Πρότυπο Σχέδιο Συγγραφής Διπλωματικής Εργασίας



ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Δεδομένων από IoT χρησιμοποιώντας Τεχνικές Μεγάλων Δεδομένων και Python

ΝΙΚΟΛΑΟΣ ΒΟΥΔΟΥΚΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΑ: ΜΙΧΑΛΗΣ ΓΕΩΡΓΙΑΔΗΣ

ΙΑΝΟΥΑΡΙΟΣ 2025



ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Δεδομένων από IoT χρησιμοποιώντας Τεχνικές Μεγάλων Δεδομένων και Python

Διπλωματική Εργασία η οποία υποβλήθηκε προς απόκτηση του Μεταπτυχιακού στα Πληροφοριακά Συστήματα και τη Ψηφιακή Καινοτομία

ΝΙΚΟΛΑΟΣ ΒΟΥΔΟΥΚΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΑ: ΜΙΧΑΛΗΣ ΓΕΩΡΓΙΑΔΗΣ

ΙΑΝΟΥΑΡΙΟΣ 2025

Πνευματικά δικαιώματα

Copyright © Νικόλαος Βουδούκης, 2025

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της Διπλωματικής Εργασίας από το Πανεπιστημίου Νεάπολις δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Πανεπιστημίου.

Σελίδα Εγκυρότητας

Ονοματεπώνυμο Φοιτητή: Νικόλαος Βουδούκης

Τίτλος Διπλωματικής Εργασίας: Ανάλυση Δεδομένων από IoT χρησιμοποιώντας Τεχνικές Μεγάλων Δεδομένων και Python

Η παρούσα Διπλωματική Εργασία εκπονήθηκε στο πλαίσιο των σπουδών για την απόκτηση εξ αποστάσεως μεταπτυχιακού τίτλου στο Πανεπιστήμιο Νεάπολις και εγκρίθηκε στις 6/2/2025 από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή:

Πρώτος επιβλέπων (Πανεπιστήμιο Νεάπολις Πάφος) ΜΙΧΑΛΗΣ ΓΕΩΡΓΙΑΔΗΣ

Μέλος Εξεταστικής Επιτροπής

Μέλος Εξεταστικής Επιτροπής

Ή ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ

Ο Νικόλαος Βουδούκης, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Ανάλυση Δεδομένων από IoT χρησιμοποιώντας Τεχνικές Μεγάλων Δεδομένων και Python», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την

κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Ο Δηλών

Νικόλαος Βουδούκης

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή 6
1.1 Εισαγωγή στο Διαδίκτυο των Πραγμάτων (IoT) 6
1.2 Η σημασία των Μεγάλων Δεδομένων στο IoT 8
1.3 Σκοπός και Σημασία της Μελέτης 10
1.5 Ερευνητικά Ερωτήματα 11
2. Ανασκόπηση Βιβλιογραφίας 11
2.1 Διαδίκτυο των Πραγμάτων (IoT) 11
2.1.1 Ιστορία και Εξέλιξη του IoT 11
2.1.2 Τεχνολογίες και Πρωτόκολλα IoT 14
2.2 Μεγάλα Δεδομένα (Big Data) 22
2.2.1 Ορισμός και Ιδιότητες των Μεγάλων Δεδομένων 22
2.2.2 Εφαρμογές Μεγάλων Δεδομένων στο IoT 25
2.3 Τεχνικές και Εργαλεία για Ανάλυση Δεδομένων 27
2.3.1 Βιβλιοθήκες Python 27
2.3.2 Τεχνικές και Εργαλεία για Ανάλυση Δεδομένων 28
2.4 Έξυπνος οικιακός έλεγχος - Πίνακες εργαλείων 31
2.5 Αποτελεσματική διαχείριση ενέργειας – Στρατηγικές 33
2.6 Ασφάλεια έξυπνων σπιτιών - Κίνδυνοι και λύσεις 37
3. Οπτικοποίηση των δεδομένων από το IoT 41
3.1 Οπτικοποίηση δεδομένων για το IoT: Μετασχηματισμός σε αξιοποιήσιμες γνώσεις 41
3.2 Πηγές δεδομένων για οπτικοποίηση στο IoT 42
3.3 Βασικά οφέλη της οπτικοποίησης δεδομένων για το IoT 43
3.4 Τεχνικές οπτικοποίησης IoT 44
3.5 Εργαλεία για την οπτικοποίηση δεδομένων IoT 44
3.6 Εφαρμογές της οπτικοποίησης IoT 45
3.7 Προκλήσεις και προβληματισμοί στην οπτικοποίηση δεδομένων IoT 46
4. Μεθοδολογία για την ανάπτυξη Πίνακα Ελέγχου οπτικοποίησης δεδομένων IoT 46
5. Υλοποίηση Πίνακα Ελέγχου οπτικοποίησης δεδομένων IoT 49
5.1 Πίνακας ελέγχου κατανάλωσης ενέργειας 50
5.2 Πίνακας ελέγχου περιβαλλοντικών δεδομένων 51
5.3 Πίνακας ελέγχου δεδομένων ασφάλειας 52

5.4 Πίνακας ελέγχου παρακολούθησης συσκευών 52
5.5 Πίνακας ελέγχου παρακολούθησης δεδομένων πληρότητας 53
5.6 Πίνακας ελέγχου παρακολούθησης δεδομένων υγείας 54
5.7 Πίνακας ελέγχου παρακολούθησης δεδομένων συσκευών 55
5.8 Πίνακας ελέγχου παρακολούθησης δεδομένων συνδεσιμότητας και δικτύου 56
5.9 Πίνακας ελέγχου με δεδομένα ειδοποιήσεων (alarms). 56
6. Συμπεράσματα 58
6. Βιβλιογραφικές Αναφορές 60
ΠΑΡΑΡΤΗΜΑ Α. Κώδικας της εφαρμογής 65

1. Εισαγωγή

1.1 Εισαγωγή στο Διαδίκτυο των Πραγμάτων (IoT)

Το διαδίκτυο των πραγμάτων (IoT) αναφέρεται σε ένα δίκτυο φυσικών συσκευών, οχημάτων, συσκευών και άλλων φυσικών αντικειμένων που είναι ενσωματωμένα με αισθητήρες, λογισμικό και συνδεσιμότητα δικτύου, επιτρέποντάς τους να συλλέγουν και να μοιράζονται δεδομένα. Οι συσκευές IoT μπορεί να κυμαίνονται από απλές συσκευές «έξυπνου σπιτιού», όπως έξυπνοι θερμοστάτες, έξυπνες μικροσυσκευές, μέχρι πολύπλοκα βιομηχανικά μηχανήματα και συστήματα μεταφορών. Το διαδίκτυο των πραγμάτων είναι ένα δίκτυο φυσικών αντικειμένων που συλλέγουν, ανταλλάσσουν και χρησιμοποιούν δεδομένα. Οι συσκευές IoT χρησιμοποιούν αισθητήρες για τη συλλογή δεδομένων του πραγματικού κόσμου και ενεργοποιητές για την ανάληψη φυσικών ενεργειών. Οι τεχνικές ανάλυσης δεδομένων εξάγουν πολύτιμες πληροφορίες από τα δεδομένα, επιτρέποντας τη λήψη τεκμηριωμένων αποφάσεων και τη βελτιστοποίηση. Επιπλέον, η δυναμική φύση του IoT φέρνει επανάσταση στις αλληλεπιδράσεις με την τεχνολογία και υπόσχεται ένα μέλλον όπου τα αντικείμενα αλληλοεπιδρούν έξυπνα, ενισχύοντας την ευκολία, την αποδοτικότητα και τη συνδεσιμότητα. Χρησιμοποιεί πολλαπλά ενσωματωμένα συστήματα για τη συλλογή δεδομένων σχετικά με την κίνηση, τη θερμοκρασία και άλλες πτυχές του περιβάλλοντος του χρήστη και είναι μια εξαιρετική επιλογή για τις επιχειρήσεις που θέλουν να προσφέρουν καλύτερη εξυπηρέτηση πελατών. Με τη δημιουργία συσκευών που γνωρίζουν τις απαιτήσεις των χρηστών, οι επιχειρήσεις μπορούν να βελτιώσουν την ακρίβεια, την αποτελεσματικότητα και την ευκολία (Atzori et al., 2010).

