οποία η ψηφοφορία υπολογίζεται με βάση την προβλεπόμενη πιθανότητα της κλάσης εξόδου.

Όπως φαίνεται και στα παρακάτω αριθμημένα στατιστικά, το μοντέλο KNN και LR εξαγωγή την πιθανότητα διάγνωσης και μετά με την βοήθεια του voting classifier βγαίνει ο μέσος όρος των πιθανοτήτων, με την μεγαλύτερη μέση πιθανότητα να χρησιμοποιείται ως τελική εξαγωγή πιθανότητας πρόβλεψης διάγνωσης του ασθενή.

Με την ολοκλήρωση της εκπαίδευσης όλων των μοντέλων, γίνεται δοκιμή των μοντέλων με ένα μικρό κομμάτι του συνόλου δεδομένων που χρησιμοποιούνται ως δεδομένα δοκιμής (test data) ώστε να υπολογιστεί η ακρίβεια πρόβλεψης του κάθε μοντέλου με την βοηθητική συνάρτηση `accuracy_score` της βιβλιοθήκης `scikit-learn`. Οι ακρίβειες του κάθε μοντέλου είναι οι εξής:

- `KNeighborsClassifier`: 74,4%
- `LogisticRegression`: 73,4%
- `VotingClassifier`: 76,5%

Με την βοήθεια της βιβλιοθήκης `pickle` αποθηκεύουμε το τελικό ensemble μοντέλο σε μορφή `pickle` ώστε να μπορεί να χρησιμοποιηθεί ξανά με τα δεδομένα που χωριστήκαν για σκοπούς παρουσιάσεις δοκιμής για την επίδειξη του συστήματος στο πέμπτο κεφάλαιο.

ΚΕΦΑΛΑΙΟ 5 Περιγραφή Εκτέλεσης

Στα προηγούμενα κεφάλαια αναλύθηκε ο τρόπος με τον οποίο αναπτύχθηκε το σύστημα με τις επιμέρους υπομονάδες του και επίσης δόθηκε μια γενική εικόνα των τεχνολογιών που χρησιμοποιούνται στο σύστημα. Στο συγκεκριμένο κεφάλαιο θα παρουσιαστεί ο τρόπος με τον οποίο λειτουργεί η κάθε επιμέρους υπομονάδα του συστήματος. Θα γίνει δηλαδή η μελέτη χρήσης της δημιουργίας της προσομοίωσης των συσκευών IoT, το περιβάλλον του μεσολαβητή που κρατάει και αποθηκεύει τα δεδομένα και τέλος την λειτουργία του Εξυπηρετητή και η αλληλεπίδραση του με τον τελικό χρήστη για εξαγωγή πρόβλεψης ποσοστού διάγνωσης.

Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια οι υπομονάδες του συστήματος λειτουργούν ανεξάρτητα μεταξύ τους για αυτό θα πρέπει να γίνει η εκτέλεση και η διαχείριση της κάθε υπομονάδας ξεχωριστά.

Αρχικά η βασική υπομονάδα του συστήματος είναι αυτήν του μεσολαβητή και θα πρέπει να εκτελεστεί πρώτη διότι επικοινωνεί με τις άλλες δύο υπομονάδες και είναι υπεύθυνη για την μεσολάβηση των δεδομένων τους. Οπότε ο προγραμματιστής ξεκινάει μέσα από το τερματικό την λειτουργία της γράφοντας την εντολή “python manage.py runserver”. Με την εκκίνηση της λειτουργίας του API λαμβάνουμε μέσα από το τερματικό ασύγχρονα την κατάσταση του μεσολαβητή για οποιαδήποτε ενέργεια και εκτελεστεί από τις υπόλοιπες υπομονάδες.

```
(env-thesis) C:\Users\Giannis\Desktop\ bachelor-thesis\API\dataApi>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 15, 2022 - 09:35:17
Django version 4.1, using settings 'dataApi.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Εικόνα 1: Terminal εκκίνηση μεσολαβητή

Η διεπαφή του μεσολαβητή εμφανίζεται όταν ο χρήστης – προγραμματιστής συνδέεται μέσω του περιηγητή (browser) του στην διεύθυνση “http://127.0.0.1:8000/data/patient_entries/” . Μέσα από την διεπαφή αυτήν προβάλλονται οι καταχωρήσεις των ασθενών – χρηστών. Η διεπαφή του μεσολαβητή μπορεί να χρησιμοποιηθεί για την προβολή και διαχείριση των καταχωρήσεων όλων των ασθενών.

Αρχικά στην εικόνα που ακολουθεί βλέπουμε την διεπαφή του μεσολαβητή κενή, χωρίς τις καταχωρήσεις των ασθενών, επειδή δεν έχει γίνει η τροφοδότηση των συσκευών με δεδομένα ακόμα και δεν έχει εκτελεστεί η λειτουργία αποστολής των δεδομένων στο API.

Patient Entry List

OPTIONS GET

GET /data/patient_entries/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{  
  "count": 0,  
  "next": null,  
  "previous": null,  
  "results": []  
}
```

