



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΠΜΣ «Ψηφιακές Επικοινωνίες και Δίκτυα»

Διπλωματική εργασία με τίτλο:

**Σχεδιασμός και Διαχείριση δικτύων αισθητήρων
και έξυπνων εφαρμογών IoT
(Design and Management of IoT/Sensor
Networks & Application)**

Δημήτριος Κατσιούλας

Επιβλέπων Καθηγητής : Κώστας Τσαγκάρης

Πειραιάς, Φεβρουάριος 2022

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ'ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, με την προϋπόθεση όμως να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται στο έγγραφο αυτό, αφορούν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιά.

Ευχαριστίες

Φτάνοντάς στο τέλος των μεταπτυχιακών μου σπουδών και με τη διαδικασία περάτωσης της παρούσας διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω θερμά τον Δρ. Κωνσταντίνο Τσαγκάρη για την εμπιστοσύνη που μου έδειξε με την ανάθεση της διπλωματικής αυτής εργασίας, καθώς και για την καθοδήγηση του καθ' όλη την διάρκεια εκπόνησής της. Ένα μεταπτυχιακό που μόνο πλουσιότερο με έκανε σε γνώσεις και κατάρτιση για τον κλάδο.

Στη συνέχεια θα ήθελα να ευχαριστήσω την οικογένεια μου και την κοπέλα μου για την στήριξη και την ενθάρρυνση τους σε κάθε μου βήμα.

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας, όπως λέει και ο τίτλος, είναι ο σχεδιασμός και η διαχείριση δικτύων αισθητήρων νερού και έξυπνων εφαρμογών IoT. Παρά την τεχνολογική εξέλιξη τόσο σε επίπεδο Hardware, όσο και σε επίπεδο Software, τα παραδοσιακά δίκτυα και ο τρόπος λειτουργίας τους όπως αυτός έχει εξελιχθεί μέχρι σήμερα είναι πιθανό να μη μπορούν να ανταπεξέλθουν στον ολοένα αυξανόμενο αριθμό συνδεδεμένων συσκευών στο διαδίκτυο. Το γεγονός αυτό, έχει στρέψει το ενδιαφέρον της επιστημονικής κοινότητας προς την κατεύθυνση της βελτιστοποίησης των υπάρχοντων πρωτοκόλλων επικοινωνίας ή και του σχεδιασμού νέων μοντέλων που θα μπορούν να υποστηρίξουν τη διαλειτουργικότητα των συσκευών του Internet of Things.

Στις σελίδες που ακολουθούν θα αναλυθεί ο όρος του Internet of Things, η αρχιτεκτονική του, οι βασικές τεχνολογίες που χρησιμοποιούνται για να το υποστηρίξουν. Στη συνέχεια γίνεται αναφορά σε προβλήματα που μπορεί να προκύψουν, αλλά και τα προβλήματα που μπορεί να βοηθήσει και να λύσει το IoT. Έχοντας πλέον φτάσει σε ένα τόσο καλό τεχνολογικό επίπεδο, το IoT μπορεί να προσφέρει ό,τι είχε εκτιμηθεί πως θα προσφέρει. Πλέον οι λέξεις έξυπνη συσκευή, έξυπνη εφαρμογή και σε μεγαλύτερη κλίμακα έξυπνη πόλη, έχει μπει για τα καλά στην ζωή μας και στην πράξη. Μέσα από τις πολλές διαφορετικές δυνατότητες και κατηγορίες επιλέχθηκε για την παρούσα εργασία η μελέτη, ο σχεδιασμός και η επεξεργασία συστημάτων αισθητήρων διαχείρισης υδάτων.

Τέλος, έπειτα από την συλλογή δεδομένων από ήδη υπάρχοντες αλγόριθμους, αναπτύχθηκε ένας αλγόριθμος για την εύρεση της καλύτερης και πιο συμφέρουσας τοποθέτησης και επιλογής τεχνολογίας και συσκευών, για τη δημιουργία ενός έξυπνου δικτύου αισθητήρων νερού IoT και ένα υποστηρικτικό εργαλείο διαχείρισης.

Abstract

The aim of this thesis, as the title says, is the design and management of water sensor networks and smart IoT applications. Despite the technological development, both at the Hardware and Software level, traditional networks and their mode of operation, as it has evolved to date, are likely to be unable to cope with the ever-increasing number of devices connected to the internet. This fact has turned the interest of the scientific community towards the optimization of existing communication protocols or the design of new models that will be able to support the interoperability of Internet of Things devices.

The following pages will analyze the term Internet of Things, its architecture, the key technologies used to support it. Then there is a reference to problems that can arise, but also the problems that IoT can help and solve. Having now reached such a good technological level, the IoT can deliver what it was estimated to deliver. Now the words smart device, smart application and on a larger scale smart city, have entered our lives. Through the many different possibilities and categories, the study, design and processing of water management sensor systems was chosen for this work.

Finally, after collecting data from already existing algorithms, an algorithm was developed to find the best and most advantageous placement and selection of technology and devices, to create a smart IoT water sensor network and a supporting management tool.

Contents

Ευχαριστίες	3
Περίληψη	4
Abstract	5
Κατάλογος Σχημάτων	8
Κατάλογος Πινάκων	8
Συνομογραφίες – Ακρωνύμια.....	9
1. Εισαγωγή στο Διαδίκτυο των πραγμάτων - Internet of Things.....	10
1.1 Internet of Things (IoT), η εμφάνισή του και οι βασικές τεχνολογίες	10
1.1.1 Αισθητήρες (Sensors).....	11
1.1.2 RFID tags.....	13
1.1.3 Bluetooth.....	13
1.1.4 Near Field Communications (NFC).....	14
1.2 Χαρακτηριστικά & Απαιτήσεις του IoT	14
1.3 Αρχιτεκτονική IoT	15
1.4 Γενικά του 5G	17
1.5 Γενικά του LoRa	18
1.6 Γενικά του NB-IoT	20
1.7 Τύποι και Πρωτόκολλα Επικοινωνίας.....	20
1.7.1 Device-to-Device Επικοινωνία.....	20
1.7.2 Device-to-Cloud Επικοινωνία	21
1.7.3 Device-to-Gateway Επικοινωνία	21
1.7.4 Back-End Data-Sharing Επικοινωνία.....	22
1.7.5 Πρωτόκολλα Επικοινωνίας.....	23
1.8 Πρωτόκολλα, Διαδίκτυο και APIs	23
1.8.1 TCP, UDP και IP	24
1.8.2 HTTP	24
1.8.3 MQTT	25
1.8.4 CoAP	26
1.8.5 XML	26
1.8.6 JSON	26
1.8.7 Online and web services	27
1.8.8 APIs	28
1.9 IoT/Sensor networks και έξυπνες εφαρμογές (smart applications)	28
1.10 Σχεδιασμός και Διαχείριση δικτύων IoT με 5G	30
1.11 Σχεδιασμός και Διαχείριση δικτύων IoT με LoRa	31
1.12 Σχεδιασμός και Διαχείριση δικτύων IoT με NB-IoT.....	32

2. Προβλήματα – Ζητήματα	34
2.1 Προβλήματα και προκλήσεις ασφαλείας του IoT.....	34
2.2 Προβλήματα σχεδιασμού και Διαχείρισης δικτύων IoT.....	35
2.3 Στόχος – Σκοπός	36
3. Water Management IoT	38
3.1 Βασικά πλεονεκτήματα	38
3.2 Σύγχρονες έξυπνες τεχνολογίες νερού.....	39
3.3 Μέθοδοι βελτιστοποίησης	40
3.4 Στόχοι της έξυπνης διαχείρισης των υδάτων.....	40
4. Διατύπωση προβλήματος και οι προτεινόμενοι αλγόριθμοι επίλυσης	45
4.1 Μαθηματικά μοντέλα του προβλήματος (Problem Formulation).....	45
4.2 Αλγόριθμοι επίλυσης του παραπάνω προβλήματος	49
4.2.1 Σχετικά με Greedy Algorithm – Άπληστος Αλγόριθμος	49
4.2.2 Σχετικά με Genetic Algorithm – Γενετικός Αλγόριθμος	61
5. Το προτεινόμενο σύστημα	78
5.1 Περιγραφή και αρχιτεκτονική του συστήματος.....	78
5.2 Οδηγίες εγκατάστασης και χρήσης	78
6. Επίλογος & Αξιολόγηση αποτελεσμάτων	93
6.1 Σενάρια/Περιπτώσεις χρήσης.....	93
6.2 Ανάλυση Αποτελεσμάτων/Συμπεράσματα	93
6.3 Μελλοντικές επεκτάσεις.....	94
References	95

Κατάλογος Σχημάτων

Εικόνα 1 Τα επίπεδα της αρχιτεκτονικής του IoT	16
Εικόνα 2 ranges vs. bandwidth [13]	18
Εικόνα 3 Παράδειγμα μοντέλου επικοινωνίας από συσκευή σε συσκευή.....	21
Εικόνα 4 Παράδειγμα μοντέλου επικοινωνίας από συσκευή σε νέφος.....	21
Εικόνα 5 Παράδειγμα μοντέλου επικοινωνίας από συσκευή σε πύλη	22
Εικόνα 6 Μοντέλο κοινής χρήσης δεδομένων back-end	23
Εικόνα 7 Το Protocol Stack του IoT	23
Εικόνα 8 Έξυπνο θερμοκήπιο	43
Εικόνα 9 Σύστημα παρακολούθησης νερού.....	43
Εικόνα 10 (Εικόνα από διαφάνειες Σχεδιασμός και Διαχείριση Δικτύων)	45
Εικόνα 11 Παράδειγμα χρωμοσωμάτων με binary κωδικοποίηση	62
Εικόνα 12 Απεικόνιση χρωμοσωμάτων [21]	63
Εικόνα 13 Παράδειγμα χρωμοσώματος με Permutation κωδικοποίηση	63
Εικόνα 14 Παράδειγμα χρωμοσώματος με Value κωδικοποίηση	63
Εικόνα 15 Roulette Wheel Selection [22].....	65
Εικόνα 16 Η κατάσταση πριν το Ranking [22]	65
Εικόνα 17 Η Κατάσταση μετά το Ranking (graph of order numbers) [22]	66
Εικόνα 18 Two points crossover [23]	68
Εικόνα 19 Παράδειγμα Ομοιόμορφου crossover	68
Εικόνα 20 Σχεδίαση crossover point και Ανταλλαγή γονιδίων μεταξύ γονέων	68
Εικόνα 21 Τελικό αποτέλεσμα	69
Εικόνα 22 Επεξήγηση διαδικασίας mutation	70
Εικόνα 23 Διάγραμμα ροής των γενετικών αλγορίθμων	70
Εικόνα 24 Αρχική σελίδα	83
Εικόνα 25 Σελίδα εγγραφής νέου χρήστη	83
Εικόνα 26 Σελίδα σύνδεσης νέου χρήστη	84
Εικόνα 27 Dashboard	84
Εικόνα 28 Σελίδα διαχείρισης τύπων συστήματος.....	85
Εικόνα 29 Σελίδα προσθήκης τύπου συστήματος	85

Κατάλογος Πινάκων

Πίνακας 1 Σύγκριση μεταξύ MQTT και HTTP	25
---	----

Απόδοση αγγλικών όρων

Internet of Things	Διαδίκτυο των Πραγμάτων
Middleware	Ενδιάμεσο Λογισμικό
Cloud Computing	Υπολογιστικό Νέφος
Data Analytics	Αναλύσεις Δεδομένων
Device	Συσκευή
Gateway	Πύλη
Software	Λογισμικό
Hardware	Υλικό
User Interface	Διεπαφή Χρήστη

Συντομογραφίες – Ακρωνύμια

IoT:	Internet of Things
RFID:	Radio Frequency Identification
GPS:	Global Positioning System
LAN:	Local Area Network
FTTx:	Fibre to the x
3G/4G/5G:	3 ^η /4 ^η /5 ^η γενιά δικτύων κινητής τηλεφωνίας
NFC:	Near Field Communications
EMFs:	Ηλεκτρομαγνητικά Πεδία
LoRa:	Long Range
LPWA:	Low-Powe Wide Area
API:	Application Programming Interface
IDE:	Integrated Development Environment
IP:	Internet Protocol
HTTP:	HyperText Transfer Protocol
REST:	Representational State Transfer
UHF:	Ultra High Frequency
PAN:	Personal Area Network
PaaS:	Platform as a Service
UI:	User Interface
DDoS:	Distributed denial-of-service

1. Εισαγωγή στο Διαδίκτυο των πραγμάτων - Internet of Things

1.1 Internet of Things (IoT), η εμφάνισή του και οι βασικές τεχνολογίες

Ο όρος 'Διαδίκτυο των πραγμάτων' ή 'Ίντερνετ των πραγμάτων' (αγγλικά: Internet of Things - IoT) αποτελεί το δίκτυο επικοινωνίας πληθώρας συσκευών όπως: οικιακών συσκευών, αυτοκινήτων, καθώς και κάθε αντικείμενου που ενσωματώνει ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα σε δίκτυο, ώστε να επιτρέπεται η σύνδεση και η ανταλλαγή δεδομένων. Απλούστερα, η φιλοσοφία του IoT είναι η σύνδεση όλων των ηλεκτρονικών συσκευών μεταξύ τους (τοπικό δίκτυο) ή με δυνατότητα σύνδεσης στο διαδίκτυο (παγκόσμιο ιστό). Το IoT έχει φέρει επανάσταση στον τρόπο που ζούμε, εργαζόμαστε και επικοινωνούμε επιτρέποντας σε αντικείμενα να συλλέγουν και να ανταλλάσσουν δεδομένα σε πραγματικό χρόνο. Αυτά τα δεδομένα μπορούν στη συνέχεια να αναλυθούν και να χρησιμοποιηθούν για την αυτοματοποίηση των διαδικασιών, τη βελτίωση της αποτελεσματικότητας και τη λήψη τεκμηριωμένων αποφάσεων.

Ένας απλός και πρακτικός ορισμός θα μπορούσε να περιγράψει τη διασύνδεση συσκευών που χρησιμοποιούνται στην καθημερινή ζωή του ανθρώπου, σε ένα ευρύτερο δίκτυο με στόχο την αύξηση αφενός της αποτελεσματικότητάς τους και αφετέρου της παραγωγικότητας των χρηστών στις διάφορες δραστηριότητές τους. Οι συσκευές IoT μπορούν να κυμαίνονται από απλούς αισθητήρες έως πολύπλοκες συσκευές όπως έξυπνα σπίτια, βιομηχανικά συστήματα ελέγχου και τεχνολογία φορητών συσκευών. Αυτές οι συσκευές επικοινωνούν μεταξύ τους και με κεντρικούς διακομιστές χρησιμοποιώντας μια ποικιλία τεχνολογιών, όπως Wi-Fi, Bluetooth και δίκτυα κινητής τηλεφωνίας.

Η ονομασία και ο ορισμός του IoT δόθηκε το 1985 από τον Peter T. Lewis, σύμφωνα με τον οποίο "IoT" είναι η ενσωμάτωση ανθρώπων, διαδικασιών, συσκευών και της τεχνολογίας, σε ένα κοινό δίκτυο, για την απομακρυσμένη παρακολούθηση, το χειρισμό και την αξιολόγηση των τάσεων των συσκευών" [1].

Παρά την τεχνολογική εξέλιξη τόσο σε επίπεδο Hardware, όσο και σε επίπεδο Software, τα παραδοσιακά δίκτυα και ο τρόπος λειτουργίας τους όπως αυτός έχει εξελιχθεί μέχρι σήμερα, είναι πιθανό να μην μπορούν να ανταπεξέλθουν στον ολοένα αυξανόμενο αριθμό συνδεδεμένων συσκευών στο διαδίκτυο. Το γεγονός αυτό, έχει στρέψει το ενδιαφέρον της επιστημονικής κοινότητας προς την κατεύθυνση της βελτιστοποίησης των υπάρχοντων πρωτοκόλλων επικοινωνίας ή και του σχεδιασμού νέων μοντέλων που θα μπορούν να υποστηρίξουν τη διαλειτουργικότητα των συσκευών του Internet of Things.

Οι λόγοι για τους οποίους η ύπαρξη του IoT σήμερα είναι δυνατή, οφείλεται τόσο σε επιχειρηματικούς, όσο και οικονομικούς. Οι κυριότεροι επιχειρηματικοί λόγοι για τους οποίους το IoT μπορεί να γνωρίσει μεγάλη άνθιση στα επόμενα χρόνια είναι οι ακόλουθοι [2]:

- Η ανάπτυξη του Cloud Computing, που παρέχει απομακρυσμένους υπολογιστικούς πόρους για την αποθήκευση, επεξεργασία και διαχείριση πληροφορίας.

- Η εξέλιξη του χώρου των Data Analytics. Ο αυξανόμενος σχεδιασμός νέων αλγορίθμων σε συνδυασμό με την εξέλιξη της τεχνολογίας του hardware, της αποθήκευσης δεδομένων και του Cloud Computing, επιτρέπουν τη συσχέτιση και την ανάλυση μεγάλου όγκου δεδομένων που ένα απλό τερματικό δεν μπορούσε μέχρι πριν μερικά χρόνια να διαχειριστεί μόνο του. Αυτές οι νέες και εξελισσόμενες δυνατότητες δίνουν την ευκαιρία για ευκολότερη, ταχύτερη και αποτελεσματικότερη συλλογή και ανάλυση πληροφορίας και γνώσης.

Έχοντας εξηγήσει εν συντομία την έννοια του IoT, κρίνεται σκόπιμο να αναλυθούν και οι τεχνολογίες από τις οποίες εξαρτάται κατά κύριο λόγο η λειτουργία του. Εκτός από τις “έξυπνες συσκευές” που έχουν τη δυνατότητα να είναι συνδεδεμένες στο διαδίκτυο, είναι απαραίτητη και η ένταξη στο σχεδιασμό της γενικότερης αρχιτεκτονικής, των τεχνολογιών οι οποίες θα μπορούν να λαμβάνουν πληροφορίες του περιβάλλοντος, να αναγνωρίζουν τη θέση του αντικειμένου και να συνδυαστούν με βέλτιστο τρόπο ώστε να γίνει δυνατή η υλοποίηση του οράματος “Internet of Things”. Οι σημαντικότερες τεχνολογίες, οι οποίες μπορούν να παίξουν το ρόλο αυτό σήμερα, θεωρούνται οι εν γένει τεχνολογίες μικρών radio chips, όπως για παράδειγμα οι Sensors, Bluetooth και τα RFID tags. Οι παραπάνω τεχνολογίες, είναι γνωστές και χρησιμοποιούνται σε διάφορους τομείς της ανθρώπινης δραστηριότητας μέχρι και σήμερα, όμως τα ιδιαίτερα χαρακτηριστικά τους όπως για παράδειγμα, το μικρό κόστος κατασκευής και το λειτουργικό κόστος, τα καθιστά ιδανικά για την χρήση τους σε συσκευές και υπηρεσίες Internet of Things.

1.1.1 Αισθητήρες (Sensors)

Οι αισθητήρες είναι συσκευές, που συλλέγουν δεδομένα από το φυσικό περιβάλλον και τα μεταδίδουν σε άλλες συσκευές μέσω του Διαδικτύου. Παίζουν σημαντικό στο Internet of Things (IoT) επιτρέποντας τη συλλογή και την ανταλλαγή δεδομένων μεταξύ συσκευών. Οι αισθητήρες μπορούν να μετρούν ένα χαρακτηριστικό ή τις συνθήκες του περιβάλλοντος στο οποίο τοποθετούνται και στη συνέχεια να μετατρέπουν αυτή την πληροφορία σε ψηφιακή μορφή, ώστε να μπορεί να χρησιμοποιηθεί για παρατήρηση, μετρήσεις και αξιολόγηση της ποιότητας του αντικειμένου.

Υπάρχουν πολλοί διαφορετικοί τύποι αισθητήρων IoT, όπως:

- Αισθητήρες θερμοκρασίας (Temperature sensors): μετρούν τη θερμοκρασία και μπορούν να χρησιμοποιηθούν σε συστήματα HVAC (HVAC Systems) [3], ψύξη και άλλες εφαρμογές όπου ο έλεγχος θερμοκρασίας είναι σημαντικός.
- Αισθητήρες υγρασίας (Humidity sensors): μετρούν την ποσότητα υγρασίας στον αέρα και μπορούν να χρησιμοποιηθούν στη γεωργία, τον αυτοματισμό κτιρίων και άλλες εφαρμογές.

- Αισθητήρες πίεσης (Pressure sensors): μετρούν την πίεση και μπορούν να χρησιμοποιηθούν σε βιομηχανικές διεργασίες, παρακολούθηση καιρού και άλλες εφαρμογές.
- Αισθητήρες φωτός (Light sensors): μετρούν την ένταση του φωτός και μπορούν να χρησιμοποιηθούν σε συστήματα ελέγχου φωτισμού, συστήματα ασφαλείας και άλλες εφαρμογές.
- Αισθητήρες κίνησης (Motion sensors): ανιχνεύουν κίνηση και μπορούν να χρησιμοποιηθούν σε συστήματα ασφαλείας, οικιακούς αυτοματισμούς και άλλες εφαρμογές.
- Αισθητήρες GPS (GPS sensors): προσδιορίζουν τη θέση μιας συσκευής χρησιμοποιώντας δορυφορικά σήματα και μπορούν να χρησιμοποιηθούν στην πλοήγηση, την παρακολούθηση και άλλες εφαρμογές.

Οι αισθητήρες IoT μπορούν να τροφοδοτούνται από μπαταρίες, γεγονός που τους καθιστά εξαιρετικά φορητούς ή μέσω σύνδεσης με το δίκτυο ηλεκτρικής ενέργειας. Μπορούν επίσης να συνδεθούν στο διαδίκτυο χρησιμοποιώντας διάφορες τεχνολογίες επικοινωνίας όπως Wi-Fi, Bluetooth και δίκτυα κινητής τηλεφωνίας.

Τα χαρακτηριστικά που θα πρέπει να διαθέτει ένας αισθητήρας είναι:

- Accuracy (Ακρίβεια): Είναι το πόσο σωστά μπορεί να μετρήσει ένας αισθητήρας ή πόσο μικρές οι διάφορες τιμές που μπορεί να μετρήσει. Για παράδειγμα ένας αισθητήρας μέτρησης βάρους να έχει τη δυνατότητα να μετρήσει το βάρος με πολλά δεκαδικά ψηφία (π.χ. η μέτρηση 1.57 είναι πιο ακριβής από την 1.5 ή 1.6).
- Range (Εύρος τιμών): Δίνει την υψηλότερη και τη χαμηλότερη τιμή της φυσικής ποσότητας εντός της οποίας ο αισθητήρας μπορεί πραγματικά να αισθανθεί.
- Resolution (Ανάλυση): Είναι μια σημαντική προδιαγραφή για την επιλογή αισθητήρων. Όσο υψηλότερη είναι η ανάλυση, τόσο καλύτερη είναι η ακρίβεια.
- Repeatability (Επαναληψιμότητα): Είναι η ικανότητα ενός αισθητήρα να παρέχει σταθερή έξοδο όταν υπάρχει σταθερή είσοδος, κατά την απόκτηση νέου δείγματος.
- Linearity (Γραμμικότητα): Είναι ένα μέτρο για το πόσο καλά η καμπύλη απόκρισης του αισθητήρα προσεγγίζει μια ευθεία.
- Sensitivity (Ευαισθησία): Η ευαισθησία ενός αισθητήρα είναι το πόσο πρέπει να αλλάξει η είσοδος στον αισθητήρα για να ανιχνεύσει οποιαδήποτε αλλαγή στην έξοδο.

- Environmental Impact (Αλλαγές στο περιβάλλον): Οι αλλαγές στο περιβάλλον μπορούν να επηρεάσουν την απόδοση και την ακρίβεια ενός αισθητήρα. Για παράδειγμα, ορισμένοι αισθητήρες είναι ιδιαίτερα ευαίσθητοι στη θερμοκρασία και την υγρασία.

1.1.2 RFID tags

Τα RFID (Radio- Frequency Identification) tags, συχνά θεωρούνται προαπαιτούμενα για το IoT. Είναι μικρές, ασύρματες συσκευές που περιέχουν ένα μικροσίπ και μια κεραία. Χρησιμοποιούνται για τον εντοπισμό και την παρακολούθηση στοιχείων μέσω της ασύρματης αποστολής δεδομένων σε μια συσκευή ανάγνωσης. Για παράδειγμα, μια ετικέτα RFID που είναι προσαρτημένη σε ένα εμπορευματοκιβώτιο αποστολής μπορεί να μεταδώσει πληροφορίες όπως η τοποθεσία, τα περιεχόμενα και ο προορισμός του σε μια συσκευή ανάγνωσης. Υπάρχουν δύο κύριοι τύποι ετικετών RFID: ενεργές και παθητικές. Οι ενεργές ετικέτες RFID έχουν τη δική τους πηγή ενέργειας και μπορούν να μεταδώσουν δεδομένα σε μεγάλες αποστάσεις, ενώ οι παθητικές ετικέτες RFID δεν έχουν εσωτερική πηγή ενέργειας και βασίζονται στον αναγνώστη για την ενεργοποίηση και την τροφοδοσία τους. Η τεχνολογία RFID προσφέρει πολλά πλεονεκτήματα σε σχέση με τα παραδοσιακά συστήματα barcode, συμπεριλαμβανομένης της δυνατότητας παρακολούθησης στοιχείων σε πραγματικό χρόνο, της δυνατότητας παρακολούθησης στοιχείων χωρίς οπτική επαφή και της δυνατότητας αποθήκευσης περισσότερων δεδομένων σε κάθε ετικέτα. Συνολικά, η τεχνολογία RFID διαδραματίζει κρίσιμο ρόλο σε πολλούς κλάδους και γίνεται ολοένα και πιο σημαντική καθώς περισσότερες εταιρείες προσπαθούν να βελτιώσουν τη διαχείριση της αλυσίδας εφοδιασμού, τον έλεγχο των αποθεμάτων και τις διαδικασίες παρακολούθησης περιουσιακών στοιχείων.

1.1.3 Bluetooth

Το Bluetooth είναι ένα πρότυπο ασύρματης τεχνολογίας για την ανταλλαγή δεδομένων σε μικρές αποστάσεις (χρησιμοποιώντας ακτινοβολία UHF μικρού μήκους κύματος) και μπορεί να εγκατασταθεί σε σταθερές και κινητές συσκευές με σκοπό τη δημιουργία προσωπικών δικτύων (PAN), επιτρέποντάς τους να επικοινωνούν και να ανταλλάσσουν δεδομένα χωρίς την ανάγκη φυσικής σύνδεσης. Αναπτύχθηκε ως μέσο ασύρματης σύνδεσης τηλεφώνων, υπολογιστών και άλλων συσκευών και έκτοτε έχει γίνει μια ευρέως χρησιμοποιούμενη τεχνολογία για μια σειρά εφαρμογών. Το Bluetooth λειτουργεί στη ζώνη συχνοτήτων των 2,4 GHz και μπορεί να μεταδώσει δεδομένα με ταχύτητες έως και 2 Mbps. Το Bluetooth χρησιμοποιείται συνήθως σε συσκευές όπως smartphones, tablet, φορητούς υπολογιστές και τεχνολογία wearable. Χρησιμοποιείται επίσης σε συστήματα οικιακού αυτοματισμού, ασύρματα ηχεία και άλλες εφαρμογές που απαιτούν ασύρματη συνδεσιμότητα μικρής εμβέλειας. Η τεχνολογία Bluetooth έχει πολλά πλεονεκτήματα, όπως χαμηλή κατανάλωση ενέργειας, χαμηλό κόστος και ευκολία στη χρήση. Υποστηρίζει επίσης ένα ευρύ φάσμα προφίλ, τα οποία είναι τυποποιημένοι ορισμοί του τρόπου χρήσης του Bluetooth για συγκεκριμένες εφαρμογές

1.1.4 Near Field Communications (NFC)

Το Near Field Communication, NFC και στα ελληνικά “Επικοινωνία Κοντινού Πεδίου”, αποτελεί μια πρότυπη τεχνολογία συνδεσιμότητας, ασύρματης επικοινωνίας μικρής εμβέλειας, η οποία διαδίδεται και εξελίσσεται ραγδαία με κύριο σκοπό τη λύση αρκετών προβλημάτων, σύγχρονων αλλά και μελλοντικών. Είναι μια μικρής εμβέλειας ασύρματη τεχνολογία, η οποία λειτουργεί στη συχνότητα των 13,56 MHz και μεταφέρει δεδομένα με ρυθμό έως και 424 Kbps και έχει γίνει γνωστή κυρίως μέσω της χρήσης της από τα κινητά τελευταίας γενιάς (smartphones). Η λειτουργία της βασίζεται στην επαφή ή στην προσέγγιση, σε απόσταση περίπου τεσσάρων με πέντε εκατοστών, της συσκευής που περιέχει το τσιπ NFC, σε κάποια άλλη συσκευή που περιλαμβάνει τον κατάλληλο αισθητήρα. Το NFC χρησιμοποιείται συχνά για εργασίες που απαιτούν απλή και ασφαλή ανταλλαγή δεδομένων μεταξύ συσκευών, όπως πληρωμή, μεταφορά δεδομένων από μια συσκευή σε άλλη ή σύνδεση σε ασύρματο δίκτυο. Χρησιμοποιείται επίσης σε συστήματα μεταφορών, όπως οι έξυπνες κάρτες για τα μέσα μαζικής μεταφοράς, και σε συστήματα ελέγχου πρόσβασης, όπως η είσοδος χωρίς κλειδί για κτίρια και οχήματα. Συνολικά, το NFC είναι μια εξαιρετικά ασφαλής και βολική τεχνολογία που έχει πολλές εφαρμογές και γίνεται ολοένα και πιο δημοφιλής ως μέσο διευκόλυνσης απλών και ασφαλών ανταλλαγών δεδομένων μεταξύ συσκευών.

1.2 Χαρακτηριστικά & Απαιτήσεις του IoT

Οι δυνατότητες και οι απαιτήσεις του Internet of Things (IoT) εξαρτώνται από τη συγκεκριμένη εφαρμογή, αλλά υπάρχουν ορισμένα κοινά στοιχεία που υπάρχουν σε πολλά συστήματα IoT. Τα γενικά χαρακτηριστικά και οι γενικές απαιτήσεις του IoT είναι [4]:

- Heterogeneity (Ετερογένεια): Το IoT χαρακτηρίζεται από μεγάλη ανομοιογένεια (ετερογένεια), λόγω του μεγάλου αριθμού διαφορετικών συσκευών που είναι συνδεδεμένες. Η διαχείριση και η υποστήριξη διαφορετικών συσκευών / τεχνολογιών / υπηρεσιών / περιβάλλοντων, αποτελεί βασική απαίτηση του IoT.
- Scalability (Επεκτασιμότητα): Η επεκτασιμότητα είναι μια βασική προϋπόθεση της υλοποίησης και της ορθής λειτουργίας του IoT. Το IoT πρέπει να κάνει μια αποτελεσματική διαχείριση των τεράστιων αριθμών δεδομένων (Big Data), των πόρων και των λειτουργιών που είναι απαραίτητη προϋπόθεση ορθής λειτουργίας του IoT. Χρόνο με το χρόνο θα αυξάνεται ο αριθμός των συσκευών και ο αριθμός των αλληλεπιδράσεων και των υπηρεσιών. Προκύπτουν λοιπόν σημαντικά ζητήματα, όπως αυτά του αποθηκευτικού χώρου και της επαρκούς (συνολικής) υπολογιστικής ισχύος.
- Data Storage (Αποθήκευση δεδομένων): Οι συσκευές IoT πρέπει να μπορούν να αποθηκεύουν δεδομένα για μεταγενέστερη ανάλυση και αναφορά. Αυτό μπορεί να περιλαμβάνει την αποθήκευση δεδομένων τοπικά στη συσκευή ή στο cloud

- Cost minimization (Ελαχιστοποίηση κόστους): Η ελαχιστοποίηση του κόστους ανάπτυξης / συντήρησης, καθώς και η ελαχιστοποίηση της κατανάλωσης ενέργειας αποτελούν βασικό μέλημα των σχεδιαστών μιας IoT αρχιτεκτονικής.
- Flexibility & Connectivity (Ευελιξία και συνδεσιμότητα): Απαιτείται δυναμική διαχείριση και επαναπρογραμματισμός των συσκευών, καθώς η κατάστασή τους μεταβάλλεται διαρκώς (π.χ. συνδεδεμένο/ αποσυνδεδεμένο, ενεργοποίηση/απενεργοποίηση λειτουργίας ύπνου). Οι συσκευές IoT πρέπει να μπορούν να συνδέονται στο διαδίκτυο και σε άλλες συσκευές προκειμένου να ανταλλάσσουν δεδομένα και να εκτελούν τις προβλεπόμενες λειτουργίες τους. Αυτό συνήθως περιλαμβάνει τη χρήση τεχνολογιών ασύρματης επικοινωνίας όπως Wi-Fi, Bluetooth και δίκτυα κινητής τηλεφωνίας.
- Quality of Service (QoS - Ποιότητα υπηρεσιών): Η εγγύηση υψηλής ποιότητας παρεχόμενων υπηρεσιών και εφαρμογών έχει μεγάλη σημασία, ιδιαίτερα όταν οι εφαρμογές αυτές πραγματεύονται real-time πληροφορίες.
- User Interface (Διεπαφές χρήστη): Οι συσκευές IoT πρέπει να διαθέτουν διεπαφές χρήστη που είναι εύχρηστες και εύχρηστες, συχνά μέσω μιας εφαρμογής για κινητά ή μιας διεπαφής που βασίζεται στον ιστό.
- Secure environment (Ασφάλεια): Το IoT πρέπει να εγγυηθεί ασφαλές περιβάλλον και αξιόπιστο δίκτυο, παρέχοντας ασφάλεια στις επικοινωνίες μέσω της ταυτοποίησης των συσκευών και των χρηστών, διατηρώντας την ακεραιότητα των δεδομένων και των συσκευών και προστατεύοντας τα προσωπικά δεδομένα.

1.3 Αρχιτεκτονική IoT

Οι αρχιτεκτονικές IoT αποτελούνται συνήθως από ένα σύνολο αισθητήρων που συλλέγουν διαφορετικούς τύπους δεδομένων και τα μεταδίδουν σε έναν «σταθμό βάσης» που ανεβάζει τα δεδομένα στο cloud. Η αρχιτεκτονική του Διαδικτύου των Πραγμάτων (IoT) αναφέρεται στον συνολικό σχεδιασμό και τη δομή ενός συστήματος IoT, συμπεριλαμβανομένων των στοιχείων και των τεχνολογιών που εμπλέκονται, και στον τρόπο με τον οποίο αλληλεπιδρούν μεταξύ τους. Η επικρατέστερη IoT αρχιτεκτονική είναι η αρχιτεκτονική πέντε επιπέδων [5][6]. Τα πέντε αυτά επίπεδα ξεκινώντας από κάτω προς τα πάνω είναι:

1. Perception Layer (Επίπεδο Αντίληψης)

Σκοπός αυτού του επιπέδου είναι η αναγνώριση των αντικειμένων και η συλλογή πληροφοριών και δεδομένων από αυτά. Στη βιβλιογραφία συναντάται συχνά και ο όρος 'επίπεδο έξυπνων αντικειμένων' ή 'επίπεδο αισθητήρων'. Το επίπεδο αντίληψης αποτελείται από φυσικά αντικείμενα όπως κάμερες, αισθητήρες, ενεργοποιητές, RFID tags, GPS, τερματικά και δίκτυα αισθητήρων.