Το IoT επιτρέπει σε αυτές τις έξυπνες συσκευές να επικοινωνούν μεταξύ τους και με άλλες συσκευές με δυνατότητα σύνδεσης στο διαδίκτυο δημιουργώντας ένα τεράστιο δίκτυο διασυνδεδεμένων συσκευών που μπορούν να ανταλλάσσουν δεδομένα και να εκτελούν διάφορες εργασίες αυτόνομα. Αυτό μπορεί να περιλαμβάνει την παρακολούθηση των περιβαλλοντικών συνθηκών, τη διαχείριση κυκλοφοριακών προτύπων με έξυπνα αυτοκίνητα και άλλες έξυπνες συσκευές αυτοκινήτων, τον έλεγχο μηχανών και διαδικασιών σε εργοστάσια και την παρακολούθηση αποθεμάτων και αποστολών σε αποθήκες. Οι πιθανές εφαρμογές του IoT είναι τεράστιες και ποικίλες και ο αντίκτυπός του είναι ήδη αισθητός σε ένα ευρύ φάσμα βιομηχανιών, όπως η μεταποίηση, οι μεταφορές, η υγειονομική περίθαλψη και η γεωργία. Καθώς ο αριθμός των συνδεδεμένων στο διαδίκτυο συσκευών συνεχίζει να αυξάνεται, το IoT είναι πιθανό να διαδραματίσει ολοένα και πιο σημαντικό ρόλο στη διαμόρφωση της κοινωνίας μετασχηματίζοντας τον τρόπο ζωής, εργασίας και αλληλεπίδρασης. Σε επιχειρησιακό πλαίσιο, οι συσκευές IoT χρησιμοποιούνται για την παρακολούθηση ενός ευρέος φάσματος παραμέτρων, όπως η θερμοκρασία, η υγρασία, η ποιότητα του αέρα, η κατανάλωση ενέργειας

και η απόδοση των μηχανημάτων. Τα δεδομένα αυτά μπορούν να αναλυθούν σε πραγματικό χρόνο για τον εντοπισμό μοτίβων, τάσεων και ανωμαλιών που μπορούν να βοηθήσουν τις επιχειρήσεις να βελτιστοποιήσουν τις λειτουργίες τους και να βελτιώσουν το τελικό τους αποτέλεσμα.

Το διαδίκτυο των πραγμάτων είναι ένα δίκτυο φυσικών αντικειμένων που είναι ενσωματωμένα με αισθητήρες, λογισμικό και άλλες τεχνολογίες για τη συλλογή και ανταλλαγή δεδομένων. Αυτά τα αντικείμενα αναφέρονται ως «πράγματα» και μπορεί να κυμαίνονται από συνηθισμένα οικιακά αντικείμενα έως εξελιγμένα βιομηχανικά εργαλεία. Το IoT έχει τη δυνατότητα να φέρει επανάσταση σε πολλούς κλάδους και πτυχές της καθημερινότητας, για παράδειγμα, μπορεί να χρησιμοποιηθεί για τη βελτίωση της αποδοτικότητας στην παραγωγή, την παρακολούθηση και τη διαχείριση περιουσιακών στοιχείων και την παροχή καλύτερης υγειονομικής περίθαλψης. Τα αντικείμενα αυτά μπορεί να κυμαίνονται από καθημερινές συσκευές, όπως τα έξυπνα τηλέφωνα και οι έξυπνες συσκευές, έως πολύπλοκα συστήματα, όπως βιομηχανικά μηχανήματα και συσκευές παρακολούθησης του περιβάλλοντος. Η κατανόηση των εννοιών πίσω από το διαδίκτυο των πραγμάτων είναι ζωτικής σημασίας για την κατανόηση των δυνατοτήτων και των επιπτώσεών του. Το IoT βασίζεται στην ικανότητα των φυσικών αντικειμένων να συνδέονται στο διαδίκτυο με την χρήση δικτύων χαμηλής ισχύος και ευρείας περιοχής. Οι συσκευές IoT είναι συνήθως εξοπλισμένες με αισθητήρες που συλλέγουν δεδομένα σχετικά με το περιβάλλον που μπορούν να χρησιμοποιηθούν για την παρακολούθηση των συνθηκών, την παρακολούθηση της κίνησης και την ανίχνευση συμβάντων. Οι συσκευές IoT πρέπει να μπορούν να επικοινωνούν μεταξύ τους και με το σύννεφο γεγονός που γίνεται με τη χρήση τυποποιημένων πρωτοκόλλων. Τα δεδομένα που συλλέγονται μπορούν να αναλυθούν για να αποκτηθούν γνώσεις σχετικά με τον φυσικό κόσμο αλλά και να χρησιμοποιηθούν για τη βελτίωση της αποδοτικότητας, τη βελτιστοποίηση των διαδικασιών και τη λήψη καλύτερων αποφάσεων (Alavi et al., 2018).

Το διαδίκτυο των πραγμάτων βασίζεται σε ένα ευρύ φάσμα τεχνολογιών που συνεργάζονται για τη λειτουργικότητα και τη συνδεσιμότητά του. Οι αισθητήρες είναι συσκευές που μετρούν φυσικές ή περιβαλλοντικές συνθήκες και τις μετατρέπουν σε ηλεκτρικά σήματα. Επιπλέον, αυτά τα σήματα μπορούν στη συνέχεια να μεταδοθούν σε άλλες συσκευές ή συστήματα. Οι αισθητήρες είναι απαραίτητοι καθώς παρέχουν τα δεδομένα που επιτρέπουν στις συσκευές IoT να λαμβάνουν αποφάσεις και να αναλαμβάνουν δράση. Τα πρωτόκολλα επικοινωνίας είναι οι κανόνες που διέπουν τον τρόπο με τον οποίο οι συσκευές επικοινωνούν μεταξύ τους. Υπάρχουν πολλά διαφορετικά πρωτόκολλα επικοινωνίας που χρησιμοποιούνται το καθένα με τα δικά του πλεονεκτήματα και μειονεκτήματα. Μερικά από τα πιο συνηθισμένα πρωτόκολλα επικοινωνίας για συσκευές IoT περιλαμβάνουν τα MQTT, CoAP και HTTP. Επίσης, υπάρχει μια ποικιλία τεχνολογιών δικτύωσης που μπορούν να χρησιμοποιηθούν για συσκευές IoT, όπως Wi-Fi, κυψελοειδής και LPWAN (low power wide area network) που είναι ένας τύπος

τεχνολογίας δικτύωσης που έχει σχεδιαστεί για συσκευές IoT που έχουν περιορισμένη ισχύ και εύρος ζώνης. Το υπολογιστικό νέφος παρέχει μια πλατφόρμα για την αποθήκευση και την επεξεργασία δεδομένων και είναι απαραίτητο, καθώς επιτρέπει να αποθηκεύουν και να επεξεργάζονται μεγάλες ποσότητες δεδομένων χωρίς να χρειάζεται να διαθέτουν το δικό τους ειδικό υλικό. Η ανάλυση δεδομένων είναι η διαδικασία ανάλυσης δεδομένων για την εξαγωγή συμπερασμάτων και επιτρέπει στις επιχειρήσεις να αποκτούν πληροφορίες από τα δεδομένα που συλλέγονται από τις συσκευές IoT (Al-Fuqaha et al., 2015).