Raw data HTML form

Gender	<input type="text"/>
Age	<input type="text"/>
Race	<input type="text"/>
Diagnosis	<input type="text"/>
MD	<input type="text"/>
Assignment	<input type="text"/>

Εικόνα 2: Διεπαφή μεσολαβητή API

Το επόμενο στάδιο του συστήματος, το οποίο αναλαμβάνει την προσομοίωση των IoT συσκευών, ξεκινά να δημιουργεί τις συσκευές αυτές μόλις ο προγραμματιστής εκτελέσει το βασικό αρχείο σε μορφή python. Με την βοήθεια του τερματικού (terminal) εμφανίζονται στον χρήστη-προγραμματιστή οι πληροφορίες των συσκευών. Οι πληροφορίες αυτές διακρίνονται σε ενημέρωση του χρήστη για την δημιουργία των συσκευών και πληροφόρηση του χρήστη για τις βασικές πληροφορίες της κάθε συσκευής για την κατανομή των δεδομένων που θα χρησιμοποιηθούν στην τροφοδοσία του API. Η προσθήκη των καταχωρήσεων των ασθενών μπορεί να γίνει μέσα από την υπομονάδα των συσκευών αυτόματα με τα ήδη αποθηκευμένα δεδομένα ή από την διεπαφή του μεσολαβητή καταχωρώντας ένα προς ένα στοιχεία Gender, Age, Race, Diagnosis κ.α.

```
Creating Devices:
Device-1 was successfully created.
Device-2 was successfully created.
Device-3 was successfully created.
Device-4 was successfully created.
Device-5 was successfully created.
Device-6 was successfully created.
Device-7 was successfully created.
Device-8 was successfully created.
Device-9 was successfully created.
Device-10 was successfully created.
[DEVICE1]: Device:1 | Name:Patient1 | Training Entries:94 | Test Entries: 2

[DEVICE2]: Device:2 | Name:Patient2 | Training Entries:94 | Test Entries: 2

[DEVICE3]: Device:3 | Name:Patient3 | Training Entries:94 | Test Entries: 2

[DEVICE4]: Device:4 | Name:Patient4 | Training Entries:94 | Test Entries: 2

[DEVICE5]: Device:5 | Name:Patient5 | Training Entries:94 | Test Entries: 2

[DEVICE6]: Device:6 | Name:Patient6 | Training Entries:94 | Test Entries: 2

[DEVICE7]: Device:7 | Name:Patient7 | Training Entries:94 | Test Entries: 2

[DEVICE8]: Device:8 | Name:Patient8 | Training Entries:94 | Test Entries: 2

[DEVICE9]: Device:9 | Name:Patient9 | Training Entries:94 | Test Entries: 2

[DEVICE10]: Device:10 | Name:Patient10 | Training Entries:94 | Test Entries: 2
```