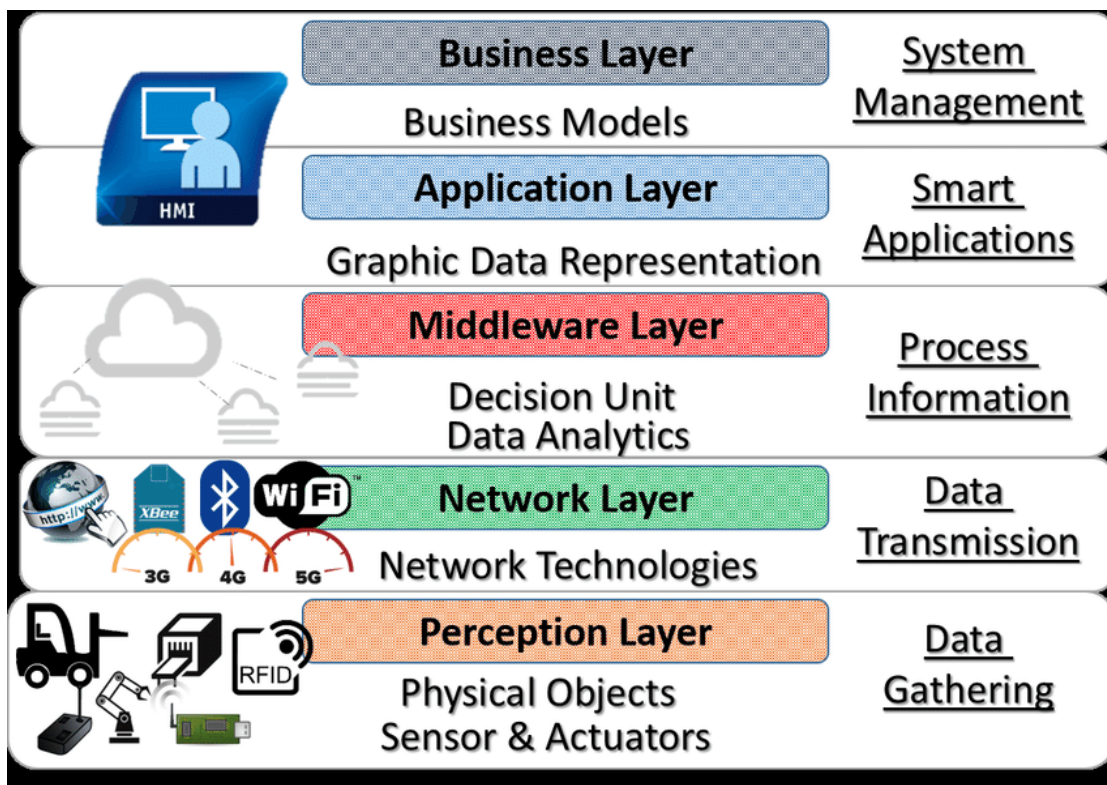
2. Network Layer (Επίπεδο Δικτύου)

Βασική λειτουργία του επιπέδου δικτύου αποτελεί η ορθή και γρήγορη δρομολόγηση και μετάδοση των πακέτων σε ένα δίκτυο. Στο επίπεδο αυτό περιλαμβάνονται τεχνολογίες δικτύων, όπως ασύρματα ή ενσύρματα δίκτυα και τοπικά δίκτυα (LAN). Τα κυριότερα μέσα, για τη μετάδοση των πληροφοριών, είναι τα FTTx, 3G/4G, WiFi, Ethernet, Bluetooth, ZigBee και η τεχνολογία υπεράυθρων (infrared). Επιπλέον, στο επίπεδο αυτό μπορεί να πραγματοποιηθεί η αποθήκευση και η επεξεργασία ενός μεγάλου αριθμού δεδομένων.

3. Application Layer (Επίπεδο Εφαρμογής)

Το επίπεδο εφαρμογής είναι υπεύθυνο για την επεξεργασία των ληφθέντων δεδομένων από το επίπεδο δικτύου. Μέσω διαφόρων εφαρμογών, ο χρήστης μπορεί να έρθει σε επαφή με τα έξυπνα αντικείμενα που παρέχει το επίπεδο αντίληψης. Επιπλέον, το επίπεδο εφαρμογής περιλαμβάνει και το ενδιάμεσο λογισμικό (middleware), το οποίο αποτελεί τον ενδιάμεσο συνδετικό κρίκο μεταξύ των δύο επιπέδων, της εφαρμογής και του δικτύου. Το middleware μπορεί να αποτελέσει ξεχωριστό επίπεδο.

Τα τρία επίπεδα που αναλύθηκαν είναι βασικά σε όλες τις αρχιτεκτονικές. Παρ'όλα αυτά, ορισμένοι προσθέτουν δύο επιπλέον επίπεδα: επίπεδο ενδιάμεσου λογισμικού και επίπεδο επιχείρησης (Εικόνα 1) [7].



Εικόνα 1 Τα επίπεδα της αρχιτεκτονικής του IoT

4. Middleware Layer (Επίπεδο Ενδιάμεσου Λογισμικού)

Το επίπεδο αυτό συνδέει το επίπεδο δικτύου με το επίπεδο εφαρμογής. Στην ουσία, αποτελεί το λογισμικό, το οποίο επιτρέπει στις εφαρμογές την πρόσβαση στα δεδομένα που παρέχουν

τα έξυπνα αντικείμενα. Στο επίπεδο αυτό πραγματοποιούνται πολλές λειτουργίες επεξεργασίας και αποθήκευσης δεδομένων. Οι πλατφόρμες λογισμικού, αποτελούν ένα τμήμα αυτού του επιπέδου. Ο κύριος στόχος είναι η επίτευξη μιας αυτόνομης διαδικασίας λήψης αποφάσεων, η οποία θα στέλνει εντολές ενεργοποίησης πίσω στα φυσικά αντικείμενα, προκειμένου να εκτελούν ενέργειες που θα επηρεάζουν τις συνολικές συνθήκες του φυσικού περιβάλλοντος. Οι πληροφορίες που συλλέγονται/αναλύονται μπορούν να παρουσιαστούν σε έναν τελικό χρήστη χρησιμοποιώντας το επίπεδο εφαρμογής.

5. Business Layer (Επίπεδο Επιχείρησης)

Το κέρδος πάντα αποτελούσε βασικό στόχο των εταιρειών και γι' αυτό το επίπεδο αυτό ασχολείται με την οικονομική διαχείριση των παρεχόμενων υπηρεσιών. Οι επιχειρήσεις, μέσω των εφαρμογών τους, παρέχουν πληροφορίες και δεδομένα, παίρνοντας ως αντάλλαγμα ένα χρηματικό αντίτιμο.

1.4 Γενικά του 5G

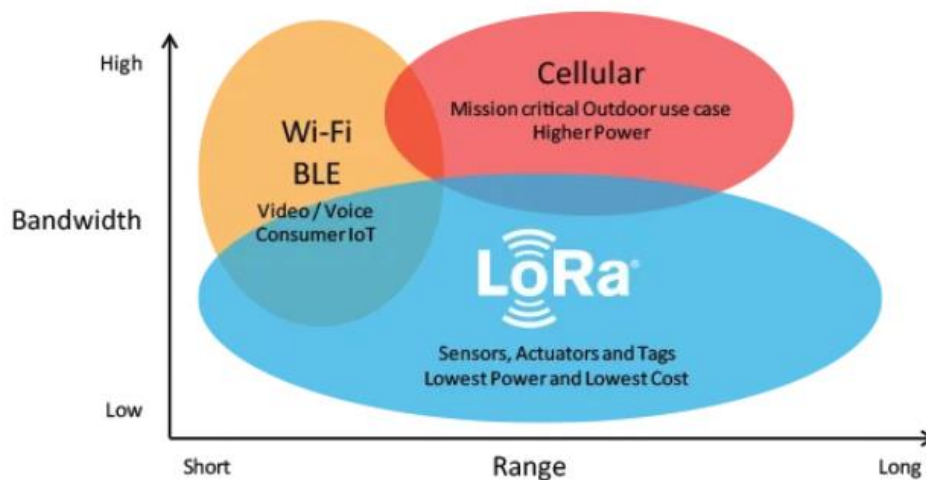
Η 5η γενιά δικτύων κινητής τηλεφωνίας (5G) αποτελεί το επόμενο βήμα στον τομέα των τηλεπικοινωνιών, υπερβαίνοντας σημαντικά τις δυνατότητες των σημερινών δικτύων. Η τεχνολογία 5G θα καλύψει τις ανάγκες που προκύπτουν από τις συνεχώς αυξανόμενες απαιτήσεις των χρηστών του δικτύου κινητής τηλεφωνίας, καθώς και των εφαρμογών του IoT. Σε σύγκριση με τα υπάρχοντα δίκτυα κινητής τηλεφωνίας, το 5G θα μπορεί να υποστηρίξει πολύ περισσότερα τερματικά με πολύ υψηλότερους ρυθμούς δεδομένων, εξαιρετικά χαμηλό λανθάνοντα χρόνο και πολύ υψηλή αξιοπιστία. Με αυτόν τον τρόπο, το 5G θα εξασφαλίσει υψηλή ποιότητα εξυπηρέτησης για τους χρήστες και θα επιτρέψει επίσης εξαιρετικά αξιόπιστη μαζική επικοινωνία μεταξύ συσκευών. Για αυτό λοιπόν και μπορεί να στηρίξει το IoT.

Με το νέο δίκτυο 5G ως δίκτυο υπηρεσιών, θέματα που σχετίζονται με τα ηλεκτρομαγνητικά πεδία (EMFs) αποτελούν αντικείμενο μεγάλου ερευνητικού ενδιαφέροντος, καθώς η εισαγωγή νέων πηγών εκπομπής που λειτουργούν παράλληλα με τις ήδη υπάρχουσες τεχνολογίες κινητής τηλεφωνίας 2G / 3G / 4G, εγείρει ανησυχίες σχετικά με την υπέρβαση των αποδεκτών ορίων έκθεσης στην ηλεκτρομαγνητική ακτινοβολία.

Με το 5G, τα δεδομένα που μεταδίδονται μέσω ασύρματων ευρυζωνικών συνδέσεων μπορούν να ταξιδέψουν με πολύ μεγάλες ταχύτητες, με πιθανές μέγιστες ταχύτητες έως και 20 gigabits ανά δευτερόλεπτο (Gbps) κατά ορισμένες εκτιμήσεις. Αυτές οι ταχύτητες υπερβαίνουν τις ταχύτητες του καλωδίου και προσφέρουν καθυστέρηση 1 χιλιοστού του δευτερολέπτου (ms) ή μικρότερη, η οποία είναι χρήσιμη για εφαρμογές που απαιτούν ανατροφοδότηση σε πραγματικό χρόνο. Το 5G θα επιτρέψει μια απότομη αύξηση του όγκου δεδομένων που μεταδίδονται μέσω ασύρματων συστημάτων λόγω του πιο διαθέσιμου εύρους ζώνης και της προηγμένης τεχνολογίας κεραιάς [8].

1.5 Γενικά του LoRa

Η τεχνολογία ασύρματης μεγάλης εμβέλειας (LoRa) είναι μια τεχνολογία ασύρματης επικοινωνίας ευρείας περιοχής χαμηλής κατανάλωσης δεδομένων, που είναι διαφορετική από το πρότυπο Wi-Fi και προέρχεται από την τεχνολογία Chirp Spread Spectrum (CSS) [9]. Κωδικοποιεί πληροφορίες για ραδιοκύματα χρησιμοποιώντας παλμούς κεληδίσματος (chirp pulses) - παρόμοιους με τον τρόπο που επικοινωνούν τα δελφίνια και οι νυχτερίδες. Η διαμορφωμένη μετάδοση LoRa είναι ανθεκτική σε διαταραχές και μπορεί να ληφθεί σε μεγάλες αποστάσεις. Αναπτύχθηκε από την Cycleo της Γκρενόμπλ της Γαλλίας και αποκτήθηκε από την Semtech, το ιδρυτικό μέλος της “Συμμαχίας LoRa” (LoRa Alliance) και είναι κατοχυρωμένο με δίπλωμα ευρεσιτεχνίας [10][11]. Το LoRa τρέχει στο ασύρματο φάσμα 900 MHz, παρέχοντας πρόσβαση στα δίκτυα LoRaWAN. Το LoRaWAN είναι ένα πρωτόκολλο επιπέδου Media Access Control (MAC) χτισμένο πάνω από τη διαμόρφωση LoRa. Είναι ένα επίπεδο λογισμικού που ορίζει τον τρόπο με τον οποίο οι συσκευές χρησιμοποιούν το υλικό LoRa, για παράδειγμα όταν μεταδίδουν, και τη μορφή των μηνυμάτων που χρησιμοποιούν. Το LoRaWAN είναι κατάλληλο για τη μετάδοση ωφέλιμων φορτίων μικρού μεγέθους (όπως δεδομένα αισθητήρων) σε μεγάλες αποστάσεις. Το παρακάτω σχήμα δείχνει ορισμένες τεχνολογίες πρόσβασης που μπορούν να χρησιμοποιηθούν για ασύρματη μετάδοση δεδομένων και το αναμενόμενο εύρος μετάδοσης σε σχέση με το εύρος ζώνης.



Εικόνα 2 ranges vs. bandwidth [13]

Το LoRa έχει σχεδιαστεί για εφαρμογές IoT και οι συσκευές που λειτουργούν σε αυτό μπορούν να έχουν μπαταρίες που διαρκούν έως και επτά χρόνια. Σε ένα σύστημα LoRa, οι ασύρματοι αισθητήρες συλλέγουν δεδομένα από σωλήνες και αντλίες σε πραγματικό χρόνο και στέλνουν τις πληροφορίες μέσω ασύρματων πυλών (wireless gateways). Από εκεί, τα δεδομένα πηγαίνουν σε έναν διακομιστή δικτύου (network server) και στη συνέχεια σε ένα κέντρο δεδομένων, είτε στο cloud, είτε εντός της εγκατάστασης.

Το LoRa της Semtech είναι μια ασύρματη πλατφόρμα μεγάλης εμβέλειας, χαμηλής ισχύος που έχει γίνει η de facto ασύρματη πλατφόρμα του Internet of Things (IoT). Τα ασύρματα προϊόντα LoRa αναπτύχθηκαν από τη Semtech και συνδέουν δισεκατομμύρια αισθητήρες, ετικέτες (tags) και ενεργοποιητές (actuators) σήμερα σε εφαρμογές βιομηχανικών

επιχειρήσεων και καταναλωτών παγκοσμίως. Η τεχνολογία LoRa ενισχύεται από την ταχέως αναπτυσσόμενη συμμαχία LoRa (LoRa alliance). Ένα ισχυρό παγκόσμιο οικοσύστημα εταιρειών τηλεπικοινωνιών, κατασκευαστών εξοπλισμού, ολοκληρωμένων συστημάτων, κατασκευαστών αισθητήρων και εταιρειών ημιαγωγών οδήγησε στην υιοθέτηση του ανοιχτού κώδικα πρωτοκόλλου επικοινωνίας LoRaWAN. Η συμμαχία LoRa, που δημιουργήθηκε πριν από μερικά χρόνια, έχει πλέον πάνω από 500 παγκόσμια μέλη. Αυτή η μεγάλη ανάπτυξη επέτρεψε την υιοθέτηση του LoRa σε βασικές περιοχές σε όλο τον κόσμο. Οι φορείς εκμετάλλευσης σε μεγάλες παγκόσμιες αγορές εφαρμόζουν περισσότερα όταν τα δίκτυα υποστηρίζουν τη συνδεσιμότητα IoT συσκευών που βασίζονται στο LoRa.

LoRa και LoRaWAN: Πώς η τεχνολογία βοηθά τις έξυπνες πόλεις

Τα έξυπνα δίκτυα στις πόλεις τροφοδοτούν τα πάντα, από αυτόνομα οχήματα μέχρι παρακολούθηση της κυκλοφορίας έως και έξυπνα φώτα δρόμων. Η τεχνολογία LoRa προσφέρει πραγματικά λύσεις δικτύων με πολύ χαμηλότερο κόστος για κτίρια διαχείρισης εγκαταστάσεων. Υπάρχουν επίσης περισσότερα βιομηχανικά υλικά όπως, υγρασία εδάφους σε αγροκτήματα, πάρκινγκ, αισθητήρες μετρητή νερού, ηλεκτρικός μετρητή, που απλά συνδέονται και λειτουργούν εύκολα και γρήγορα για την επίλυση οποιασδήποτε περίπτωσης χρήσης.

Οι συσκευές και τα δίκτυα LoRa όπως το LoRaWAN επιτρέπουν στις έξυπνες εφαρμογές IoT να επιλύουν μερικές από τις μεγαλύτερες προκλήσεις που αντιμετωπίζει ο πλανήτης μας: διαχείριση ενέργειας (energy management), μείωση φυσικών πόρων (natural resource reduction), έλεγχος της ρύπανσης (pollution control), αποδοτικότητα υποδομής (infrastructure efficiency) και πρόληψη καταστροφών (disaster prevention). Οι συσκευές LoRa της Semtech έχουν συγκεντρώσει αρκετές εκατοντάδες γνωστές περιπτώσεις χρήσεων για έξυπνες πόλεις, σπίτια και κτίρια, κοινότητες, μετρήσεις, εφοδιαστική αλυσίδα και logistics, γεωργία και πολλά άλλα. Με εκατοντάδες εκατομμύρια συσκευές συνδεδεμένες σε δίκτυα, σε περισσότερες από 100 χώρες, και να αυξάνονται, η LoRa δημιουργεί έναν πιο έξυπνο πλανήτη, παρέχοντας κάλυψη εκεί όπου οι παραδοσιακές λύσεις Wi-Fi δεν μπορούν. Η μεγάλη εμβέλεια που υποστηρίζει το LoRaWAN μειώνει τόσο το κόστος των συσκευών, όσο και των gateways.

Επιπλέον, η Cisco έχει συνεργαστεί με πόλεις σε όλο τον κόσμο σε έργα έξυπνου νερού χρησιμοποιώντας την τεχνολογία LoRaWAN. Ένα παράδειγμα αποτελεί μια μικρή πόλη στην Πορτογαλία που ονομάζεται Caldas da Rainha όπου, παρακολουθείται η ποιότητα του νερού και πόσο νερό πίνουν οι κάτοικοι, φτάνοντας σε μείωση του κόστους διαχείρισης του νερού κατά 33%, του συνολικού κόστους απορριμμάτων κατά 67% και μείωση του χρόνου εύρεσης θέσης στάθμευσης στο μισό. Οι πόλεις αποτελούν σπουδαίο φορέα αλλαγής γύρω από τις τεχνολογίες IoT και έχουν στην πραγματικότητα ένα ευρύ φάσμα περιπτώσεων χρήσης που μπορούν να καλυφθούν από το LoRaWAN.

Παρατηρώντας το παραπάνω παράδειγμα, φαίνεται πως μια ολόκληρη πόλη των 250 km² πλέον καλύπτεται πλήρως από ένα τυποποιημένο δίκτυο LoRaWAN με πύλες Cisco IXM.

1.6 Γενικά του NB-IoT

Το NB-IoT (Narrowband IoT) είναι μια τεχνολογία ασύρματης επικοινωνίας χαμηλής κατανάλωσης, ευρείας περιοχής (LPWA) που έχει σχεδιαστεί ειδικά για το Internet of Things (IoT). Είναι ένας τύπος τεχνολογίας κυψελοειδούς δικτύου που λειτουργεί σε αδειοδοτημένο φάσμα και παρέχει έναν οικονομικά αποδοτικό, ασφαλή και αξιόπιστο τρόπο για τις συσκευές IoT να επικοινωνούν σε μεγάλες αποστάσεις.

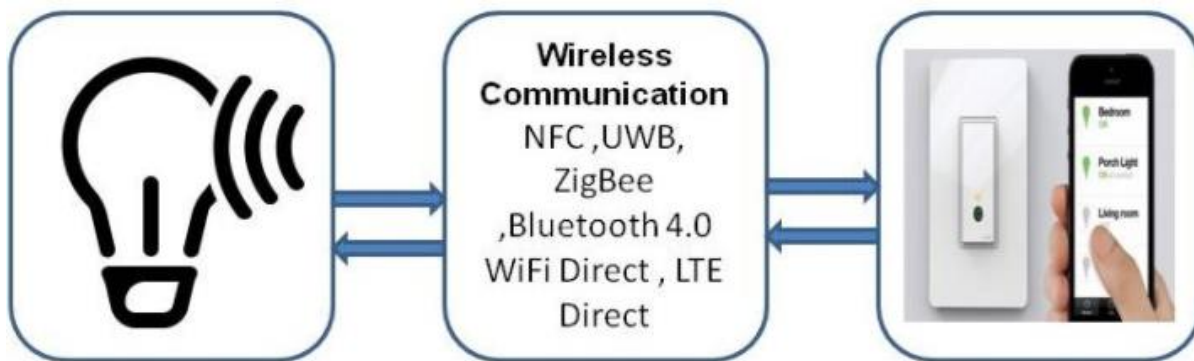
Το NB-IoT έχει σχεδιαστεί για να παρέχει επικοινωνίες χαμηλού ρυθμού δεδομένων (low-data rate communication), καθιστώντας το κατάλληλο για εφαρμογές IoT που απαιτούν σπάνιες μεταφορές δεδομένων, όπως έξυπνους μετρητές, παρακολούθηση περιουσιακών στοιχείων και τηλεχειρισμό. Η τεχνολογία είναι βελτιστοποιημένη για χαμηλή κατανάλωση ενέργειας, η οποία επιτρέπει στις συσκευές να λειτουργούν για εκτεταμένες χρονικές περιόδους με μία μόνο φόρτιση της μπαταρίας. Το NB-IoT λειτουργεί σε ένα φάσμα συχνοτήτων στενής ζώνης, το οποίο του επιτρέπει να υποστηρίζει μεγάλο αριθμό συσκευών σε μία μόνο κυψέλη και παρέχει καλύτερη κάλυψη σε δυσπρόσιτες περιοχές. Αυτό το καθιστά ιδανικό για εφαρμογές IoT σε αγροτικές ή απομακρυσμένες περιοχές, όπου ενδέχεται να μην είναι διαθέσιμες άλλες τεχνολογίες επικοινωνίας. Το NB-IoT έχει επίσης σχεδιαστεί για να είναι εξαιρετικά ασφαλές, με ενσωματωμένους μηχανισμούς κρυπτογράφησης και ελέγχου ταυτότητας για την αποτροπή μη εξουσιοδοτημένης πρόσβασης στα δεδομένα. Η τεχνολογία είναι επίσης επεκτάσιμη, καθιστώντας την κατάλληλη για μεγάλης κλίμακας αναπτύξεις IoT. Συνολικά, το NB-IoT είναι μια οικονομικά αποδοτική και αξιόπιστη τεχνολογία επικοινωνίας που είναι κατάλληλη για εφαρμογές IoT που απαιτούν επικοινωνίες χαμηλού ρυθμού δεδομένων και μεγάλη διάρκεια ζωής της μπαταρίας.

1.7 Τύποι και Πρωτόκολλα Επικοινωνίας

Από λειτουργικής πλευράς, είναι χρήσιμο να σκεφτούμε πως οι συσκευές του IoT συνδέονται και επικοινωνούν σε σχέση με τα τεχνικά μοντέλα επικοινωνίας. Παρακάτω παρουσιάζονται κάποια μοντέλα επικοινωνίας που χρησιμοποιούνται από συσκευές IoT και τα βασικά χαρακτηριστικά τους [12][13]:

1.7.1 Device-to-Device Επικοινωνία

Το μοντέλο επικοινωνίας device-to-device αντιπροσωπεύει δύο ή περισσότερες συσκευές που συνδέονται άμεσα και επικοινωνούν μεταξύ τους, χωρίς τη χρήση ενός ενδιάμεσου application server. Αυτές οι συσκευές επικοινωνούν σε πολλούς τύπους δικτύων, συμπεριλαμβανομένων των δικτύων IP ή το Internet. Συχνά, ωστόσο, αυτές οι συσκευές χρησιμοποιούν πρωτόκολλα όπως το Bluetooth-Wave, ή ZigBee για την καθιέρωση της μεταξύ τους επικοινωνίας, όπως φαίνεται στην Εικόνα 3 [14].

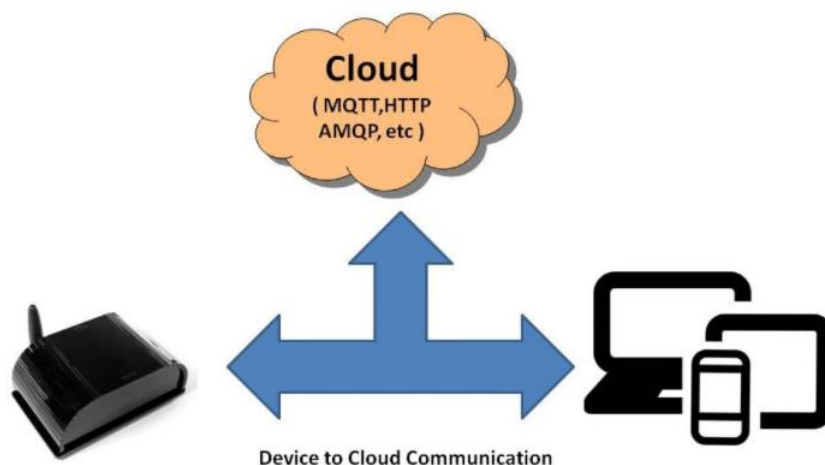


Εικόνα 3 Παράδειγμα μοντέλου επικοινωνίας από συσκευή σε συσκευή

1.7.2 Device-to-Cloud Επικοινωνία

Σε ένα μοντέλο επικοινωνίας μεταξύ συσκευής και υπολογιστικού νέφους, (device-to-cloud) η συσκευή IoT συνδέεται απευθείας με μια υπηρεσία υπολογιστικού νέφους του Διαδικτύου, όπως ένας πάροχος υπηρεσιών εφαρμογών, για την ανταλλαγή δεδομένων και τον έλεγχο της κίνησης μηνυμάτων. Αυτή η προσέγγιση συχνά εκμεταλλεύεται τους υπάρχοντες μηχανισμούς επικοινωνίας, όπως οι παραδοσιακές ενσύρματες συνδέσεις Ethernet ή Wi-Fi, για τη δημιουργία μιας σύνδεσης μεταξύ της συσκευής και του δικτύου IP, το οποίο τελικά συνδέεται με την υπηρεσία νέφους. Όπως φαίνεται στην Εικόνα 4.

Αυτό το μοντέλο επικοινωνίας χρησιμοποιείται από κάποια δημοφιλή καταναλωτικά προϊόντα (π.χ. SmartTV). Τα προϊόντα αυτά συνδέονται στο Internet και μεταξύ άλλων μεταδίδουν δεδομένα σε μια cloud υπηρεσία για ανάλυση ή/και αποθήκευση, παρέχουν στο χρήστη απόμακρο έλεγχο της συσκευής μέσω κάποιου smartphone ή web interface και κάνουν αναβαθμίσεις λογισμικού.



Εικόνα 4 Παράδειγμα μοντέλου επικοινωνίας από συσκευή σε νέφος

1.7.3 Device-to-Gateway Επικοινωνία

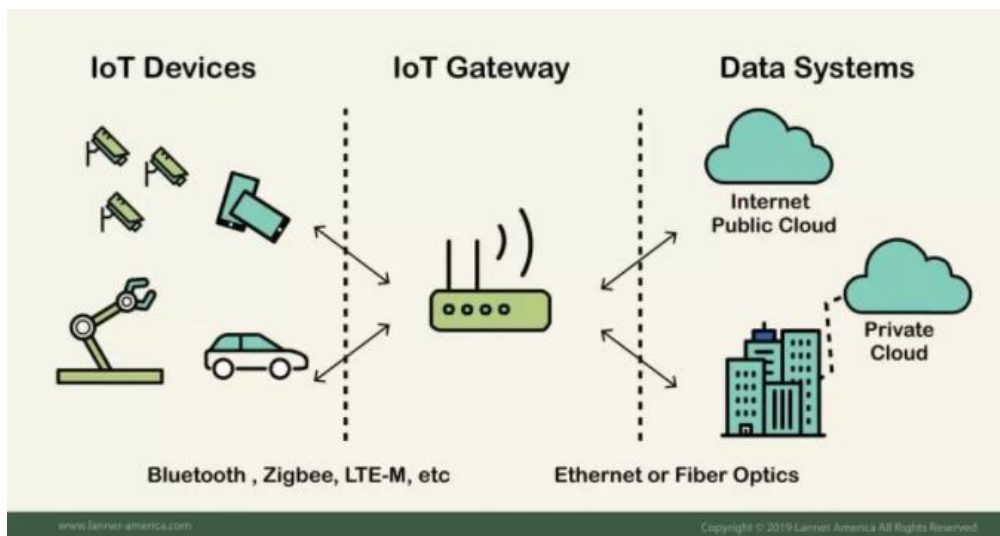
Αρχικά δίνεται ο όρος gateway. Είναι ένας κόμβος του δικτύου που συνδέει δύο ή περισσότερα δίκτυα που χρησιμοποιούν διαφορετικά πρωτόκολλα. Τα gateways μπορούν να λάβουν διάφορες μορφές (router ή υπολογιστές) και μπορούν να εκτελέσουν μια ποικιλία εργασιών (traffic filtering, proxies, protocol translations).

Στο μοντέλο Device-to-Gateway, οι συσκευές IoT συνδέονται βασικά σε μια ενδιάμεση συσκευή για να αποκτήσουν πρόσβαση σε μια υπηρεσία cloud. Αυτό το μοντέλο περιλαμβάνει συχνά λογισμικό εφαρμογών που λειτουργεί σε μια τοπική συσκευή πύλης (όπως ένα smartphone ή ένας "κόμβος"), η οποία ενεργεί ως ενδιάμεσος μεταξύ μιας συσκευής IoT και μιας υπηρεσίας cloud.

Αυτή η πύλη μπορεί να παρέχει ασφάλεια και άλλες λειτουργίες, όπως η μετάφραση δεδομένων ή πρωτοκόλλων. Εάν η πύλη επιπέδου εφαρμογής είναι ένα smartphone, αυτό το λογισμικό εφαρμογής μπορεί να έχει τη μορφή μιας εφαρμογής που συνδυάζεται με τη συσκευή IoT και επικοινωνεί με μια υπηρεσία cloud.

Αυτό θα μπορούσε να είναι μια συσκευή γυμναστικής που συνδέεται με το σύννεφο μέσω μιας εφαρμογής smartphone, όπως η Nike+ [15], ή εφαρμογές οικιακού αυτοματισμού που περιλαμβάνουν συσκευές που συνδέονται σε έναν κόμβο, όπως το οικοσύστημα SmartThings της Samsung [16].

Οι συσκευές πύλης μπορούν επίσης να γεφυρώσουν το χάσμα διαλειτουργικότητας μεταξύ συσκευών που επικοινωνούν με διαφορετικά πρότυπα. Για παράδειγμα, οι πομποδέκτες Z-Wave και Zigbee της SmartThings μπορούν να επικοινωνούν και με τις δύο οικογένειες συσκευών.

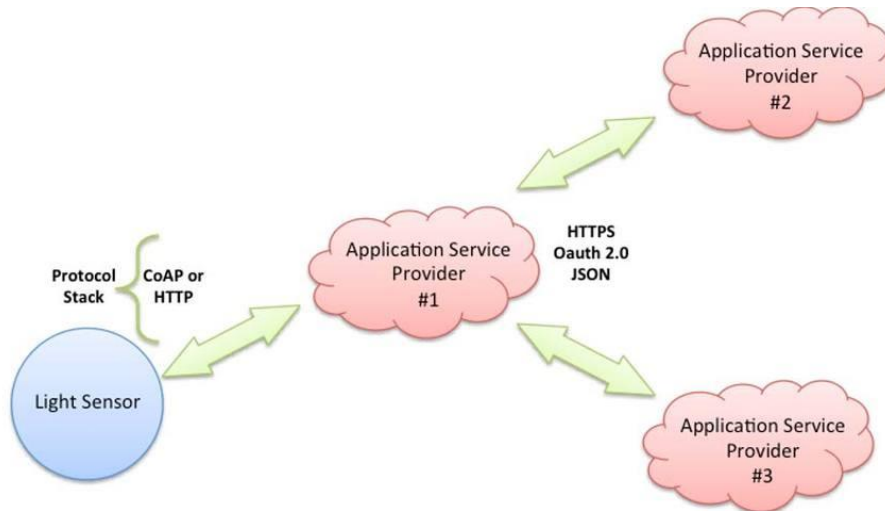


Εικόνα 5 Παράδειγμα μοντέλου επικοινωνίας από συσκευή σε πύλη

1.7.4 Back-End Data-Sharing Επικοινωνία

Ο διαμοιρασμός δεδομένων στο back-end επεκτείνει ουσιαστικά το μοντέλο επικοινωνίας μεταξύ μιας συσκευής και του cloud, ώστε να μπορούν να έχουν πρόσβαση σε συσκευές IoT και δεδομένα αισθητήρων εξουσιοδοτημένα τρίτα μέρη. Σύμφωνα με αυτό το μοντέλο, οι χρήστες μπορούν να εξάγουν και να αναλύουν δεδομένα έξυπνων αντικειμένων από μια υπηρεσία cloud σε συνδυασμό με δεδομένα από άλλες πηγές και να τα στέλνουν σε άλλες υπηρεσίες για συγκέντρωση και ανάλυση. Η προσέγγιση αυτή, αποτελεί μια προέκταση του μοντέλου επικοινωνίας device-to-cloud, στο οποίο οι IoT συσκευές ανεβάζουν τα δεδομένα μόνο σε έναν πάροχο υπηρεσιών. Το μοντέλο back-end data-sharing υποδηλώνει ότι

χρειάζονται cloud APIs και μια πιο συνολική προσέγγιση των cloud υπηρεσιών για την επίτευξη της αμοιβαίας χρήσης των δεδομένων των smart devices που φιλοξενούνται στο cloud. Αυτό απεικονίζεται στην Εικόνα 6.



Εικόνα 6 Μοντέλο κοινής χρήσης δεδομένων back-end

1.7.5 Πρωτόκολλα Επικοινωνίας

Από τα παραπάνω έγινε εμφανές πως το IoT σχετίζεται με πληθώρα πρωτοκόλλων επικοινωνίας, τα οποία κατηγοριοποιούνται συνήθως με βάση το μοντέλο αναφοράς OSI, όπως φαίνεται και στην Εικόνα 7.

Application Layer	CoAP	HTTP	MQTT	Resource Directory	Cache	C2C Proxy	H2C Proxy	Replica Manager
Transport Layer	UDP	TCP						
Network Layer	IP			Border Router				
Physical/Link Layer	IEEE 802.15.4	...	IEEE 802.11					
	Protocol stack			Functional plane				

Εικόνα 7 Το Protocol Stack του IoT

Περεταίρω ανάλυση των παραπάνω πρωτοκόλλων, καθώς και γενικά των σημαντικότερων πρωτοκόλλων, που σχετίζονται με διαδίκτυο εξετάζονται στην Ενότητα 1.8 που ακολουθεί.

1.8 Πρωτόκολλα, Διαδίκτυο και APIs

Το Διαδίκτυο των Πραγμάτων (IoT) περιλαμβάνει επικοινωνία μεταξύ συσκευών, αισθητήρων και άλλων εξαρτημάτων και ως εκ τούτου απαιτεί μια ποικιλία τεχνολογιών και πρωτοκόλλων επικοινωνίας για να λειτουργήσει αποτελεσματικά. Μερικοί από τους πιο συχνά χρησιμοποιούμενους τύπους και πρωτόκολλα επικοινωνίας στο IoT παρουσιάζονται παρακάτω.

1.8.1 TCP, UDP και IP

Το **TCP** (Transmission Control Protocol) αποτελεί ένα από τα δυο βασικά πρωτόκολλα του επιπέδου μεταφοράς στο μοντέλο OSI. Παρέχει αξιόπιστη αμφίδρομη επικοινωνία, ενώ διέπεται από τα ακόλουθα χαρακτηριστικά:

- Μεταφορά δεδομένων.
- Σύνδεση δύο τερματικών (endpoints).
- Τα δεδομένα να παραλαμβάνονται όπως στάλθηκαν.
- Αποστολή δεδομένων σε ακολουθία.
- Αξιόπιστη εκκίνηση και τερματισμός.

Το **UDP** (User Datagram Protocol) είναι το άλλο βασικό πρωτόκολλο στο επίπεδο μεταφοράς που χρησιμοποιείται στο διαδίκτυο. Σε σύγκριση με το TCP, το UDP είναι λιγότερο πολύπλοκο, αλλά ταυτόχρονα και λιγότερο αξιόπιστο όσον αφορά τη λήψη των πακέτων που αποστέλλονται στο άλλο τερματικό. Το πρωτόκολλο αυτό χρησιμοποιείται κυρίως σε περιπτώσεις όπου απαιτείται μεγάλη ταχύτητα στην αποστολή των δεδομένων, όπως η ζωντανή μετάδοση ομιλίας ή βίντεο, με στόχο να μην διακοπεί η συνεχής ροή των δεδομένων.

Το **IP** (Internet Protocol) παρέχει έναν μηχανισμό επικοινωνίας για υπολογιστές και συσκευές. Το πρωτόκολλο αυτό διαχωρίζει τον κάθε συνδεδεμένο στο διαδίκτυο υπολογιστή ή συσκευή μέσω ενός μοναδικού αριθμού γνωστού ως IP address (διεύθυνση IP). Τα TCP ή UDP πακέτα περικλείονται συνήθως σε πακέτα IP.

1.8.2 HTTP

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP) είναι ένα ευρέως διαδεδομένο πρωτόκολλο επικοινωνίας που χρησιμοποιείται στο IoT για τη μετάδοση δεδομένων μεταξύ συσκευών και του cloud. Υλοποιεί μια τεχνική αίτησης-απάντησης (request-response), χρησιμοποιώντας μεθόδους που θα αναλυθούν στη συνέχεια. Υποστηρίζει τη μεταφορά και TCP και UDP πακέτων.

Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού (Web browsers), με στόχο τη διασύνδεση του διακομιστή (server) και του πελάτη (client), επιτρέποντας τη μεταξύ τους μεταφορά δεδομένων. Οι βασικοί μέθοδοι που χρησιμοποιούνται ευρέως είναι:

- **POST**: Δημιουργία νέου πόρου.
- **GET**: Ανάκτηση πληροφορίας.
- **PUT**: Ενημέρωση του περιεχομένου του πόρου.
- **DELETE**: Διαγραφή ενός πόρου.

1.8.3 MQTT

Το MQTT (Message Queue Telemetry Transport) είναι ένα απλό και ελαφρύ (lightweight) πρωτόκολλο επικοινωνίας δημοσίευσης/εγγραφής που είναι κατάλληλο για εφαρμογές IoT λόγω της αποτελεσματικής χρήσης πόρων δικτύου. Από το πρώτο κίολας έτος δημιουργίας του άρχισε να χρησιμοποιείται ευρέως σε ποικίλες βιομηχανικές εφαρμογές. Είναι βασισμένο στην τεχνική publish-subscribe, η οποία διευκολύνει σε μεγάλο βαθμό το σχεδιασμό IoT εφαρμογών. Οι αρχές σχεδίασης που ακολουθεί το πρωτόκολλο είναι η ελαχιστοποίηση του εύρους ζώνης του δικτύου, καθώς και των απαιτούμενων πόρων (ενέργειας, μνήμης). Παράδειγμα τέτοιου πρωτοκόλλου αποτελεί το Facebook messenger.

Publish-Subscribe

Η τεχνική Publish-Subscribe (PubSub) αποτελεί ένα πρότυπο αποστολής μηνυμάτων το οποίο εμπεριέχει τρεις ρόλους:

1. *Publishers*: Οι Publishers αποτελούν αυτούς που αποστέλλουν τα μηνύματα. Κατά την αποστολή μηνυμάτων δεν καθορίζουν ποιος θα είναι ο παραλήπτης του μηνύματος, αντ' αυτού το μήνυμα αποστέλλεται σε κάποιο θέμα (topic) στον Broker.
2. *Subscribers*: Οι Subscribers αποτελούν τους παραλήπτες των μηνυμάτων. Ένας subscriber δηλώνει στον Broker το ενδιαφέρον του σε ένα συγκεκριμένο θέμα. Οποιοδήποτε μήνυμα σταλεί στο συγκεκριμένο θέμα, αποστέλλεται αυτομάτως και στους subscribers του θέματος.
3. *Broker*: Ο Broker στην ουσία αποτελεί τον ενδιάμεσο κόμβο που συνδέει όλους τους χρήστες (publishers και subscribers). Είναι ένας δρομολογητής (router) ο οποίος είναι ο υπεύθυνος να επιτρέπει την εγγραφή (subscription) των subscribers σε θέματα, και να δρομολογεί τα εισερχόμενα γεγονότα (events) από τους publishers στους subscribers ανάλογα με το θέμα. Παράδειγμα αποτελεί ο MQTT Broker.

Σύγκριση MQTT-HTTP

Αν και συχνά γίνεται σύγκριση του MQTT με άλλα κοινά πρωτόκολλα, η πιο χρήσιμη σύγκριση είναι με το HTTP (Πίνακας 1). Και τα δυο αυτά πρωτόκολλα είναι βασισμένα στο TCP/IP, αλλά το HTTP είναι πολύ διαδεδομένο και σχεδόν όλες οι συσκευές με TCP/IP το διαθέτουν [17].

	MQTT	HTTP
Design orientation	Data centric	Document centric
Pattern	Publish/subscribe	Request/response
Complexity	Simple	More complex
Message size	Small, with a compact binary header just two bytes in size	Larger, partly because status detail is text-based
Service levels	Three quality of service settings	All messages get the same level of service
Extra libraries	Libraries for C (30 KB) and Java (100 KB)	Depends on the application (JSON, XML), but typically not small
Data distribution	Supports 1 to zero, 1 to 1, and 1 to n	1 to 1 only

Πίνακας 1 Σύγκριση μεταξύ MQTT και HTTP

1.8.4 CoAP

CoAP: Το CoAP (Constrained Application Protocol) είναι ένα πρωτόκολλο επικοινωνίας RESTful που έχει σχεδιαστεί ειδικά για συσκευές IoT με περιορισμένες δυνατότητες επεξεργασίας και αποθήκευσης.

1.8.5 XML

Η XML είναι μια γλώσσα σήμανσης παρόμοια με την HTML, αλλά χωρίς προκαθορισμένα tags για χρήση. Αντ' αυτού, ο καθένας μπορεί να ορίσει τις δικές του ετικέτες σχεδιασμένες ειδικά για τις ανάγκες του. Η XML σχεδιάστηκε για να αποθηκεύει και να μεταφέρει δεδομένα από μηχανήματα (machines). Η XML καθορίζει ένα σύνολο κανόνων για την κωδικοποίηση εγγράφων και τη μετατροπή τους σε αναγνώσιμη μορφή και από τον άνθρωπο και από τη μηχανή. Αυτός είναι ένας ισχυρός τρόπος αποθήκευσης δεδομένων σε μορφή που μπορεί να αποθηκευτεί, να αναζητηθεί και να διαμοιραστεί. Το πιο σημαντικό είναι ότι, δεδομένου ότι η θεμελιώδης μορφή της XML είναι τυποποιημένη, εάν μοιραστείτε ή μεταδώσετε XML σε διάφορα συστήματα ή πλατφόρμες, είτε τοπικά είτε μέσω του διαδικτύου, ο παραλήπτης μπορεί να αναλύσει τα δεδομένα λόγω της τυποποιημένης σύνταξης XML.

1.8.6 JSON

Το JSON (JavaScript Object Notation) είναι μια ελαφριά μορφή δεδομένων που χρησιμοποιείται ευρέως στο Internet of Things (IoT) για την ανταλλαγή δεδομένων μεταξύ συσκευών και εφαρμογών. Το JSON είναι μια μορφή βασισμένη σε κείμενο που διαβάζεται και γράφεται εύκολα και υποστηρίζεται ευρέως από γλώσσες προγραμματισμού, καθιστώντας το μια δημοφιλή επιλογή για εφαρμογές IoT. Στο IoT, το JSON χρησιμοποιείται συχνά για τη μετάδοση δεδομένων μεταξύ συσκευών και του cloud, καθώς και μεταξύ συσκευών και άλλων στοιχείων. Για παράδειγμα, ένας αισθητήρας μπορεί να στείλει δεδομένα στο cloud σε μορφή JSON, όπου μπορούν να αποθηκευτούν, να υποβληθούν σε επεξεργασία και να αναλυθούν. Ομοίως, μια συσκευή μπορεί να λάβει εντολές ή ενημερώσεις διαμόρφωσης από το cloud σε μορφή JSON, τις οποίες μπορεί να αναλύσει και να ενεργήσει. Ένα από τα βασικά πλεονεκτήματα του JSON στο IoT είναι η απλότητα και η ευελιξία του. Το JSON μπορεί να χρησιμοποιηθεί για την αναπαράσταση μιας μεγάλης ποικιλίας data structures (δομών δεδομένων), συμπεριλαμβανομένων πινάκων και (objects) αντικειμένων, καθιστώντας το μια ευέλικτη μορφή για τη μετάδοση και την επεξεργασία δεδομένων IoT. Επιπλέον, το JSON είναι εύκολο να αναλυθεί, πράγμα που σημαίνει ότι οι συσκευές με περιορισμένες δυνατότητες επεξεργασίας και αποθήκευσης μπορούν να χρησιμοποιούν JSON χωρίς να απαιτούν πολλούς πόρους. Αυτό καθιστά το JSON μια καλή επιλογή για συσκευές IoT με περιορισμένο υλικό, όπως αισθητήρες και ενεργοποιητές. Συνολικά, το JSON είναι μια ευρέως χρησιμοποιούμενη και σημαντική μορφή στο IoT, παρέχοντας έναν απλό και ευέλικτο τρόπο ανταλλαγής δεδομένων μεταξύ συσκευών και εφαρμογών και επιτρέποντας την ανάπτυξη και την καινοτομία του οικοσυστήματος IoT.

1.8.7 Online and web services

Οι διαδικτυακές υπηρεσίες και οι υπηρεσίες ιστού αποτελούν σημαντικό μέρος του οικοσυστήματος του Διαδικτύου των Πραγμάτων (IoT), επιτρέποντας την επικοινωνία και την ενοποίηση μεταξύ συσκευών, εφαρμογών και του cloud. Στο IoT, οι διαδικτυακές υπηρεσίες και οι υπηρεσίες Ιστού χρησιμοποιούνται για την παροχή μιας σειράς υπηρεσιών και λειτουργιών, όπως:

- Αποθήκευση και διαχείριση δεδομένων (Data storage and management):

Οι διαδικτυακές υπηρεσίες μπορούν να χρησιμοποιηθούν για την αποθήκευση και τη διαχείριση μεγάλων ποσοτήτων δεδομένων IoT, επιτρέποντας στις συσκευές να εκφορτώνουν αυτά τα δεδομένα στο cloud για επεξεργασία, ανάλυση και μακροπρόθεσμη αποθήκευση

- Επεξεργασία και ανάλυση δεδομένων (Data processing and analysis):

Οι υπηρεσίες Ιστού μπορούν να χρησιμοποιηθούν για την επεξεργασία και ανάλυση δεδομένων IoT σε πραγματικό χρόνο, επιτρέποντας τη δημιουργία πληροφοριών και τον εντοπισμό προτύπων και τάσεων.

- Απομακρυσμένη διαχείριση συσκευών (Remote device management):

Οι διαδικτυακές υπηρεσίες και οι υπηρεσίες Ιστού μπορούν να χρησιμοποιηθούν για την απομακρυσμένη διαχείριση συσκευών IoT, συμπεριλαμβανομένης της ενημέρωσης υλικολογισμικού, της αλλαγής διαμορφώσεων και της παρακολούθησης της κατάστασης της συσκευής.

- Ανάπτυξη εφαρμογών (Application development and deployment):

Οι υπηρεσίες Ιστού μπορούν να χρησιμοποιηθούν για τη δημιουργία και την ανάπτυξη εφαρμογών IoT, παρέχοντας μια πλατφόρμα στους προγραμματιστές για τη δημιουργία νέων εφαρμογών και υπηρεσιών που μπορούν να χρησιμοποιηθούν για την αλληλεπίδραση και τον έλεγχο συσκευών IoT.

- Ενοποίηση με άλλα συστήματα και υπηρεσίες (Integration with other systems and services):

Οι διαδικτυακές υπηρεσίες μπορούν να χρησιμοποιηθούν για την ενοποίηση συσκευών και δεδομένων IoT με άλλα συστήματα και υπηρεσίες, επιτρέποντας τη δημιουργία νέων και καινοτόμων λύσεων.

Συνολικά, οι διαδικτυακές υπηρεσίες και οι διαδικτυακές υπηρεσίες αποτελούν κρίσιμο στοιχείο του οικοσυστήματος IoT, παρέχοντας την υποδομή και τη λειτουργικότητα που απαιτούνται για την υποστήριξη της ανάπτυξης και της καινοτομίας του IoT και για τη δημιουργία νέων και συναρπαστικών εφαρμογών και υπηρεσιών IoT.

1.8.8 APIs

Τα API (Application Programming Interfaces) διαδραματίζουν βασικό ρόλο στο IoT επιτρέποντας σε διαφορετικές συσκευές και στοιχεία να επικοινωνούν και να ανταλλάσσουν δεδομένα μεταξύ τους. Τα API παρέχουν μια τυποποιημένη διεπαφή επικοινωνίας, επιτρέποντας στις συσκευές να αλληλεπιδρούν μεταξύ τους ανεξάρτητα από τις υποκείμενες τεχνολογίες και τα πρωτόκολλά τους. Στο IoT, τα API χρησιμοποιούνται για να επιτρέψουν την επικοινωνία μεταξύ συσκευών και του cloud, καθώς και μεταξύ συσκευών και άλλων στοιχείων, όπως αισθητήρες και gateways. Τα API παρέχουν επίσης έναν τρόπο στους προγραμματιστές να έχουν πρόσβαση στα δεδομένα και τη λειτουργικότητα των συσκευών IoT και να δημιουργήσουν νέες εφαρμογές και υπηρεσίες πάνω από το οικοσύστημα IoT. Συνολικά, τα πρωτόκολλα και τα API είναι κρίσιμα στοιχεία του IoT, επιτρέποντας την επικοινωνία, την ολοκλήρωση και τη διαλειτουργικότητα μεταξύ συσκευών και εξαρτημάτων και παρέχουν τη βάση για την ανάπτυξη και την καινοτομία του οικοσυστήματος IoT.

1.9 IoT/Sensor networks και έξυπνες εφαρμογές (smart applications)

Το IoT και τα δίκτυα αισθητήρων μεταμορφώνουν τον τρόπο που ζούμε και εργαζόμαστε, επιτρέποντας νέες και καινοτόμες εφαρμογές και υπηρεσίες. Συνδέοντας φυσικές συσκευές, αισθητήρες και συστήματα στο διαδίκτυο, το IoT δημιουργεί ένα τεράστιο δίκτυο πληροφοριών και νοημοσύνης που μπορεί να αξιοποιηθεί για να προσφέρει νέες και πολύτιμες γνώσεις και εμπειρίες.

Τα δίκτυα αισθητήρων διαδραματίζουν κρίσιμο ρόλο στο IoT, παρέχοντας τη βάση για τη συλλογή δεδομένων και την επικοινωνία. Ενσωματώνοντας ένα ευρύ φάσμα αισθητήρων σε συνδεδεμένες συσκευές και συστήματα, τα δίκτυα αισθητήρων παρέχουν μια πλούσια και ποικιλόμορφη ροή δεδομένων που μπορούν να χρησιμοποιηθούν για την προώθηση νέων εφαρμογών και υπηρεσιών.

Οι έξυπνες εφαρμογές είναι το τελικό αποτέλεσμα του IoT και των δικτύων αισθητήρων, αξιοποιώντας τα δεδομένα και την ευφυΐα που δημιουργούνται από συνδεδεμένες συσκευές για να προσφέρουν νέες και καινοτόμες εμπειρίες. Αυτές οι εφαρμογές καλύπτουν ένα ευρύ φάσμα τομέων, από τον οικιακό αυτοματισμό και τη διαχείριση ενέργειας, μέχρι την υγειονομική περίθαλψη, τις μεταφορές και όχι μόνο. Για παράδειγμα, ένα έξυπνο σύστημα διαχείρισης κτιρίου μπορεί να χρησιμοποιήσει δεδομένα αισθητήρων για τη βελτιστοποίηση της χρήσης ενέργειας, τη μείωση του κόστους και τη βελτίωση της βιωσιμότητας. Ομοίως, ένα έξυπνο σύστημα υγειονομικής περίθαλψης μπορεί να χρησιμοποιήσει δεδομένα ασθενών για να παρέχει πιο εξατομικευμένη και αποτελεσματική θεραπεία.

Όπως προαναφέρθηκε παραπάνω, οι πιθανές εφαρμογές του IoT είναι πολλές. Οι εφαρμογές IoT απευθύνονται στις ανάγκες της κοινωνίας και την εξέλιξη των τεχνολογιών. Έξυπνες πόλεις (smart cities), έξυπνη ενέργεια, έξυπνα δίκτυα διανομής ηλεκτρικής ενέργειας, έξυπνη κινητικότητα και μεταφορά, έξυπνα σπίτια, έξυπνα κτήρια, έξυπνη λιανική και Logistics,

έξυπνο εργοστάσιο και έξυπνη βιομηχανοποίηση, έξυπνη γεωργία και εκτροφή. Οι έξυπνες εφαρμογές IoT με έξυπνους αισθητήρες είναι εφαρμογές λογισμικού που αξιοποιούν τα δεδομένα και τη νοημοσύνη που δημιουργούνται από συνδεδεμένους αισθητήρες για να προσφέρουν νέες και καινοτόμες εμπειρίες. Ο συνδυασμός IoT και έξυπνων αισθητήρων επιτρέπει ένα ευρύ φάσμα νέων και πολύτιμων εφαρμογών, όπως:

- Προγνωστική συντήρηση (Predictive maintenance):

Οι έξυπνοι αισθητήρες IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση της απόδοσης διαφόρων συστημάτων και συσκευών και για την παροχή πληροφοριών σε πραγματικό χρόνο για πιθανά προβλήματα. Για παράδειγμα, ένας έξυπνος αισθητήρας σε ένα εργοστασιακό μηχάνημα μπορεί να παρακολουθεί τα μοτίβα των κραδασμών του και να ειδοποιεί το προσωπικό συντήρησης για πιθανά προβλήματα προτού γίνουν σοβαρά προβλήματα.

- Περιβαλλοντική παρακολούθηση (Environmental monitoring):

Οι έξυπνοι αισθητήρες IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση διαφόρων πτυχών του περιβάλλοντος, όπως η ποιότητα του αέρα, η θερμοκρασία και η υγρασία. Για παράδειγμα, ένα έξυπνο δίκτυο αισθητήρων σε μια πόλη μπορεί να παρακολουθεί την ποιότητα του αέρα σε πραγματικό χρόνο και να παρέχει στους πολίτες πολύτιμες πληροφορίες και ειδοποιήσεις.

- Παρακολούθηση (Tracking):

Οι έξυπνοι αισθητήρες IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση της τοποθεσίας και της κατάστασης διαφόρων περιουσιακών στοιχείων, όπως οχήματα, μηχανήματα και εξοπλισμός. Για παράδειγμα, ένας έξυπνος αισθητήρας σε ένα φορτηγό παράδοσης μπορεί να παρέχει ενημερώσεις σε πραγματικό χρόνο για την τοποθεσία και την κατάσταση παράδοσης, επιτρέποντας πιο αποτελεσματική και αποτελεσματική εφοδιαστική.

- Γεωργία (Agriculture):

Οι έξυπνοι αισθητήρες IoT μπορούν να χρησιμοποιηθούν για τη βελτιστοποίηση διαφόρων πτυχών της γεωργίας, όπως η ανάπτυξη των καλλιεργειών και η υγρασία του εδάφους. Για παράδειγμα, ένα έξυπνο δίκτυο αισθητήρων σε ένα αγρόκτημα μπορεί να παρακολουθεί τα επίπεδα υγρασίας του εδάφους και να παρέχει σε πραγματικό χρόνο πληροφορίες για τις ανάγκες άρδευσης, επιτρέποντας πιο αποτελεσματικές και βιώσιμες γεωργικές πρακτικές.

- Υγειονομική περίθαλψη (Healthcare):

Οι έξυπνοι αισθητήρες IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση διαφόρων πτυχών της προσωπικής υγείας, όπως η φυσική κατάσταση και η ευεξία. Για παράδειγμα, μια έξυπνη φορητή συσκευή μπορεί να παρακολουθεί τη φυσική δραστηριότητα και τα μοτίβα ύπνου ή μια έξυπνη συσκευή παρακολούθησης γλυκόζης αίματος μπορεί να παρέχει πληροφορίες σε πραγματικό χρόνο για τα επίπεδα σακχάρου στο αίμα.

Αυτά είναι μερικά μόνο παραδείγματα από το ευρύ φάσμα εφαρμογών που μπορούν να ενεργοποιηθούν από έξυπνες εφαρμογές IoT με έξυπνους αισθητήρες. Συμπερασματικά, τα δίκτυα IoT και αισθητήρων οδηγούν την ανάπτυξη νέων και καινοτόμων εφαρμογών και υπηρεσιών, δημιουργώντας νέες ευκαιρίες και παρέχοντας νέα αξία σε άτομα και οργανισμούς. Καθώς το οικοσύστημα IoT συνεχίζει να εξελίσσεται, μπορούμε να αναμένουμε να αναδύονται ακόμη πιο ισχυρές και αποτελεσματικές εφαρμογές, προσφέροντας νέα επίπεδα νοημοσύνης και αποτελεσματικότητας.

Η παρούσα εργασία ασχολείται με τις έξυπνες πόλεις και συγκεκριμένα με ένα δίκτυο αισθητήρων για τη διαχείριση των υδάτων μια πόλης.

1.10 Σχεδιασμός και Διαχείριση δικτύων IoT με 5G

Ο σχεδιασμός και η διαχείριση δικτύων IoT μέσω της 5G τεχνολογίας, παρουσιάζει ένα μοναδικό σύνολο προκλήσεων και ευκαιριών. Το 5G, όπως αναφέρθηκε και παραπάνω είναι μια τεχνολογία ασύρματης επικοινωνίας υψηλής ταχύτητας και χαμηλής καθυστέρησης που επιτρέπει την ανάπτυξη δικτύων IoT μεγάλης κλίμακας. Για να σχεδιαστεί και να διαχειριστεί αποτελεσματικά ένα δίκτυο IoT στο 5G, πρέπει να ληφθούν υπόψη πολλά βασικά ζητήματα όπως:

- Κάλυψη δικτύου (Network coverage):

Τα δίκτυα 5G έχουν σχεδιαστεί για να παρέχουν κάλυψη ευρείας περιοχής και ως εκ τούτου μπορούν να υποστηρίξουν την ανάπτυξη συσκευών και αισθητήρων IoT σε μεγάλες γεωγραφικές περιοχές. Ωστόσο, για να διασφαλιστεί ότι όλες οι συσκευές IoT λαμβάνουν αξιόπιστη και συνεχή κάλυψη, απαιτείται προσεκτικός σχεδιασμός και διαχείριση της υποδομής δικτύου.

- Ασφάλεια (Security):

Τα δίκτυα 5G είναι εξαιρετικά ασφαλή, αλλά η ανάπτυξη δικτύων IoT μεγάλης κλίμακας μπορεί να δημιουργήσει νέους κινδύνους για την ασφάλεια. Για τον μετριασμό αυτών των κινδύνων, είναι σημαντικό να διασφαλιστεί ότι όλες οι συσκευές IoT είναι ασφαλείς και ότι η μετάδοση των δεδομένων είναι κρυπτογραφημένη.

- Τεμαχισμός δικτύου (Network slicing):

Τα δίκτυα 5G υποστηρίζουν την έννοια του τεμαχισμού δικτύου, η οποία επιτρέπει τη δημιουργία πολλαπλών λογικών δικτύων σε ένα ενιαίο φυσικό δίκτυο. Αυτό μπορεί να χρησιμοποιηθεί για τη δημιουργία αποκλειστικών δικτύων IoT με συγκεκριμένες απαιτήσεις απόδοσης, όπως χαμηλή καθυστέρηση ή υψηλή αξιοπιστία.

- Διαχείριση δεδομένων (Data management):

Τα δίκτυα IoT παράγουν τεράστιες ποσότητες δεδομένων και η αποτελεσματική διαχείριση αυτών των δεδομένων είναι κρίσιμη για την επιτυχία των εφαρμογών IoT. Τα δίκτυα 5G παρέχουν τη συνδεσιμότητα υψηλής ταχύτητας και τη χαμηλή καθυστέρηση που

απαιτούνται για την επεξεργασία και ανάλυση δεδομένων IoT σε πραγματικό χρόνο, επιτρέποντας την ανάπτυξη νέων και καινοτόμων εφαρμογών.

Λαμβάνοντας υπόψη αυτές τις σκέψεις κατά τον σχεδιασμό και τη διαχείριση ενός δικτύου IoT στο 5G, μπορεί να διασφαλιστεί ότι τα δίκτυά τους IoT αναπτύσσονται και λειτουργούν αποτελεσματικά, παρέχοντας αξιόπιστη και ασφαλή συνδεσιμότητα σε όλες τις συνδεδεμένες συσκευές και αισθητήρες.

1.11 Σχεδιασμός και Διαχείριση δικτύων IoT με LoRa

Ο σχεδιασμός και η διαχείριση δικτύων IoT με την τεχνολογία LoRa είναι μια σημαντική πτυχή του τοπίου του Internet of Things (IoT). Η τεχνολογία LoRa (Long Range) είναι μια τεχνολογία ασύρματης επικοινωνίας που έχει σχεδιαστεί για συσκευές IoT, παρέχοντας δυνατότητες επικοινωνίας μεγάλης εμβέλειας και χαμηλής ισχύος.

- Κάλυψη δικτύου (Network coverage):

Η τεχνολογία LoRa παρέχει κάλυψη μεγάλης εμβέλειας, επιτρέποντας στις συσκευές IoT να επικοινωνούν σε αποστάσεις έως και 15 χιλιομέτρων σε αστικά περιβάλλοντα και έως 30 χιλιομέτρων σε αγροτικά περιβάλλοντα. Αυτό καθιστά το LoRa ιδανική τεχνολογία για δίκτυα IoT που πρέπει να καλύπτουν μεγάλες γεωγραφικές περιοχές, όπως γεωργικές ή εξορμητικές δραστηριότητες.

- Χαμηλή κατανάλωση ενέργειας (Low power consumption):

Η τεχνολογία LoRa έχει σχεδιαστεί για να είναι εξαιρετικά ενεργειακά αποδοτική, καθιστώντας την ιδανική για συσκευές IoT που απαιτούν μεγάλη διάρκεια μπαταρίας. Αυτό είναι ιδιαίτερα σημαντικό για συσκευές που αναπτύσσονται σε απομακρυσμένες τοποθεσίες ή σε περιβάλλοντα όπου η ισχύς είναι περιορισμένη.

- Επεκτασιμότητα (Scalability):

Τα δίκτυα LoRa έχουν σχεδιαστεί για να είναι επεκτάσιμα, επιτρέποντας στους οργανισμούς να προσθέτουν νέες συσκευές και αισθητήρες στο δίκτυό τους όπως απαιτείται. Αυτό καθιστά το LoRa ιδανική τεχνολογία για οργανισμούς που πρέπει να διαχειρίζονται δίκτυα IoT μεγάλης κλίμακας, όπως οι έξυπνες πόλεις.

- Ασφάλεια (Security):

Η τεχνολογία LoRa περιλαμβάνει προηγμένες λειτουργίες ασφαλείας, συμπεριλαμβανομένης της κρυπτογράφησης και του ελέγχου ταυτότητας, για προστασία από μη εξουσιοδοτημένη πρόσβαση σε συσκευές και δεδομένα IoT. Αυτό είναι κρίσιμο για καταστάσεις από όπου πρέπει να διαχειριστούν ευαίσθητα δεδομένα, όπως ιατρικές ή οικονομικές πληροφορίες.

- Διαλειτουργικότητα (Interoperability):

Η τεχνολογία LoRa έχει σχεδιαστεί για να είναι διαλειτουργική με άλλες τεχνολογίες IoT, όπως Wi-Fi και δίκτυα κινητής τηλεφωνίας. Αυτό επιτρέπει να ενσωματωθούν εύκολα συσκευές LoRa σε ήδη υπάρχουσα υποδομή.

- Διαχείριση και παρακολούθηση (Management and monitoring):

Τα δίκτυα LoRa μπορούν εύκολα να διαχειρίζονται και να παρακολουθούνται μέσω της χρήσης πλατφορμών διαχείρισης που βασίζονται σε σύννεφο. Αυτές οι πλατφόρμες παρέχουν στους οργανισμούς τη δυνατότητα να παρακολουθούν και να διαχειρίζονται την υγεία και την απόδοση των συσκευών και των αισθητήρων τους LoRa, δίνοντάς τους τη δυνατότητα να εντοπίζουν και να επιλύουν γρήγορα τυχόν ζητήματα που μπορεί να προκύψουν.

Ο σχεδιασμός και η διαχείριση δικτύων IoT με την τεχνολογία LoRa παρέχει μια οικονομικά αποδοτική και ευέλικτη λύση για τις ανάγκες τους στο IoT. Αξιοποιώντας τις δυνατότητες μεγάλης εμβέλειας και χαμηλής κατανάλωσης της τεχνολογίας LoRa, οι οργανισμοί μπορούν να αναπτύξουν εφαρμογές IoT που προσφέρουν πραγματική επιχειρηματική αξία και προσφέρουν ανταγωνιστικό πλεονέκτημα. Είτε αναπτύσσει δίκτυα IoT για έξυπνα σπίτια, βιομηχανικό IoT ή έξυπνες πόλεις, η τεχνολογία LoRa είναι μια ιδανική λύση για οργανισμούς που θέλουν να επωφεληθούν από την επανάσταση του IoT.

1.12 Σχεδιασμός και Διαχείριση δικτύων IoT με NB-IoT

Ο σχεδιασμός και η διαχείριση δικτύων IoT με την τεχνολογία NB-IoT (Στενής ζώνης Internet of Things) είναι μια βασική πτυχή του τοπίου του Διαδικτύου των Πραγμάτων (IoT). Το NB-IoT είναι μια τεχνολογία χαμηλής κατανάλωσης, ευρείας περιοχής δικτύου (LPWAN) που έχει σχεδιαστεί ειδικά για συσκευές IoT, παρέχοντας οικονομικά αποδοτικές και αξιόπιστες δυνατότητες επικοινωνίας. Αυτή η τεχνολογία επιτρέπει στους οργανισμούς να σχεδιάζουν και να διαχειρίζονται δίκτυα IoT που υποστηρίζουν ένα ευρύ φάσμα περιπτώσεων χρήσης και εφαρμογών, από έξυπνα σπίτια έως βιομηχανικά IoT.

- Κάλυψη δικτύου:

Το NB-IoT παρέχει κάλυψη δικτύου ευρείας περιοχής, επιτρέποντας στις συσκευές IoT να επικοινωνούν σε μεγάλες αποστάσεις, ακόμη και σε περιοχές με ασθενή ή αποσπασματική κάλυψη δικτύου. Αυτό καθιστά το NB-IoT ιδανική τεχνολογία για οργανισμούς που πρέπει να διαχειρίζονται δίκτυα IoT σε απομακρυσμένες τοποθεσίες ή περιοχές με περιορισμένη κάλυψη δικτύου.

- Χαμηλή κατανάλωση ενέργειας:

Το NB-IoT έχει σχεδιαστεί για να είναι εξαιρετικά ενεργειακά αποδοτικό, καθιστώντας το ιδανικό για συσκευές IoT που απαιτούν μεγάλη διάρκεια μπαταρίας. Αυτό είναι ιδιαίτερα σημαντικό για συσκευές που τοποθετούνται σε απομακρυσμένες τοποθεσίες ή σε περιβάλλοντα όπου η ισχύς είναι περιορισμένη.

- Ασφάλεια:

Το NB-IoT περιλαμβάνει προηγμένες λειτουργίες ασφαλείας, συμπεριλαμβανομένης της κρυπτογράφησης και του ελέγχου ταυτότητας, για προστασία από μη εξουσιοδοτημένη πρόσβαση σε συσκευές και δεδομένα IoT. Αυτό είναι κρίσιμο για όταν χρειαστεί να διαχειριστούν ευαίσθητα δεδομένα, όπως ιατρικές ή οικονομικές πληροφορίες.

- Διαχείριση και παρακολούθηση:

Τα δίκτυα NB-IoT μπορούν εύκολα να διαχειρίζονται και να παρακολουθούνται μέσω της χρήσης πλατφορμών διαχείρισης που βασίζονται σε σύννεφο. Αυτές οι πλατφόρμες παρέχουν στους οργανισμούς τη δυνατότητα να παρακολουθούν και να διαχειρίζονται την υγεία και την απόδοση των συσκευών και αισθητήρων NB-IoT, δίνοντάς τους τη δυνατότητα να εντοπίζουν και να επιλύουν γρήγορα τυχόν ζητήματα που μπορεί να προκύψουν.

2. Προβλήματα – Ζητήματα

Όπως σε οτιδήποτε, έτσι και στο IoT αδημονούν προκλήσεις και προβλήματα, τα οποία θα αναλυθούν σε αυτήν την ενότητα.

2.1 Προβλήματα και προκλήσεις ασφαλείας του IoT

Η ασφάλεια δεδομένων και γενικότερα η ασφάλεια είναι το παν και κυρίως σε μια τεχνολογία σαν το IoT όπου προκαλεί ανησυχία. Λόγω της φύσης της τεχνολογίας του Internet of Things, μια συσκευή ή ένα Πληροφοριακό Σύστημα IoT, απειλείται τόσο από τους κινδύνους των Δικτύων κινητής τηλεφωνίας, τους κινδύνους των δικτύων των αισθητήρων (sensors) όσο και από το Ίντερνετ. Η συσχέτιση μιας συσκευής IoT με τις παραπάνω περιοχές δικτύων δεν γίνεται με την πρώτη ματιά αντιληπτή, αρκεί όμως να ληφθούν υπόψιν τα εξής.

Μια συσκευή IoT, για παράδειγμα ένα έξυπνο ρολόι που μετρά την απόσταση που τρέχουμε, τους παλμούς κ.ο.κ., είναι μόνιμα συνδεδεμένο στο διαδίκτυο μέσω κάποιας σύνδεσης. Αυτό σημαίνει ότι πληροφορίες αποστέλλονται και λαμβάνονται συνεχώς, καταγράφεται η διαδρομή που διανύει ο άνθρωπος, καταχωρούνται τα ζωτικά του σημεία και έμμεσα καταγράφεται η καθημερινή του ρουτίνα. Επιπλέον, η βάση στην οποία στηρίζεται η τεχνολογία του IoT όπως αναφέρθηκε και προηγουμένως, είναι τα RFID tags, οι sensors, το Bluetooth και Near Field Communications. Στην εικόνα αυτή, εισέρχεται ο κίνδυνος της ιδιωτικότητας, της ασφάλειας της πληροφορίας που μεταδίδεται και αποθηκεύεται σε κάποια cloud υπηρεσία. Αν κάποιος έχει τη δυνατότητα να παρέμβει σε οποιοδήποτε κομμάτι της επικοινωνίας μεταξύ του smart watch και του server με τον οποίο επικοινωνεί, μπορεί να εξάγει πληροφορίες όπως, πότε και πόσο ο χρήστης λείπει από το σπίτι, πού συχνάζει, τις συνήθειες του κλπ. Τέτοιες πληροφορίες μπορούν να χρησιμοποιηθούν κακόβουλα για να βλάψουν με κάποιον τρόπο τον χρήστη (π.χ. ληστεία) ή και να τον παρακολουθούν.

Τα παρακάτω είναι μερικά από τα βασικά ζητήματα ασφαλείας και προκλήσεις που πρέπει να αντιμετωπίσουν οι οργανισμοί κατά την ανάπτυξη δικτύων IoT.

- Μη ασφαλείς συσκευές (Unsecured devices):

Πολλές συσκευές IoT αποστέλλονται με αδύναμους ή εύκολα μαντέψιμους κωδικούς πρόσβασης, γεγονός που τις καθιστά ευάλωτες σε επιθέσεις. Επιπλέον, ορισμένες συσκευές μπορεί να έχουν ξεπερασμένο υλικολογισμικό ή λογισμικό που μπορεί εύκολα να γίνει αντικείμενο εκμετάλλευσης, αφήνοντάς τες ανοιχτές σε επιθέσεις.

- Μη ασφαλής μετάδοση δεδομένων (Insecure data transmission):

Τα δεδομένα που μεταδίδονται μεταξύ συσκευών και δικτύων IoT μπορούν να υποκλαπούν από κακόβουλους παράγοντες, θέτοντας ευαίσθητες πληροφορίες σε κίνδυνο. Αυτό είναι ιδιαίτερα ανησυχητικό για οργανισμούς που συλλέγουν και αποθηκεύουν ευαίσθητα δεδομένα, όπως ιατρικές ή οικονομικές πληροφορίες.

- Εσωτερικές απειλές (Insider threats):

Τα δίκτυα IoT είναι ευάλωτα σε επιθέσεις τόσο από εξωτερικούς όσο και από εσωτερικούς παράγοντες. Οι εσωτερικές απειλές μπορεί να περιλαμβάνουν υπαλλήλους που έχουν πρόσβαση σε ευαίσθητες πληροφορίες ή που είναι σε θέση να παραβιάσουν συσκευές ή δίκτυα IoT.

- Επιθέσεις κατανεμημένης άρνησης υπηρεσίας (DDoS):

Οι συσκευές IoT μπορούν να χρησιμοποιηθούν για την εκτόξευση επιθέσεων DDoS, κατακλύζοντας ιστότοπους και δίκτυα με κίνηση, καθιστώντας τα μη διαθέσιμα στους χρήστες. Αυτοί οι τύποι επιθέσεων μπορούν να έχουν σημαντικό αντίκτυπο προκαλώντας διακοπές λειτουργίας.

- Κακώς σχεδιασμένα API:

Πολλές συσκευές και δίκτυα IoT βασίζονται σε API για να επιτρέπουν τη μετάδοση δεδομένων και εντολών μεταξύ συσκευών. Τα κακώς σχεδιασμένα API μπορούν να αφήσουν τα δίκτυα IoT ανοιχτά σε επιθέσεις, επιτρέποντας σε κακόβουλους παράγοντες να έχουν πρόσβαση σε ευαίσθητες πληροφορίες ή συσκευές ελέγχου.