1.2 Η σημασία των Μεγάλων Δεδομένων στο IoT

Τα μεγάλα δεδομένα και το διαδίκτυο των πραγμάτων είναι δύο σημαντικές τάσεις στην τεχνολογία σήμερα. Τα μεγάλα δεδομένα αναφέρονται στον τεράστιο όγκο δεδομένων που παράγουμε και αναλύουμε καθημερινά, ενώ το διαδίκτυο των πραγμάτων είναι το δίκτυο συσκευών που συλλέγουν και μοιράζονται αυτά τα δεδομένα. Η σύνδεσή τους μεταμορφώνει τον τρόπο με τον οποίο συλλέγονται, αναλύονται και χρησιμοποιούνται τα δεδομένα στις επιχειρήσεις. Τα μεγάλα δεδομένα και το διαδίκτυο των πραγμάτων είναι στενά συνδεδεμένα και η κατανόηση της σύνδεσής τους βοηθά στο να αντιληφθεί ο χρήστης τη μεγαλύτερη εικόνα του ψηφιακού κόσμου. Το διαδίκτυο των πραγμάτων αναφέρεται στο δίκτυο φυσικών συσκευών που είναι συνδεδεμένες στο διαδίκτυο και διαθέτουν αισθητήρες, λογισμικό και άλλες τεχνολογίες. Αυτές οι συσκευές κυμαίνονται από οικιακά αντικείμενα έως εξελιγμένα μηχανήματα στην παραγωγή, μπορούν να συλλέγουν, να στέλνουν και να λαμβάνουν δεδομένα. Οι συσκευές IoT παράγουν συνεχώς νέες πληροφορίες, προσθέτοντας στην τεράστια δεξαμενή των μεγάλων δεδομένων. Τα εργαλεία μεγάλων δεδομένων επεξεργάζονται αυτές τις πληροφορίες, μετατρέποντας μια ροή ακατέργαστων δεδομένων από τις συσκευές IoT σε χρήσιμη γνώση. Έτσι, τα δεδομένα μετατρέπονται σε πολύτιμες γνώσεις, επιτρέποντας στις επιχειρήσεις και τα άτομα να λαμβάνουν τεκμηριωμένες αποφάσεις, να βελτιώνουν τις υπηρεσίες και να αυξάνουν την αποτελεσματικότητα των λειτουργιών (Palattella et al., 2016). Αν και το διαδίκτυο των πραγμάτων και τα μεγάλα δεδομένα είναι δύο εντελώς διαφορετικές έννοιες, αλληλοσυμπληρώνονται στην περίπτωση της σύνθεσης και επεξεργασίας πληροφοριών για την εξυπηρέτηση επωφελών σκοπών για τις επιχειρήσεις. Το IoT λειτουργεί ως η κύρια πηγή δεδομένων για τα μεγάλα δεδομένα, με τη σειρά τους τα μεγάλα δεδομένα είναι μια συλλογή τεράστιων ποσοτήτων πληροφοριών που συλλέγονται από διαφορετικές πηγές, συμπεριλαμβανομένου του IoT. Επειδή τα δεδομένα IoT είναι ως επί το πλείστων δεδομένα πραγματικού χρόνου που βασίζονται στη συμπεριφορά των πελατών των συνδεδεμένων συσκευών, μπορεί να ειπωθεί ότι τα δεδομένα πελατών που προέρχονται από δίκτυα IoT παρέχουν καλύτερη ποιότητα λεπτομέρειας σε πραγματικό χρόνο από ό,τι οι πληροφορίες από ορισμένες άλλες πηγές. Εκτός αυτού, ο καθαρός όγκος των δεδομένων που παράγονται από τις συσκευές χρηστών του IoT μπορεί να είναι λιγότερο χρήσιμος εάν η

επιχείρηση εφαρμόζει μόνο τις συνήθεις μεθόδους ανάλυσης και δεν χρησιμοποιεί τα κατάλληλα εργαλεία και τις κατάλληλες διαδικασίες που υπάρχουν. Για να ξεκλειδώσουν πλήρως το δυναμικό των δεδομένων IoT και να τα μετατρέψουν σε πολύτιμες πληροφορίες, οι ιδιοκτήτες των επιχειρήσεων πρέπει να περάσουν από μια διαδικασία όπου οι πληροφορίες αυτές εξάγονται. Η αυτόματη αλληλεπίδραση μεταξύ φυσικών συσκευών που συνδέονται στο διαδίκτυο σε μια συγκεκριμένη περίοδο θα βοηθήσει τις επιχειρήσεις να συλλέξουν μεγάλες ποσότητες δεδομένων από τους πελάτες που χρησιμοποιούν τα προϊόντα τους.

Η συλλογή των παραπάνω συλλεχθέντων κομματιών θα συγκεντρωθεί σε αρχεία εντός της βάσης δεδομένων μεγάλων δεδομένων και θα χρησιμοποιηθεί σύμφωνα με τις ανάγκες της επιχείρησης. Οι επιχειρήσεις χρησιμοποιούν αναλυτικά συστήματα για να αυτοματοποιήσουν τη διαδικασία ανάλυσης αυτού του τεράστιου όγκου πληροφοριών. Οι επιχειρήσεις ομαδοποιούν τα δεδομένα μετά την ανάλυση κατηγοριοποιώντας τα σε τρεις διαφορετικούς τύπους πληροφοριών, συμπεριλαμβανομένων των περιγραφικών αναλύσεων, πώς οι συνδεδεμένες συσκευές αποδίδουν σε πραγματικό χρόνο, των διαγνωστικών αναλύσεων γιατί οι συνδεδεμένες συσκευές παράγουν αυτές τις εκροές, των προγνωστικών αναλύσεων, πώς οι συνδεδεμένες συσκευές λειτουργούν στο μέλλον, και των προγνωστικών αναλύσεων, πώς να επηρεάσουν τα προβλεπόμενα πράγματα, δημιουργώντας έτσι αναφορές για την ανάπτυξη προϊόντων λογισμικού και τις ανάγκες συντονισμού και τη βελτίωση της λήψης αποφάσεων. Υπάρχουν πολλοί παράγοντες που επηρεάζουν τη διαδικασία λήψης αποφάσεων ενός ιδιοκτήτη επιχείρησης. Ειδικότερα, οι χρήστες διαδραματίζουν τον σημαντικότερο ρόλο, διότι ο απώτερος σκοπός ενός προϊόντος λογισμικού εξακολουθεί να είναι η εξυπηρέτηση των αναγκών των πελατών-στόχων. Οι ανάγκες των χρηστών είναι διαφορετικές από καιρό σε καιρό και αυτός είναι και ο λόγος για τον οποίο τα προϊόντα δεδομένων σε πραγματικό χρόνο από τα δίκτυα IoT έχουν τόσο μεγάλη ισχύ (Wang et al., 2014).

Οι πληροφορίες πραγματικού χρόνου αντικατοπτρίζουν την τρέχουσα κατάσταση του χρήστη και καταγράφουν τυχόν ειδικά γεγονότα που συνέβησαν κατά τη διάρκεια της εμπειρίας του προϊόντος. Συλλέγοντας και αναλύοντας τα δεδομένα που σχετίζονται με τις συνήθειες των χρηστών κατά τη στιγμή της δημιουργίας των δεδομένων χωρίς καθυστέρηση, η επιχείρηση κατανοεί τις επιθυμίες και τα σημεία πόνου του πελάτη και πραγματοποιεί τις αντίστοιχες βελτιώσεις για τη βελτίωση της συνολικής εμπειρίας του χρήστη. Εκτός από την πραγματοποίηση βελτιώσεων, που συναντώνται περισσότερο στις πλατφόρμες ηλεκτρονικού εμπορίου και στους ιστότοπους κοινωνικής δικτύωσης, τα μεγάλα δεδομένα που συλλέγονται από το τεράστιο δίκτυο IoT παρέχουν επίσης εξατομικευμένο περιεχόμενο, διαφημίσεις και συστάσεις σε πραγματικό χρόνο με βάση το ιστορικό περιήγησης του χρήστη.