Εικόνα 3: Βασικές πληροφορίες συσκευών

Με την ολοκλήρωση της δημιουργίας των συσκευών και την κατανομή των δεδομένων σε αυτές, δημιουργούνται νήματα (threads) τα οποία τρέχουν ταυτόχρονα και αντιπροσωπεύει την λειτουργία της κάθε συσκευής IoT ξεχωριστά. Στο τερματικό ο χρήστης μπορεί να δει την κατάσταση και την λειτουργία της κάθε λειτουργίας ξεχωριστά, την χρονική στιγμή που κάθε συσκευή ξεκίνησε να στέλνει τα δεδομένα της στον μεσολαβητή μέχρι και την χρονική στιγμή που η συσκευή ολοκλήρωσε την αποστολή των δεδομένων της.

```
13:10:06: [MAIN]-train: create and start thread 1
13:10:06: [Thread 1-train][Device1]: starting
13:10:06: [Thread 1-test][Device1]: starting
13:10:06: [MAIN]-train: create and start thread 2
13:10:06: [Thread 2-train][Device2]: starting
13:10:06: [Thread 2-test][Device2]: starting
13:10:06: [MAIN]-train: create and start thread 3
13:10:06: [Thread 3-train][Device3]: starting
13:10:06: [Thread 3-test][Device3]: starting
13:10:06: [MAIN]-train: create and start thread 4
13:10:06: [Thread 4-train][Device4]: starting
13:10:06: [Thread 4-test][Device4]: starting
13:10:06: [MAIN]-train: create and start thread 5
13:10:06: [Thread 5-train][Device5]: starting
13:10:06: [Thread 5-test][Device5]: starting
13:10:06: [MAIN]-train: create and start thread 6
13:10:06: [Thread 6-train][Device6]: starting
13:10:06: [Thread 6-test][Device6]: starting
13:10:06: [MAIN]-train: create and start thread 7
13:10:06: [Thread 7-train][Device7]: starting
13:10:06: [Thread 7-test][Device7]: starting
13:10:06: [MAIN]-train: create and start thread 8
13:10:06: [Thread 8-train][Device8]: starting
13:10:06: [Thread 8-test][Device8]: starting
13:10:06: [MAIN]-train: create and start thread 9
13:10:06: [Thread 9-train][Device9]: starting
13:10:06: [Thread 9-test][Device9]: starting
13:10:06: [MAIN]-train: create and start thread 10
13:10:06: [Thread 10-train][Device10]: starting
13:10:06: [Thread 10-test][Device10]: starting
13:10:08: [Thread 9-test][Device9]: finishing
13:10:08: [Thread 1-test][Device1]: finishing
13:10:10: [Thread 2-test][Device2]: finishing
13:10:10: [Thread 5-test][Device5]: finishing
13:10:11: [Thread 7-test][Device7]: finishing
13:10:12: [Thread 4-test][Device4]: finishing
13:10:12: [Thread 8-test][Device8]: finishing
13:10:13: [Thread 10-test][Device10]: finishing
13:10:14: [Thread 3-test][Device3]: finishing
13:10:15: [Thread 6-test][Device6]: finishing
```

Εικόνα 4: Διαργασίες των συσκευών IoT

Κατά την διάρκεια της αποστολής των δεδομένων της κάθε συσκευής έχοντας το τερματικό παράθυρο του μεσολαβητή ανοιχτό, εμφανίζονται όλα τα http αιτήματα (requests) που λαμβάνει ο μεσολαβητής API από την υπομονάδα διαχείρισης IoT συσκευών. Στην παρακάτω εικόνα φαίνονται όλα τα αιτήματα καταχωρήσεων (POST) στην τελική διεύθυνση /data/patient_entries/ και την αποδοχή των αιτημάτων αυτών από τον μεσολαβητή εμφανίζοντας τον κωδικό 201.

```
[14/Sep/2022 13:12:04] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:04] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 394
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 391
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 387
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 391
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 391
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 391
[14/Sep/2022 13:12:05] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 393
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 391
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 387
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 389
[14/Sep/2022 13:12:06] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 388
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 390
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 391
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 393
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 387
[14/Sep/2022 13:12:07] "POST /data/patient_entries/ HTTP/1.1" 201 388
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 397
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 387
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 389
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 395
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:08] "POST /data/patient_entries/ HTTP/1.1" 201 394
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 388
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 393
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 386
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 396
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 395
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 393
[14/Sep/2022 13:12:09] "POST /data/patient_entries/ HTTP/1.1" 201 390
[14/Sep/2022 13:12:10] "POST /data/patient_entries/ HTTP/1.1" 201 396
```