Οι υπεύθυνοι πρέπει να αντιμετωπίσουν αυτές τις προκλήσεις ασφαλείας εφαρμόζοντας ισχυρά μέτρα ασφαλείας που προστατεύουν τα δίκτυά τους IoT. Αυτό μπορεί να περιλαμβάνει την εφαρμογή ισχυρών μηχανισμών ελέγχου ταυτότητας, τη χρήση κρυπτογράφησης για την προστασία ευαίσθητων δεδομένων και την εφαρμογή τείχους προστασίας και συστημάτων ανίχνευσης εισβολής για τον εντοπισμό και την πρόληψη επιθέσεων. Επιπλέον, πρέπει να είναι επιμελείς στο να διατηρούν τις συσκευές και τα δίκτυά τους IoT ενημερωμένα, με τις πιο πρόσφατες ενημερώσεις κώδικα υλικολογισμικού και λογισμικού, ώστε να διασφαλίζουν ότι προστατεύονται από γνωστά τρωτά σημεία.

2.2 Προβλήματα σχεδιασμού και Διαχείρισης δικτύων IoT

Ο σχεδιασμός και η διαχείριση ενός δικτύου IoT μπορεί να παρουσιάσει μια σειρά από προκλήσεις. Αυτές οι προκλήσεις μπορεί να περιλαμβάνουν ζητήματα που σχετίζονται με την αρχιτεκτονική δικτύου, τη διαχείριση δεδομένων και την ασφάλεια. Τα ακόλουθα είναι μερικά από τα βασικά ζητήματα σχεδιασμού και διαχείρισης που πρέπει να αντιμετωπίσουν οι οργανισμοί κατά την ανάπτυξη ενός δικτύου IoT:

- Αρχιτεκτονική δικτύου (Network architecture):

Μία από τις μεγαλύτερες προκλήσεις στο σχεδιασμό ενός δικτύου IoT είναι η επιλογή της σωστής αρχιτεκτονικής για την υποστήριξη των συσκευών και των δεδομένων που θα συνδεθούν σε αυτό. Αυτό μπορεί να περιλαμβάνει την επιλογή μεταξύ κεντρικών (centralized), αποκεντρωμένων (decentralized) και υβριδικών αρχιτεκτονικών (hybrid architectures) και τον καθορισμό του καλύτερου τρόπου για την ενσωμάτωση συσκευών και δεδομένων IoT σε υπάρχοντα δίκτυα.

- Διαχείριση δεδομένων (Data management):

Τα δίκτυα IoT παράγουν τεράστιες ποσότητες δεδομένων, τα οποία μπορεί να είναι δύσκολο να διαχειριστούν και να αναλυθούν. Οι οργανισμοί πρέπει να διαθέτουν αποτελεσματικά συστήματα διαχείρισης δεδομένων για να διασφαλίζουν ότι είναι σε θέση να αποθηκεύουν, να επεξεργάζονται και να αναλύουν τα δεδομένα που δημιουργούνται από τις συσκευές IoT τους σε πραγματικό χρόνο.

- Επεκτασιμότητα (Scalability):

Καθώς ο αριθμός των συσκευών IoT που είναι συνδεδεμένες σε ένα δίκτυο αυξάνεται, η αποτελεσματική διαχείριση του δικτύου μπορεί να γίνει όλο και πιο δύσκολη. Οι οργανισμοί πρέπει να διασφαλίσουν ότι ο σχεδιασμός του δικτύου IoT τους είναι επεκτάσιμος, ώστε να μπορεί να υποστηρίξει τον αυξανόμενο αριθμό συσκευών και ροών δεδομένων που θα συνδεθούν σε αυτό. Θα πρέπει να διασφαλισθεί ότι ο σχεδιασμός του δικτύου IoT θα είναι επεκτάσιμος, ώστε να μπορεί να υποστηρίξει τον αυξανόμενο αριθμό συσκευών και ροών δεδομένων που θα συνδεθούν σε αυτό.

- Ασφάλεια (Security):

Τα δίκτυα IoT είναι ευάλωτα σε απειλές και οι υπεύθυνοι πρέπει να λάβουν μέτρα για να προστατεύσουν τα δίκτυα και τις συσκευές τους από πιθανές επιθέσεις. Αυτό μπορεί να περιλαμβάνει την εφαρμογή ισχυρών μηχανισμών ελέγχου ταυτότητας, τη χρήση κρυπτογράφησης για την προστασία ευαίσθητων δεδομένων και την ανάπτυξη τείχους προστασίας και συστημάτων ανίχνευσης εισβολής για τον εντοπισμό και την πρόληψη επιθέσεων.

- Κόστος (Cost):

Η εφαρμογή ενός δικτύου IoT μπορεί να είναι δαπανηρή και οι οργανισμοί πρέπει να λάβουν μέτρα για τη διαχείριση του κόστους που σχετίζεται με την ανάπτυξη και τη συντήρηση των δικτύων. Αυτό μπορεί να περιλαμβάνει την επιλογή οικονομικά αποδοτικών τεχνολογιών και τη μείωση της πολυπλοκότητας του σχεδιασμού του δικτύου τους για την ελαχιστοποίηση του κόστους.

Συμπερασματικά, ο σχεδιασμός και η διαχείριση ενός δικτύου IoT μπορεί να παρουσιάσει μια σειρά από προκλήσεις. Αντιμετωπίζοντας αυτά τα ζητήματα και εφαρμόζοντας αποτελεσματικές λύσεις, μπορεί να διασφαλιστεί ότι τα δίκτυα IoT είναι επεκτάσιμα, ασφαλή και οικονομικά αποδοτικά και ότι είναι σε θέση να επωφεληθούν πλήρως από τα οφέλη που έχει να προσφέρει το IoT.

2.3 Στόχος – Σκοπός

Ο σκοπός και ο στόχος της παρούσας εργασίας είναι να γίνει ο σχεδιασμός ενός πλάνου και συγκεκριμένα αλγορίθμων, οι οποίοι θα κάνουν μια αποτίμηση και θα βγάλουν σαν αποτέλεσμα την καλύτερη και την πιο συμφέρουσα θέση και τεχνολογία για το δίκτυο. Όπως το λιγότερο κόστος και την λιγότερη κατανάλωση ενέργειας. Με άλλα λόγια, θα πρέπει να βρεθεί η τεχνολογία και οι σένσορες που θα περιλαμβάνονται στο πιο αποδοτικό IoT δίκτυο

αισθητήρων ύδρευσης σε μια συγκεκριμένη περιοχή. Επιπρόσθετα στοχεύετε η δημιουργία ενός βοηθητικού εργαλείου για την συλλογή των δεδομένων και το reporting των αποτελεσμάτων. Ο τρόπος, η δομή και η διαδικασία που ακολουθήθηκε περιγράφονται σε παρακάτω κεφάλαια.

3. Water Management IoT

Οι έξυπνες τεχνολογίες νερού είναι ένας τρόπος εξοικονόμησης, δίνοντάς τους τη δυνατότητα να μειώσουν το νερό που χάνεται λόγω διαρροών και να κάνουν τα συστήματα ύδρευσης που λειτουργούν στην πόλη πιο αποτελεσματικά. Το Διαδίκτυο των Πραγμάτων έχει κάνει τις έξυπνες λύσεις νερού ελκυστικές για επιχειρήσεις κοινής ωφέλειας. Η λειψυδρία θα επηρεάσει άμεσα σχεδόν το 20% του ανθρώπινου πληθυσμού μέχρι το 2025, σύμφωνα με αρκετές εκθέσεις του ΟΗΕ, και θα επηρεάσει έμμεσα όλους τους κατοίκους του πλανήτη, καθώς και τις οικονομίες και το σύνολο των οικοσυστημάτων. Η εταιρεία ερευνών IDC έχει προβλέψει ότι για την καταπολέμηση της λειψυδρίας, έως το 2024 το 45% των πόλεων και των κοινοτήτων θα υιοθετήσουν διαχείριση νερού με δυνατότητα IoT για παρακολούθηση χρήσης και ποιότητας και ανίχνευση διαρροών.

Η Παγκόσμια Τράπεζα έχει υπολογίσει ότι, κατά μέσο όρο, έως και 25% με 30% του νερού μιας εταιρείας κοινής ωφέλειας χάνεται στο δίκτυο ύδρευσης ως «νερό χωρίς έσοδα» (nonrevenue water - NRW). Σε ορισμένες πόλεις, το ποσοστό αυτό είναι υψηλότερο. Το νερό χωρίς έσοδα αναφέρεται στο νερό που έχει παραχθεί αλλά «χάνεται» πριν φτάσει στον καταναλωτή, είτε λόγω διαρροών είτε λόγω ανακριβιών μέτρησης. Λέγεται ότι η απομακρυσμένη παρακολούθηση του NRW μπορεί να επιτευχθεί χρησιμοποιώντας μια ποικιλία τεχνολογιών αισθητήρων ή ιστορικών αναλύσεων προτύπων για τη μείωση της σπατάλης νερού και του συναφούς κόστους. Ακόμα, η IDC εκτιμά ότι μόνο στις Ηνωμένες Πολιτείες, η μείωση των απωλειών NRW κατά το ήμισυ θα αποφέρει 2,9 δισεκατομμύρια δολάρια και θα παρέχει νερό σε επιπλέον 90 εκατομμύρια κατοίκους.

3.1 Βασικά πλεονεκτήματα

Τα έξυπνα συστήματα νερού που βασίζονται στο συνδυασμό τεχνολογιών IoT, μεγάλων δεδομένων και τεχνολογιών τεχνητής νοημοσύνης μπορούν να βοηθήσουν να σταματήσουν αυτές οι προβλέψεις και να αναιρέσουν τη ζημιά που έχει ήδη προκαλέσει η αλόγιστη χρήση των υδάτινων πόρων.

Η ασφάλεια είναι ένας σημαντικός παράγοντας για τις επιχειρήσεις κοινής ωφέλειας, οι οποίες αποτελούν συχνά στόχο. Εάν μια σημαντική εγκατάσταση δεχθεί επίθεση, θα μπορούσε να επηρεάσει τη δημόσια υγεία και το περιβάλλον για εκατομμύρια ανθρώπους. Η τεχνολογία έξυπνης διαχείρισης του νερού επιτρέπει στις πόλεις και τις επιχειρήσεις κοινής ωφέλειας να κατανοήσουν καλύτερα την προσφορά των πλεονεκτημάτων των συστημάτων ύδρευσης. Απαιτεί την ενοποίηση συστημάτων και ένα σύνολο μέτρων για την παρακολούθηση, τον έλεγχο και τη ρύθμιση της χρήσης και της ποιότητας των υδάτινων πόρων καθώς και τη συντήρηση του σχετικού εξοπλισμού (σωλήνες, αντλίες κ.λπ.). Υπάρχει μια μεγάλη γκάμα οργάνων υλικού και λογισμικού, συμπεριλαμβανομένων αισθητήρων, μετρητών, εργαλείων επεξεργασίας δεδομένων και οπτικοποίησης, ενεργοποιητές και χειριστήρια ιστού και κινητών που συνδέουν άτομα με συστήματα νερού.

Το IoT βοηθά τις επιχειρήσεις ύδρευσης παρέχοντας νέα δεδομένα και πληροφορίες που δεν είχαν οι εργαζόμενοι στο παρελθόν (όπως ειδοποιήσεις διαρροής σε πραγματικό χρόνο), πιο αναλυτικά δεδομένα, πιο έγκαιρα δεδομένα και πιο ακριβή δεδομένα. Μπορεί επίσης να

προσφέρει εξοικονόμηση λειτουργικού κόστους, καθώς η απομακρυσμένη παρακολούθηση και ο αυτοματισμός μειώνουν, το κόστος προσωπικού, αποτρέπουν την δαπανηρή απώλεια νερού κ.λπ.

Οι έξυπνες λύσεις νερού «εντοπίζουν» τις διαρροές νωρίτερα, αποτρέπουν διαρροές, διορθώνουν και συντηρούν προληπτικά τον εξοπλισμό και εντοπίζουν ανώμαλα μοτίβα στη χρήση του νερού για να επισημαίνουν τις ανακρίβειες της μέτρησης. Οι έξυπνες λύσεις νερού παρέχουν ζωτικής σημασίας υποδομές με ανθεκτικότητα απέναντι σε φυσικές καταστροφές. Παρέχουν επίσης καλύτερη εξυπηρέτηση πελατών και ορατότητα, καθώς οι επιχειρήσεις κοινής ωφέλειας έχουν καλύτερη αίσθηση του πόσο νερό χρησιμοποιούν οι άνθρωποι και πού είναι οι διαρροές. Επίσης, οι έξυπνες λύσεις νερού συμβάλλουν στην προστασία της δημόσιας και περιβαλλοντικής υγείας και επίσης συμβάλλουν στην ασφάλεια των υδάτων και διασφαλίζουν ότι το νερό δεν σπαταλάται.

3.2 Σύγχρονες έξυπνες τεχνολογίες νερού

Σήμερα, η τεχνολογία smart water φέρνει διαφάνεια και βελτιωμένο έλεγχο σε ολόκληρη την αλυσίδα παροχής νερού, ξεκινώντας από μια δεξαμενή γλυκού νερού έως τη συλλογή και την ανακύκλωση λυμάτων. Αυτή η κατηγορία περιλαμβάνει συσκευές IoT για διαχείριση νερού, συστήματα και εργαλεία λογισμικού που συμβάλλουν στη βελτιστοποίηση της παραγωγής, διανομής και κατανάλωσης νερού και επιτρέπουν έξυπνες πρακτικές επεξεργασίας νερού.

Οι αισθητήρες (sensors) έχουν ευρείες εφαρμογές στην έξυπνη διαχείριση νερού λόγω της μεγάλης ποικιλομορφίας και των σκοπών τους. Σε μια πολύ βασική αλυσίδα παροχής νερού, οι αισθητήρες μετρούν:

- την ποιότητα του ακατέργαστου νερού λεκάνης απορροής,
- τη χημική σύνθεση στο νερό μετά την επεξεργασία και τα λύματα κ.λπ.
- την αλλαγή της ποσότητας στη δεξαμενή αποθήκευσης,
- τη πίεση στους σωλήνες στον αγωγό διανομής,
- τη φθορά του εξοπλισμού και των μηχανημάτων που επεξεργάζονται και διανέμουν νερό στους τελικούς χρήστες και πολλά άλλα.

Επίσης, οι έξυπνοι μετρητές και οι κόμβοι παρακολούθησης επιτρέπουν τη μέτρηση της κατανάλωσης νερού σε πραγματικό χρόνο, βοηθούν στον εντοπισμό σημείων υπερβολικής χρήσης και σπατάλης, καθώς και σωστών μοτίβων χρήσης και κάνουν προβλέψεις για μελλοντική κατανάλωση. Αυτή η τεχνολογία διαχείρισης νερού είναι χρήσιμη για τους διαχειριστές παραγωγής και διανομής νερού αλλά και για τα νοικοκυριά. Χρησιμοποιώντας έξυπνους μετρητές και συστήματα παρακολούθησης νερού, μπορούμε να διορθώσουμε τις περιττές καταναλώσεις νερού και να επιταχύνουμε τους στόχους βιωσιμότητας και προϋπολογισμού. Όλο και περισσότερες εταιρείες στρέφονται σε πλήρως αυτόματες πρακτικές διαχείρισης νερού. Χρησιμοποιώντας περιβαλλοντικούς αισθητήρες και προκαθορισμένους (predefined) αλγόριθμους ή αλγόριθμους μηχανικής μάθησης (machine learning), τα συστήματα διανομής μπορούν να ρυθμίζουν και να ελέγχουν δυναμικά την παροχή νερού. Στην περίπτωση της έξυπνης άρδευσης, για παράδειγμα, οι ψεκασθήρες

παρέχουν αρκετό νερό ανάλογα με τις ενδείξεις από τους αισθητήρες υγρασίας του εδάφους, υγρασίας αέρα και κατάσταση καλλιέργειας.

3.3 Μέθοδοι βελτιστοποίησης

Η βέλτιστη τοποθέτηση αισθητήρων για τον εντοπισμό συμβάντων μόλυνσης στα συστήματα διανομής νερού είναι πολύ σημαντική. Άρα, θα πρέπει να μελετηθεί το πρόβλημα της τοποθέτησης αισθητήρων σε δίκτυα νερού (water networks) για την ελαχιστοποίηση της κατανάλωσης μολυσμένου νερού πριν από την ανίχνευση της μόλυνσης.

3.4 Στόχοι της έξυπνης διαχείρισης των υδάτων

Ο πρωταρχικός στόχος της έξυπνης διαχείρισης του νερού είναι η λογική και βιώσιμη χρήση και ανακύκλωση των υδάτινων πόρων. Ο αυξανόμενος πληθυσμός, τα αυξανόμενα περιβαλλοντικά ζητήματα και η πίεση στον τομέα των τροφίμων και της γεωργίας καθιστούν το νερό ακόμη πιο πολύτιμο αγαθό. Από αυτή την άποψη, οι τεχνολογίες και οι δραστηριότητες διαχείρισης των υδάτων επιδιώκουν τους ακόλουθους στόχους:

1. **Μείωση της σπατάλης νερού** που χρησιμοποιείται σε μεγάλες ποσότητες για τη γεωργία, τη μεταποίηση, την παραγωγή ηλεκτρικής ενέργειας. Συνεπάγεται την εισαγωγή πρακτικών υψηλής τεχνολογίας όπως η γεωργία ακριβείας, η έξυπνη άρδευση, η διαχείριση του νερού των καλλιεργειών, η μέτρηση νερού σε πραγματικό χρόνο και άλλες εφαρμογές του IoT στη γεωργία [18][19].

2. **Βελτίωση της ποιότητας του νερού** και απότρεψη της μόλυνσης από χημικά απόβλητα και φυσική ρύπανση όπως η όξυνση. Προκειμένου να βελτιώσουν και να διατηρήσουν την ποιότητα του νερού, οι εταιρείες χρησιμοποιούν αισθητήρες και τεχνολογία IoT για παρακολούθηση και έλεγχο σε πραγματικό χρόνο.

3. **Βελτίωση της αποτελεσματικότητας των συστημάτων ύδρευσης** όπως οι συλλέκτες νερού, οι μονάδες επεξεργασίας, τα δίκτυα διανομής και τα κέντρα ανακύκλωσης λυμάτων. Χρησιμοποιώντας λύσεις IoT και δεδομένων για διαχείριση περιουσιακών στοιχείων, οι εταιρείες μπορούν να κρατούν σημαντικές μετρήσεις όπως η πίεση του νερού, η θερμοκρασία, η ροή κ.λπ., να ενσωματώνουν προγνωστική συντήρηση και να αποφεύγουν τη θραύση και το χρόνο διακοπής λειτουργίας του εξοπλισμού.

4. **Εφαρμογή ελέγχου διαρροών** χρησιμοποιώντας έξυπνες συσκευές διαχείρισης νερού εξοπλισμένες με αισθητήρες διαρροής και υγρασίας. Δεδομένου ότι σχεδόν 3 δισεκατομμύρια δολάρια δαπανώνται για την αποκατάσταση της ζημιάς που προκαλείται από διαρροή ετησίως, ο έλεγχος των διαρροών είναι απαραίτητος για την ασφαλή διατήρηση των υδάτινων πόρων και των προϋπολογισμών.

5. **Εξάσκηση στην παρακολούθηση** της κατανάλωσης μέσω συστημάτων διαχείρισης νερού που βασίζονται στο IoT. Βοηθά στη βελτιστοποίηση και τη διατήρηση υπό έλεγχο της χρήσης των υδάτινων πόρων σε διαφορετικά επίπεδα όπως νοικοκυριά, κοινότητες, χώρες και ολόκληρο τον πλανήτη.

Οφέλη από τη χρήση του IoT για τη διαχείριση του νερού

Οι λύσεις του IoT για τη διαχείριση του νερού βοηθούν τους ενδιαφερόμενους φορείς του κλάδου, τις κυβερνήσεις και τους μέσους καταναλωτές να επιτύχουν τους στόχους βιωσιμότητας και αποδοτικότητας. Σήμερα, η ιδέα του IoT σε αυτόν τον τομέα μεταφράζεται ήδη σε μια ολοκαίνουρια ιδέα - το Διαδίκτυο του Νερού. Απαιτεί λοιπόν, τη σύνδεση όλων των συστημάτων και των φορέων στην αλυσίδα παροχής νερού. Πηγές νερού, εγκαταστάσεις επεξεργασίας και βιομηχανικά συστήματα διαχείρισης νερού, εγκαταστάσεις διανομής, εταιρείες κοινής ωφελείας και καταναλωτές κ.λπ. Ίσως, δεν υπάρχει καλύτερη τεχνολογία για τον εξοπλισμό της αλυσίδας εφοδιασμού με τις απαραίτητες διαδικασίες, εφαρμογές δεδομένων και εργαλεία από το IoT. Και οι λόγοι είναι οι εξής:

Διαφάνεια (Transparency): Ένα από τα μεγαλύτερα οφέλη της έξυπνης διαχείρισης νερού με χρήση IoT είναι η βελτίωση της διαφάνειας όλων των διαδικασιών στην αλυσίδα παροχής νερού. Χάρη στα δεδομένα που συλλέγονται σε όλη την αλυσίδα εφοδιασμού, διαφορετικοί ενδιαφερόμενοι λαμβάνουν σημαντικές πληροφορίες για τους πόρους και την απόδοση του συστήματός τους. Ως αποτέλεσμα μπορούν να λαμβάνουν τεκμηριωμένες αποφάσεις για το πώς να βελτιώσουν τις δραστηριότητές τους.

Άμεση απάντηση: (Immediate response): Ένα άλλο πλεονέκτημα της ενσωμάτωσης έξυπνων συστημάτων διαχείρισης νερού είναι η ικανότητα εντοπισμού ή ακόμη και πρόβλεψης προβλημάτων και άμεσης απόκρισης για την ελαχιστοποίηση της ζημιάς. Για παράδειγμα, η παρακολούθηση της ποιότητας του νερού και της χημικής σύστασης σε πραγματικό χρόνο επιτρέπει την ανίχνευση έστω και της ελαφριάς μόλυνσης και την έναρξη ταχείας απόκρισης προτού καταστεί επικίνδυνη.

Αυτοματοποίηση και βελτιστοποιημένη χρήση του ανθρώπινου δυναμικού: Οι λύσεις διαχείρισης νερού IoT επιτρέπουν στους διαχειριστές να αυτοματοποιήσουν εν μέρει ή πλήρως ορισμένες διαδικασίες και να βελτιστοποιήσουν τη χρήση της ανθρώπινης δύναμης. Το βάθος και το εύρος του αυτοματισμού ποικίλλει ανάλογα με τον κλάδο και τις συγκεκριμένες επιχειρηματικές ανάγκες. Για παράδειγμα, οι έξυπνες εταιρείες παροχής νερού και τα δίκτυα κοινής ωφέλειας μπορούν να αυτοματοποιήσουν ολόκληρο τον κύκλο ζωής της παροχής νερού στους καταναλωτές χρησιμοποιώντας συνδεδεμένους μετρητές, συστήματα παρακολούθησης σε πραγματικό χρόνο και δυναμικά μοντέλα τιμολόγησης.

Βελτιστοποιημένο κόστος (Optimized cost): Η αυτοματοποίηση, η βελτιστοποιημένη χρήση των ανθρώπινων πόρων, η στρατηγική που βασίζεται σε δεδομένα και η προληπτική προσέγγιση στη συντήρηση του εξοπλισμού και τη χρήση των πόρων μεταφράζονται τελικά σε σημαντική εξοικονόμηση πόρων. Ένας από τους λόγους για τους οποίους οι εταιρείες ύδρευσης εξετάζουν τη χρήση του IoT στη διαχείριση του νερού είναι η μείωση του λειτουργικού κόστους μακροπρόθεσμα.

Βιωσιμότητα: Οι στόχοι βιωσιμότητας βρίσκονται στο επίκεντρο πολλών έργων αναβάθμισης και καινοτομίας όχι μόνο στη βιομηχανία έξυπνου νερού, αλλά σε οποιονδήποτε άλλο τομέα, όπως η ενέργεια, οι κατασκευές, τα logistics κ.λπ. Οι τεχνολογίες για έξυπνη χρήση νερού δεν θεωρούνται πλέον μόνο πηγή εξοικονόμησης και υψηλότερης αποδοτικότητας, αλλά και μέσο για την επίτευξη διαφορετικών περιβαλλοντικών στόχων, συμπεριλαμβανομένου του μειωμένου αποτυπώματος άνθρακα, της ρύπανσης και, ουσιαστικά, της διατήρησης του νερού.

Μελλοντική στρατηγική (Forward-looking strategy): Όσον αφορά την εξοικονόμηση νερού, η έξυπνη τεχνολογία είναι επωφελής όχι μόνο για τις επιχειρήσεις και τους καταναλωτές, αλλά και για τους ερευνητές που είναι υπεύθυνοι για τη δημιουργία στρατηγικών εξοικονόμησης νερού με γνώμονα το μέλλον. Χρησιμοποιώντας τα δεδομένα που παρέχονται από τη διαχείριση της παροχής νερού και άλλα συνδεδεμένα συστήματα σε όλη την αλυσίδα εφοδιασμού, οι ερευνητές μπορούν να δημιουργήσουν στρατηγικές βάσει δεδομένων για το πώς να βελτιστοποιήσουν τη χρήση των υδάτινων πόρων προς όφελος των κοινοτήτων, των οικοσυστημάτων και ολόκληρου του πλανήτη.

Εφαρμογές IoT στη διαχείριση υδάτων: Υπάρχουν πολλά συστήματα IoT διαχείρισης νερού και λύσεις μεγάλων δεδομένων (big data) στην αγορά, που καταδεικνύουν τον αντίκτυπο που έχουν αυτές οι τεχνολογίες σε ολόκληρο τον κλάδο.

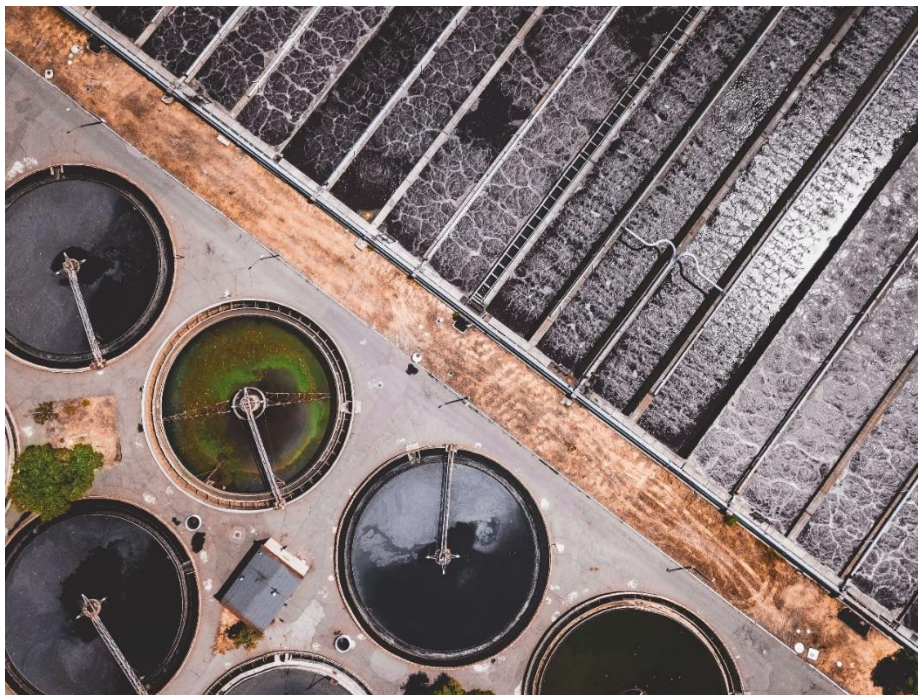
Ακεραιότητα του συστήματος ύδρευσης: Υπάρχουν ανιχνευτές διαρροών που βασίζονται σε αισθητήρες που βοηθούν στην άμεση αναγνώριση της ζημιάς του σωλήνα ή της σύνδεσης και αποτρέπουν τη μεγάλη διαρροή και τη σπατάλη των υδάτινων πόρων. Αυτές οι ταινίες είναι απίστευτα απλές στην εγκατάσταση και στη χρήση, τόσο σε εσωτερικούς όσο και σε εξωτερικούς χώρους και επομένως είναι ιδανικές για νοικοκυριά, γραφεία και δημόσιους χώρους.

Έξυπνη άρδευση: Η Bosch παρέχει μια λύση βασισμένη σε αισθητήρες για έξυπνη άρδευση κατ' απαίτηση. Μετρά την κατάσταση του νερού στα φυτά για να βεβαιωθεί ότι λαμβάνουν αρκετό νερό για την καλύτερη θρεπτική αξία και την υψηλότερη απόδοση. Τα δεδομένα που συλλέγονται για τη «δίψα» του φυτού συνδυάζονται με την πρόγνωση του καιρού. Στη συνέχεια, οι αλγόριθμοι τεχνητής νοημοσύνης υπολογίζουν το ιδανικό πρόγραμμα άρδευσης με βάση αυτά τα δεδομένα και βοηθούν τους αγρότες να διατηρήσουν τις καλλιέργειές τους στην απόλυτη υγεία.



Εικόνα 8 Έξυπνο θερμοκήπιο

Έξυπνη παρακολούθηση του νερού: Η Adcon είναι μια εταιρεία ύδρευσης που παρέχει ένα ευρύ φάσμα υπηρεσιών διαχείρισης νερού από ανιχνευτές διαρροών έως διαχείριση άρδευσης και παρακολούθηση όμβριων υδάτων. Μία από τις λύσεις της εταιρείας επικεντρώνεται στην έξυπνη μέτρηση του νερού και την παρακολούθηση της ποιότητας για διαφορετικές επιχειρήσεις στην αλυσίδα εφοδιασμού όπως αγρότες, μετεωρολόγους, υπηρεσίες κοινής ωφέλειας κ.λπ. Η λύση περιλαμβάνει αισθητήρες, σταθμούς, μονάδες τηλεμετρίας και λογισμικό που επεξεργάζεται τα δεδομένα που δημιουργούνται και δημιουργεί πληροφορίες για όσους λαμβάνουν αποφάσεις.



Εικόνα 9 Σύστημα παρακολούθησης νερού

Έξυπνη διαχείριση βροχής και όμβριων-βρόχινων υδάτων: Γίνεται η μετατροπή της βροχής και των όμβριων υδάτων σε νερό ικανό να καλύψει πλήρως τις ανάγκες σε νερό για

ανεξάρτητα νοικοκυριά και ολόκληρες τις γειτονιές. Επίσης μπορούν να εξεταστούν δεδομένα για τη συλλογή του βρόχινου νερού και τη μετατροπή του σε σημαντική πηγή νερού για κοινότητες εκτός δικτύου. Αυτή η προσέγγιση δείχνει πώς η εφαρμογή του στη διαχείριση των υδάτινων πόρων βοηθά να ξεκλειδωθούν οι νέες επιλογές για πιο βιώσιμη και ανθεκτική ζωή.

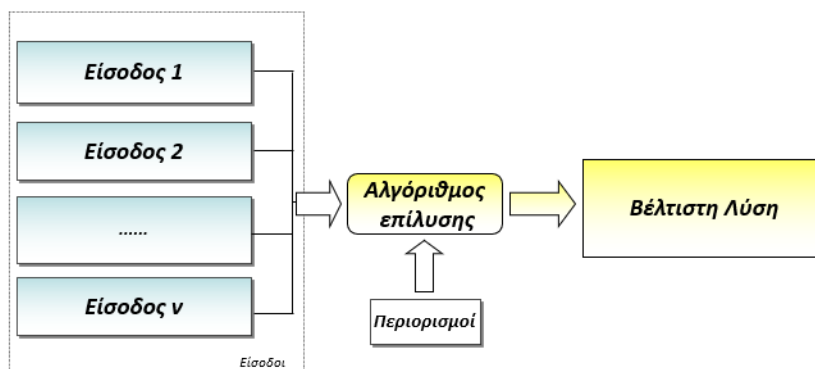
4. Διατύπωση προβλήματος και οι προτεινόμενοι αλγόριθμοι επίλυσης

4.1 Μαθηματικά μοντέλα του προβλήματος (Problem Formulation)

Σε αυτό το κεφάλαιο περιγράφονται πολύ αναλυτικά το μαθηματικό μοντέλο (Problem Formulation) που ακολουθήθηκε, οι αλγόριθμοι και το προτεινόμενο application – tool που αναπτύχθηκε.

Το πρόβλημα που καλούμαστε να λύσουμε είναι ένα πρόβλημα σχεδιασμού δικτύου. Η γενική ιδέα είναι ότι επιθυμούμε να τοποθετήσουμε IoT αισθητήρες και gateways για την δημιουργία ενός έξυπνου συστήματος νερού για την καλύτερη διαχείριση των υδάτων σε μια γειτονιά ή και μια έξυπνη πόλη (smart city).

Όπως σε κάθε πρόβλημα, πρέπει να οριστούν οι μεταβλητές αποφάσεις (decision variables). Έτσι έχει γίνει και στο παρόν πρόβλημα. Αυτές οι μεταβλητές καθορίζουν κατά κάποιο τρόπο την απόφαση για την βέλτιστη λύση. Υπάρχει μια “περιοχή” με όλες τις λύσεις που λύνουν το πρόβλημα. Πρέπει να γίνει προσπέλαση όλων αυτών των λύσεων, ώστε να βρεθεί η καλύτερη και να επιλεγεί. Για να γίνει αυτό θα πρέπει να γίνει χρήση κάποιου αλγόριθμου επίλυσης. Έχοντας λοιπόν το ζητούμενο και ό,τι δεδομένα χρειάζεται, καταλήγουμε σε λύσεις όπου μπορούν να λύσουν το πρόβλημα. Αυτές οι λύσεις είναι οι εισοδοί - λύσεις για έναν αλγόριθμο επίλυσης, που διαθέτει κάποιους περιορισμούς για να φτάσει στην βέλτιστη ή στις βέλτιστες λύσεις.



Εικόνα 10 (Εικόνα από διαφάνειες Σχεδιασμός και Διαχείριση Δικτύων)

Διατύπωση του προβλήματος

Το πρόβλημα μας ξεκινάει με τον διαχωρισμό του συστήματος. Έστω ότι έχουμε να επιλέξουμε ανάμεσα σε 3 συστήματα, οπότε ορίζουμε μια μεταβλητή $type\ of\ system = X$ όπου $X \in N$. Το εύρος επιλογών μας είναι:

Type of system: 1 = Water Quality, 2 = Water Leak, 3 = Soil Moisture.

Στη συνέχεια, ορίζουμε την περιοχή όπου θέλουμε να εξετάσουμε και συγκεκριμένα την ορίζουμε σε km^2 (είναι προαιρετικό).

Ορίστηκαν 3 διαφορετικά σενάρια όπου κάθε σενάριο αντιστοιχεί σε μια τεχνολογία. Τα σενάρια – τεχνολογίες που συμπεριλάβαμε είναι 5G, NB-IoT και LoRa. Μελετάμε όλα τα διαφορετικά σενάρια με τον ίδιο τρόπο και με την ίδια λογική προσέγγιση. Έπειτα φτιάξαμε λύσεις για κάθε μια τεχνολογία, βασιζόμενοι σε κάποιους περιορισμούς (constraints), με βάση κάποια χαρακτηριστικά της κάθε τεχνολογίας ως προς το τι προσφέρει και τι σένσορες ταιριάζουν. Ο βασικός περιορισμός για κάθε πιθανή λύση ώστε να είναι “έγκυρη”, είναι η **κάλυψη** (coverage), η οποία δεν θέλουμε να είναι μικρότερη από το 75%.

Έχουμε επιλέξει για κάθε τεχνολογία κάποια είδη αισθητήρων, από 3 έως 5 και επιπλέον τα gateways για την τεχνολογία LoRa. Οπότε η κάθε λύση έχει διάφορους συνδυασμούς αισθητήρων, πάντα τηρώντας τους αρχικούς περιορισμούς. Έχοντας λοιπόν τα σενάρια και τις διαφορετικές προτεινόμενες λύσεις για το καθένα, στη διαδικασία πλέον μπορούν να “εμπλακούν” και οι αλγόριθμοι. Οπότε έχουμε 3 σενάρια με N λύσεις το κάθε ένα. Επίσης, έχουμε δημιουργήσει και κάποιους βοηθητικούς πίνακες. Η πληροφορία που έχουν αυτοί οι πίνακες είναι το κόστος του αισθητήρα, το κόστος εγκατάστασης και συντήρησης, η χωρητικότητα (capacity) της μπαταρίας του αισθητήρα σε mAh και η κατανάλωση της μπαταρίας για κάθε διαφορετικό τύπο αισθητήρα ανά τεχνολογία. Η μορφή των βοηθητικών αυτών πινάκων φαίνεται στην παρακάτω εικόνα. Ο τρόπος με τον οποίον αξιοποιούμε τα δεδομένα αυτά, περιγράφεται στη συνέχεια.