Αντιλαμβανόμενοι τη σημασία του να συμβαδίζουν με τις τάσεις, οι υπάρχοντες οργανισμοί χρησιμοποιούν την υπεροχή του διαδικτύου των πραγμάτων και τους συνδυασμούς μεγάλων δεδομένων για να προβλέψουν και να πρωτοπορήσουν στις τάσεις για να προσελκύσουν

χρήστες. Συγκεκριμένα, οι αναλύσεις μεγάλων δεδομένων χρησιμοποιούν αλγόριθμους μηχανικής μάθησης για την πρόβλεψη μελλοντικών τάσεων με βάση ιστορικά δεδομένα που τροφοδοτούνται σε μεγάλο βαθμό από τα ίδια δεδομένα σε πολλαπλές συνδεδεμένες συσκευές. Αναλύοντας τις τάσεις και τα μοτίβα του παρελθόντος, οι επιχειρήσεις μπορούν να κάνουν συγκεκριμένες προβλέψεις για τις επόμενες τάσεις. Τα δεδομένα σε πραγματικό χρόνο από τα δίκτυα IoT είναι επίσης πολύτιμες πληροφορίες που βοηθούν τις επιχειρήσεις να ανταποκρίνονται γρήγορα στις αναδυόμενες τάσεις. Για παράδειγμα, οι αισθητήρες μπορούν να χρησιμοποιηθούν για την παρακολούθηση των περιβαλλοντικών συνθηκών. Σε περίπτωση δυσμενούς συμβάντος, όπως υψηλή υγρασία ή ακόμη και ο κίνδυνος καταιγίδας, τα δεδομένα IoT βοηθούν στην αποστολή σήματος στις συσκευές των πελατών και παρέχουν έγκαιρη προειδοποίηση για πιθανά προβλήματα (Gubbi et al., 2013).

Εκτός από τα πολυάριθμα οφέλη, εξακολουθούν να υπάρχουν με μια δέσμη προκλήσεων στο IoT με ανάλυση μεγάλων δεδομένων. Τα δεδομένα που συλλέγονται από τις συνδεδεμένες συσκευές θεωρούνται γεγονότα σχετικά με τα προϊόντα και τους χρήστες. Ωστόσο, με βάση την πολυπλοκότητά τους, τα μεγάλα δεδομένα μπορεί να είναι αναξιόπιστα και να μην αποφέρουν καν αξία στην επιχείρηση εάν αναλυθούν εσφαλμένα. Ως εκ τούτου, οι επιστήμονες δεδομένων πρέπει να δίνουν ιδιαίτερη προσοχή στην ταξινόμηση και την εξαγωγή δεδομένων σφαλμάτων, διπλών πληροφοριών και μιας σειράς άλλων ανωμαλιών, ώστε να διασφαλίζεται η εύρεση πραγματικών γνώσεων και η απρόσκοπτη παρακολούθηση των δεδομένων.

1.3 Σκοπός και Σημασία της Μελέτης

Η παρούσα πτυχιακή εργασία στοχεύει στην ανάπτυξη μιας πλατφόρμας ανάλυσης δεδομένων που συλλέγονται από συσκευές στο διαδίκτυο των πραγμάτων χρησιμοποιώντας τεχνικές μεγάλων δεδομένων και τη γλώσσα προγραμματισμού Python. Η πλατφόρμα αυτή θα επιτρέπει επεξεργασία και ανάλυση των δεδομένων σε πραγματικό χρόνο, με σκοπό την εξαγωγή χρήσιμων πληροφοριών. Ο στόχος είναι η δημιουργία ενός μηχανισμού προσομοίωσης συλλογής δεδομένων από διάφορες IoT συσκευές. Ο επόμενος στόχος είναι η εφαρμογή τεχνικών ανάλυσης δεδομένων με τη χρήση της Python για την εξαγωγή χρήσιμων πληροφοριών. Η πλατφόρμα θα παρέχει δυνατότητες οπτικοποίησης δεδομένων με εργαλεία δημιουργώντας γραφήματα και διαδραστικά dashboards που απεικονίζουν τα αποτελέσματα της ανάλυσης σε πραγματικό χρόνο.

Η επίδραση της παρούσας μελέτης, η οποία επικεντρώνεται στην ανάλυση δεδομένων από συσκευές Internet of Things (IoT) χρησιμοποιώντας τεχνικές μεγάλων δεδομένων και τη γλώσσα προγραμματισμού Python, αναμένεται να είναι σημαντική και πολυδιάστατη. Πρώτον, η ανάπτυξη μιας πλατφόρμας ανάλυσης δεδομένων IoT μπορεί να προσφέρει μεγάλες δυνατότητες σε διάφορους τομείς, όπως η βιομηχανία, η υγεία, οι μεταφορές και η ενέργεια.

Δεύτερον, η χρήση τεχνικών μεγάλων δεδομένων επιτρέπει την επεξεργασία και ανάλυση τεράστιων όγκων δεδομένων σε πραγματικό χρόνο. Αυτό παρέχει τη δυνατότητα εξαγωγής σημαντικών πληροφοριών και προγνωστικών μοντέλων που μπορούν να βελτιώσουν τη λήψη αποφάσεων και να προωθήσουν την καινοτομία. Τρίτον, η χρήση της γλώσσας προγραμματισμού Python προσφέρει σημαντικά πλεονεκτήματα λόγω της ευελιξίας και της ισχυρής κοινότητας υποστήριξης.

1.5 Ερευνητικά Ερωτήματα

Ποιες είναι οι βέλτιστες πρακτικές για τη συλλογή και αποθήκευση μεγάλων ποσοτήτων δεδομένων από συσκευές IoT;

Ποιες τεχνικές επεξεργασίας και ανάλυσης δεδομένων είναι κατάλληλες για την ανάλυση δεδομένων IoT;

Πώς μπορεί να χρησιμοποιηθεί η Python για την υλοποίηση ενός αποτελεσματικού και αποδοτικού συστήματος ανάλυσης μεγάλων δεδομένων από IoT συσκευές;

Ποιες είναι δυνατότητες οπτικοποίησης δεδομένων με εργαλεία δημιουργώντας γραφήματα και διαδραστικά dashboards που απεικονίζουν τα αποτελέσματα της ανάλυσης σε πραγματικό χρόνο;

2. Ανασκόπηση Βιβλιογραφίας

2.1 Διαδίκτυο των Πραγμάτων (IoT)

2.1.1 Ιστορία και Εξέλιξη του IoT

Το διαδίκτυο των πραγμάτων βρίσκεται στο επίκεντρο του ψηφιακού κόσμου και αλλάζει ριζικά τον τρόπο ζωής και εργασίας. Αναφέρεται στη δικτύωση φυσικών συσκευών που συλλέγουν, αναλύουν και αξιολογούν δεδομένα από τις δικτυωμένες οικιακές συσκευές μέχρι τα βιομηχανικά μηχανήματα. Το IoT επιτρέπει την αυτοματοποίηση και την αύξηση της αποδοτικότητας που ξεπερνά κατά πολύ τις συμβατικές τεχνολογίες, με την ανάπτυξη του να μην έχει θέσει μόνο τεχνολογικά ορόσημα, αλλά να έχει επίσης δημιουργήσει νέες ευκαιρίες για τις επιχειρήσεις και τους καταναλωτές. Το διαδίκτυο των πραγμάτων αναφέρεται στη δικτύωση φυσικών αντικειμένων που μπορούν να επικοινωνούν μεταξύ τους και με το διαδίκτυο. Οι συσκευές IoT δικτυώνονται είτε τοπικά είτε μέσω του διαδικτύου γεγονός που τους επιτρέπει την άμεση επικοινωνία μεταξύ τους, επιτρέποντάς τους να εκτελούν εργασίες ανεξάρτητα και να μαθαίνουν από την εμπειρία (Kumar et al., 2017).