Εικόνα 5: Αιτήματα καταχωρήσεων δεδομένων στο τερματικό του μεσολαβητή

Μόλις ληφθούν όλα τα αιτήματα από την υπομονάδα των IoT συσκευών και γίνει η αποδοχή τους από την υπομονάδα του μεσολαβητή, ο χρήστης-προγραμματιστής μπορεί να δει με την βοήθεια της διεπαφής του μεσολαβητή API τα δεδομένα των ασθενών που καταχωρήθηκαν από τις IoT συσκευές. Οι καταχωρήσεις εμφανίζονται σε JSON μορφή και μπορούν να προβληθούν συνολικά όλες οι καταχωρήσεις μαζί αλλά και η κάθε καταχώρηση ξεχωριστά αφού ο χρήστης-προγραμματιστής πρόσθεση στην τελική διεύθυνση endpoint τον αριθμό id της καταχώρησης του ασθενή. Για την πρώτη καταχώρηση με παράδειγμα την παρακάτω εικόνα, ο χρήστης θα έπρεπε να επισκεφτεί την διεύθυνση http://127.0.0.1:8000/data/patient_entries/10354 για να δει μόνο την συγκεκριμένη καταχώρηση με τα δημογραφικά στοιχεία του και τα υπόλοιπα πεδία που καταχώρησε.

Django REST framework

Api Root / Patient Entry List

Patient Entry List

OPTIONS GET

« 1 2 3 ... 10 »

GET /data/patient_entries/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "count": 938,
  "next": "http://127.0.0.1:8000/data/patient_entries/?limit=100&offset=100",
  "previous": null,
  "results": [
    {
      "id": 10354,
      "Gender": "M",
      "Age": 25,
      "Race": "W",
      "Diagnosis": "MDD, Single Episode",
      "MD": "D",
      "Assignment": "G",
      "EMR": "C",
      "LOS": 121,
      "RAR": 0,
      "A": 0,
      "B": 0,
      "C": 0,
      "D": 0,
      "E": 0,
    }
  ]
}
```

Εικόνα 6: Διεπαφή μεσολαβητή API με δεδομένα

Με την ολοκλήρωση της συλλογής των δεδομένων από τις συσκευές IoT στην βάση δεδομένων του μεσολαβητή API, ακολουθεί η πρόβλεψη ποσοστού της διάγνωσης του ασθενή.

Αρχικά γίνεται εκκίνηση της ξεχωριστής υπομονάδας του εξυπηρετητή. Κατά την εκτέλεση αυτής της υπομονάδας, συλλέγονται όλες οι συνολικές καταχωρήσεις των ασθενών για να ακολουθήσει η εκπαίδευση του μοντέλου μηχανικής μάθησης. Κατά την διάρκεια της συλλογής των δεδομένων εμφανίζονται στο τερματικό (terminal) όλα τα αιτήματα GET που λαμβάνονται από τον μεσολαβητή API και την αποδοχή

αυτών με την αποστολή του κωδικού 200 που σημαίνει ότι η αποστολή ήταν επιτυχής.

```
system check identified no issues (0 silenced).
September 16, 2022 - 15:39:51
Django version 4.1, using settings 'dataApi.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[16/Sep/2022 15:45:03] "GET /data/patient_entries_test/ HTTP/1.1" 200 36586
[16/Sep/2022 15:45:46] "GET /data/patient_entries/ HTTP/1.1" 200 39280
[16/Sep/2022 15:45:46] "GET /data/patient_entries/?limit=100&offset=100 HTTP/1.1" 200 39360
[16/Sep/2022 15:45:46] "GET /data/patient_entries/?limit=100&offset=200 HTTP/1.1" 200 39305
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=300 HTTP/1.1" 200 39415
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=400 HTTP/1.1" 200 39328
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=500 HTTP/1.1" 200 39295
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=600 HTTP/1.1" 200 39317
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=700 HTTP/1.1" 200 39389
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=800 HTTP/1.1" 200 39317
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=900 HTTP/1.1" 200 14992
[16/Sep/2022 15:45:47] "GET /data/patient_entries/ HTTP/1.1" 200 39280
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=100 HTTP/1.1" 200 39360
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=200 HTTP/1.1" 200 39305
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=300 HTTP/1.1" 200 39415
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=400 HTTP/1.1" 200 39328
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=500 HTTP/1.1" 200 39295
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=600 HTTP/1.1" 200 39317
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=700 HTTP/1.1" 200 39389
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=800 HTTP/1.1" 200 39317
[16/Sep/2022 15:45:47] "GET /data/patient_entries/?limit=100&offset=900 HTTP/1.1" 200 14992
```

Εικόνα 7: : Συλλογή δεδομένων από τον εξυπηρετητή

Τέλος αφού ολοκληρωθεί η εκπαίδευση του μοντέλου μηχανικής μάθησης, ο εξυπηρετητής λαμβάνει τα επιπλέον δεδομένα δοκιμής από τον μεσολαβητή και τα χρησιμοποιεί για την πρόβλεψη της διάγνωσης του ασθενή. Ο χρήστης μέσα από το τερματικό μπορεί να δει την έναρξη της δοκιμής των καταχωρήσεων στο μοντέλο μηχανικής μάθησης και τα αποτελέσματα αυτών για τον κάθε ασθενή ξεχωριστά. Στα αποτελέσματα αυτά αναφέρεται το αναγνωριστικό του ασθενή ακολουθούμενο από το ποσοστό της διάγνωσης που ασθενής μπορεί να εμφανίσει στο μέλλον.