Όπως έχει ήδη αναφερθεί και παραπάνω, επιθυμούμε να βρούμε την καλύτερη λύση ως προς το κόστος, την κατανάλωση και φυσικά την κάλυψη. Έτσι, κατά την προσπέλαση των αλγορίθμων δημιουργούνται αποτελέσματα για ήδη υπάρχουσες λύσεις που έχουν συγκριθεί, είτε δημιουργούνται και προτείνονται ως καλύτερες καινούριες λύσεις.

Μαθηματική διατύπωση προβλήματος

Παράδειγμα υπολογισμού κόστους κάθε λύσης:

Έχουμε N λύσεις για κάθε σενάριο $X_1, X_2, X_3, \dots, X_N$

και $X_i = [X_{i,1}, X_{i,2}, X_{i,3}]$, $X_{i,1}, X_{i,2}, X_{i,3} \in \mathbb{N}$ και είναι ένας ακέραιος αριθμός.

Όπου $X_1 = [a, b, c]$, $X_2 = [d, e, f]$, ..., $X_N = [x, y, z]$,

Σε κάθε X_i αντιστοιχεί και ένα C_i , οπότε έχουμε $C_1, C_2, C_3, \dots, C_N$.

Και $C_i = [C_{i,1}, C_{i,2}, C_{i,3}]$, $C_{i,1}, C_{i,2}, C_{i,3} \in \mathbb{N}$

Όπου $C_{i,j} = [C_{i,j,1}, C_{i,j,2}]$ και $C_{i,j,k} = [(X_{i,1} \cdot C_{i,j,k,1}) + (X_{i,1} \cdot C_{i,j,k,2})]$ όπου $C_{i,j,k,y} \in \mathbb{N}$

Στόχος: Εύρεση του μικρότερου κόστους C. Minimize $C_i = \sum_{i \in N} []$

Ένα παράδειγμα με αριθμούς

Έστω X_i μια λύση για 5G Τεχνολογία με 3 τύπους αισθητήρων.

Έστω ότι έχουμε 20 αισθητήρες (τύπου A), 30 αισθητήρες (τύπου B), και 5 αισθητήρες (τύπου Γ). Οπότε η λύση είναι της μορφής $X_i = [20, 30, 5]$. Κάθε ένας τύπος αισθητήρα έχει και ένα κόστος με αποτέλεσμα αθροίζοντας τα 3 αυτά κόστη να καταλήγουμε στο τελικό κόστος αυτής της λύσης.

Οπότε $\text{Solution}_{\text{TotalCost}} = \text{cost}_1 + \text{cost}_2 + \text{cost}_3$ ή αλλιώς $C_i = (\text{cost}_1 + \text{cost}_2 + \text{cost}_3)$

Τελικά $C_i =$ με κάποιον πραγματικό αριθμό

Όπου: $\text{cost}_i = (\text{αριθμός αισθητήρων}_i \cdot \text{κόστος αισθητήρων}_i) + (\text{αριθμός αισθητήρων}_i \cdot \text{κόστος εγκατάστασης} - \text{συντήρησης}_i)$

Έχουμε έναν πίνακα με κόστη

Κόστος αγοράς	Κόστος εγκατάστασης – συντήρησης
100	100
200	50
50	100

Οπότε καταλήγουμε στο παρακάτω

$$\text{cost}_1 = (20 \cdot 100) + (20 \cdot 100)$$

$$\text{cost}_2 = (30 \cdot 200) + (30 \cdot 50)$$

$$\text{cost}_3 = (5 \cdot 50) + (5 \cdot 100)$$

Άρα, $C_i = [(20 \cdot 100) + (20 \cdot 100), (30 \cdot 200) + (30 \cdot 50), (5 \cdot 50) + (5 \cdot 100)] \Rightarrow$
 $C_i = [4.000 + 7.500 + 750] \Rightarrow C_i = [12.750]$

Παράδειγμα υπολογισμού ενέργειας – μπαταρίας κάθε λύσης:

Έχουμε όπως και παραπάνω N λύσεις για κάθε σενάριο $X_1, X_2, X_3, \dots, X_N$

και $X_i = [X_{i,1}, X_{i,2}, X_{i,3}]$, $X_{i,1}, X_{i,2}, X_{i,3} \in \mathbb{N}$ και είναι ένας ακέραιος αριθμός.

Όπου $X_1 = [a, b, c]$, $X_2 = [d, e, f]$, ..., $X_N = [x, y, z]$,

Σε κάθε X_i αντιστοιχεί και ένα Energy_i , οπότε έχουμε $\text{Energy}_{i,1}, \text{Energy}_{i,2}, \text{Energy}_{i,3}, \dots, \text{Energy}_{i,N}$.

Και $\text{Energy}_{i,j} = [BL_{i,1}, BL_{i,2}, BL_{i,3}]$, $BL_{i,1}, BL_{i,2}, BL_{i,3} \in \mathbb{N}$

Όπου $BL_{i,1} = BL_{i,j,1}$ και $BL_{i,j,1} = \left\lfloor \frac{\text{Battery Capacity (in mAh)}}{\text{Device Consumption (in mAh)}} \right\rfloor$.

Τέλος, έχουμε τη ζωή της μπαταρίας για κάθε αισθητήρα της προτεινόμενης λύσης. Σαν τελικό αποτέλεσμα ενέργειας της κάθε λύσης, θεωρούμε το μέσο όρο ζωής της μπαταρίας κάθε αισθητήρα.

Άρα $BL_i = \frac{BL_{i,j,1} + BL_{i,j,2} + BL_{i,j,\text{length}_i}}{\text{length}_i}$.

Ένα παράδειγμα με αριθμούς

Σε συνέχεια του προηγούμενου παραδείγματος με την εξής λύση.

Έστω λύση $X_i = [20, 30, 5]$

$$\text{Οπότε } \text{Solution}_{\text{TotalEnergy}} = \left(\frac{bl_1 + bl_2 + bl_3}{3} \right)$$

Τελικά Energy_i = με κάποιον πραγματικό αριθμό

$$\text{Όπου } bl_1 = \frac{\text{χωρητικότητα μπαταρίας αισθητήρα (σε mAh)}}{\text{κατανάλωση ενέργειας μπαταρίας αισθητήρα (σε mA)}}$$

Έχουμε ένα βοηθητικό πίνακα.

Χωρητικότητα μπαταρίας (σε mAh)	Κατανάλωση ενέργειας μπαταρίας (σε mA)
13000	0.018
30000	0.021
18000	0.017

Οπότε καταλήγουμε στο παρακάτω

$$BL_1 = \frac{13000}{0.018}$$

$$BL_2 = \frac{30000}{0.021}$$

$$BL_3 = \frac{18000}{0.017}$$

$$\begin{aligned} \text{Energy}_i &= \frac{\frac{13000}{0.018} + \frac{30000}{0.021} + \frac{18000}{0.017}}{3} \Rightarrow \text{Energy}_i \\ &= \frac{72.222,22 + 1.428.571,42 + 105.882,35}{3} \Rightarrow \end{aligned}$$

$\text{Energy}_i = \frac{1.606.675,99}{3} \Rightarrow \text{Energy}_i = [535.558,66] \Rightarrow \text{Energy}_i = 535.558 \text{ h}$ και αν θέλουμε κάνουμε την μετατροπή σε μήνες ή χρόνια (θεωρούμε το πιο ασφαλές ότι είναι οι μήνες, οπότε η υλοποίησή μας ακολουθεί αυτόν τον κανόνα).

$$\text{Τέλος } \text{Solution}_{\text{TotalEnergy(month)}} = \frac{535.558 \text{ h}}{24 \text{ h}} = 22.314,91 \text{ ημέρες} \Rightarrow$$

$$\text{Solution}_{\text{TotalEnergy(month)}} = \frac{22.314 \text{ ημέρες}}{30 \text{ ημέρες}} = 743,8 \text{ μήνες}$$

Έχοντας εισάγει και υλοποιήσει αυτούς τους υπολογισμούς μέσα στους αλγορίθμους που χρησιμοποιήθηκαν εξάγουμε τα αποτελέσματα και τα συγκρίνουμε, ώστε να καταλήξουμε στον σχεδιασμό και στην υλοποίηση του δικτύου βασιζόμενοι σε αυτά τα αποτελέσματα.

4.2 Αλγόριθμοι επίλυσης του παραπάνω προβλήματος

Όσον αφορά την επίλυση του προβλήματος, όπως αναφέρθηκε και παραπάνω χρησιμοποιήθηκαν κάποιοι αλγόριθμοι. Η χρήση αλγορίθμων ανέκαθεν ήταν πολύ σημαντική και απαραίτητη για την επίλυση προβλημάτων και ιδιαίτερα προβλημάτων που έχουν να κάνουν με τέτοιου είδους ζητήματα, όπως αυτό που έχει επιλεγεί στην παρούσα διπλωματική εργασία. Για αυτό το σκοπό υπάρχουν πολλές εριστικές μέθοδοι.

Ευριστικοί μέθοδοι: Είναι τεχνικές που αναζητούν και βρίσκουν καλές λύσεις σε δύσκολα προβλήματα, με λογικό υπολογισμό κόστους. Θεωρούνται προσεγγιστικές μέθοδοι, διότι παρέχουν αρκετά καλές λύσεις, με μικρή σχετικά προσπάθεια, αλλά δεν εγγυώνται την βέλτιστη λύση. Έχουν όμως ευρεία χρήση σε πολλούς τομείς. Η “άπληστη αναζήτηση” (greedy search) είναι μια τέτοια μέθοδος, με βάση την οποία υλοποιήθηκε ένας αλγόριθμος (άπληστος αλγόριθμος – greedy algorithm). Για την επίλυση του παρόντος προβλήματος επιλέχθηκαν και υλοποιήθηκαν **άπληστοι αλγόριθμοι (greedy algorithms) και γενετικοί αλγόριθμοι (genetic algorithms)**.

4.2.1 Σχετικά με Greedy Algorithm – Άπληστος Αλγόριθμος

Οι άπληστοι αλγόριθμοι ή αλλιώς όπως είναι ευρέως γνωστοί ως greedy algorithms, είναι απλοί και ευκολονόητοι αλγόριθμοι που χρησιμοποιούνται για τη βελτιστοποίηση των προβλημάτων και την εύρεση του βέλτιστου τρόπου λύσης ενός προβλήματος. Ένας άπληστος αλγόριθμος όπως λέει και το όνομά του, πάντα κάνει την επιλογή που φαίνεται να είναι η καλύτερη τη συγκεκριμένη στιγμή όπου τρέχει. Αυτό σημαίνει ότι κάνει μια τοπικά βέλτιστη επιλογή. Οι άπληστοι αλγόριθμοι είναι πολύ αποδοτικοί σε προβλήματα εύρεσης της καλύτερης διαδρομής σε γράφο ή προβλήματα εξοικονόμησης κόστους. Γι’ αυτό το λόγο, χρησιμοποιείται και ένας τέτοιος αλγόριθμος στο πρόβλημα που παρουσιάζεται και λύνεται σε αυτήν την εργασία. Οι άπληστοι αλγόριθμοι συλλέγουν όλα τα δεδομένα από ένα πρόβλημα και στη συνέχεια ορίζουν κανόνες για το ποια στοιχεία θα προστεθούν στη λύση σε κάθε βήμα του αλγορίθμου. Τους κανόνες τους ορίζουμε εμείς, αναλόγως με το τι θέλουμε να πετύχουμε. Οι αλγόριθμοι αυτοί τρέχουν σε βήματα και σε κάθε βήμα επιλέγουν την πιο βέλτιστη λύση. Με άλλα λόγια, οι άπληστοι αλγόριθμοι λειτουργούν σε προβλήματα για τα οποία, σε κάθε βήμα, υπάρχει μια επιλογή που είναι βέλτιστη για το πρόβλημα μέχρι εκείνο το βήμα, και μετά το τελευταίο βήμα, ο αλγόριθμος παράγει τη συνολική βέλτιστη λύση του προβλήματος. Υπάρχουν πολλές δυνατότητες με την χρήση τέτοιου είδους αλγορίθμων και πολλές διαφορετικές μέθοδοι. Οι άπληστοι αλγόριθμοι ταιριάζουν καλύτερα για προβλήματα όπου μπορεί να βρεθεί μια συνολική βέλτιστη λύση κάνοντας τοπικές βέλτιστες επιλογές. Για προβλήματα με πολλαπλούς στόχους, μια άπληστη προσέγγιση μπορεί να μην είναι κατάλληλη.

Μέθοδος, οπτική, στρατηγικές και αποφάσεις

Για να λυθεί ένα πρόβλημα βασισμένο στην άπληστη προσέγγιση, υπάρχουν 2 στάδια. Το πρώτο στάδιο είναι η σάρωση της λίστας των αντικειμένων και το δεύτερο στάδιο είναι η βελτιστοποίηση. Δυο συνθήκες καθορίζουν το άπληστο παράδειγμα. Κάθε σταδιακή

(stepwise) λύση, πρέπει να δομεί ένα πρόβλημα προς την καλύτερη αποδεκτή λύση του. Είναι επαρκές, αν η δομή του προβλήματος μπορεί να σταματήσει σε έναν πεπερασμένο αριθμό άπληστων βημάτων.

Η λογική στην πιο απλή μορφή της συνοψίζεται σε “άπληστη” ή “μη άπληστη”. Αυτές οι δηλώσεις ορίστηκαν από την προσέγγιση που ακολουθήθηκε για να προχωρήσουμε σε κάθε στάδιο του αλγορίθμου.

Για παράδειγμα, ο αλγόριθμος Dijkstra χρησιμοποίησε μια σταδιακά άπληστη στρατηγική, που εντοπίζει κεντρικούς υπολογιστές στο Διαδίκτυο υπολογίζοντας μια συνάρτηση κόστους. Η τιμή που επιστρέφεται από την συνάρτηση κόστους καθορίζει εάν η επόμενη διαδρομή είναι “άπληστη” ή “μη άπληστη”.

Πλεονεκτήματα

Οι άπληστοι αλγόριθμοι έχουν πολλά πλεονεκτήματα. Πρώτο και βασικότερο είναι η απλότητά τους. Είναι απλοί στην κατανόηση αλλά και εύκολοι στο σχεδιασμό και στην εφαρμογή τους. Οι άπληστοι αλγόριθμοι μπορούν να εφαρμοστούν σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης, συμπεριλαμβανομένων προβλημάτων συνδυαστικής βελτιστοποίησης και προβλημάτων προγραμματισμού (scheduling problems). Επίσης, μπορούν να χρησιμοποιηθούν για γρήγορη και αποτελεσματική επίλυση προβλήματος. Τέλος, σε πολλές περιπτώσεις, οι άπληστοι αλγόριθμοι μπορούν να παρέχουν καλές προσεγγίσεις στη βέλτιστη λύση, η οποία μπορεί να είναι επαρκής για πρακτικούς σκοπούς.

Μειονεκτήματα

Ωστόσο, υπάρχουν και αρνητικά. Μερικές φορές οι άπληστοι αλγόριθμοι αποτυγχάνουν να βρουν τη βέλτιστη λύση, επειδή δεν λαμβάνουν υπόψιν όλα τα δεδομένα. Η επιλογή που γίνεται από έναν άπληστο αλγόριθμο μπορεί να εξαρτάται από τις επιλογές που έχει κάνει μέχρι τώρα, αλλά δεν γνωρίζει τις μελλοντικές επιλογές που θα μπορούσε να κάνει. Υπάρχει έλλειψη ευελιξίας στους άπληστους αλγορίθμους, διότι ακολουθούν μια μονόδρομη διαδικασία λήψης αποφάσεων και μόλις επιλεγεί μια λύση δεν μπορεί να αλλάξει.

Τι υλοποιήθηκε

Υλοποιήθηκαν 2 άπληστοι αλγόριθμοι, ένας για την εύρεση της λύσης με το ελάχιστο κόστος και ένας για την εύρεση της πιο συμφέρουσας λύσης, όσον αναφορά τη ζωή της μπαταρίας. Ο κώδικας αναπτύχθηκε και τρέχει μέσω MATLAB. Με άλλα λόγια, αυτό που κάνει ο συγκεκριμένος άπληστος αλγόριθμος είναι να ελέγχει όλες τις λύσεις βήμα-βήμα και να υποδεικνύει την καλύτερη – φθηνότερη. Πέραν αυτών των διαφορετικών αλγορίθμων, όπως αναφέρθηκε παραπάνω, ανάλογα με το τι ζητάμε (μικρότερο κόστος και ζωή μπαταρίας), έχουν υλοποιηθεί και δύο παραλλαγές των αλγορίθμων. Η διαφορά έγκειται στα αρχικά δεδομένα και στον αριθμό των λύσεων που θα τρέξει ο αλγόριθμος. Στο πλαίσιο της εργασίας αυτής, υπάρχει εκδοχή με 20 λύσεις ανά σενάριο. Προφανώς, δεν

υπάρχει περιορισμός για παραπάνω λύσεις, εξαρτάται από το μέγεθος της βάσης δεδομένων.

Άπληστος αλγόριθμος – βέλτιστη λύση, λιγότερο κόστος.

Βήμα 1: Διαβάζει τα δεδομένα και δημιουργεί βοηθητικούς πίνακες

Τα δεδομένα αυτά είναι αποθηκευμένα σε .json αρχεία και περιέχουν τα σενάρια, τις λύσεις και τα κόστη των αισθητήρων. Μετά τη συλλογή των δεδομένων, ο αλγόριθμος δημιουργεί πίνακες από κάθε ένα από τα .json αρχεία.

Βήμα 2: Υπολογισμός κόστους

Σε αυτό το βήμα ο αλγόριθμος παίρνει τους πίνακες που δημιουργήθηκαν στο προηγούμενο βήμα και υπολογίζει το κόστος ανά λύση και ανά σενάριο σύμφωνα με το format της παραπάνω υποενότητας (problem formulation).

Βήμα 3: Προσπέλαση και επιλογή φθηνότερης λύσης

Σε αυτό το βήμα, βλέπουμε και την πραγματική υλοποίηση του άπληστου αλγόριθμου όπου προσπελαύνονται οι πίνακες, ελέγχεται το κόστος ανά βήμα και επιλέγεται αυτό με τη μικρότερη τιμή. Κάθε βήμα (step) του αλγορίθμου είναι και μια διαφορετική λύση. Επιλέγει σε μια μεταβλητή την μικρότερη τιμή και σε έναν πίνακα ταξινομημένες όλες τις μικρότερες τιμές που έχουν επιλεγεί προηγουμένως. Αυτό είναι και το τέλος του αλγορίθμου.

Στην συνέχεια, για να μπορέσουμε να προβάλουμε και να διαχειριστούμε τα αποτελέσματα, με σκοπό να τα ενσωματώσουμε στο εργαλείο που έχει υλοποιηθεί αυτό που γίνεται στον κώδικα είναι να δημιουργεί json files ανά σενάριο με τις βέλτιστες λύσεις ως προς το κόστος.

Ακολουθεί ο κώδικας.

```
clc;
clear;
close all;
%% GREEDY ALGORITHM: FIND THE BEST COST-EFFECTIVE SOLUTION FOR EACH SENARION

%% STEP 1: Fetch data from database which is in json format and creation of tables to help the process.

% Get data from 5g_solutions json file
jsonText_5g_solutions = fileread("Inputs-json\5g_scenario.json");
jsonData_5g_solutions = jsondecode(jsonText_5g_solutions); % Convert JSON formatted text to MATLAB data types
Table_5g_solutions = table(jsonData_5g_solutions);

% Get data from nb_solutions json file
jsonText_nb_solutions = fileread("Inputs-json\nb_scenario.json");
jsonData_nb_solutions = jsondecode(jsonText_nb_solutions); % Convert JSON formatted text to MATLAB data types
Table_nb_solutions = table(jsonData_nb_solutions);

% Get data from lora_solutions json file
jsonText_lora_solutions = fileread("Inputs-json\lora_scenario.json");
jsonData_lora_solutions = jsondecode(jsonText_lora_solutions); % Convert JSON formatted text to MATLAB data types
Table_lora_solutions = table(jsonData_lora_solutions);

% Get cost data from json file
jsonText_Cost = fileread("Inputs-json\costs.json");
jsonData_Cost = jsondecode(jsonText_Cost); % Convert JSON formatted text to MATLAB data types
jsonDataCostTable = table(jsonData_Cost);
```

```
%% STEP 2: Access the solution tables and Cost calculation
```

```
% 5G Scenario
```

```
for j=1:1:20
```

```
    numberOf5gSensorsTypeA = Table_5g_solutions.jsonData_5g_solutions(j,1).numberOf5gSensorsTypeA;  
    cost_5g_type_a = jsonDataCostTable.jsonData_Cost.cost_5g_type_a;  
    cost_a = numberOf5gSensorsTypeA * cost_5g_type_a;  
    installation_cost_5g_type_a = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_5g_type_a;  
    costA = (cost_a + installation_cost_5g_type_a);
```

```
    numberOf5gSensorsTypeB = Table_5g_solutions.jsonData_5g_solutions(j,1).numberOf5gSensorsTypeB;  
    cost_5g_type_b = jsonDataCostTable.jsonData_Cost.cost_5g_type_b;  
    cost_b = numberOf5gSensorsTypeB * cost_5g_type_b;  
    installation_cost_5g_type_b = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_5g_type_b;  
    costB = (cost_b + installation_cost_5g_type_b);
```

```
    numberOf5gSensorsTypeC = Table_5g_solutions.jsonData_5g_solutions(j,1).numberOf5gSensorsTypeC;  
    cost_5g_type_c = jsonDataCostTable.jsonData_Cost.cost_5g_type_c;  
    cost_c = numberOf5gSensorsTypeC * cost_5g_type_c;  
    installation_cost_5g_type_c = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_5g_type_c;  
    costC = (cost_c + installation_cost_5g_type_c);
```

```
    solution_5g_FinalCost(j) = costA + costB + costC;
```

```
end
```

```
% LORA Scenario
```

```
for j=1:1:20
```

```
    numberOfLoraSensorsTypeA = Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfLoraSensorsTypeA;  
    cost_lora_type_a = jsonDataCostTable.jsonData_Cost.cost_lora_type_a;  
    cost_a = numberOfLoraSensorsTypeA * cost_lora_type_a;  
    installation_cost_lora_type_a = jsonDataCostTable.jsonData_Cost.installation_cost_lora_type_a;  
    costA = (cost_a + installation_cost_lora_type_a);
```

```
    numberOfLoraSensorsTypeB = Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfLoraSensorsTypeB;  
    cost_lora_type_b = jsonDataCostTable.jsonData_Cost.cost_lora_type_b;  
    cost_b = numberOfLoraSensorsTypeB * cost_lora_type_b;  
    installation_cost_lora_type_b = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_lora_type_b;  
    costB = (cost_b + installation_cost_lora_type_b);
```

```
    numberOfLoraSensorsTypeC = Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfLoraSensorsTypeC;  
    cost_lora_type_c = jsonDataCostTable.jsonData_Cost.cost_lora_type_c;  
    cost_c = numberOfLoraSensorsTypeC * cost_lora_type_c;  
    installation_cost_lora_type_c = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_lora_type_c;  
    costC = (cost_c + installation_cost_lora_type_c);
```

```
    numberOfGatewaysTypeA = Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfGatewaysTypeA;  
    cost_lora_gateway_type_a = jsonDataCostTable.jsonData_Cost.cost_lora_gateway_type_a;  
    cost_g_a = numberOfGatewaysTypeA * cost_lora_gateway_type_a;  
    installation_lora_gateway_type_a = jsonDataCostTable.jsonData_Cost(j,1).installation_lora_gateway_type_a;  
    costGA = (cost_g_a + installation_lora_gateway_type_a);
```

```
    numberOfGatewaysTypeB = Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfGatewaysTypeB;  
    cost_lora_gateway_type_b = jsonDataCostTable.jsonData_Cost.cost_lora_gateway_type_b;  
    cost_g_b = numberOfGatewaysTypeB * cost_lora_gateway_type_b;  
    installation_lora_gateway_type_b = jsonDataCostTable.jsonData_Cost(j,1).installation_lora_gateway_type_b;  
    costGB = (cost_g_b + installation_lora_gateway_type_b);
```

```
    solution_lora_FinalCost(j) = costA + costB + costC + costGA + costGB;
```

```
    % solution_lora_FinalCost(j) = costA + costB + costC;
```

```
end
```

```
% NB-IoT Scenario
```

```

for j=1:1:20

    numberOfNBSensorsTypeA = Table_nb_solutions.jsonData_nb_solutions(j,1).numberOfNBSensorsTypeA;
    cost_nb_type_a = jsonDataCostTable.jsonData_Cost(j,1).cost_nb_type_a;
    cost_a = numberOfNBSensorsTypeA * cost_nb_type_a;
    installation_cost_nb_type_a = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_nb_type_a;
    costA = (cost_a + installation_cost_nb_type_a);

    numberOfNBSensorsTypeB = Table_nb_solutions.jsonData_nb_solutions(j,1).numberOfNBSensorsTypeB;
    cost_nb_type_b = jsonDataCostTable.jsonData_Cost(j,1).cost_nb_type_b;
    cost_b = numberOfNBSensorsTypeB * cost_nb_type_b;
    installation_cost_nb_type_b = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_nb_type_b;
    costB = (cost_b + installation_cost_nb_type_b);

    numberOfNBSensorsTypeC = Table_nb_solutions.jsonData_nb_solutions(j,1).numberOfNBSensorsTypeC;
    cost_nb_type_c = jsonDataCostTable.jsonData_Cost(j,1).cost_nb_type_c;
    cost_c = numberOfNBSensorsTypeC * cost_nb_type_c;
    installation_cost_nb_type_c = jsonDataCostTable.jsonData_Cost(j,1).installation_cost_nb_type_c;
    costC = (cost_c + installation_cost_nb_type_c);

    solution_nb_FinalCost(j) = costA + costB + costC;
end

%% Calling the algorithm

% Each function's parameter is a table with the final cost for each scenario and for each solution
find_the_cheapest(solution_5g_FinalCost, solution_lora_FinalCost, solution_nb_FinalCost, Table_5g_solutions, Table_nb_solutions, Table_lora_solutions);

%% STEP 3: Start of Greedy Algorithm - Finds the best solutions for each scenario
function find_the_cheapest(solution_5g_FinalCost, solution_lora_FinalCost, solution_nb_FinalCost, Table_5g_solutions, Table_nb_solutions, Table_lora_solutions)
    minimum_cost_5g = solution_5g_FinalCost(1);
    minimum_cost_lora = solution_lora_FinalCost(1);
    minimum_cost_nb = solution_nb_FinalCost(1);
    c1 = 0;
    c2 = 0;
    c3 = 0;
    for l=1:1:20
        if solution_5g_FinalCost(l) < minimum_cost_5g
            c1 = c1 + 1;
            minimum_cost_5g = solution_5g_FinalCost(l);
            cheapest_5g_solution = l;
            cheapest_5g_solutionTable(c1) = cheapest_5g_solution;
            cheapest_5g_solutionTableCost(c1) = minimum_cost_5g;
        elseif solution_5g_FinalCost(l) == minimum_cost_5g
            c1 = c1 + 1;
            minimum_cost_5g = solution_5g_FinalCost(l);
            cheapest_5g_solution = l;
            cheapest_5g_solutionTable(c1) = cheapest_5g_solution;
            cheapest_5g_solutionTableCost(c1) = minimum_cost_5g;
        end
        if solution_lora_FinalCost(l) < minimum_cost_lora
            c2 = c2 + 1;
            minimum_cost_lora = solution_lora_FinalCost(l);
            cheapest_lora_solution = l;
            cheapest_lora_solutionTable(c2) = cheapest_lora_solution;
            cheapest_lora_solutionTableCost(c2) = minimum_cost_lora;
        elseif solution_lora_FinalCost(l) == minimum_cost_lora
            c2 = c2 + 1;
            minimum_cost_lora = solution_lora_FinalCost(l);
            cheapest_lora_solution = l;
            cheapest_lora_solutionTable(c2) = cheapest_lora_solution;
            cheapest_lora_solutionTableCost(c2) = minimum_cost_lora;
        end
        if solution_nb_FinalCost(l) < minimum_cost_nb
            c3 = c3 + 1;

```

```

        minimum_cost_nb = solution_nb_FinalCost(1);
        cheapest_nb_solution = 1;
        cheapest_nb_solutionTable(c3) = cheapest_nb_solution;
        cheapest_nb_solutionTableCost(c3) = minimum_cost_nb;
    elseif solution_nb_FinalCost(1) == minimum_cost_nb
        c3 = c3 + 1;
        minimum_cost_nb = solution_nb_FinalCost(1);
        cheapest_nb_solution = 1;
        cheapest_nb_solutionTable(c3) = cheapest_nb_solution;
        cheapest_nb_solutionTableCost(c3) = minimum_cost_nb;
    end
end
json_file_generator_greedy(cheapest_5g_solutionTable,cheapest_5g_solutionTableCost, ...
    cheapest_lora_solutionTable,cheapest_lora_solutionTableCost,cheapest_nb_solutionTable,cheapest_nb_solutionTableCost,
    Table_5g_solutions, Table_nb_solutions, Table_lora_solutions);
end

```

%% - Start creating json files!

```

function json_file_generator_greedy(cheapest_5g_solutionTable,cheapest_5g_solutionTableCost, ...
    cheapest_lora_solutionTable,cheapest_lora_solutionTableCost,cheapest_nb_solutionTable, ...
    cheapest_nb_solutionTableCost, Table_5g_solutions, Table_nb_solutions, Table_lora_solutions)

```

```

cheapest_5g_solutionTable = cheapest_5g_solutionTable;
solutions_5g = table(cheapest_5g_solutionTable);
%   for l = 1:length(solutions_5g)
%       solution_number = solutions_5g(l);
%       solutions_5g_data(l) = Table_5g_solutions(l);
%   end
solution_5g_cost = table(cheapest_5g_solutionTableCost);
cheapest_lora_solutionTable = table(cheapest_lora_solutionTable);
cheapest_lora_solutionTableCost = table(cheapest_lora_solutionTableCost);
cheapest_nb_solutionTable = table(cheapest_nb_solutionTable);
cheapest_nb_solutionTableCost = table(cheapest_nb_solutionTableCost);

% cost-effective (cheapest) 5g solution (sort-by-cost)

jsonText = jsonencode(solution_5g_cost); % Convert to JSON text
jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':', sprintf(:[\r\t\t]));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, '}', sprintf('\r}'));
jsonText = strrep(jsonText, ']', sprintf('\r\t]'));
fid = fopen('cost-effective-5g-solutions_by_cost.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);

movefile('cost-effective-5g-solutions_by_cost.json', 'Results\Greedy')

% cost-effective (cheapest) 5g solution (sort-by-solution)
jsonText = jsonencode(solutions_5g); % Convert to JSON text
jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':', sprintf(:[\r\t\t]));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, '}', sprintf('\r}'));
jsonText = strrep(jsonText, ']', sprintf('\r\t]'));
fid = fopen('cost-effective-5g-solutions_by_solution.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);

movefile('cost-effective-5g-solutions_by_solution.json', 'Results\Greedy')

% cost-effective (cheapest) LoRa solution (sort-by-cost)
jsonText = jsonencode(cheapest_lora_solutionTableCost); % Convert to JSON text
jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':', sprintf(:[\r\t\t]));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));

```

```

jsonText = strrep(jsonText, '}}', sprintf('\r'));
jsonText = strrep(jsonText, ']', sprintf('\r\t'));

fid = fopen('cost-effective-lora-solutions_by_cost.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);

movefile('cost-effective-lora-solutions_by_cost.json', 'Results\Greedy')

% cost-effective (cheapest) LoRa solution (sort-by-solution)
jsonText = jsonencode(cheapest_lora_solutionTable); % Convert to JSON text
jsonText = strrep(jsonText, '[[', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':[', sprintf('{:\r\t\t'));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, ']]', sprintf('\r'));
jsonText = strrep(jsonText, ']', sprintf('\r\t'));

fid = fopen('cost-effective-lora-solutions_by_solution.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);

movefile('cost-effective-lora-solutions_by_solution.json', 'Results\Greedy')

% cost-effective (cheapest) NB-IoT solution (sort-by-cost)
jsonText = jsonencode(cheapest_nb_solutionTableCost); % Convert to JSON text
jsonText = strrep(jsonText, '[[', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':[', sprintf('{:\r\t\t'));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, ']]', sprintf('\r'));
jsonText = strrep(jsonText, ']', sprintf('\r\t'));

fid = fopen('cost-effective-nb-solutions_by_cost.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);

movefile('cost-effective-nb-solutions_by_cost.json', 'Results\Greedy')

% cost-effective (cheapest) NB-IoT solution (sort-by-solution)
jsonText = jsonencode(cheapest_nb_solutionTable); % Convert to JSON text
jsonText = strrep(jsonText, '[[', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':[', sprintf('{:\r\t\t'));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, ']]', sprintf('\r'));
jsonText = strrep(jsonText, ']', sprintf('\r\t'));
%jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
% jsonText = strrep(jsonText, '[{', sprintf('[\r\t\r'));
% jsonText = strrep(jsonText, '}', sprintf('\r\r'));
%jsonText = strrep(jsonText, '}', sprintf('\r'));

fid = fopen('cost-effective-nb-solutions_by_solution.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);

movefile('cost-effective-nb-solutions_by_solution.json', 'Results\Greedy')

disp('Completed! all files was saved');

```

end

Άπληστος αλγόριθμος – βέλτιστη ενεργειακά λύση, ζωή μπαταρίας - καλύτερη απόδοση μπαταρίας.

Τα βήματα και η όλη διαδικασία όσον αφορά την προετοιμασία των δεδομένων, την τελική διαχείριση των αποτελεσμάτων, αλλά και τον ίδιο τον άπληστο αλγόριθμο είναι ίδια με τα παραπάνω βήματα που αναφέρθηκαν. Το μόνο που αλλάζει είναι το ζητούμενο, εννοώντας ότι πλέον ασχολούμαστε με την εύρεση της ενέργειας, δηλαδή τη ζωή της μπαταρίας για κάθε λύση σε κάθε διαφορετικό σενάριο κι όχι του κόστους, οπότε σαν δεδομένα αντί για κόστος παίρνουμε χωρητικότητα μπαταρίας και κατανάλωση μπαταρίας.