Το διαδίκτυο, το οποίο αποτελεί από μόνο του σημαντικό στοιχείο του IoT, ξεκίνησε ως μέρος της DARPA (Defense Advanced Research Projects Agency) το 1962 και εξελίχθηκε σε ARPANET (Advanced Research Projects Agency Network) το 1969. Τη δεκαετία του 1980, οι εμπορικοί πάροχοι υπηρεσιών άρχισαν να υποστηρίζουν τη δημόσια χρήση του ARPANET,

επιτρέποντάς του να εξελιχθεί στο σύγχρονο διαδίκτυο. Οι δορυφόροι και οι σταθερές γραμμές παρέχουν βασικές επικοινωνίες για μεγάλο μέρος του IoT. Οι παγκόσμιοι δορυφόροι εντοπισμού θέσης (GPS) έγιναν πραγματικότητα στις αρχές του 1993. Αυτό ακολουθήθηκε γρήγορα από την τοποθέτηση ιδιωτικών, εμπορικών δορυφόρων σε τροχιά, καθιστώντας το IIoT πολύ πιο λειτουργικό (Salukhe, 2023).

Το Διαδίκτυο των Πραγμάτων, ως έννοια, δεν ονομάστηκε επίσημα μέχρι το 1999, αλλά ένα από τα πρώτα παραδείγματα IoT είναι από τις αρχές της δεκαετίας του 1980 και ήταν ένα μηχάνημα της Coca Cola, που βρισκόταν στο Πανεπιστήμιο Carnegie Mellon. Ο Κέβιν Άστον, εκτελεστικός διευθυντής των Auto-ID Labs του MIT, επινόησε τη φράση «Internet of Things» το 1999. Ήταν ο πρώτος που περιέγραψε το IoT, κάνοντας μια παρουσίαση για την Procter & Gamble, αλλά ο ορισμός του IoT έχει εξελιχθεί με την πάροδο του χρόνου. Ο κ. Ashton δήλωσε: «Το IoT έχει αλλάξει από την αρχή μέχρι σήμερα. Σήμερα οι υπολογιστές και ως εκ τούτου, το διαδίκτυο, εξαρτώνται σχεδόν εξ ολοκλήρου από τον άνθρωπο για πληροφορίες». Σχεδόν όλα τα περίπου 50 petabytes δεδομένων που είναι διαθέσιμα στο διαδίκτυο καταγράφηκαν και δημιουργήθηκαν για πρώτη φορά από ανθρώπους με την πληκτρολόγηση, το πάτημα ενός κουμπιού εγγραφής, τη λήψη μιας ψηφιακής φωτογραφίας ή τη σάρωση ενός barcode. Ο Kevin Ashton πίστευε ότι η αναγνώριση ραδιοσυχνοτήτων αποτελούσε προϋπόθεση για το διαδίκτυο των πραγμάτων κυρίως ως λύση εντοπισμού αποθεμάτων. Εκ των υστέρων, η παρακολούθηση αποθεμάτων έχει γίνει ένα από τα πιο προφανή πλεονεκτήματα του IoT. Κατέληξε στο συμπέρασμα ότι αν όλες οι συσκευές είχαν «ετικέτα», οι υπολογιστές θα μπορούσαν να τις διαχειρίζονται, να τις παρακολουθούν και να τις απογράφουν. Σε κάποιο βαθμό, η σήμανση των πραγμάτων έχει επιτευχθεί μέσω τεχνολογιών όπως η ψηφιακή υδατογράφηση, οι γραμμωτοί κώδικες και οι κωδικοί QR.

Ένα πρόσθετο και σημαντικό στοιχείο για την ανάπτυξη ενός λειτουργικού IoT έλαβε χώρα τον Ιούνιο του 2012, όταν οι μεγάλοι πάροχοι υπηρεσιών διαδικτύου και οι εταιρείες διαδικτύου συμφώνησαν να αυξήσουν τον χώρο διευθύνσεων στο παγκόσμιο διαδίκτυο, ενεργοποιώντας το IPV6 για τις υπηρεσίες και τα προϊόντα τους. Οι «έξυπνες πόλεις» μπορούν να χρησιμοποιήσουν το IoT για να μειώσουν τα απόβλητα και να μεγιστοποιήσουν την αποδοτική χρήση της ενέργειας και μπορεί επίσης να χρησιμοποιηθεί για τον εξορθολογισμό των κυκλοφοριακών ροών και τον εντοπισμό διαθέσιμων χώρων στάθμευσης. Μια καλά σχεδιασμένη έξυπνη πόλη υποστηρίζει όλα τα είδη αισθητήρων που είναι συνδεδεμένοι στο διαδίκτυο και παρέχει παρακολούθηση της κυκλοφορίας, παρακολούθηση και αναφορά της κυκλοφορίας σε πραγματικό χρόνο, παρακολούθηση της ποιότητας του αέρα- Ενσωματωμένοι αισθητήρες IoT μπορούν να εντοπίσουν τους ρυπαντές, έξυπνες μεταφορές, τα έξυπνα φανάρια βελτιώνουν την αποτελεσματικότητα της κυκλοφορίας και των δημόσιων μεταφορών, έξυπνη στάθμευση, αισθητήρες εγκατεστημένοι στο πεζοδρόμιο κ.λπ. για τον προσδιορισμό της πληρότητας του χώρου στάθμευσης, η οποία κοινοποιείται στους οδηγούς, έξυπνος

δημόσιος φωτισμός, φωτισμός χαμηλής κατανάλωσης ενέργειας σε συνδυασμό με χρονομέτρηση και αισθητήρες. έξυπνα κτίρια, όταν συνδέονται με την έξυπνη πόλη μέσω του διαδικτύου, γίνονται μέρος της υποδομής της πόλης. Ένα έξυπνο κτίριο, από μόνο του, χρησιμοποιεί αισθητήρες και αυτοματοποιημένες διαδικασίες για τον έλεγχο των λειτουργιών του κτιρίου, οι οποίες περιλαμβάνουν κλιματισμό, θέρμανση, εξαερισμό, ασφάλεια, φωτισμό και άλλα συστήματα. Τα έξυπνα κτίρια είναι ολοκληρωμένα συστήματα και μοιράζονται ζωτικής σημασίας πληροφορίες (Sanja et al., 2020).

Το Βιομηχανικό Διαδίκτυο των Πραγμάτων (IIoT) αποτελεί επέκταση του IoT και χρησιμοποιεί ενεργοποιητές και έξυπνους αισθητήρες, οι οποίοι είναι δικτυωμένοι μαζί με τις βιομηχανικές εφαρμογές μιας εταιρείας. Στόχος είναι να προσδώσει στις βιομηχανίες μεγαλύτερη αποτελεσματικότητα και αξιοπιστία. Το IIoT περιλαμβάνει τη ρομποτική και τις διαδικασίες παραγωγής που καθορίζονται από λογισμικό. Η τεράστια αποθηκευτική ικανότητα του cloud (2002) ήταν απαραίτητη για να γίνει πραγματικότητα η σύγχρονη εκδοχή του IIoT. Το IIoT δημιουργήθηκε περίπου το 2010, με αρκετές μεγάλες εταιρείες να αναπτύσσουν τα δικά τους συστήματα. Στην GE αποδίδονται τα εύσημα για τη δημιουργία του όρου «Βιομηχανικό Διαδίκτυο των Πραγμάτων», το 2012. Μέχρι το έτος 2013, το IoT είχε γίνει ένα σύστημα που χρησιμοποιούσε πολλαπλές τεχνολογίες, από το Διαδίκτυο μέχρι την ασύρματη επικοινωνία και από τα μικρο-ηλεκτρομηχανικά συστήματα (MEMS) μέχρι τα ενσωματωμένα συστήματα. Αυτό περιλαμβάνει σχεδόν οτιδήποτε μπορείτε να φανταστείτε, από κινητά τηλέφωνα μέχρι τη συντήρηση κτιρίων και τον κινητήρα ενός αεροπλάνου. Οι ιατρικές συσκευές, όπως ένα εμφύτευμα καρδιακής παρακολούθησης μπορούν να μεταφέρουν δεδομένα μέσω δικτύου και αποτελούν μέλη του IoT (Mahantesh et al, 2017).