```
Running Model
[08:10:29]-Patient with ID:0 has 61.03% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:1 has 60.66% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:2 has 59.56% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:3 has 59.68% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:4 has 59.45% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:5 has 55.19% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:6 has 57.49% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:7 has 58.01% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:8 has 62.15% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:9 has 73.14% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:10 has 62.17% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:11 has 56.94% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:12 has 50.87% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:13 has 66.72% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:14 has 57.67% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:15 has 50.49% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:16 has 73.94% Probability of occurrence of symptoms (MDD, Single Episode)
[08:10:29]-Patient with ID:17 has 73.79% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:18 has 57.26% Probability of occurrence of symptoms (MDD, Recurrent)
[08:10:29]-Patient with ID:19 has 71.64% Probability of occurrence of symptoms (MDD, Recurrent)
```

Εικόνα 8: Τελικά αποτελέσματα προβλέψεων

ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα και Μελλοντικές Προεκτάσεις

6.1 Συμπεράσματα

Με την ανάπτυξη ποικίλων συστημάτων και μοντέλων μηχανικής μάθησης σε συνδυασμό την συλλογή και διαχείριση δεδομένων που προέρχονται από συσκευές IoT έχουν θετικό αντίκτυπο σε διάφορους επιστημονικούς ή και μη τομείς και κατά συνέπεια την επίλυση διάφορων προβλημάτων. Ο τομέας υγείας, θεωρείται ένας από τους πιο σημαντικούς και κρίσιμους τομείς διότι με την κάθε απόφαση που λαμβάνεται για την πρόβλεψη κάποιας διάγνωσης ή ασθένειας, το αποτέλεσμα της πρόβλεψης έχει μεγάλη σημασία.

Για την ανάπτυξη ενός βέλτιστου μοντέλου μηχανικής μάθησης δεν αρκούν μόνο τα εργαλεία και οι βιβλιοθήκες που προσφέρει μια γλώσσα προγραμματισμού ώστε να επιτευχθεί η σωστή παραγωγή αποτελεσμάτων. Θεωρείται απαραίτητη η συνεργασία με κάποιον ειδικό στον συγκεκριμένο τομέα αλλά και στην συγκεκριμένη ειδικότητα που σχετίζονται τα δεδομένα. Επομένως με την βοήθεια κάποιου ειδικού ιατρού θα επιλεχθούν τα κατάλληλα δεδομένα που σχετίζονται με το πρόβλημα που θέλουμε να αντιμετωπίσουμε.

6.2 Βελτιώσεις

Σε αυτήν την ενότητα θα αναφερθούν κάποιες σκέψεις και ιδέες, οι οποίες θα μπορούσαν να εξελίξουν και να επεκτείνουν την συγκεκριμένη υλοποίηση αλλά να την διαμορφώσουν σε επίπεδο ικανό για εμπορική αλλά και ερευνητική χρήση.

6.2.1 Συσκευές IoT

Όπως αναφέρθηκε και στο Κεφάλαιο 4, δεν χρησιμοποιήθηκαν συσκευές υλικού χαρακτήρα αλλά έγινε η προσομοίωση τους σε μορφή προγράμματος, με την κάθε συσκευή να διαθέτει ήδη δεδομένα τα οποία έχουν εισαχθεί από τον προγραμματιστή τα οποία .