Ακολουθεί ο κώδικας.

```
clc;
clear;
close all;
%% GREEDY ALGORITHM FIND THE BEST ENERGY-EFFECTIVE SOLUTION FOR EACH SENARION

%% STEP 1: Fetch data from database which is in json format and creation of tables to help the process.

% Get data from 5g_solutions json file
jsonText_5g_solutions = fileread("Inputs-json\5g_scenario.json");
jsonData_5g_solutions = jsondecode(jsonText_5g_solutions); % Convert JSON formatted text to MATLAB data types
Table_5g_solutions = table(jsonData_5g_solutions);

% Get data from nb_solutions json file
jsonText_nb_solutions = fileread("Inputs-json\nb_scenario.json");
jsonData_nb_solutions = jsondecode(jsonText_nb_solutions); % Convert JSON formatted text to MATLAB data types
Table_nb_solutions = table(jsonData_nb_solutions);
% Get data from lora_solutions json file
jsonText_lora_solutions = fileread("Inputs-json\lora_scenario.json");
jsonData_lora_solutions = jsondecode(jsonText_lora_solutions); % Convert JSON formatted text to MATLAB data types
Table_lora_solutions = table(jsonData_lora_solutions);

% Get cost data from json file
jsonText_Battery = fileread("Inputs-json\battery.json");
jsonData_Battery = jsondecode(jsonText_Battery); % Convert JSON formatted text to MATLAB data types
jsonDataBatteryTable = table(jsonData_Battery);

%% STEP 2: Access the solution tables and Cost calculation

% 5G Scenario
for j=1:1:20

    if(Table_5g_solutions.jsonData_5g_solutions(j,1).numberOf5gSensorsTypeA > 0)
        battery_capacity_5g_a = double(jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_5g_type_a);
        battery_consumption_5g_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_5g_type_a;
        battery_life_5g_a = battery_capacity_5g_a / battery_consumption_5g_a;
    else
        battery_life_5g_a = 0;
    end

    if(Table_5g_solutions.jsonData_5g_solutions(j,1).numberOf5gSensorsTypeB > 0)
        battery_capacity_5g_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_5g_type_b;
        battery_consumption_5g_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_5g_type_b;
        battery_life_5g_b = battery_capacity_5g_b / battery_consumption_5g_b;
    else
        battery_life_5g_b = 0;
    end

    if(Table_5g_solutions.jsonData_5g_solutions(j,1).numberOf5gSensorsTypeC > 0)
        battery_capacity_5g_c = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_5g_type_c;
        battery_consumption_5g_c = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_5g_type_c;
        battery_life_5g_c = battery_capacity_5g_c / battery_consumption_5g_c;
    end
end
```



```

else
    battery_life_5g_c = 0;
end
battery_life_5g_solution(j) = (battery_life_5g_a + battery_life_5g_b + battery_life_5g_c)/3;
end

% LORA Scenario
for j=1:1:20
if(Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfLoraSensorsTypeA > 0)
    battery_capacity_lora_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_lora_type_a;
    battery_consumption_lora_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_lora_type_a;
    battery_life_lora_a = battery_capacity_lora_a / battery_consumption_lora_a;
else
    battery_life_lora_a = 0;
end
if(Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfLoraSensorsTypeB > 0)
    battery_capacity_lora_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_lora_type_b;
    battery_consumption_lora_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_lora_type_b;
    battery_life_lora_b = battery_capacity_lora_b / battery_consumption_lora_b;
else
    battery_life_lora_b = 0;
end

if(Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfLoraSensorsTypeC > 0)
    battery_capacity_lora_c = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_lora_type_c;
    battery_consumption_lora_c = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_lora_type_c;
    battery_life_lora_c = battery_capacity_lora_c / battery_consumption_lora_c;
else
    battery_life_lora_c = 0;
end
sensors_battery_life_lora_solution(j) = (battery_life_lora_a + battery_life_lora_b + battery_life_lora_c)/3;

if(Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfGatewaysTypeA > 0)
    battery_capacity_lora_gateway_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_lora_gateway_type_a;
    battery_consumption_lora_gateway_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_lora_gateway_type_a;
    battery_life_lora_gateway_a = battery_capacity_lora_gateway_a / battery_consumption_lora_gateway_a;
else
    battery_life_lora_gateway_a = 0;
end

if(Table_lora_solutions.jsonData_lora_solutions(j,1).numberOfGatewaysTypeB > 0)
    battery_capacity_lora_gateway_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_lora_gateway_type_b;
    battery_consumption_lora_gateway_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_lora_gateway_type_b;
    battery_life_lora_gateway_b = battery_capacity_lora_gateway_b / battery_consumption_lora_gateway_b;
else
    battery_life_lora_gateway_b = 0;
end
gateway_battery_life_lora_solution(j) = (battery_life_lora_gateway_a + battery_life_lora_gateway_b)/2;
end

% NB-IoT Scenario
for j=1:1:20
if(Table_nb_solutions.jsonData_nb_solutions(1,1).numberOfNBSensorsTypeA > 0)
    battery_capacity_nb_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_nb_type_a;
    battery_consumption_nb_a = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_nb_type_a;
    battery_life_nb_a = battery_capacity_nb_a / battery_consumption_nb_a;
else
    battery_life_nb_a = 0;
end

if(Table_nb_solutions.jsonData_nb_solutions(1,1).numberOfNBSensorsTypeB > 0)
    battery_capacity_nb_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_nb_type_b;
    battery_consumption_nb_b = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_nb_type_b;

```

```

        battery_life_nb_b = battery_capacity_nb_b / battery_consumption_nb_b;
else
    battery_life_nb_b = 0;
end

if(Table_nb_solutions.jsonData_nb_solutions(1,1).numberOfNBSensorsTypeC > 0)
    battery_capacity_nb_c = jsonDataBatteryTable.jsonData_Battery(j,1).battery_capacity_nb_type_c;
    battery_consumption_nb_c = jsonDataBatteryTable.jsonData_Battery(j,1).battery_consumption_nb_type_c;
    battery_life_nb_c = battery_capacity_nb_c / battery_consumption_nb_c;
else
    battery_life_nb_c = 0;
end
battery_life_nb_solution(j) = (battery_life_nb_a + battery_life_nb_b + battery_life_nb_c)/3;
end

```

%% Calling the algorithm

% Each function's parameter is a table with the final cost for each scenario and for each solution

```

find_the_best(battery_life_5g_solution, sensors_battery_life_lora_solution,
gateway_battery_life_lora_solution,battery_life_nb_solution);

```

%% STEP 3: Start of Greedy Algorithm - Finds the best solutions for each scenario

```

function find_the_best(battery_life_5g_solution, sensors_battery_life_lora_solution,
gateway_battery_life_lora_solution,battery_life_nb_solution)
    maximum_battery_life_5g_solution = battery_life_5g_solution(1);
    maximum_sensors_battery_life_lora_solution = sensors_battery_life_lora_solution(1);
    maximum_gateway_battery_life_lora_solution = gateway_battery_life_lora_solution(1);
    maximum_battery_life_nb_solution = battery_life_nb_solution(1);
    c1 = 0;
    c2 = 0;
    c3 = 0;
    c4 = 0;
    for l=1:1:20
        if battery_life_5g_solution(l) > maximum_battery_life_5g_solution
            c1 = c1 + 1;
            maximum_battery_life_5g_solution = battery_life_5g_solution(l);
            best_5g_solution = l;
            best_5g_solutionTable(c1) = best_5g_solution;
            best_5g_solutionTableBL(c1) = maximum_battery_life_5g_solution;
        elseif battery_life_5g_solution(l) == maximum_battery_life_5g_solution
            c1 = c1 + 1;
            maximum_battery_life_5g_solution = battery_life_5g_solution(l);
            best_5g_solution = l;
            best_5g_solutionTable(c1) = best_5g_solution;
            best_5g_solutionTableBL(c1) = maximum_battery_life_5g_solution;
        end

        if sensors_battery_life_lora_solution(l) > maximum_sensors_battery_life_lora_solution
            c2 = c2 + 1;
            maximum_sensors_battery_life_lora_solution = sensors_battery_life_lora_solution(l);
            best_lora_solution = l;
            best_lora_solutionTable(c2) = best_lora_solution;
            best_lora_solutionTableBL(c2) = maximum_sensors_battery_life_lora_solution;
            best_lora_solutionTableBL_GatewaysBL(c2) = gateway_battery_life_lora_solution(l);
        elseif sensors_battery_life_lora_solution(l) == maximum_sensors_battery_life_lora_solution
            c2 = c2 + 1;
            maximum_sensors_battery_life_lora_solution = sensors_battery_life_lora_solution(l);
            best_lora_solution = l;
            best_lora_solutionTable(c2) = best_lora_solution;
            best_lora_solutionTableBL(c2) = maximum_sensors_battery_life_lora_solution;
            best_lora_solutionTableBL_GatewaysBL(c2) = gateway_battery_life_lora_solution(l);
        end

        if gateway_battery_life_lora_solution(l) > maximum_gateway_battery_life_lora_solution
            c4 = c4 + 1;
            maximum_gateway_battery_life_lora_solution = gateway_battery_life_lora_solution(l);
            best_lora_gateway_solution = l;

```

```

        best_lora_gateway_solutionTable(c4) = best_lora_gateway_solution;
        best_lora_gateway_solutionTableBL(c4) = maximum_gateway_battery_life_lora_solution;
        best_lora_gateway_solutionTableBL_sensorsBL(c4) = sensors_battery_life_lora_solution(1);
elseif gateway_battery_life_lora_solution(1) == maximum_gateway_battery_life_lora_solution
    c4 = c4 + 1;
    maximum_gateway_battery_life_lora_solution = gateway_battery_life_lora_solution(1);
    best_lora_gateway_solution = 1;
    best_lora_gateway_solutionTable(c4) = best_lora_gateway_solution;
    best_lora_gateway_solutionTableBL(c4) = maximum_gateway_battery_life_lora_solution;
    best_lora_gateway_solutionTableBL_sensorsBL(c4) = sensors_battery_life_lora_solution(1);
end

if battery_life_nb_solution(1) > maximum_battery_life_nb_solution
    c3 = c3 + 1;
    maximum_battery_life_nb_solution = battery_life_nb_solution(1);
    best_nb_solution = 1;
    best_nb_solutionTable(c3) = best_nb_solution;
    best_nb_solutionTableBL(c3) = maximum_battery_life_nb_solution;
elseif battery_life_nb_solution(1) == maximum_battery_life_nb_solution
    c3 = c3 + 1;
    maximum_battery_life_nb_solution = battery_life_nb_solution(1);
    best_nb_solution = 1;
    best_nb_solutionTable(c3) = best_nb_solution;
    best_nb_solutionTableBL(c3) = maximum_battery_life_nb_solution;
end
end

json_file_generator_greedy(best_5g_solutionTable,best_5g_solutionTableBL, ...
    best_lora_solutionTable,best_lora_solutionTableBL,best_lora_solutionTableBL_GatewaysBL,
best_lora_gateway_solutionTable, ...
    best_lora_gateway_solutionTableBL,
best_lora_gateway_solutionTableBL_sensorsBL,best_nb_solutionTable,best_nb_solutionTableBL);
end

%% - Start creating json files!
function json_file_generator_greedy(best_5g_solutionTable,best_5g_solutionTableBL, ...
    best_lora_solutionTable,best_lora_solutionTableBL,best_lora_solutionTableBL_GatewaysBL,
best_lora_gateway_solutionTable, ...
    best_lora_gateway_solutionTableBL,
best_lora_gateway_solutionTableBL_sensorsBL,best_nb_solutionTable,best_nb_solutionTableBL);

best_solutions_5g = table(best_5g_solutionTable);
best_solution_5g_battery_life = table(best_5g_solutionTableBL);

best_solutions_lora = table(best_lora_solutionTable);
best_solutions_lora_gateways = table(best_lora_gateway_solutionTable);
best_solution_lora_battery_life = table(best_lora_solutionTableBL,best_lora_solutionTableBL_GatewaysBL);
best_solution_lora_gateways_battery_life =
table(best_lora_gateway_solutionTableBL,best_lora_gateway_solutionTableBL_sensorsBL);

best_solutions_nb = table(best_nb_solutionTable);
best_solution_nb_battery_life = table(best_nb_solutionTableBL);

% battery-effective 5g solution (sort-by-solution)
jsonText = jsonencode(best_solutions_5g); % Convert to JSON text
jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':', sprintf(:[\r\t\t]));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, '}', sprintf('\r}'));
jsonText = strrep(jsonText, ']', sprintf('\r\t]'));
fid = fopen('battery-effective-5g-solutions_by_solution.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);
movefile('battery-effective-5g-solutions_by_solution.json','Results\Greedy')

% battery-effective 5g solution (sort-by-battery life)

```

```

    jsonText = jsonencode(best_solution_5g_battery_life); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t'));
    fid = fopen('battery-effective-5g-solutions_by_battery_life.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);
    movefile('battery-effective-5g-solutions_by_battery_life.json', 'Results\Greedy')

% battery-effective lora solution (sort-by-solution and sensors efficiency)
    jsonText = jsonencode(best_solutions_lora); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t'));
    fid = fopen('battery-effective-lora-solutions_by_solution.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);
    movefile('battery-effective-lora-solutions_by_solution.json', 'Results\Greedy')

% battery-effective lora solution (sort-by-battery life)
    jsonText = jsonencode(best_solution_lora_battery_life); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t'));
    fid = fopen('battery-effective-lora-solutions_by_battery_life_sensors.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);
    movefile('battery-effective-lora-solutions_by_battery_life_sensors.json', 'Results\Greedy')

% battery-effective lora solution (sort-by-solution and gateways efficiency)
    jsonText = jsonencode(best_solutions_lora_gateways); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t'));
    fid = fopen('battery-effective-lora-solutions_by_solution_gateways.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);
    movefile('battery-effective-lora-solutions_by_solution_gateways.json', 'Results\Greedy')

% battery-effective lora solution (sort-by-battery life) - GATEWAYS
    jsonText = jsonencode(best_solution_lora_gateways_battery_life); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t'));
    fid = fopen('battery-effective-lora-solutions_by_battery_life_gateways.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);
    movefile('battery-effective-lora-solutions_by_battery_life_gateways.json', 'Results\Greedy')

% battery-effective 5g solution (sort-by-solution)
    jsonText = jsonencode(best_solutions_nb); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t'));
    fid = fopen('battery-effective-nb-solutions_by_solution.json', 'w'); % Write to a json file

```

```

fprintf(fid, '%s', jsonText);
fclose(fid);
movefile('battery-effective-nb-solutions_by_solution.json', 'Results\Greedy')

% battery-effective 5g solution (sort-by-battery life)
jsonText = jsonencode(best_solution_nb_battery_life); % Convert to JSON text
jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
jsonText = strrep(jsonText, ':', sprintf('{\r\t\t'));
jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
jsonText = strrep(jsonText, '}', sprintf('\r}'));
jsonText = strrep(jsonText, ']', sprintf('\r\t]'));
fid = fopen('battery-effective-nb-solutions_by_battery_life.json', 'w'); % Write to a json file
fprintf(fid, '%s', jsonText);
fclose(fid);
movefile('battery-effective-nb-solutions_by_battery_life.json', 'Results\Greedy')

disp('Completed! all files was saved');

end

```

4.2.2 Σχετικά με Genetic Algorithm – Γενετικός Αλγόριθμος

Γενικά

Οι γενετικοί αλγόριθμοι (GAs – Genetic Algorithms) [20] είναι προσαρμοστικοί – ευριστικοί (heuristic) αλγόριθμοι αναζήτησης που ανήκουν στο μεγαλύτερο μέρος των εξελικτικών αλγορίθμων. Οι γενετικοί αλγόριθμοι βασίζονται στις ιδέες της φυσικής επιλογής στην βιολογία και της γενετικής. Πρόκειται για έξυπνη εκμετάλλευση της τυχαίας αναζήτησης που παρέχεται με ιστορικά δεδομένα για να κατευθύνει την αναζήτηση στην περιοχή καλύτερης απόδοσης στο χώρο λύσεων. Χρησιμοποιούνται συνήθως για τη δημιουργία λύσεων υψηλής ποιότητας για προβλήματα αναζήτησης. Ο αλγόριθμος χρησιμοποιεί μια σειρά κανόνων, που είναι γνωστοί ως "genetic operator", για να χειριστούν τη γενετική πληροφορία που κωδικοποιήθηκε, όπως η μετάλλαξη (mutation) και ο διασταύρωση (crossover). Σε κάθε επανάληψη, ο γενετικός αλγόριθμος αξιολογεί κάθε λύση (solution) του πληθυσμού (δηλ. το πλήθος όλων των δηγμάτων – λύσεων) και επιλέγονται οι πιο ικανοί γονείς για αναπαραγωγή. Η γενετική πληροφορία αυτών των προτάσεων – λύσεων στη συνέχεια συνδέεται για να δημιουργηθούν νέα στοιχεία που αντιπροσωπεύουν νέες λύσεις για την επόμενη επανάληψη. Οι γενετικοί αλγόριθμοι προσομοιώνουν τη διαδικασία της φυσικής επιλογής, πράγμα που σημαίνει ότι εκείνα τα είδη που μπορούν να προσαρμοστούν στις αλλαγές του περιβάλλοντος, μπορούν να επιβιώσουν, να αναπαραχθούν και να πάνε στην επόμενη γενιά. Με απλά λόγια, προσομοιώνουν την "επιβίωση του ισχυρότερου" μεταξύ ατόμων για την επίλυση ενός προβλήματος. Κάθε γενιά αποτελείται από έναν **πληθυσμό ατόμων** και κάθε άτομο αντιπροσωπεύει ένα σημείο στον χώρο αναζήτησης και μια **πιθανή λύση**. Κάθε άτομο αναπαρίσταται ως μια **συμβολοσειρά** χαρακτήρων/ακεραίων/float/bit. Αυτή η συμβολοσειρά είναι ανάλογη με το **χρωμόσωμα**. Τα χρωμοσώματα αποτελούνται από διάφορα στοιχεία που ονομάζονται **γονίδια**. Κάθε γονίδιο επηρεάζει την **κληρονομικότητα** ενός ή περισσοτέρων χαρακτηριστικών. Ο πληθυσμός έχει ένα σταθερό μέγεθος. Καθώς σχηματίζονται νέες γενιές, τα άτομα με τη λιγότερη φυσική κατάσταση πεθαίνουν, παρέχοντας χώρο για νέους απογόνους. Η ακολουθία των φάσεων επαναλαμβάνεται για να παραχθούν άτομα σε κάθε

νέα γενιά τα οποία είναι καλύτερα από την προηγούμενη γενιά. Αυτή η διαδικασία συνεχίζεται μέχρι να βρεθεί μια ικανοποιητική λύση ή μέχρι να περατωθεί ένα προκαθορισμένο κριτήριο διακοπής. Οι γενετικοί αλγόριθμοι χρησιμοποιούνται συχνά σε μια ευρεία περιοχή πεδίων, συμπεριλαμβανομένων της μηχανικής, τα χρηματοοικονομικά και τη μηχανική μάθηση, για να λύσουν περίπλοκα προβλήματα βελτίωσης.

Η έννοια της φυσικής επιλογής

Η διαδικασία της φυσικής επιλογής ξεκινά με την επιλογή των πιο ικανών ατόμων από έναν πληθυσμό. Παράγουν απογόνους που κληρονομούν τα χαρακτηριστικά των γονέων και θα προστεθούν στην επόμενη γενιά. Εάν οι γονείς έχουν καλύτερη φυσική κατάσταση (fitness), οι απόγονοι τους θα είναι καλύτεροι από τους γονείς και θα έχουν περισσότερες πιθανότητες να επιβιώσουν.

Αυτή η έννοια, μπορεί να εφαρμοστεί για ένα πρόβλημα αναζήτησης, όπου εξετάζεται ένα σύνολο λύσεων για ένα πρόβλημα και επιλέγεται το σύνολο των καλύτερων από αυτές.

Λογική και στάδια των γενετικών αλγορίθμων

Οι γενετικοί αλγόριθμοι αποτελούνται από 5 φάσεις.

1. Αρχικός πληθυσμός (Initial population)

Η διαδικασία ξεκινά με τη δημιουργία ενός σύνολο ατόμων που ονομάζεται πληθυσμός (population). Κάθε άτομο είναι μια λύση στο πρόβλημα που θέλουμε να λύσουμε. Ένα άτομο χαρακτηρίζεται από ένα σύνολο παραμέτρων (μεταβλητών) γνωστών ως γονίδια. Τα γονίδια ενώνονται για να σχηματίσουμε ένα χρωμόσωμα. Σε έναν γενετικό αλγόριθμο, το σύνολο των γονιδίων ενός ατόμου αναπαρίσταται χρησιμοποιώντας μια συμβολοσειρά. Συνήθως είναι δυαδικές τιμές (συμβολοσειρά 1 και 0 - binary), αλλά μπορεί να αναπαρασταθεί και με άλλους τρόπος, όπως ακέραιοι πραγματικοί αριθμοί. Βέβαια, οι μη δυαδικές ακολουθίες λύσεων αναλύονται δυσκολότερα στην θεωρία, αλλά στην πράξη αποδίδουν καλύτερα.

Κωδικοποίηση ενός χρωμοσώματος (Chromosomes Encoding)

Η επιλογή της κωδικοποίησης είναι το πιο σημαντικό για να επιλεγεί σωστά από όποιον θέλει να επιλύσει κάποιο πρόβλημα με χρήση ενός GA. Με άλλα λόγια η κωδικοποίηση βασίζεται και εξαρτάται από το πρόβλημα. Το χρωμόσωμα θα πρέπει κατά κάποιο τρόπο να περιέχει πληροφορίες σχετικά με τη λύση την οποία αντιπροσωπεύει.

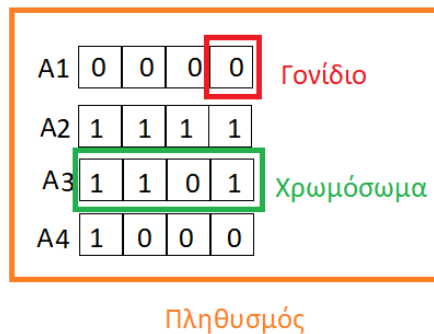
Δυαδική κωδικοποίηση (Binary Encoding): Όπως αναφέρθηκε και παραπάνω, είναι ο πιο συνηθισμένος τρόπος κωδικοποίησης. Κάθε χρωμόσωμα είναι μια συμβολοσειρά χαρακτήρων (string of bits) 0 ή 1.

Χρωμόσωμα A	101100101100101
Χρωμόσωμα B	111111100000101

Εικόνα 11 Παράδειγμα χρωμοσωμάτων με binary κωδικοποίηση

Η δυαδική κωδικοποίηση δίνει πολλά διαφορετικά χρωμοσώματα και δυνατότητες. Από την άλλη μεριά αυτή η κωδικοποίηση δεν είναι ιδανική για πολλά προβλήματα και μερικές φορές

πρέπει να γίνονται διορθώσεις μετά από τη διασταύρωση (crossover) ή και τη μετάλλαξη mutation.



Εικόνα 12 Απεικόνιση χρωμοσωμάτων [21]

Κωδικοποίηση μετάθεσης (Permutation Encoding): Σε αυτήν την κωδικοποίηση, κάθε χρωμόσωμα είναι μια συμβολοσειρά αριθμών, η οποία αντιπροσωπεύει τον αριθμό σε μια ακολουθία.

Χρωμόσωμα A	1 5 3 2 6 4 7 9
Χρωμόσωμα B	8 5 6 7 3 2 1 4

Εικόνα 13 Παράδειγμα χρωμοσώματος με Permutation κωδικοποίηση

Value Encoding: Σε αυτό το είδος κωδικοποίησης, κάθε χρωμόσωμα είναι μια συμβολοσειρά κάποιων τιμών. Οι τιμές αυτές μπορεί να είναι πραγματικοί αριθμοί ή χαρακτήρες.

Χρωμόσωμα A	1.232 5.345 0.456 2.345
Χρωμόσωμα B	ABDFHJLNOJFETA

Εικόνα 14 Παράδειγμα χρωμοσώματος με Value κωδικοποίηση

2. Συνάρτηση φυσικής κατάστασης – φόρμας (Fitness function)

Στους γενετικούς αλγορίθμους η συνάρτηση φυσικής κατάστασης (fitness function) είναι πολύ βασικό στοιχείο διότι αξιολογεί την ποιότητα των λύσεων στον πληθυσμού. Άρα, η συνάρτηση fitness εφαρμόζεται σε κάθε μέλος του πληθυσμού και καθορίζει το πόσο κατάλληλο είναι ένα άτομο. Κάποια άτομα - λύσεις θα είναι καλύτερες από τις άλλες (fitter). Η συνάρτηση φόρμας δίνει βαθμολογία φυσικής κατάστασης σε κάθε άτομο. Η πιθανότητα να επιλεγεί ένα άτομο για αναπαραγωγή βασίζεται στη βαθμολογία φυσικής του κατάστασης. Ισχύει ότι: high fitness = low cost.

Η συνάρτηση fitness πρέπει να σχεδιαστεί προσεκτικά για να προσεγγίσει τους στόχους του προβλήματος βελτίωσης. Θα πρέπει να εξετάζει όλους τους σχετικούς περιορισμούς και στόχους, όπως η μεγιστοποίηση του κέρδους ή η ελαχιστοποίηση του κόστους. Η λειτουργία φυσικής κατάστασης θα πρέπει επίσης να είναι αποτελεσματική και γρήγορη στην αξιολόγηση, καθώς χρησιμοποιείται πολλές φορές κατά τη διαδικασία βελτιστοποίησης. Σε γενικές γραμμές, όσο υψηλότερος είναι ο βαθμός καταλληλότητας μιας υποψήφιας λύσης, τόσο καλύτερη θεωρείται και είναι πιο πιθανό να επιλεγεί για αναπαραγωγή. Στη συνέχεια, οι γενετικές πληροφορίες των καταλληλότερων λύσεων συνδυάζονται για να δημιουργήσουν

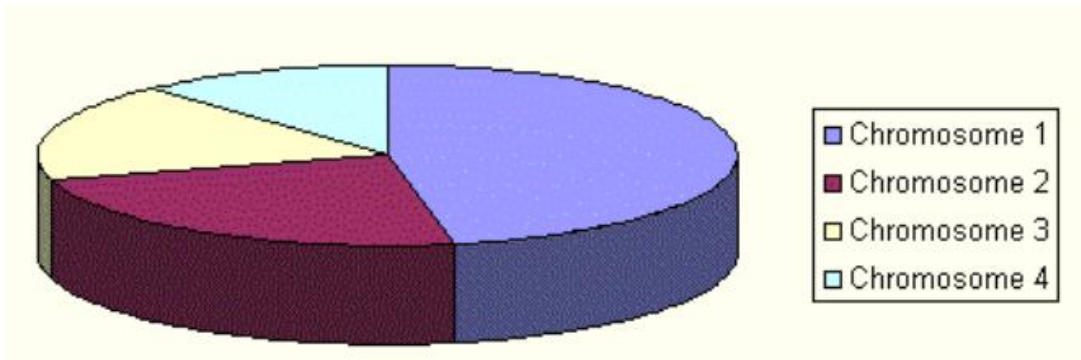
νέους απογόνους και η διαδικασία συνεχίζεται μέχρι να βρεθεί μια ικανοποιητική λύση ή να εκπληρωθεί ένα προκαθορισμένο κριτήριο διακοπής.

3. Επιλογή (Selection)

Η επιλογή είναι ένα σημαντικό βήμα σε έναν γενετικό αλγόριθμο που περιλαμβάνει την επιλογή των καταλληλότερων λύσεων από τον τρέχοντα πληθυσμό για αναπαραγωγή. Το στάδιο της «επιλογής – selection» είναι η επιλογή των κατάλληλων ατόμων, ώστε να επιτραπεί να περάσουν τα γονίδιά τους στην επόμενη γενιά. Δύο ζευγάρια ατόμων (γονείς) επιλέγονται με βάση τα σκορ της φυσικής τους κατάστασης, όπως καθορίζεται από τη συνάρτηση φυσικής κατάστασης. Τα άτομα με υψηλή φυσική κατάσταση έχουν περισσότερες πιθανότητες να επιλεγούν για αναπαραγωγή. Οι λύσεις με τις υψηλότερες βαθμολογίες φυσικής κατάστασης επιλέγονται στη συνέχεια για αναπαραγωγή, ενώ οι λύσεις με χαμηλότερες βαθμολογίες απορρίπτονται. Υπάρχουν πολλές διαφορετικές μέθοδοι επιλογής που μπορούν να χρησιμοποιηθούν σε έναν γενετικό αλγόριθμο, όπως η επιλογή τροχού ρουλέτας (roulette wheel selection), η επιλογή τουρνουά (tournament selection), η επιλογή θέσης (rank selection) και η επιλογή σταθερής κατάστασης (Steady-State selection). Κάθε μέθοδος έχει τα δικά της πλεονεκτήματα και μειονεκτήματα και η επιλογή της μεθόδου θα εξαρτηθεί από το συγκεκριμένο πρόβλημα βελτιστοποίησης και τις επιθυμητές ιδιότητες του αλγορίθμου. Γενικά, στόχος της επιλογής (selection) είναι να διασφαλιστεί ότι οι καλύτερες λύσεις θα περάσουν στην επόμενη γενιά, έτσι ώστε η ποιότητα των λύσεων στον πληθυσμό να συνεχίσει να βελτιώνεται με την πάροδο του χρόνου.

Επιλογή ρουλέτας (Roulette Wheel Selection): Η επιλογή τροχού ρουλέτας είναι μια δημοφιλής μέθοδος επιλογής που χρησιμοποιείται στους γενετικούς αλγόριθμους. Είναι ένας τύπος στοχαστικής επιλογής που προσομοιώνει τη διαδικασία της φυσικής επιλογής, όπου τα πιο κατάλληλα άτομα είναι πιο πιθανό να επιλεγούν για αναπαραγωγή.

Θεωρούμε έναν κύκλο που χωρίζεται σε κομμάτια ανάλογα με την τιμή της φόρμας κάθε λύσης που αναπαρίστανται ως τμήματα ενός εικονικού τροχού ρουλέτας. Το μέγεθος κάθε τμήματος είναι ανάλογο με την καταλληλότητα της αντίστοιχης λύσης, έτσι ώστε οι λύσεις υψηλότερης καταλληλότητας να έχουν μεγαλύτερα τμήματα. Για την επιλογή της λύσης, δημιουργείται ένας τυχαίος αριθμός και περιστρέφεται ο τροχός της ρουλέτας και επιλέγει ποιο άτομο είναι ο νικητής. Φυσικά το χρωμόσωμα με την καλύτερη φόρμα είναι και αυτό που έχει την μεγαλύτερη πιθανότητα να επιλεγεί.

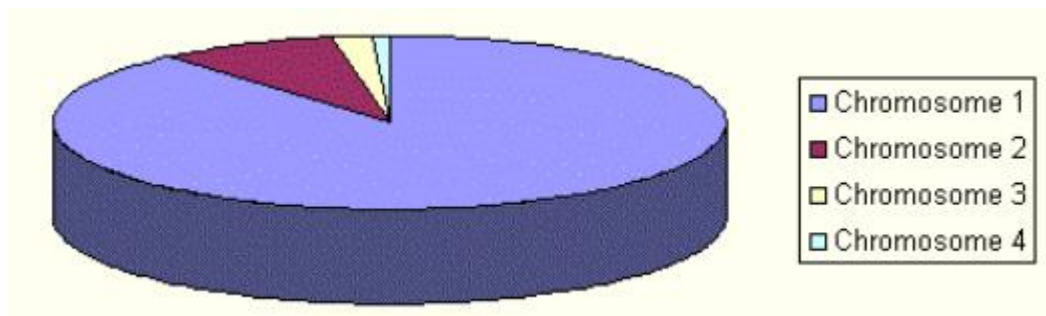


Εικόνα 15 Roulette Wheel Selection [22]

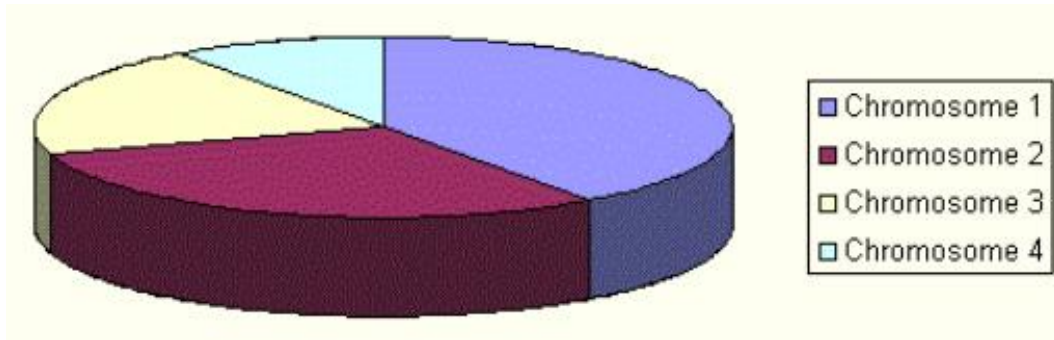
Η επιλογή τροχού ρουλέτας είναι απλή στην εφαρμογή και έχει χρησιμοποιηθεί ευρέως σε γενετικούς αλγόριθμους λόγω της αποτελεσματικότητάς της και της ικανότητάς της να διατηρεί την ποικιλία των λύσεων στον πληθυσμό. Ωστόσο, έχει το μειονέκτημα ότι είναι προκατειλημμένο προς τις λύσεις με υψηλότερες βαθμολογίες φυσικής κατάστασης, γεγονός που μπορεί να οδηγήσει σε πρόωρη σύγκλιση.

Επιλογή Θέσεις (Rank Selection): Σε αυτόν τον τύπο επιλογής, τα άτομα στον πληθυσμό κατατάσσονται με βάση τις τιμές φυσικής κατάστασης και τα καλύτερα άτομα λαμβάνουν υψηλότερη κατάταξη. Πιο συγκεκριμένα θεωρούνται N άτομα και μπαίνουν σε μια αύξουσα σειρά από το χειρότερο στο καλύτερο. Παίρνουν μια βαθμολογία από 1 έως και N . Το χειρότερο θα έχει fitness 1, το επόμενο 2 και ούτω καθεξής, το καλύτερο θα έχει N , όπου είναι και ο αριθμός των χρωμοσωμάτων στον πληθυσμό. Αυτό διασφαλίζει ότι τα καλύτερα άτομα έχουν πολύ περισσότερες πιθανότητες να επιλεγούν σε σύγκριση με τα χειρότερα άτομα. Η επιλογή κατάταξης είναι μια απλή και αποτελεσματική μέθοδος επιλογής που μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης. Θεωρείται πιο δίκαιη από την παραπάνω μέθοδο (roulette wheel selection), όταν οι διαφορές των τιμών της φόρμας των ατόμων είναι πολύ μεγάλες, όμως είναι πολύ πιο αργή.

Στην παρακάτω εικόνα φαίνεται η διαφορά όταν αλλάζει η φόρμα με τον τρόπο που αναφέρθηκε παραπάνω.



Εικόνα 16 Η κατάσταση πριν το Ranking [22]



Εικόνα 17 Η Κατάσταση μετά το Ranking (graph of order numbers) [22]

Επιλογή σταθερής κατάστασης (Steady-State Selection): Αυτή η μέθοδος δεν είναι κάποια συγκεκριμένη μέθοδος επιλογής. Η κύρια ιδέα αυτής της επιλογής είναι ότι ένα μεγάλο μέρος των χρωμοσωμάτων θα πρέπει να επιβιώσει στην επόμενη γενιά. Ο γενετικός αλγόριθμος λειτουργεί με τον ακόλουθο τρόπο. Σε κάθε γενιά επιλέγονται λίγα καλά χρωμοσώματα (με υψηλό fitness) για τη δημιουργία ενός νέου απογόνου. Στην συνέχεια αφαιρούνται κάποια κακά χρωμοσώματα (με χαμηλό fitness) και στη θέση τους τοποθετούνται οι νέοι απόγονοι. Ο υπόλοιπος πληθυσμός επιβιώνει στην νέα γενιά. Η μέθοδος επιλογής σταθερής κατάστασης είναι χρήσιμη όταν το πρόβλημα βελτιστοποίησης έχει μεγάλο μέγεθος πληθυσμού και υψηλό υπολογιστικό κόστος για την αξιολόγηση της καταλληλότητας των ατόμων. Επιτρέπει στον αλγόριθμο να διατηρεί ποικίλο πληθυσμό και να αποφεύγει την πρόωρη σύγκλιση, ενώ ταυτόχρονα μειώνει το υπολογιστικό κόστος.

Επιλογή τουρνουά (tournament selection): Στην επιλογή τουρνουά, ένα μικρό υποσύνολο λύσεων, γνωστό ως τουρνουά, επιλέγεται τυχαία από τον πληθυσμό. Η καλύτερη λύση μεταξύ αυτών των υποψηφίων επιλέγεται στη συνέχεια ως γονέας για αναπαραγωγή. Ο νικητής κάθε τουρνουά (αυτός με την καλύτερη φυσική κατάσταση) επιλέγεται για crossover. Με αυτόν τον τρόπο γίνονται πολλά τέτοια τουρνουά και έχουμε την τελική μας επιλογή υποψηφίων που περνούν στην επόμενη γενιά. Έχει επίσης μια παράμετρο που ονομάζεται πίεση επιλογής που είναι η πιθανότητα της συμμετοχής ενός υποψηφίου σε ένα τουρνουά. Η παράμετρος πίεσης επιλογής καθορίζει το ρυθμό σύγκλισης του γενετικού αλγορίθμου. Όσο περισσότερο η πίεση επιλογής θα είναι ο ρυθμός σύγκλισης. Οι γενετικοί αλγόριθμοι είναι σε θέση να προσδιορίζουν βέλτιστες ή σχεδόν βέλτιστες λύσεις σε ένα ευρύ φάσμα πιέσεων επιλογής. Η επιλογή τουρνουά λειτουργεί επίσης για αρνητικές τιμές φυσικής κατάστασης.

Η επιλογή τουρνουά χρησιμοποιείται συχνά σε συνδυασμό με άλλες μεθόδους επιλογής, όπως η επιλογή κατάταξης ή η επιλογή τροχού ρουλέτας, για τη δημιουργία ενός υβριδικού σχήματος επιλογής που εξισορροπεί τα δυνατά σημεία διαφορετικών μεθόδων επιλογής. Είναι μια δημοφιλής μέθοδος επιλογής λόγω της απλότητας, της αποτελεσματικότητας και της ευελιξίας της. Η μέθοδος επιλογής τουρνουά μπορεί να περιγραφεί και με τον παρακάτω ψευδοκώδικα.

1. Select k individuals from the population and perform a tournament amongst them

2. Select the best individual from the k individuals

3. Repeat process 1 and 2 until you have the desired amount of population

4. Διασταύρωση (Crossover)

Το crossover [23] είναι η πιο σημαντική φάση σε έναν γενετικό αλγόριθμο. Για κάθε ζευγάρι γονέων που πρόκειται να ζευγαρώσουν, επιλέγεται τυχαία ένα σημείο διασταύρωσης-crossover μέσα από τα γονίδια. Με άλλα λόγια, συνδυάζει μέρη από δύο χρωμοσώματα (γονείς), για να δημιουργήσει δύο νέους απογόνους (παιδιά). Ο στόχος του crossover είναι η δημιουργία λύσεων απόγονων που κληρονομούν τα καλύτερα χαρακτηριστικά των γονικών λύσεων, οδηγώντας σε βελτιωμένες λύσεις με την πάροδο του χρόνου. Η διαδικασία διασταύρωσης τυπικά περιλαμβάνει την επιλογή δύο μητρικών λύσεων και στη συνέχεια την ανταλλαγή γενετικών πληροφοριών μεταξύ τους. Φυσικά υπάρχουν διάφοροι τύποι crossover και εκτελούνται λίγο διαφορετικά ανάλογα και με την κωδικοποίηση των χρωμοσωμάτων.