Τα smartphones αποτελούν μέρος του IoT και έχουν γίνει σημαντικό εργαλείο επικοινωνίας για πολλά άτομα. Το 2015, εντάχθηκαν στο IoT με μεγάλο βαθμό ενθουσιασμού από τους εμπόρους. Οι αισθητήρες μέσα σε αυτές τις συσκευές παρακολουθούνται από τα τμήματα μάρκετινγκ, τα οποία στέλνουν συγκεκριμένες προωθητικές ενέργειες με βάση τον πελάτη και την τοποθεσία του προϊόντος. Ο κλάδος της υγειονομικής περίθαλψης έχει επίσης επωφεληθεί από αυτή την τάση. Συσκευές, όπως τα έξυπνα ρολόγια, τα smartphones και οι οθόνες μπορούν να παρακολουθούν τα δεδομένα ενός ασθενούς σχετικά με την αρτηριακή πίεση, τον καρδιακό ρυθμό και άλλες ανησυχίες σε πραγματικό χρόνο. Τα αυτοκίνητα και τα φορτηγά έχουν γίνει μέλη του IoT. Ένα συνδεδεμένο όχημα συνεργάζεται με άλλες συσκευές μέσω ασύρματων δικτύων. Αυτή η τεχνολογία επιτρέπει σε διάφορα «συνδεδεμένα δίκτυα» να έχουν πρόσβαση και να επικοινωνούν με τα οχήματα. Τα αυτοκίνητα και τα φορτηγά είναι ήδη φορτωμένα με αισθητήρες και τεχνολογία, συμπεριλαμβανομένων των OBD (on-board diagnostics) και του GPS. Με τη μεγιστοποίηση της χρήσης αυτών των τεχνολογιών, οι επιχειρήσεις μπορούν να αντλούν πληροφορίες από τους στόλους τους σχετικά με τις απαιτήσεις συντήρησης, τις συνθήκες οδήγησης και τις διαδρομές σε πραγματικό χρόνο. Τα

αυτοοδηγούμενα αυτοκίνητα χρησιμοποιούν το cloud για να ανταποκρίνονται στα γειτονικά αυτοκίνητα, στα δεδομένα κυκλοφορίας, στους χάρτες, στον καιρό, στις συνθήκες του οδοστρώματος. Η χρήση του νέφους βοηθά τα οχήματα να παρακολουθούν το περιβάλλον τους και να λαμβάνουν καλύτερες αποφάσεις. Τα αυτοοδηγούμενα αυτοκίνητα είναι νέα μέλη του IOT. Το πρώτο πραγματικά αυτοκινούμενο όχημα εμφανίστηκε τη δεκαετία του 1980. Τον Οκτώβριο του 2021, η May Mobility ξεκίνησε ένα πιλοτικό πρόγραμμα για να δοκιμάσει το αυτοοδηγούμενο λογισμικό της (Al-Joboury et al., 2017).

2.1.2 Τεχνολογίες και Πρωτόκολλα IoT

Το διαδίκτυο των πραγμάτων συνδέει φυσικές συσκευές και επιτρέπει την ανταλλαγή δεδομένων μέσω του διαδικτύου, ενώ τα πρωτόκολλα IoT είναι υπεύθυνα για την επικοινωνία και τη μεταφορά δεδομένων μεταξύ των συσκευών. Τα πρωτόκολλα IoT είναι ένα σύνολο κανόνων και προτύπων που διέπουν την επικοινωνία μεταξύ συσκευών στο διαδίκτυο των πραγμάτων (IoT). Ο κύριος σκοπός αυτών των τεχνολογιών είναι να διασφαλίζουν την ομαλή επικοινωνία μεταξύ διαφορετικών συσκευών, ανεξαρτήτως κατασκευαστή ή πλατφόρμας. Αποτελούν το θεμέλιο της μετάδοσης δεδομένων και είναι ζωτικής σημασίας για την επιτυχή υλοποίηση έργων IoT. Χωρίς αυτά τα κατάλληλα πρωτόκολλα, οι εφαρμογές IoT δεν θα μπορούσαν να επικοινωνήσουν σωστά μεταξύ τους (Al-Fuqaha et al., 2015).

Ενώ τα παραδοσιακά πρωτόκολλα, όπως το HTTP και το FTP, σχεδιάστηκαν για εφαρμογές διαδικτύου, τα πρότυπα IoT είναι ειδικά προσαρμοσμένα στις ανάγκες των συσκευών IoT. Αυτές περιλαμβάνουν την εμβέλεια, τον όγκο δεδομένων και την ενεργειακή απόδοση. Στην πράξη, πολλές εφαρμογές IoT χρησιμοποιούν συμβατικές τεχνολογίες επειδή η απόδοσή τους είναι επαρκής ή επειδή νέα πρωτόκολλα εξακολουθούν να αναπτύσσονται. Μια άλλη διαφορά είναι ο τύπος της επικοινωνίας. Τα παραδοσιακά πρότυπα χρησιμοποιούν κυρίως πρωτόκολλα προσανατολισμένα στη σύνδεση, όπως το TCP, ωστόσο οι τεχνολογίες IoT υποστηρίζουν τόσο την επικοινωνία προσανατολισμένη στη σύνδεση όσο και την επικοινωνία χωρίς σύνδεση, επιτρέποντας την ευέλικτη και προσαρμόσιμη μεταφορά δεδομένων σε διάφορες εφαρμογές IoT. Τα πρωτόκολλα προσανατολισμένα στη σύνδεση δημιουργούν πρώτα μια σύνδεση από άκρο σε άκρο μεταξύ των συσκευών και στη συνέχεια αποστέλλουν τα δεδομένα, ώστε να διασφαλίζεται ότι κάθε συσκευή μπορεί να στείλει δεδομένα χωρίς σφάλματα και με τη σωστή σειρά. Από την άλλη πλευρά, η μεταφορά δεδομένων χωρίς σύνδεση επικοινωνεί χωρίς αυτή την προηγούμενη εγκαθίδρυση της σύνδεσης από άκρο σε άκρο. Αυτά τα πρωτόκολλα χρησιμοποιούνται όταν η γρήγορη μετάδοση είναι σημαντική και η περιστασιακή απώλεια δεδομένων είναι αποδεκτή. Η επικοινωνία του IoT βασίζεται σε μια πολυεπίπεδη αρχιτεκτονική που οργανώνει αποτελεσματικά την ανταλλαγή δεδομένων σε διαφορετικά πρωτόκολλα. Η ανταλλαγή δεδομένων μεταξύ των συσκευών περνά από διάφορα επίπεδα, από την αποθήκευση, την επεξεργασία και τη διεπαφή χρήστη. Κάθε στρώμα είναι

υπεύθυνο για μια συγκεκριμένη εργασία και επικοινωνεί με τα στρώματα που βρίσκονται κάτω και πάνω από αυτό (Sethi et al., 2017).

Πρωτόκολλα στρώματος εφαρμογής IoT

Το επίπεδο εφαρμογής είναι το ανώτερο επίπεδο του μοντέλου OSI. Σε αυτό το επίπεδο, τα πρωτόκολλα εκτελούν τις πιο απαιτητικές σε δεδομένα διεργασίες που είναι υπεύθυνες για την ανταλλαγή μηνυμάτων μεταξύ τελικών συσκευών και εφαρμογών λογισμικού.