Μια βασική αλλαγή στο κομμάτι της υλοποίησης και στην τροφοδοτήσει δεδομένων, θα μπορούσε να γίνει από πραγματικές συσκευές που θα συμμετέχουν στην καταγραφή δεδομένων σε πραγματικό χρόνο. Οι συσκευές αυτές θα μπορούσαν να είναι απλές όπως η χρήση εφαρμογών σε έξυπνα τηλέφωνα (smartphones) με απλή συμπλήρωση ερωτηματολογίου από τον χρήστη – ασθενή, αλλά και πολύπλοκες συσκευές με κατάλληλους αισθητήρες που θα ενσωματώνονταν στον σώμα του κάθε ασθενή και θα καταγράφανε φυσιολογικά, βιομετρικά ή δεδομένα συμπεριφοράς. Και οι δύο τύποι συσκευών θα είχαν ως αποτέλεσμα την συλλογή τεράστιων μεγεθών από δεδομένα, το οποίο θα αποτελούσε χρήσιμο στην εκπαίδευση ενός καλύτερου μοντέλου.

6.2.2 Μοντέλο πρόβλεψης

Στο κομμάτι της εξαγωγής αποτελεσμάτων, η μελέτη και η δοκιμή περισσότερων αλγόριθμων συσταδοποίησης θα μπορούσαν να βελτιώσουν την απόδοση του τελικού μοντέλου και της ακρίβειας του, έχοντας ως αποτέλεσμα πιο ακριβείς πρόβλεψης της διάγνωσης του ασθενή.

BIBΛΙΟΓΡΑΦΙΑ

- 1) Alexander L. Fradkov, "Early History of Machine Learning" (2020) [online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320325027>
- 2) Akash, "What is Unsupervised Learning and How does it Work?" (2020) [online]. Available: <https://www.edureka.co/blog/unsupervised-learning/>
- 3) Akash, "What is Supervised Learning and its different types?" (2020) [online]. Available: <https://www.edureka.co/blog/supervised-learning/>
- 4) Official Django website, "Meet Django" [online]. Available: <https://www.djangoproject.com/>
- 5) Wikipedia, "Python (programming language)" [online]. Available: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- 6) Shashwat Tiwari, "Clinical Dataset of the CYP-GUIDES Trial"(2020) [online]. Available: <https://www.kaggle.com/datasets/shashwatwork/clinical-dataset-of-the-cypguides-trial>
- 7) Vitalin Mokin, "Heart Disease - Automatic AdvEDA & FE & 20 models" (2020) [online]. Available: <https://www.kaggle.com/code/vbmokin/heart-disease-automatic-aveda-fe-20-models>
- 8) Andreas C. Müller & Sarah Guido, "Introduction to Machine Learning with Python" (2017). [ebook].
- 9) Shai Shalev-Shwartz and Shai Ben-David, "Understanding Machine Learning: From Theory to Algorithms" (2014) [ebook]
- 10) Ethem Alpaydm, "Introduction To Machine Learning, Second Edition" (2010) [ebook].
- 11) Εργαστήριο Ανάλυσης Συστημάτων και Τεχνολογίας Λογισμικού, "REST Υπηρεσίες", [online]. Available: <https://eclass.uoa.gr/modules/document/index.php?course=D226&download=/541c23e5gapN/5436a2c67Ybk/54579aacLG6U.pdf>
- 12) Erwin Adi, Adnan Anwar, Zubair Baig, Sherali Zeadally, Machine learning and data analytics for the IoT(2020) [online]. Available: <https://link.springer.com/article/10.1007/s00521-020-04874-y>
- 13) TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU, "SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS" (2012). Available: <https://www.itu.int/rec/T-REC-Y>
- 14) Qusay F. Hassan, "Internet of Things A to Z" (2018), [ebook].
- 15) Adamu Murtala Zungeru, S. Subashini, P. Vetrivelan, "Wireless Communication Networks and Internet of Things" (2019) [ebook][online]. Available: <https://link.springer.com/book/10.1007/978-981-10-8663-2>