Διασταύρωση ενός σημείου - Single point crossover: Στη διασταύρωση ενός σημείου, επιλέγεται ένα μόνο τυχαίο σημείο στο χρωμόσωμα κάθε γονέα. Στη συνέχεια, τα χρωμοσώματα χωρίζονται σε δύο μέρη σε αυτό το σημείο και τα μέρη ανταλλάσσονται μεταξύ των δύο γονέων για να δημιουργηθούν δύο λύσεις απογόνων. Αν για παράδειγμα η κωδικοποίηση που έχει επιλεγεί είναι binary, τότε η διαδικασία που ακολουθείται για κάθε ζευγάρι γονέων είναι η εξής. Το γονίδιο δημιουργείται από την αντιγραφή των πρώτων x bit του 1^{ου} γονέα και x+1 μέχρι το τέλος της συμβολοσειράς από τον 2^ο γονέα.

Γονέας 1: 10110111

Γονέας 2: 01101100

Σημείο διασταύρωσης ενός σημείου: 4

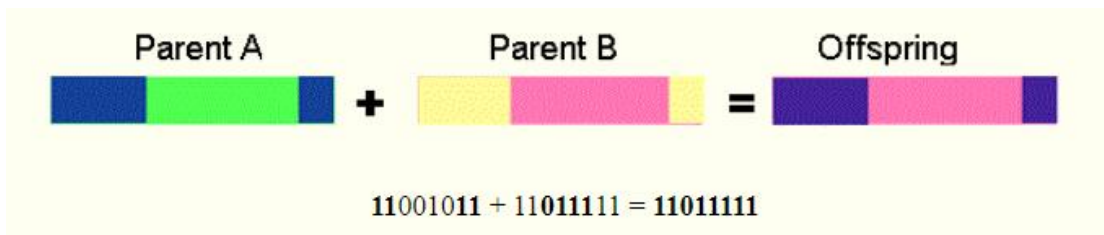
Απόγονοι 1: 10110100

Απόγονοι 2: 01101111

Η διασταύρωση ενός σημείου είναι μια απλή αλλά αποτελεσματική μέθοδος για την επανασύνθεση γενετικών πληροφοριών και την παραγωγή νέων λύσεων σε έναν γενετικό αλγόριθμο. Με την ανταλλαγή πληροφοριών σε ένα μόνο σημείο, επιτρέπει την εξερεύνηση διαφορετικών περιοχών του χώρου λύσεων και μπορεί να βοηθήσει στη διατήρηση της ποικιλομορφίας στον πληθυσμό.

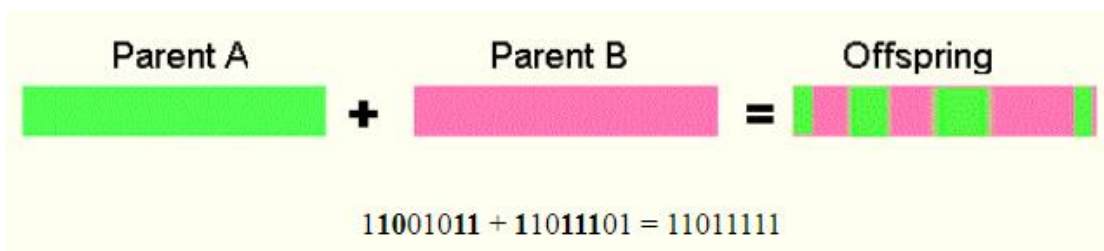
2 Σημείων - Two points crossover: Η διασταύρωση δύο σημείων, λειτουργεί επιλέγοντας δύο σημεία στη γενετική αναπαράσταση των γονέων και ανταλλάσσοντας τις γενετικές πληροφορίες μεταξύ των δύο γονέων για τη δημιουργία δύο λύσεων απογόνων. Αυτές οι λύσεις απογόνων αντιπροσωπεύουν νέες υποψήφιας λύσεις για την επόμενη γενιά του πληθυσμού. Για παράδειγμα, εάν δύο λύσεις γονέων αντιπροσωπεύονται ως δυαδικές

συμβολοσειρές, ο τελεστής διασταύρωσης δύο σημείων θα επέλεγε τυχαία δύο σημεία στις συμβολοσειρές και θα ανταλλάξει τα bit μεταξύ αυτών των σημείων. Αυτό δημιουργεί δύο λύσεις απογόνων που συνδυάζουν τις γενετικές πληροφορίες και από τους δύο γονείς. Η διασταύρωση δύο σημείων είναι ένας δημοφιλής γενετικός τελεστής στους γενετικούς αλγόριθμους επειδή επιτρέπει μια πιο ομοιόμορφη ανταλλαγή γενετικών πληροφοριών μεταξύ των γονέων σε σύγκριση με άλλους τελεστές διασταύρωσης, όπως η διασταύρωση ενός σημείου.



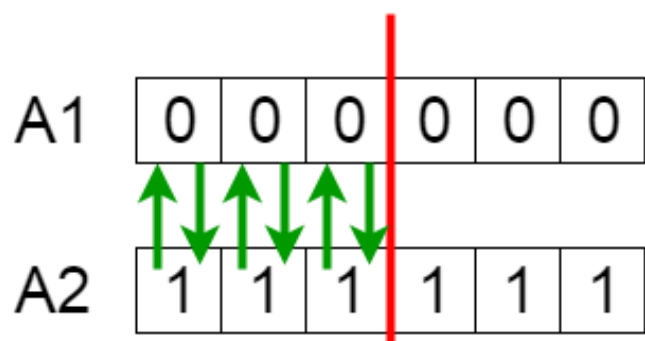
Εικόνα 18 Two points crossover [23]

Ομοιόμορφη - Uniform crossover: Στην ομοιόμορφη διασταύρωση, χρησιμοποιείται μια γεννήτρια τυχαίων αριθμών για να καθοριστεί ποια γενετικά στοιχεία από κάθε γονέα θα συμπεριληφθούν στους απογόνους. Για κάθε γονίδιο στους απογόνους, υπάρχει ίση πιθανότητα να προέρχεται από οποιονδήποτε γονέα.

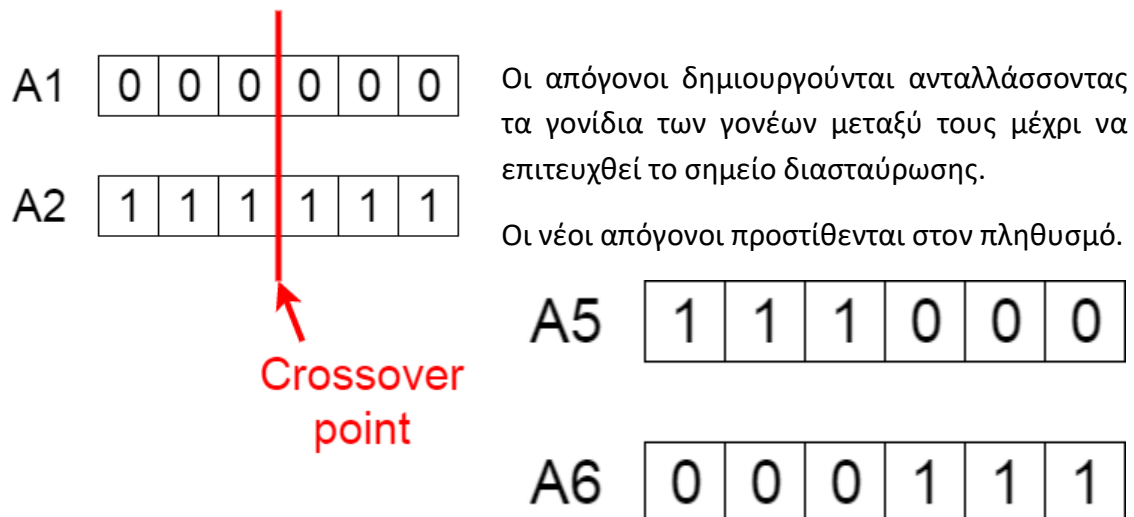


Εικόνα 19 Παράδειγμα Ομοιόμορφου crossover

Arithmetic crossover:



Εικόνα 20 Σχεδίαση crossover point και Ανταλλαγή γονιδίων μεταξύ γονέων



Εικόνα 21 Τελικό αποτέλεσμα

5. Μετάλλαξη (Mutation)

Στους γενετικούς αλγόριθμους, η μετάλλαξη είναι ένας γενετικός τελεστής που αλλάζει τυχαία τις τιμές ενός ή περισσότερων γονιδίων σε μια υποψήφια λύση. Η μετάλλαξη χρησιμεύει ως μέσο εισαγωγής νέας γενετικής πληροφορίας στον πληθυσμό, η οποία βοηθά στην αποφυγή κολλήματος της διαδικασίας βελτιστοποίησης σε ένα τοπικό βέλτιστο. Η μετάλλαξη λειτουργεί επιλέγοντας τυχαία ένα ή περισσότερα γονίδια σε μια υποψήφια λύση και αλλάζοντας την τιμή του. Υπάρχει μια μεταβλητή που ονομάζεται ρυθμός μετάλλαξης, ο οποίος καθορίζεται ως παράμετρος του γενετικού αλγορίθμου και καθορίζει την πιθανότητα μετάλλαξης των γονιδίων. Συνήθως, ο ρυθμός μετάλλαξης ορίζεται χαμηλός, καθώς η υπερβολική μετάλλαξη μπορεί να προκαλέσει τη διαδικασία βελτιστοποίησης να καταστεί αναποτελεσματική ή να οδηγήσει σε κακές λύσεις. Η μετάλλαξη βοηθά στην πρόληψη της πρόωρης σύγκλισης με την εισαγωγή νέων γενετικών πληροφοριών στον πληθυσμό, η οποία μπορεί να οδηγήσει στην ανακάλυψη νέων, καλύτερων λύσεων. Γενικά, μια καλή ισορροπία μεταξύ μετάλλαξης και διασταύρωσης είναι απαραίτητη για την αποτελεσματική βελτιστοποίηση με χρήση γενετικών αλγορίθμων.

Αν για παράδειγμα η συμβολοσειρά είναι σε δυαδική μορφή, κάποια από τα bit στη συμβολοσειρά μπορούν να αναστραφούν. Το πόσα bit θα αλλάξουν το καθορίζει ο συντελεστής – μεταβλητή μετάλλαξης ή αλλιώς ρυθμός μετάλλαξης. Το παρακάτω παράδειγμα των εικόνων () έχει συντελεστή μετάλλαξης 3.

Before Mutation

A5

1	1	1	0	0	0
---	---	---	---	---	---

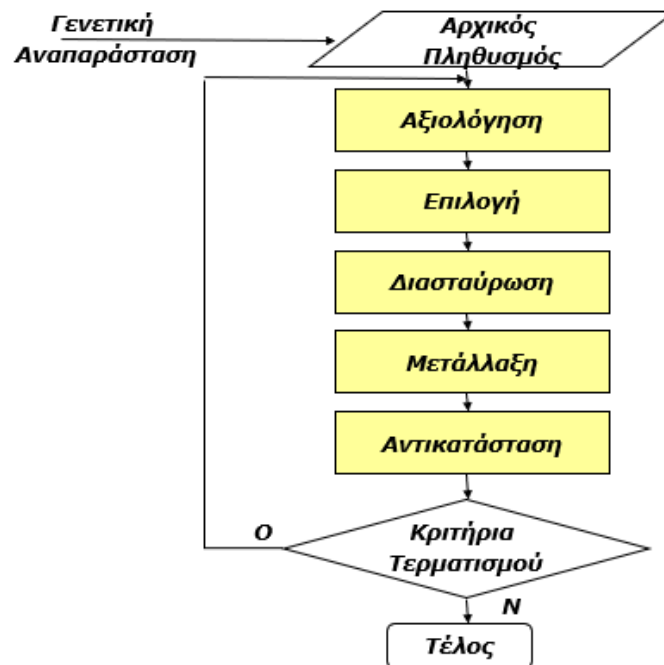
After Mutation

A5

1	1	0	1	1	0
---	---	---	---	---	---

Εικόνα 22 Επεξήγηση διαδικασίας mutation

Διάγραμμα ροής – Flow chart



Εικόνα 23 Διάγραμμα ροής των γενετικών αλγορίθμων

Ψευδοκώδικας

START

Generate randomly the initial population

Compute fitness of population

REPEAT

Selection (Select parents from population)

Crossover

Mutation

Compute fitness

UNTIL population has converged

STOP

Τι υλοποιήθηκε για την εργασία

Στο παρόν πρόβλημα υλοποιήθηκε ένας γενετικός αλγόριθμος, για τη δημιουργία βέλτιστων λύσεων για το πρόβλημά μας, για την ελαχιστοποίηση δηλαδή του κόστους, για το βέλτιστο συνδυασμό και τοποθέτηση αισθητήρων και τη μεγιστοποίηση της ζωής της μπαταρίας. Ο προτεινόμενος κώδικας αλγορίθμου, υλοποιήθηκε και τρέχει μέσω MATLAB.

Ο κώδικας του γενετικού αλγορίθμου.

Main

```
clc;
clear;
close all;

%% Problem Definition - GA Parameters (here is defined the parameters of the genetic algorithm)
scenarios = 3;
for sc=1:1:scenarios
    if (sc == 1) % 5g scenario

        nVar = 3; % Number of variables
        VarMin = [20, 20, 20]; % The lower bound of the chromosomes
        VarMax = [80, 80, 80]; % The upper bound of the chromosomes

        generation = 50; % MaxIt - Maximum number of generation / maximum number of iterations.
        population_size = 50; % Population size. Our initial population and the population size will be constant
            for any generation.

        population = zeros(population_size,nVar); % Initializing the size of the population
        temp_population = zeros(population_size,nVar); % Initializing the size of the temporary population
        mu = 0.1; % Mutation Rate => 10% and that means, i am going to mutate and alter 10% of gene
            in average.

        S_population = zeros (population_size,nVar+2); % Initializing the size of the sorted population using
            the problem objective function
        Cost = zeros(population_size,1); % initializing the size of the objective function based
on the population size
        Energy = zeros(population_size,1); % initializing the size of the objective function based
on the population size
        crossover_times = 4; % Number of crossovers
        mutation_times = 5; % Number of mutations

    elseif( sc == 2) % LoRa scenario

        nVar = 4; % Number of variables
        VarMin = [20, 20, 20, 10]; % The lower bound of the chromosomes
        VarMax = [80, 80, 80, 30]; % The upper bound of the chromosomes

        generation = 50; % MaxIt - Maximum number of generation / maximum number of iterations.
        population_size = 50; % Population size. Our initial population and the population size will be
constant for any generation.

        population = zeros(population_size,nVar); % Initializing the size of the population
        temp_population = zeros(population_size,nVar); % Initializing the size of the temporary population
        mu = 0.1; % Mutation Rate => 10% and that means, i am going to mutate and alter 10% of gene
in average.
```



```

    S_population = zeros (population_size,nVar+1); % Initializing the size of the sorted population using
the problem objective function
    Cost = zeros(population_size,1); % initializing the size of the objective function
based on the population size
    crossover_times = 4; % Number of crossovers
    mutation_times = 5; % Number of mutations

elseif( sc == 3) % NB scenario

    nVar = 3;
    VarMin = [20, 20, 20]; % The lower bound of the chromosomes
    VarMax = [80, 80, 80]; % Number of variables
    %VarMin = 20; % The lower bound of the chromosomes
    %VarMax = 80; % The upper bound of the chromosomes

    generation = 50; % MaxIt - Maximum number of generation / maximum number of iterations.
    population_size = 50; % Population size. Our initial population and the population size will be
constant for any generation.

    population = zeros(population_size,nVar); % Initializing the size of the population
    temp_population = zeros(population_size,nVar); % Intializing the size of the temporary population
    mu = 0.1; % Mutation Rate => 10% and that means, i am going to mutate and alter 10% of gene
in average.

    S_population = zeros (population_size,nVar+1); % Initializing the size of the sorted population using
the problem objective function
    Cost = zeros(population_size,1); % initializing the size of the objective function
based on the population size
    crossover_times = 4; % Number of crossovers
    mutation_times = 5; % Number of mutations

end

for i=1:1:population_size
    for n=1:1:nVar
        population(i,n)= unifrnd(VarMin(1,n),VarMax(1,n));
        %population(i,n)= unifrnd(VarMin(1),VarMax(1));
        population(i,n)= uint8(population(i,n));
    end
    if (sc == 1)
        [Cost(i), Energy(i)]= objectiveFunction5g(population(i,:));
    elseif (sc == 2)
        [Cost(i), Energy(i)]= objFunLoRa(population(i,:));
    elseif (sc == 3)
        [Cost(i), Energy(i)]= objFunNB(population(i,:));
    end
end

end

% Sorting of the population
S_population(:,1:nVar)= population(:,:);
S_population(:,nVar+1)=Cost;
S_population(:,nVar+2)=Energy;
S_population= sortrows(S_population,nVar+1);
S_population_energy= sortrows(S_population,nVar+2);

% Generation
for ii=1:1:generation
    k=1; % This is used to initialize the location of the temp_population;

```



```

% Elitism
temp_population(k,1:nVar) = S_population(1,1:nVar);
k= k + 1;

% Selection and Crossover
for j=1:1:crossover_times
    % Parents selection
    y1(j) = geornd(0.1)+1;
    while y1(j) > population_size
        y1(j) = geornd(0.1)+1 ;
    end
    y2(j) = geornd(0.1)+1;
    while y2(j)> population_size
        y2(j) = geornd(0.1)+1;
    end
end

for u=1:1:crossover_times
    parent1= S_population(y1(u),1:nVar);
    parent2= S_population(y2(u),1:nVar);

    % Arithmetic Crossover
    [Children] = arithmetic_crossover(parent1, parent2);
    temp_population(k,1:nVar) = Children(1,:);

    temp_population(k,1:nVar) = max(temp_population(k,1:nVar),VarMin);
    temp_population(k,1:nVar) = min(temp_population(k,1:nVar),VarMax);

    k=k+1;
    temp_population(k,1:nVar) = Children(2,:);

    temp_population(k,1:nVar)=max(temp_population(k,1:nVar),VarMin);
    temp_population(k,1:nVar)=min(temp_population(k,1:nVar),VarMax);

    k=k+1;
end

% Mutation
for e=1:1:mutation_times
    parent=S_population(unidrnd(population_size),1:nVar);
    [child]= gene_mutation (parent,mu,VarMax);

    temp_population(k,1:nVar)=child;
    temp_population(k,1:nVar)=max(temp_population(k,1:nVar),VarMin);
    temp_population(k,1:nVar)=min(temp_population(k,1:nVar),VarMax);

    k=k+1;
end

%Replication
for k=k:1:population_size
    replicated_child = S_population(randi([1 population_size]),1:nVar);
    temp_population(k,1:nVar) = replicated_child;
    k=k+1;
end

% Calculating the temporary population objective function values
for iii=1:1:population_size
    %[Cost(iii), Energy(iii)]= objectiveFunction(temp_population(iii,:));

```

```

if (sc == 1)
    [Cost(iii), Energy(iii)] = objectiveFunction5g(temp_population(iii,:));
elseif (sc == 2)
    [Cost(iii), Energy(iii)] = objFunLoRa(temp_population(iii,:));
elseif (sc == 3)
    [Cost(iii), Energy(iii)] = objFunNB(temp_population(iii,:));
end
end

%% Sorting for the next generation and selection of the population best
S_population(:,1:nVar) = temp_population;
S_population(:,nVar+1) = Cost(:,:);
S_population(:,nVar+2) = Energy(:,:);
S_population = sortrows(S_population,nVar+1);
S_population_energy = sortrows(S_population,nVar+2);

Best_F(ii) = S_population(1,nVar+1);

disp(['Iteration ' num2str(ii) ': Best Cost = ' num2str(Best_F(ii))]);
end

%% Generate Json files
if (sc == 1) % 5g scenario
    % cost-effective (cheapest) 5g solution (sort-by-solution)
    %s = table2struct(S_population);
    jsonText = jsonencode(S_population); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf(':\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t]'));

    fid = fopen('GA-cost-effective-5g-solutions.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);

    movefile('GA-cost-effective-5g-solutions.json','Results\GA')
elseif (sc == 2)
    % cost-effective (cheapest) 5g solution (sort-by-solution)
    %s = table2struct(S_population);
    jsonText = jsonencode(S_population); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf(':\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));
    jsonText = strrep(jsonText, ']', sprintf('\r\t]'));

    fid = fopen('GA-cost-effective-lora-solutions.json', 'w'); % Write to a json file
    fprintf(fid, '%s', jsonText);
    fclose(fid);

    movefile('GA-cost-effective-lora-solutions.json','Results\GA')
elseif (sc == 3)
    % cost-effective (cheapest) 5g solution (sort-by-solution)
    %s = table2struct(S_population);
    jsonText = jsonencode(S_population); % Convert to JSON text
    jsonText = strrep(jsonText, '{', sprintf('{\r\t'));
    jsonText = strrep(jsonText, ':', sprintf(':\r\t\t'));
    jsonText = strrep(jsonText, ',', sprintf(',\r\t\t'));
    jsonText = strrep(jsonText, '}', sprintf('\r'));

```

```
jsonText = strrep(jsonText, ']', sprintf('\r\t]'));
```

```
fid = fopen('GA-cost-effective-NB-solutions.json', 'w'); % Write to a json file  
fprintf(fid, '%s', jsonText);  
fclose(fid);
```

```
movefile('GA-cost-effective-NB-solutions.json', 'Results\GA')
```

end

end

objfunLora

```
1 function [Cost, Energy] = objFunLoRa(x)  
2 % get cost data from json file  
3 jsonText_Cost = fileread("Inputs-json\costs.json");  
4 jsonData_Cost = jsondecode(jsonText_Cost); % Convert JSON formatted text to MATLAB data types  
5 jsonDataCostTable = table(jsonData_Cost);  
6 jsonText_Energy = fileread("Inputs-json\battery.json");  
7 jsonData_Energy = jsondecode(jsonText_Energy); % Convert JSON formatted text to MATLAB data types  
8 jsonDataEnergyTable = table(jsonData_Energy);  
9  
10 x1 = x(1);  
11 x2 = x(2);  
12 x3 = x(3);  
13 x4 = x(4);  
14  
15 c1 = jsonDataCostTable.jsonData_Cost.cost_lora_type_a;  
16 c2 = jsonDataCostTable.jsonData_Cost.installation_cost_lora_type_a;  
17 c3 = jsonDataCostTable.jsonData_Cost.cost_lora_type_b;  
18 c4 = jsonDataCostTable.jsonData_Cost.installation_cost_lora_type_b;  
19 c5 = jsonDataCostTable.jsonData_Cost.cost_lora_type_c;  
20 c6 = jsonDataCostTable.jsonData_Cost.installation_cost_lora_type_c;  
21 c7 = jsonDataCostTable.jsonData_Cost.cost_lora_gateway_type_a;  
22 c8 = jsonDataCostTable.jsonData_Cost.installation_lora_gateway_type_a;  
23  
24 capacity1 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_type_a;  
25 consumption1 = jsonDataEnergyTable.jsonData_Energy.battery_consumption_lora_type_a;  
26 capacity2 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_type_b;  
27 consumption2 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_type_b;  
28 capacity3 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_type_c;  
29 consumption3 = jsonDataEnergyTable.jsonData_Energy.battery_consumption_lora_type_c;  
30 capacity4 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_gateway_type_a;  
31 consumption4 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_gateway_type_a;  
32 capacity5 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_lora_gateway_type_b;  
33 consumption5 = jsonDataEnergyTable.jsonData_Energy.battery_consumption_lora_gateway_type_b;  
34  
35 cost1 = ( x1 * c1 ) + ( x1 * c2 );  
36 cost2 = ( x2 * c3 ) + ( x2 * c4 );  
37 cost3 = ( x3 * c5 ) + ( x3 * c6 );  
38 cost4 = ( x4 * c7 ) + ( x4 * c8 );  
39  
40 batteryLife_1 = capacity1/consumption1;  
41 batteryLife_2 = capacity2/consumption2;  
42 batteryLife_3 = capacity3/consumption3;  
43 batteryLife_4 = capacity4/consumption4;  
44 batteryLife_5 = capacity5/consumption5;  
45  
46 Cost = cost1 + cost2 + cost3 + cost4;  
47 Energy = (batteryLife_1 + batteryLife_2 + batteryLife_3 + batteryLife_4 + batteryLife_5) / 5;  
48 end
```

```
objectiveFunction5g.m  +
1 function [Cost, Energy] = objectiveFunction5g(x)
2
3 % Battery Life= (Battery Capacity (in mAh))/(Device Consumption (in mAh))
4
5 % get cost data from json file
6 jsonText_Cost = fileread("Inputs-json\costs.json");
7 jsonData_Cost = jsondecode(jsonText_Cost); % Convert JSON formatted text to MATLAB data types
8 jsonDataCostTable = table(jsonData_Cost);
9
10 jsonText_Energy = fileread("Inputs-json\battery.json");
11 jsonData_Energy = jsondecode(jsonText_Energy); % Convert JSON formatted text to MATLAB data types
12 jsonDataEnergyTable = table(jsonData_Energy);
13
14 x1 = x(1);
15 x2 = x(2);
16 x3 = x(3);
17
18 c1 = jsonDataCostTable.jsonData_Cost.cost_5g_type_a;
19 c2 = jsonDataCostTable.jsonData_Cost.installation_cost_5g_type_a;
20 c3 = jsonDataCostTable.jsonData_Cost.cost_5g_type_b;
21 c4 = jsonDataCostTable.jsonData_Cost.installation_cost_5g_type_b;
22 c5 = jsonDataCostTable.jsonData_Cost.cost_5g_type_c;
23 c6 = jsonDataCostTable.jsonData_Cost.installation_cost_5g_type_c;
24
25 capacity1 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_5g_type_a;
26 consumption1 = jsonDataEnergyTable.jsonData_Energy.battery_consumption_5g_type_a;
27 capacity2 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_5g_type_b;
28 consumption2 = jsonDataEnergyTable.jsonData_Energy.battery_consumption_5g_type_b;
29 capacity3 = jsonDataEnergyTable.jsonData_Energy.battery_capacity_5g_type_c;
30 consumption3 = jsonDataEnergyTable.jsonData_Energy.battery_consumption_5g_type_c;
31
32 cost1 = ( x1 * c1 ) + ( x1 * c2 );
33 cost2 = ( x2 * c3 ) + ( x2 * c4 );
34 cost3 = ( x3 * c5 ) + ( x3 * c6 );
35
36 batteryLife_1 = capacity1/consumption1;
37 batteryLife_2 = capacity2/consumption2;
38 batteryLife_3 = capacity3/consumption3;
39
40 Cost = cost1 + cost2 + cost3;
41 Energy = (batteryLife_1 + batteryLife_2 + batteryLife_3) / 3;
42 Energy = Energy / 24; % in days
43 end
```

```
arithmetic_crossover.m  +
1 function [Children]= arithmetic_crossover (parent1, parent2)
2     alpha=rand;
3     child1= alpha* parent1 + (1-alpha)* parent2;
4     child2=(1-alpha)*parent1 + alpha* parent2;
5     Children=[child1;child2];
6 end
```

```
gene_mutation.m x +
1 function [child] = gene_mutation (parent, mu, VarMin, VarMax)
2
3 % This parent must be relected at random child is parent having a gene transformed
4
5     nVar=numel(parent);
6     nmu=ceil(mu*nVar);
7     j=randsample(nVar,1);
8     child=parent;
9     child(j)=parent(j) - rand*parent(j);
10
11 end
```

5. Το προτεινόμενο σύστημα

Για τη δημιουργία του IoT συστήματος αυτής της διπλωματικής εργασίας, χρησιμοποιήθηκαν κάποιες τεχνολογίες για τη δημιουργία ενός βοηθητικού tool, ούτως ώστε να διευκολύνει στην διαχείριση και να κάνει πιο ολοκληρωμένη την υλοποίηση της διπλωματικής.

5.1 Περιγραφή και αρχιτεκτονική του συστήματος

Το προτεινόμενο εργαλείο αυτής της εργασίας είναι ένα planning - reporting tool όπου οι χρήστες μπορούν να σχεδιάσουν και να προσομοιάσουν σενάρια για τον σχεδιασμό ή την ανάπτυξη έξυπνων δικτύων αισθητήρων, και να λάβουν αποτελέσματα σε μορφή διαγραμμάτων. Πιο συγκεκριμένα έχει σχεδιαστεί για να δώσει λύσεις εύκολα και γρήγορα σε προβλήματα σχεδιασμού και διαχείρισης υπάρχοντων και μη δικτύων αισθητήρων. Για αυτό τον λόγο είναι πολύ εύκολο και φιλικό προς την χρήση του. Το εργαλείο αυτό διαχειρίζεται τα δεδομένα και τρέχει διαδικασίες (αλγορίθμους) για την εύρεση της αποδοτικότερης και φθηνότερης λύσης, βάση κάποιων συγκεκριμένων προδιαγραφών. Σε επόμενη ενότητα προβάλλονται εικόνες και περιγράφονται πολύ αναλυτικά όλες οι λειτουργίες του.

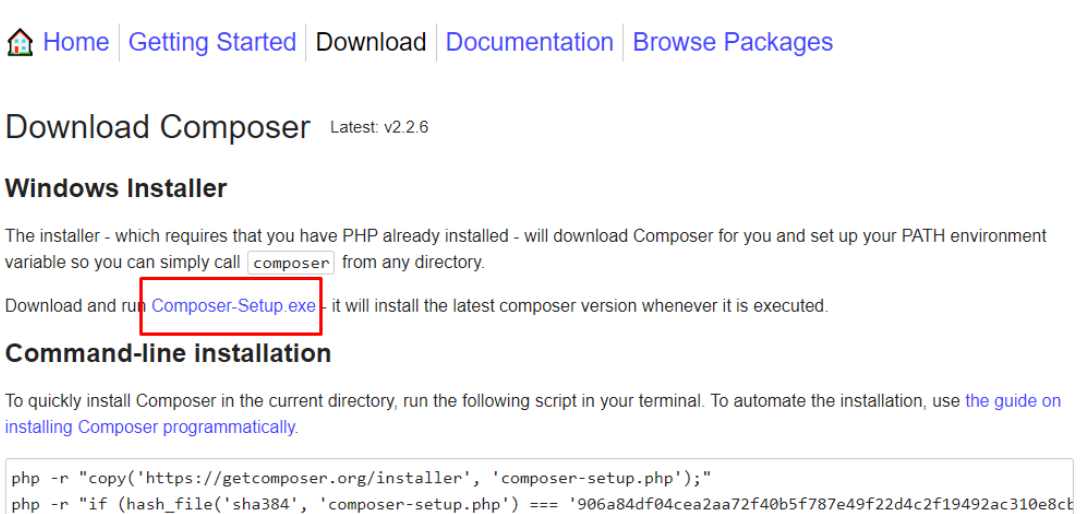
5.2 Οδηγίες εγκατάστασης και χρήσης

Σε αυτήν την υποενότητα περιγράφονται 2 οδηγοί. Ο πρώτος είναι ένας οδηγός εγκατάστασης (installation documentation) και ο δεύτερος είναι ένας οδηγός χρήσης για το βοηθητικό εργαλείο.

INSTALLATION DOCUMENTATION (For Windows – and Xampp Apache Server with MySQL)

Install Composer

Click on this link: <https://getcomposer.org/download/>

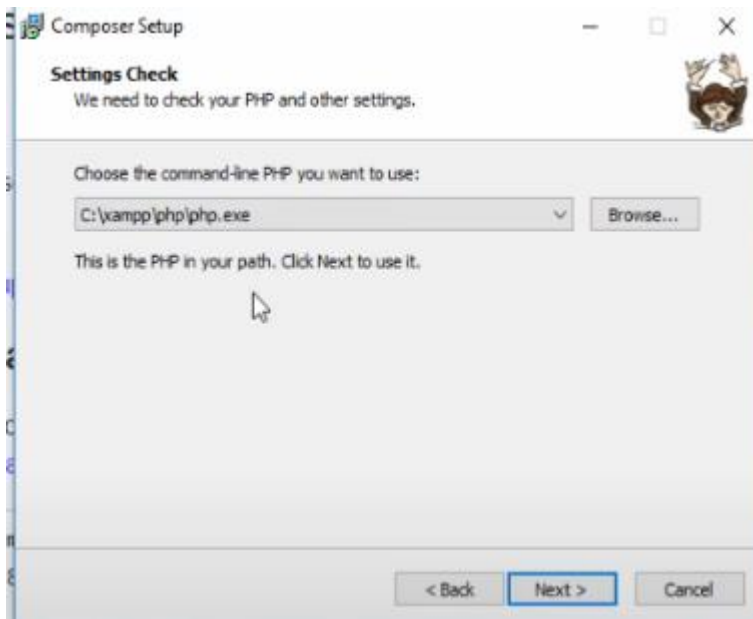


The screenshot shows the 'Download Composer' page for Windows. It includes a navigation bar with links for Home, Getting Started, Download, Documentation, and Browse Packages. The main heading is 'Download Composer' with 'Latest: v2.2.6' next to it. Under the heading, there is a section for 'Windows Installer' which explains that the installer will download Composer and set up the PATH environment variable. A red box highlights the 'Composer-Setup.exe' file. Below this is a section for 'Command-line installation' which provides a terminal script to download and install Composer programmatically.