AMQP (προηγμένο πρωτόκολλο ουράς μηνυμάτων)

Το AMQP είναι ένα ανοικτό πρότυπο πρωτόκολλο που χρησιμοποιείται για την ανταλλαγή μηνυμάτων και τη διαχείριση ουρών αναμονής. Παρέχει αξιόπιστη και ομαλή παράδοση μηνυμάτων, υποστηρίζει ποικίλα πρότυπα επικοινωνίας και προσφέρει υψηλό βαθμό διαλειτουργικότητας. Το AMQP εφαρμόζεται ευρέως στον χρηματοπιστωτικό τομέα και στην εταιρική επικοινωνία.

HTTP (Πρωτόκολλο μεταφοράς υπερκειμένου)

Το HTTP είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο για τη μεταφορά ιστοσελίδων και δεδομένων μέσω του Διαδικτύου. Αυτή η τεχνολογία είναι εύκολη στην εφαρμογή και αξιοποιεί την υπάρχουσα υποδομή του Web. Το HTTP αναπτύσσεται συχνά σε έξυπνα οικιακά συστήματα που απαιτούν ενσωμάτωση με υπηρεσίες ιστού.

WebSocket

Το πρωτόκολλο WebSocket παρέχει χαμηλή καθυστέρηση και αποτελεσματική μεταφορά δεδομένων. Είναι ιδανικό για εφαρμογές πραγματικού χρόνου, όπως διαδικτυακά παιχνίδια, πλατφόρμες χρηματοοικονομικών συναλλαγών και εφαρμογές συνομιλίας.

LwM2M (Lightweight Machine-to-Machine)

Το πρωτόκολλο LwM2M έχει σχεδιαστεί ειδικά για τη διαχείριση συσκευών IoT. Είναι αποδοτικό ως προς τους πόρους και επιτρέπει την αποτελεσματική απομακρυσμένη διαμόρφωση και παρακολούθηση εφαρμογών ενέργειας και κοινής ωφέλειας.

XMPP (Πρωτόκολλο εκτεταμένης ανταλλαγής μηνυμάτων και παρουσίας)

Το πρωτόκολλο εκτεταμένης ανταλλαγής μηνυμάτων και παρουσίας - Extensible Messaging and Presence Protocol είναι μια τεχνολογία διαχείρισης μηνυμάτων και παρουσίας. Η διαχείριση παρουσίας επιτρέπει στους εταίρους επικοινωνίας να δηλώνουν τη διαθεσιμότητά τους σε έναν αγγελιοφόρο. Το XMPP είναι ευέλικτο, ασφαλές, ευρέως διαδεδομένο και επεκτάσιμο. Το πρωτόκολλο χρησιμοποιείται συνήθως για εφαρμογές συνομιλίας και κοινωνικά δίκτυα, καθώς και για την επικοινωνία συσκευών σε δίκτυα IoT.

SMS/SMPP (Short Message Service/Short Message Peer-to-Peer)

Το πρωτόκολλο SMS είναι γνωστό για την αποστολή σύντομων μηνυμάτων μέσω κυψελοειδών δικτύων. Το SMPP είναι το πρότυπο για την ανταλλαγή μηνυμάτων SMS μεταξύ κέντρων Short Message Service και εξωτερικών διακομιστών μηνυμάτων. Και οι δύο τεχνολογίες είναι αξιόπιστες και ευρέως διαδεδομένες, καθώς επιτρέπουν την επικοινωνία

μεταξύ συσκευών IoT μέσω κυψελοειδών δικτύων. Οι τεχνολογίες χρησιμοποιούνται επίσης ευρέως στην τηλεματική και τη διαχείριση στόλου οχημάτων.

USSD (μη δομημένα συμπληρωματικά δεδομένα υπηρεσιών)

Το πρότυπο USSD είναι ένα πρωτόκολλο επικοινωνίας μεταξύ κινητών συσκευών και διακομιστών εφαρμογών των φορέων εκμετάλλευσης κινητών δικτύων. Είναι γρήγορο, προωθεί την ασφαλή επικοινωνία σε πραγματικό χρόνο, δεν απαιτεί μόνιμη σύνδεση στο Διαδίκτυο και δεν αποθηκεύει μηνύματα. Το πρωτόκολλο USSD χρησιμοποιείται επίσης για πληρωμές μέσω κινητής τηλεφωνίας.

SSI (Simple Sensor Interface)

Η τεχνολογία SSI είναι ένα απλό πρωτόκολλο επικοινωνίας για τη μεταφορά δεδομένων σε πραγματικό χρόνο μεταξύ αισθητήρων και υπολογιστών. Το SSI είναι εύκολο στην υλοποίηση και υποστηρίζει την άμεση σύνδεση αισθητήρων και συστημάτων ελέγχου. Αυτό το πρωτόκολλο IoT χρησιμοποιείται συχνά σε βιομηχανικές εφαρμογές που απαιτούν αποτελεσματική μετάδοση δεδομένων αισθητήρων.

CoAP (Πρωτόκολλο περιορισμένης εφαρμογής)

Το CoAP έχει σχεδιαστεί ειδικά για τοπικά περιορισμένες συσκευές και δίκτυα. Είναι ελαφρύ, αποδοτικό, αξιόπιστο και μπορεί να ενσωματωθεί σε υπάρχουσες υπηρεσίες ιστού. Το πρωτόκολλο IoT είναι ιδανικό για συστήματα έξυπνων πόλεων και έξυπνων σπιτιών, τα οποία συνήθως περιορίζονται σε μια συγκεκριμένη περιοχή και απαιτούν τεχνολογία αποδοτικής χρήσης πόρων.

DDS (Υπηρεσία διανομής δεδομένων)

Η τεχνολογία DDS είναι ένα πρωτόκολλο για αξιόπιστη ανταλλαγή δεδομένων σε πραγματικό χρόνο μεταξύ συσκευών IoT. Αυτό το πρότυπο χρησιμοποιείται συχνά για κρίσιμες για την ασφάλεια εφαρμογές στην αεροδιαστημική, την άμυνα και τον βιομηχανικό αυτοματισμό.

MQTT (Message Queuing Telemetry Transport)

Το MQTT είναι ένα αξιόπιστο πρωτόκολλο παράδοσης μηνυμάτων. Είναι εύκολο στην υλοποίηση και κατάλληλο για συσκευές IoT με περιορισμένο εύρος ζώνης και ισχύ. Η τεχνολογία MQTT χρησιμοποιείται ευρέως στον οικιακό αυτοματισμό, την παρακολούθηση της υγείας και τη βιομηχανική παρακολούθηση.

Πρωτόκολλα επιπέδου μεταφοράς για τεχνολογίες IoT

Το επίπεδο μεταφοράς είναι το τέταρτο επίπεδο στο μοντέλο OSI και εξασφαλίζει την αξιόπιστη μετάδοση δεδομένων μεταξύ τελικών συστημάτων. Τα πρωτόκολλα σε αυτό το επίπεδο ελέγχουν τη ροή των δεδομένων και διασφαλίζουν ότι τα πακέτα δεδομένων φθάνουν σωστά και με τη σωστή σειρά.

TCP (Πρωτόκολλο ελέγχου μετάδοσης)

Το πρότυπο TCP είναι ένα πρωτόκολλο προσανατολισμένο στη σύνδεση. Χρησιμοποιεί μηνύματα επιβεβαίωσης για να εγγυηθεί ότι τα πακέτα δεδομένων φθάνουν με τη σωστή σειρά

και χωρίς απώλειες. Το πρωτόκολλο χρησιμοποιείται κυρίως στους βιομηχανικούς αυτοματισμούς, στην υγειονομική περίθαλψη και σε κρίσιμες για την ασφάλεια εφαρμογές. Η ακεραιότητα των δεδομένων, η οποία αναφέρεται στην ορθότητα, την πληρότητα και την ασφάλεια των δεδομένων, είναι κρίσιμη σε αυτούς τους τομείς του IoT.