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8c...
```

Download and install the setup.exe

You need to have already installed PHP and add the path of the php from xampp folder in installation when it is required



Open cmd (command prompt) to check if it is installed and run:

Composer

download node and npm

go to <https://nodejs.org/en/> and download the LTS version for node.

go to the cmd and run

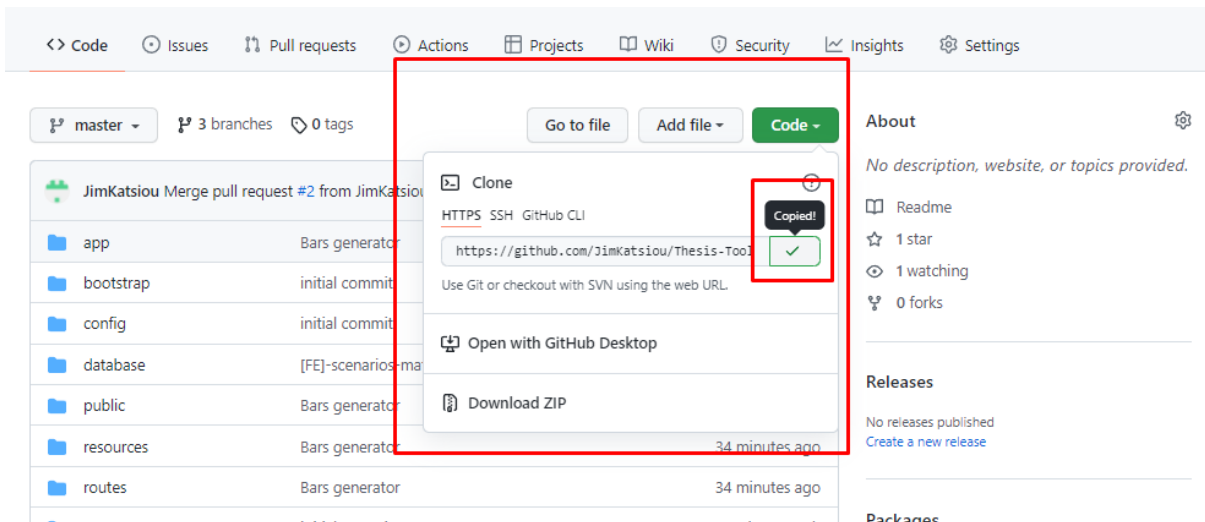
`node -v`

`npm -v`

If you don't have error, you have already installed the above in your computer. If npm version command response an error you should go to the terminal at project directory and write `npm i npm@8.1.2`

INSTALL THE PROJECT

Open <https://github.com/JimKatsiou/Master-Thesis-Tool> and copy the link that show in the picture.



Open Terminal in the directory you want

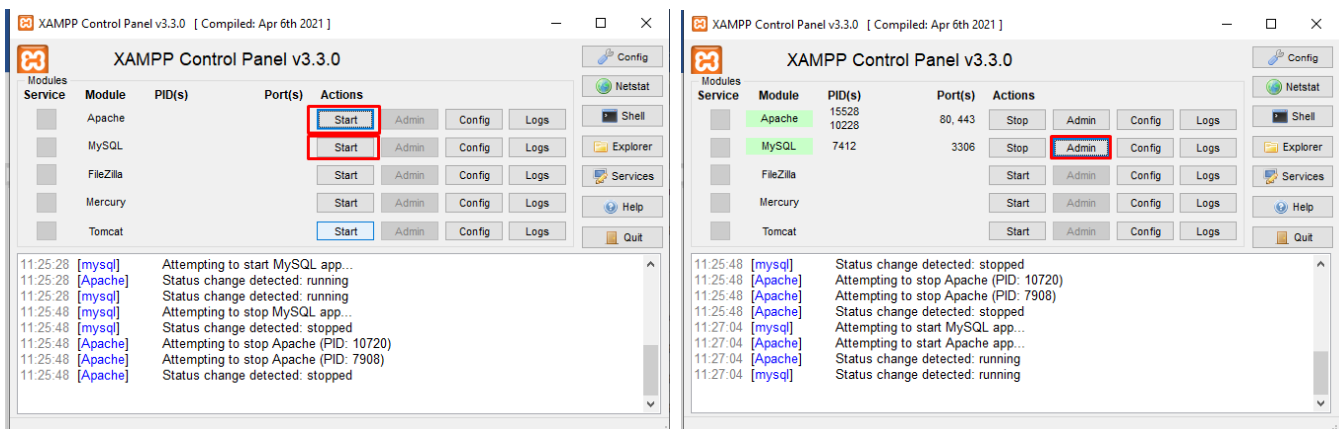
```
git clone https://github.com/JimKatsiou/Master-Thesis-Tool.git
```

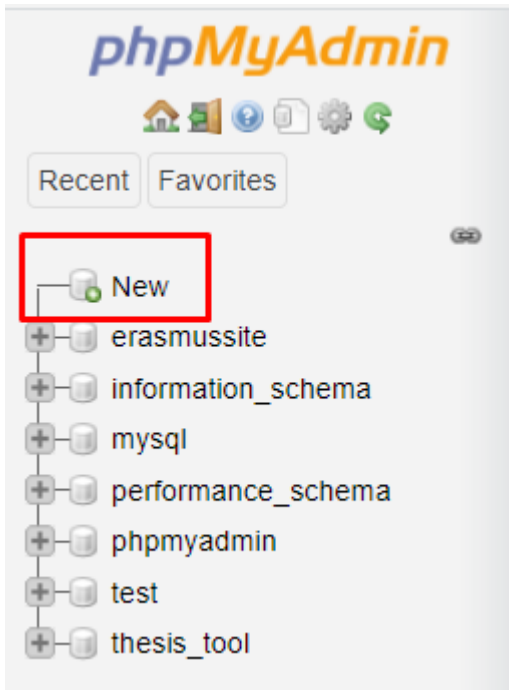
```
cd Master-Thesis-Tool
```

```
composer install
```

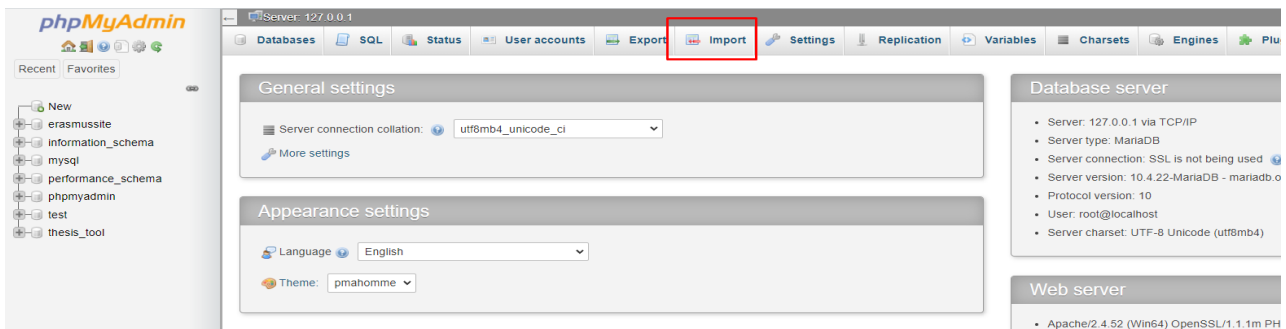
```
npm install
```

open Xampp control panel and start Apache server and MySQL. Then click admin button as shown in the left picture below. It will open phpMyAdmin for mysql and creates database with name reporting-tool as it seems in the right picture below.





Import database



Importing into the current server

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.format[.compression]**. Example: **.sql.zip**

Browse your computer: No file chosen (Max: 40MiB)

You may also drag and drop a file on any page.

Character set of the file:

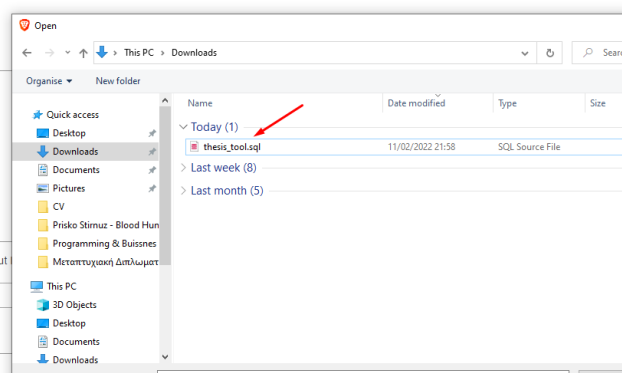
Partial import:

Allow the interruption of an import in case the script detects it is close to the PHP timeout

Skip this number of queries (for SQL) starting from the first one:

Other options:

Enable foreign key checks



Scroll down press Go button

copy the .env file to the project's .env file

.env file content

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:UiK3dvV7IsExd5v/CauizXPkoF1I0P32CbRvVjph0kc=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=reporting-tool
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DRIVER=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=cookie
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailhog
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

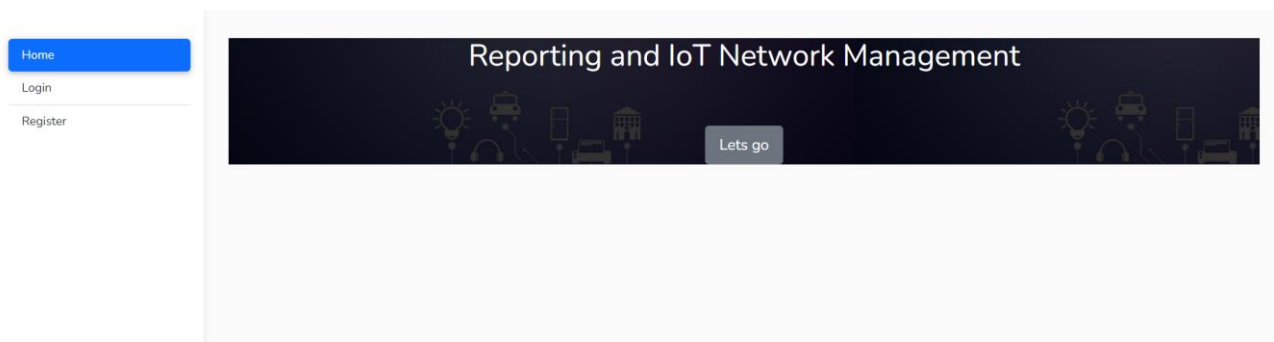
PUSHER_APP_ID=
PUSHER_APP_KEY=
```

```
PUSHER_APP_SECRET=  
PUSHER_APP_CLUSTER=mt1  
  
MIX_PUSHER_APP_KEY="{PUSHER_APP_KEY}"  
MIX_PUSHER_APP_CLUSTER="{PUSHER_APP_CLUSTER}"
```

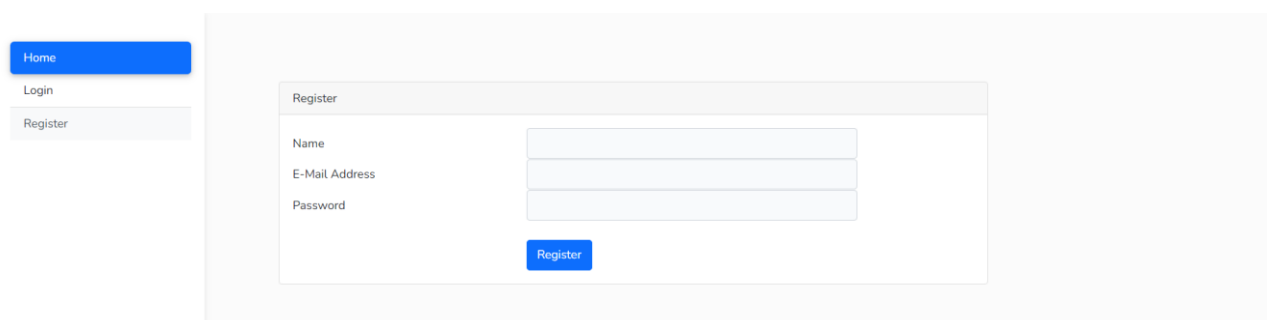
Finally you should run **php artisan serve** and **npm run watch** in project folder. Alternately, you should automate this process by having an Apache server always on.

Οδηγίες χρήσης του εργαλείου (user manual)

Η εικόνα (24) είναι το αρχικό menu (Home page) όπου είναι η πρώτη σελίδα που βλέπει ο χρήστης όταν ανοίγει την εφαρμογή. Το μόνο που πρέπει να κάνει ο χρήστης είναι να πατήσει το κουμπί *Register* από το sidebar menu, αν δεν είναι ήδη εγγεγραμμένος χρήστης, όπου θα τον οδηγήσει στην σελίδα της εγγραφής όπως φαίνεται στην εικόνα (25). Αν έχει ήδη δημιουργήσει λογαριασμό, τότε μπορεί να πατήσει το κουμπί *Login* από το sidebar menu είτε το κουμπί *Let's go* στο κέντρο της σελίδας. Και οι δύο αυτοί τρόποι τον οδηγούν στην σελίδα του Login εικόνα (26).

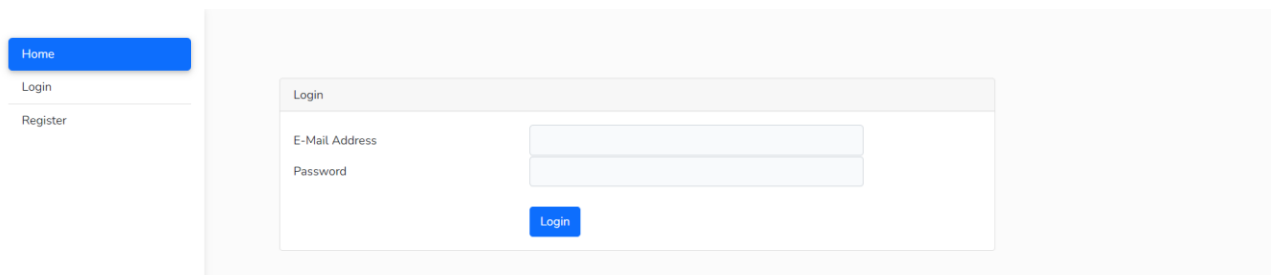


Εικόνα 24 Αρχική σελίδα



Εικόνα 25 Σελίδα εγγραφής νέου χρήστη

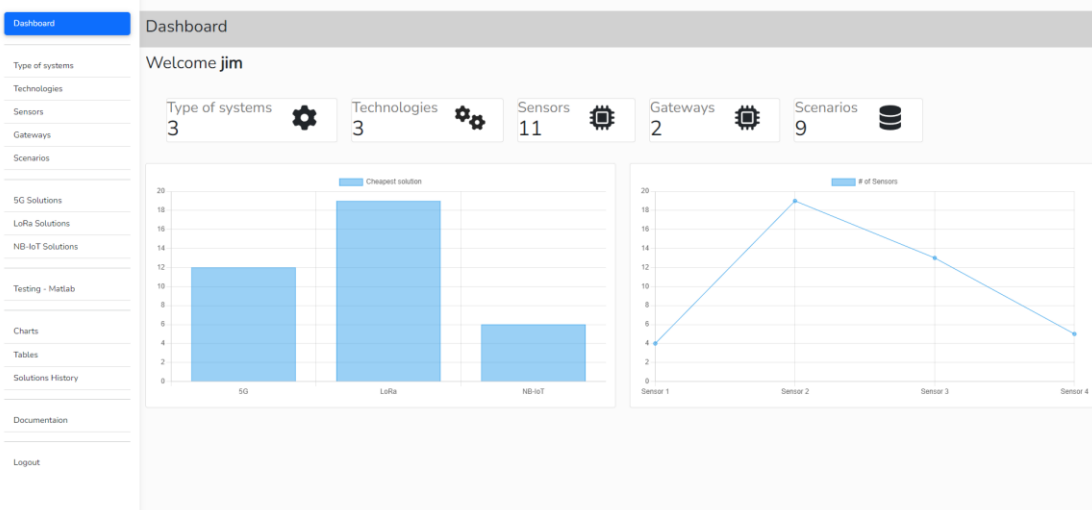
Για να εγγραφεί ο νέος χρήστης χρειάζεται να συμπληρώσει ένα email και έναν κωδικό πρόσβασης. Και για την είσοδο – σύνδεση απλά χρειάζεται να τα συμπληρώσει τα πεδία email και κωδικό.



Εικόνα 26 Σελίδα σύνδεσης νέου χρήστη

Αφού ολοκληρωθεί η σύνεση ο χρήστης οδηγείται σε ένα Dashboard στο οποίο προβάλλονται κάποια στατιστικά και στα δεξιά ένα Sidebar με πολλές επιλογές βλέπε εικόνα (27).

Το sidebar περιλαμβάνει όλες τις επιλογές και τις δυνατότητες του προτεινόμενου εργαλείου. Είναι χωρισμένο και ομαδοποιημένο με τέτοιο τρόπο ώστε να βοηθάει τον χρήστη ακόμα περισσότερο στην χρήση. Πιο συγκεκριμένα, πρώτες επιλογές είναι για την διαχείριση των δεδομένων και των στοιχείων, όπως, τύποι συστήματος (type of systems), τεχνολογίες (technologies), αισθητήρες (sensors), gateways, μπαταρίες (batteries) σενάρια (scenarios). Παρακάτω είναι ίδιες οι διαδικασίες αλλά για τις λύσεις (solutions). Στην συνέχεια φαίνεται η επιλογή Testing – Matlab όπου είναι η επιλογή για τις προσομοιώσεις και να τρέξουν όλες οι απαραίτητες διαδικασίες τους. Η επόμενη ομάδα επιλογών είναι τα διαγράμματα (Charts) και οι πίνακες (Tables) όπου εκεί προβάλλονται τα αποτελέσματα. Τελευταία ομάδα επιλογών, είναι πιο γενική όπου περιέχεται οδηγός (documentation) για την εφαρμογή και φυσικά το κουμπί της αποσύνδεσης.



Εικόνα 27 Dashboard

Sidebar – Type of system

All Type of system				
ID	Name	Description	Updated At	Actions
3	Soil Moisture Recording		05/Feb/2023	Edit Delete
2	Water Leak		24/Sep/2022	Edit Delete
1	Water Quality		25/Sep/2022	Edit Delete

[Add System](#)

Εικόνα 28 Σελίδα διαχείρισης τύπων συστήματος

Όπως φαίνεται και στην εικόνα (28) σε αυτήν την σελίδα γίνεται η διαχείριση των τύπων του συστήματος που θέλουμε να μελετήσουμε και να βρούμε λύσεις για αυτούς. Πιο συγκεκριμένα, σε αυτή την σελίδα ο χρήστης μπορεί να προσθέσει νέο σύστημα πατώντας το γαλάζιο κουμπί “Add System” όπου τον οδηγεί στην σελίδα που φαίνεται στην εικόνα (29). Ο χρήστης μπορεί επίσης να κάνει επεξεργασία του ονόματος ή της περιγραφής των ήδη υπάρχοντων συστημάτων που έχει δημιουργήσει πατώντας το μπλε κουμπί “Edit”. Τέλος, μπορεί φυσικά να τα διαγράψει πατώντας το κόκκινο κουμπί “Delete”.

Add Type of system	
Type of System	<input type="text"/>
Description	<input type="text"/>
Add System	Back

Εικόνα 29 Σελίδα προσθήκης τύπου συστήματος

Sidebar – Technologies

All Technologies			
ID	Technology	Updated At	Actions
3	NB-IoT	2022-09-24T07:35:15.000000Z	Edit Delete
2	LoRa	2022-09-24T07:35:09.000000Z	Edit Delete
1	5G	2022-09-24T07:35:04.000000Z	Edit Delete

[Add Technology](#)

Add Technology	
Technology Name	<input type="text"/>
Add Technology	Back

Sidebar – Sensors

All Sensors

ID	Name	Technology	Cost	Installation cost	Description	Updated At	Actions
11	Type D	5G	1000	300		2023-02-16T21:47:24.000000Z	Edit Delete
10	Y-Type CSolowarm	NB-IoT	400	50		2022-10-27T19:58:54.000000Z	Edit Delete
9	Type C	NB-IoT	20	3		2022-09-25T14:23:17.000000Z	Edit Delete
8	Type B	NB-IoT	100	50		2022-09-25T14:23:08.000000Z	Edit Delete
7	Type A	NB-IoT	60	10		2022-09-25T14:22:59.000000Z	Edit Delete
6	Type C	LoRa	40	20		2022-09-25T14:21:53.000000Z	Edit Delete
5	Type B	LoRa	1000	50		2022-09-25T14:21:40.000000Z	Edit Delete
4	Type A	LoRa	100	50		2022-09-25T14:21:27.000000Z	Edit Delete
3	Type C	5G	30	5		2022-09-25T14:21:14.000000Z	Edit Delete
2	Type B	5G	10	1		2022-09-25T14:19:57.000000Z	Edit Delete
1	Type A	5G	1000	30		2022-09-25T14:19:43.000000Z	Edit Delete

[Add Sensor](#)

Add Sensor

Name

Technology

Sensor cost

Installation cost

Battery

Description

[Add sensor](#) [Back](#)

Sidebar – Gateways

All Gateways

ID	Name	Technology	Description	Updated At	Actions
2	GatewayType B	2		2022-09-25T19:29:14.000000Z	Edit Delete
1	GatewayType A	2		2022-09-25T19:29:14.000000Z	Edit Delete

[Add Gateway](#)

Add Gateway

Type of System

Technology

Description

[Add Gateway](#) [Back](#)

Sidebar – Scenarios

All Scenarios

ID	Type of System	Technology	Description	Updated At	Actions
9	Soil Moisture recording	NB-IoT		2022-12-20T18:30:23.000000Z	Edit Delete
8	Soil Moisture recording	LoRa		2022-12-20T18:30:17.000000Z	Edit Delete
7	Soil Moisture recording	5G		2022-12-20T18:30:13.000000Z	Edit Delete
6	Water Quality	NB-IoT		2022-09-25T16:34:53.000000Z	Edit Delete
5	Water Quality	LoRa		2022-09-25T16:34:33.000000Z	Edit Delete
4	Water Quality	5G		2022-09-25T16:34:43.000000Z	Edit Delete
3	Water Leak	NB-IoT		2022-09-25T13:39:26.000000Z	Edit Delete
2	Water Leak	LoRa		2022-09-25T13:39:42.000000Z	Edit Delete
1	Water Leak	5G		2022-09-25T09:11:44.000000Z	Edit Delete

[Add Scenario](#)

Add Scenarios

Technology

Type of System

Description

[Add Scenario](#) [Back](#)

Sidebar – 5G Solution

All 5G Solutions

[Add Solution](#)

ID	Sensor 1	Number of sensors	Sensor 2	Number of sensors	Sensor 3	Number of sensors	Updated_at	Actions
50	Type A	29	Type B	40	Type C	31		Edit Delete
49	Type A	57	Type B	45	Type C	33		Edit Delete
48	Type A	41	Type B	25	Type C	54		Edit Delete
47	Type A	64	Type B	56	Type C	42		Edit Delete
46	Type A	72	Type B	32	Type C	56		Edit Delete
45	Type A	68	Type B	18	Type C	51		Edit Delete
44	Type A	45	Type B	16	Type C	69		Edit Delete
43	Type A	36	Type B	41	Type C	68		Edit Delete
42	Type A	15	Type B	32	Type C	65		Edit Delete
41	Type A	24	Type B	18	Type C	27		Edit Delete
40	Type A	20	Type B	57	Type C	59		Edit Delete
39	Type A	74	Type B	22	Type C	19		Edit Delete
38	Type A	71	Type B	45	Type C	50		Edit Delete
37	Type A	48	Type B	74	Type C	63		Edit Delete

Add 5G Solution

Sensor 1

Number of sensors

Sensor 2

Number of sensors

Sensor 3

Number of sensors

[Add Solution](#) [Back](#)

Edit 5G Solutions

Sensor 1

Number of sensors

Sensor 2

Number of sensors

Sensor 3

Number of sensors

[Update Solution](#) [Back](#)

Sidebar – LoRa Solution

All Lora Solutions

[Add Solution](#)

ID	Sensor 1	Number of sensors	Sensor 2	Number of sensors	Sensor 3	Number of sensors	Gateways 1	Number of gateway	Gateways 2	Number of gateway	Updated_at	Actions
50	Type A	16	Type B	81	Type C	38	GatewayType A	19	GatewayType B	0		Edit Delete
49	Type A	23	Type B	39	Type C	18	GatewayType A	38	GatewayType B	0		Edit Delete
48	Type A	82	Type B	96	Type C	28	GatewayType A	30	GatewayType B	0		Edit Delete
47	Type A	43	Type B	16	Type C	79	GatewayType A	43	GatewayType B	0		Edit Delete
46	Type A	18	Type B	98	Type C	30	GatewayType A	52	GatewayType B	0		Edit Delete
45	Type A	48	Type B	99	Type C	44	GatewayType A	73	GatewayType B	0		Edit Delete
44	Type A	48	Type B	59	Type C	65	GatewayType A	33	GatewayType B	0		Edit Delete
43	Type A	31	Type B	61	Type C	99	GatewayType A	65	GatewayType B	0		Edit Delete
42	Type A	68	Type B	78	Type C	17	GatewayType A	16	GatewayType B	0		Edit Delete
41	Type A	23	Type B	68	Type C	89	GatewayType A	75	GatewayType B	0		Edit Delete
40	Type A	56	Type B	75	Type C	44	GatewayType A	94	GatewayType B	0		Edit Delete
39	Type A	48	Type B	65	Type C	23	GatewayType A	41	GatewayType B	0		Edit Delete
38	Type A	91	Type B	94	Type C	91	GatewayType A	79	GatewayType B	0		Edit Delete
37	Type A	19	Type B	50	Type C	19	GatewayType A	79	GatewayType B	0		Edit Delete

Add LoRa Solution

Sensor 1

Number of sensors

Sensor 2

Number of sensors

Sensor 3

Number of sensors

Gateway 1

Number of gateways

Gateway 2

Number of gateways

[Add Solution](#) [Back](#)

Edit LoRa Solutions

Sensor 1
 Type A

Number of sensors

Sensor 2
 Type B

Number of sensors

Sensor 3
 Type C

Number of sensors

Gateway 1
 GatewayType A

Number of gateway

Gateway 2
 GatewayType B

Number of gateway

[Update Solution](#) [Back](#)

Sidebar – NB-IoT Solution

All NB Solutions

[Add Solution](#)

ID	Sensor 1	Number of sensors	Sensor 2	Number of sensors	Sensor 3	Number of sensors	Updated_Lat	Actions
50	Type A	69	Type B	16	Type C	70		Edit Delete
49	Type A	51	Type B	27	Type C	70		Edit Delete
48	Type A	50	Type B	43	Type C	40		Edit Delete
47	Type A	43	Type B	36	Type C	35		Edit Delete
46	Type A	68	Type B	20	Type C	18		Edit Delete
45	Type A	41	Type B	43	Y-Type CSolowarm	72		Edit Delete
44	Type A	47	Type B	73	Type C	69		Edit Delete
43	Type A	29	Type B	26	Type C	73		Edit Delete
42	Type A	57	Type B	69	Type C	38		Edit Delete
41	Type A	19	Type B	27	Type C	28		Edit Delete
40	Type A	58	Type B	20	Type C	35		Edit Delete
39	Type A	55	Type B	30	Type C	71		Edit Delete
38	Type A	66	Type B	56	Type C	64		Edit Delete
37	Type A	39	Type B	72	Type C	70		Edit Delete

Add 5G Solution

Sensor 1

Number of sensors

Sensor 2

Number of sensors

Sensor 3

Number of sensors

Gateway 1

Number of gateways

Gateway 2

Number of gateways

Edit LoRa Solutions

Sensor 1
Type A

Number of sensors

Sensor 2
Type B

Number of sensors

Sensor 3
Type C

Number of sensors

Gateway 1
GatewayType A

Number of gateway

Gateway 2
GatewayType B

Number of gateway

Sidebar – Testing-Matlab

Prepare Testing

Type of System

Select technology

Prepare Testing

Type of System

Select system

- Water Quality
- Water Leak
- Soil Moisture Recording

MATLAB was opened, wait a while for the necessary processes to load.

STEP 1: Update input with the current enabled scenarios

Update

Scenario: **5G Technology**

28-02-2023_5g_scenario.json file was created successfully!

Scenario: **NB-IoT Technology**

28-02-2023_lora_scenario.json file was created successfully!

Scenario: **LoRa Technology**

28-02-2023_nb_scenario.json file was created successfully!

Press the buttons below to generate data:

Calculate Battery

Calculate Cost

STEP 2: Run whichever algorithm you want. (Selected system: **Water Quality**)

Greedy code, find cheapest scenarios

- For cost-effective scenario:
Directory: `storage\MatlabCodes\Greedy_CostEffective_Scenrio.m`
- For energy-effective scenario:
Directory: `storage\MatlabCodes\Greedy_EnergyEffective_Scenrio.m`

Genetic Algorithm

- For most-effective scenario:
Directory: `storage\MatlabCodes>Main_GA.m`

STEP 3: Results creation.

Results

Fetch results

Sidebar – Charts

Charts

Results from Greedy Algorithm

Bars - Cheapest solution (5G)

Cheapest Scenarios using 5G Technology (Greedy Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Bars - Cheapest solution (5g)

Battery Efficient Scenarios using 5g Technology (Greedy Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Bars - Cheapest solution (LoRa)

Cheapest Scenarios using LoRa Technology (Greedy Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Bars - Battery Efficient solution (LoRa)

Battery Efficient Scenarios using LoRa Technology (Greedy Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Bars - Cheapest solution (NB-IoT)

Cheapest Scenarios using NB IoT Technology (Greedy Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Bars - Cheapest solution (NB-IoT)

Cheapest Scenarios using NB IoT Technology (Greedy Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Results from Genetic Algorithm

Bars - Most effective solution (Cost)

Cheapest Scenarios using 5G, LoRa, NB-IoT technologies (Genetic Algorithm Results)

Date

2023-02-28

Simulation Number

Enter simulation number

Plot chart

Bars - Most effective solution (Battery Life)

Battery efficient using 5G, LoRa, NB-IoT technologies (Genetic Algorithm Results)

Date

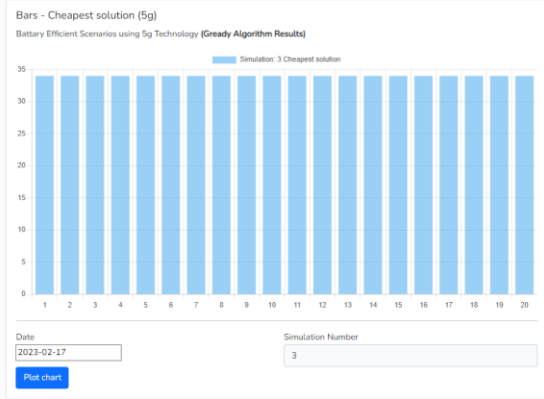
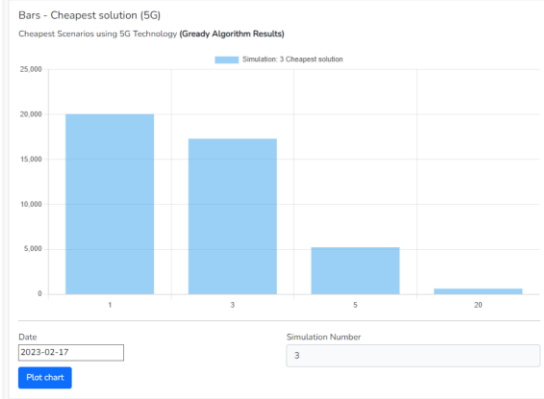
2023-02-28

Simulation Number

Enter simulation number

Plot chart

Results from Greedy Algorithm



Results from Genetic Algorithm



Sidebar – Tables

All Results

Simulation_number	technology	Solution number	Type of system	Execution date	Cost	Battery
Add						

6. Επίλογος & Αξιολόγηση αποτελεσμάτων

Σε αυτήν την ενότητα αναφέρονται τα σενάρια, τα οποία μελετήθηκαν, και τα αποτελέσματα που προέκυψαν μετά από τις διαδικασίες που εκτελέστηκαν. Γίνεται μια μικρή ανάλυση και περιγραφή των αποτελεσμάτων με κάποια συμπληρωματικά συμπεράσματα τα οποία παρατηρήθηκαν. Τέλος, αναφέρονται μελλοντικές επεκτάσεις και στο κομμάτι της έρευνας και των αλγορίθμων, αλλά και του βοηθητικού εργαλείου που προτείνεται.

6.1 Σενάρια/Περιπτώσεις χρήσης

Τα αποτελέσματα τα οποία μπορούμε να δούμε μέσα από το tool έχουν δημιουργηθεί έπειτα από έναν πρώτο διαχωρισμό σε σενάρια και περιπτώσεις. Κάθε σενάριο αποτελείται από κάποια χαρακτηριστικά, όπως, το είδος του συστήματος και την τεχνολογία. Τα σενάρια και οι περιπτώσεις που μελετήθηκαν και συμπεριλήφθηκαν είναι τα εξής:

- 5G – Διαρροή νερού (Water Leaking)
- LoRa – Διαρροή νερού (Water Leaking)
- NB-IoT – Διαρροή νερού (Water Leaking)
- 5G – Ποιότητα νερού (Water Quality)
- LoRa – Ποιότητα νερού (Water Quality)
- NB-IoT – Ποιότητα νερού (Water Quality)
- 5G – Καταγραφή υγρασίας εδάφους (Soil Moisture recording)
- LoRa – Καταγραφή υγρασίας εδάφους (Soil Moisture recording)
- NB-IoT – Καταγραφή υγρασίας εδάφους (Soil Moisture recording)

Φυσικά όλα τα παραπάνω είναι παραμετροποιήσιμα μέσα από το tool. Μπορεί να αφαιρεθούν και να προστεθούν νέα. Επίσης, κάθε πείραμα εκτελείται για 3 σενάρια με κοινό χαρακτηριστικό το είδος του συστήματος. Υπάρχει και η επιλογή να τρέξουν και οι 2 αλγόριθμοι. Τα αποτελέσματα βγαίνουν με τον παραπάνω τρόπο γιατί αυτό που θέλουμε είναι συγκρίνουμε τα αποτελέσματα και να καταλήξουμε στην πιο συμφέρουσα και αποδοτική λύση.

6.2 Ανάλυση Αποτελεσμάτων/Συμπεράσματα

Σε αυτήν την υποενότητα περιγράφονται τα αποτελέσματα και κάποια συμπεράσματα. Αρχικά τα αποτελέσματα προβάλλονται σε ξεχωριστά διαγράμματα, ανά σενάριο. Έτσι μπορούμε πολύ εύκολα να διακρίνουμε τα αποτελέσματα και να προβούμε σε συγκρίσεις διαβάζοντας τα διαγράμματα. Παραπάνω στην εικόνα της σελίδας με τα charts φαίνεται ένας τρόπος με τον οποίο το προτεινόμενο εργαλείο εμφανίζει τα αποτελέσματα. Τέλος, διαπιστώθηκε ότι τα αποτελέσματα του γενετικού αλγορίθμου είναι πιο καλά από αυτά του άπληστου αλγορίθμου. Ένα λόγος είναι ότι στον άπληστο αλγόριθμο παίζει και ο ανθρώπινος παράγοντας διότι τα δεδομένα που λαμβάνει και επεξεργάζεται, έχουν δημιουργηθεί από κάποιον χρήστη. Σε αντίθεση με τον γενετικό όπου “φτιάχνει” τα αποτελέσματα χωρίς να πάρει δεδομένα από κάποια πηγή, όπως γίνεται με τον άπληστο αλγόριθμο. Ο άπληστος αλγόριθμος βρίσκει την βέλτιστη - φθηνότερη λύση από μια “τράπεζα” σεναρίων ενώ ο γενετικός γεννά σενάρια και βγάζει τα αποτελέσματα βάση αυτών.

6.3 Μελλοντικές επεκτάσεις

Η όλη διαδικασία και η όλη ιδέα αυτής της διπλωματικής εργασίας είναι πάρα πολύ επεκτάσιμη και παραμετροποιήσιμη και ακόμα περισσότερο με τη χρήση του προτεινόμενου βοηθητικού εργαλείου που παρουσιάστηκε. Πιο συγκεκριμένα, οι επεκτάσεις που μπορούν να γίνουν είναι αρχικά σε επίπεδο σεναρίων, επιλέγοντας άλλες τεχνολογίες και άλλα συστήματα προς μελέτη και όχι μόνο για το νερό. Επίσης, πολύ βασικότερη είναι η επέκταση και σε άλλους ευριστικούς ή αλγορίθμους βελτιστοποίησης, είτε προσθέτοντας νέους είτε ακόμα παραμετροποιώντας τον ήδη υπάρχον, προτεινόμενο γενετικό αλγόριθμο. Για παράδειγμα, αλλαγές στον αριθμό του πληθυσμού, τον τρόπο επιλογής (selection), τον τρόπο διασταύρωσης (crossover), και τον τρόπο μετάλλαξης (mutation). Πέρα από το θεωρητικό και το αλγοριθμικό κομμάτι, στο εργαλείο έχουν προνοηθεί κάποιες επεκτάσεις για την δημιουργία, επεξεργασία και διαγραφή κάποιων δεδομένων. Όμως δυνητικά μπορεί να εξελιχθεί σε ένα πολύ ισχυρό εργαλείο που θα είναι για πολλαπλή χρήση. Πιο συγκεκριμένα μπορούν να δημιουργηθούν και άλλα διαγράμματα και αναφορές.

References

- [1] Mattern, F., & Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. *IEEE Pervasive Computing*, 9(1), 44-51.
- [2] Rose, K., Eldridge, S., & Chapin, L. (2015). Internet of Things – An overview.
- [3] TechTarget. (n.d.). HVAC definition
<https://www.techtarget.com/searchdatacenter/definition/HVAC>
- [4] Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues. *Computer Communications*, 54, 1-31. <https://doi.org/10.1016/j.comcom.2014.09.008>
- [5] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*, vol. 2017, p. 25, 2017. Article ID 9324035, doi:10.1155/2017/9324035.
- [6] N. C. Gaitan, V. G. Gaitan, and I. Ungurean, "A Survey on the Internet of Things Software Architecture," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 12, 2015.
- [7] Antão, Liliana & Pinto, Rui & Reis, João Pedro & Gonçalves, Gil. (2018). Requirements for Testing and Validating the Industrial Internet of Things. 10.1109/ICSTW.2018.00036.
- [8] TechTarget. (n.d.). 5G definition
<https://www.techtarget.com/searchnetworking/definition/5G>
- [9] Semtech Corporation. (n.d.). What is LoRa? <https://www.semtech.com/lora/what-is-lora>
- [10] Design And Reuse. (2019, October 17). Semtech Acquires Wireless Long Range IP Provider Cycleo. <https://www.design-reuse.com/news/45552/semtech-acquires-wireless-long-range-ip-provider-cycleo.html>
- [11] Wikipedia. (n.d.). LoRa, from <https://en.wikipedia.org/wiki/LoRa>
- [12] iSecurion. (2017, May 11). IoT Communication Protocols, <http://blog.isecurion.com/2017/05/11/iot-communication-protocols>
- [13] Inet Services Cloud. (n.d.). The Four Internet of Things Connectivity Models Explained. <http://www.inetservicescloud.com/the-four-internet-of-things-connectivity-models-explained>
- [14] iSecurion. (2017, May 11). IoT Communication Protocols. <http://blog.isecurion.com/2017/05/11/iot-communication-protocols>
- [15] Nike. (n.d.). NTC App. <https://www.nike.com/ntc-app>
- [16] SmartThings. (n.d.). SmartThings. <https://www.smartthings.com>
- [17] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," *IEEE Internet Initiative*, iot.ieee.org, May 2015

- [18] Digiteum. (n.d.). IoT in Agriculture. <https://www.digiteum.com/iot-agriculture>
- [19] Digiteum. (n.d.). Agriculture Software Development. <https://www.digiteum.com/agriculture-software-development>
- [20] Computer Science and Engineering, The University of New South Wales. (n.d.). Overview. <https://www.cse.unsw.edu.au/~cs9417ml/GA2/overview.html>
- [21] Introduction to Genetic Algorithms (Including Example Code). Towards Data Science. <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>
- [22] Obitko, M. (n.d.). Genetic Algorithms: Selection. <https://www.obitko.com/tutorials/genetic-algorithms/selection.php>
- [23] Obitko, M. (n.d.). Genetic Algorithms: Crossover & Mutation. <https://www.obitko.com/tutorials/genetic-algorithms/crossover-mutation.php>