UDP (Πρωτόκολλο δεδομένων χρήστη)

Το UDP είναι ένα πρωτόκολλο χωρίς σύνδεση, το οποίο στέλνει πακέτα δεδομένων απευθείας στον παραλήπτη χωρίς να έχει προηγηθεί η δημιουργία σύνδεσης. Η τεχνολογία αυτή παρέχει χαμηλή καθυστέρηση και γρήγορη μετάδοση δεδομένων, ενώ χρησιμοποιείται επίσης για μη κρίσιμες εφαρμογές, όπως η ροή βίντεο και η φωνή μέσω πρωτοκόλλου διαδικτύου (VoIP), όπου η περιστασιακή διακοπή λειτουργίας δεν επηρεάζει απαραίτητα την ακεραιότητα των δεδομένων.

Πρότυπα επιπέδου δικτύου IoT

Το επίπεδο δικτύου είναι το τρίτο επίπεδο του μοντέλου OSI. Είναι υπεύθυνο για την προώθηση πακέτων δεδομένων μεταξύ διαφορετικών δικτύων. Τα κατάλληλα πρωτόκολλα αποφασίζουν ποια διαδρομή θα ακολουθήσουν τα πακέτα δεδομένων για να φτάσουν αποτελεσματικά στον προορισμό τους.

IP (Πρωτόκολλο Διαδικτύου)

Το IP είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο για τη διευθυνσιοδότηση και τη δρομολόγηση πακέτων δεδομένων στο Διαδίκτυο. Είναι ευέλικτο και συμβατό με σχεδόν κάθε τύπο δικτύου και συσκευής. Το πρότυπο IP εφαρμόζεται σχεδόν σε όλες τις εφαρμογές IoT, συμπεριλαμβανομένων των έξυπνων σπιτιών, των wearables και του βιομηχανικού αυτοματισμού.

6LoWPAN (IPv6 μέσω ασύρματων προσωπικών δικτύων χαμηλής ισχύος)

Το πρωτόκολλο 6LoWPAN προωθεί την αποδοτική χρήση του εύρους ζώνης και της ισχύος. Είναι ειδικά σχεδιασμένο για ασύρματα δίκτυα χαμηλής ισχύος και επιτρέπει την απρόσκοπτη ενσωμάτωση μικρών συσκευών IoT στο Διαδίκτυο με ελάχιστη κατανάλωση ενέργειας. Η τεχνολογία υιοθετείται ευρέως για έξυπνες πόλεις, οικιακούς αυτοματισμούς και βιομηχανικά δίκτυα αισθητήρων, όπου η ενεργειακή απόδοση και το χαμηλό κόστος ιδιοκτησίας είναι κρίσιμα.

Πρωτόκολλα IoT στο επίπεδο ζεύξης δεδομένων

Το επίπεδο σύνδεσης δεδομένων είναι το δεύτερο επίπεδο του μοντέλου OSI. Οι τεχνολογίες σε αυτό το επίπεδο διατηρούν την χωρίς σφάλματα μετάδοση δεδομένων μεταξύ δύο άμεσα συνδεδεμένων συσκευών.

Πρότυπα IEEE

Το Ινστιτούτο Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE) αναπτύσσει και διατηρεί διάφορα πρότυπα για τη μετάδοση δεδομένων. Είναι ισχυρά, συμβατά με πολλούς τύπους δικτύων και διαλειτουργικά μεταξύ συσκευών και προμηθευτών. Ένα τέτοιο πρωτόκολλο είναι

το IEEE 802.15.4, το βασικό πρότυπο για το ZigBee, ένα ενεργειακά αποδοτικό πρωτόκολλο IoT για συσκευές χαμηλής ισχύος, και άλλα ασύρματα δίκτυα. Το IEEE 802.15.4 χρησιμοποιείται κυρίως στον οικιακό αυτοματισμό, τη βιομηχανική παρακολούθηση και τα δίκτυα αισθητήρων.

LPWAN (Low Power Wide Area Network)

Η τεχνολογία LPWAN περιλαμβάνει διάφορα πρωτόκολλα όπως το LoRaWAN και το NB-IoT. Το πρότυπο LoRaWAN υποστηρίζει τη μετάδοση δεδομένων σε μεγάλες αποστάσεις, ενώ το NB-IoT είναι ιδανικό για συσκευές χαμηλής ισχύος. Τα LPWAN δεν προσφέρουν μόνο μεγάλη εμβέλεια και ενεργειακή απόδοση, αλλά συνδέουν επίσης συσκευές σε μεγάλες αποστάσεις και παρατείνουν τη διάρκεια ζωής της μπαταρίας τους. Οι εφαρμογές περιλαμβάνουν τις έξυπνες πόλεις, τη γεωργία, την περιβαλλοντική παρακολούθηση και τα logistics.

Πρότυπα φυσικού επιπέδου για επικοινωνίες IoT

Το φυσικό επίπεδο είναι το χαμηλότερο επίπεδο του μοντέλου OSI. Τα πρωτόκολλα σε αυτό το επίπεδο διέπουν τη μετάδοση των bits σε ένα μέσο μετάδοσης, όπως χάλκινο καλώδιο, καλώδιο οπτικών ινών ή ραδιοκύματα.

Bluetooth/BLE (Bluetooth Low Energy)

Το Bluetooth είναι μια ασύρματη τεχνολογία για τη μετάδοση δεδομένων μικρής εμβέλειας. Το BLE είναι η έκδοση χαμηλής κατανάλωσης ενέργειας του Bluetooth και είναι ιδανικό για συσκευές που λειτουργούν με μπαταρία. Και τα δύο πρωτόκολλα είναι φθηνές και απλές επιλογές για τη μετάδοση δεδομένων. Οι φορητές συσκευές, οι έξυπνες οικιακές συσκευές και οι εφαρμογές υγειονομικής περίθαλψης χρησιμοποιούν συχνά Bluetooth και BLE.

Ethernet

Η τεχνολογία Ethernet είναι ένα κοινό πρότυπο για ενσύρματα δίκτυα. Ως πρωτόκολλο IoT, το Ethernet παρέχει στιβαρή και γρήγορη μετάδοση δεδομένων, ιδανική για τη βιομηχανία 4.0 και τα έξυπνα κτίρια όπου απαιτείται υψηλό εύρος ζώνης.

Ασύρματες τεχνολογίες: 2G, GPRS/Edge, 3G, 4G/LTE και 5G

Τα πρότυπα κινητών επικοινωνιών επιτρέπουν την ασύρματη επικοινωνία σε μεγάλες αποστάσεις. Τα LTE και 5G παρέχουν υψηλούς ρυθμούς δεδομένων και χαμηλή καθυστέρηση, ενώ αντίθετα, τα 2G, GPRS και 3G έχουν χαμηλότερες επιδόσεις αλλά χρησιμοποιούνται ευρέως. Οι εφαρμογές IoT περιλαμβάνουν έξυπνες πόλεις, αυτόνομη οδήγηση και κινητές συσκευές IoT.

NFC (Near Field Communication)

Το πρωτόκολλο NFC υποστηρίζει ασύρματη επικοινωνία σε πολύ μικρές αποστάσεις, συνήθως λίγων εκατοστών. Το NFC είναι εύκολο στη χρήση, προσφέρει γρήγορη μετάδοση και δεν απαιτεί ζεύξη συσκευών όπως το Bluetooth. Η τεχνολογία χρησιμοποιείται σε συστήματα ανέπαφων πληρωμών και για την ανταλλαγή πληροφοριών μεταξύ συσκευών.

PLC (Powerline Communication)

Το πρότυπο PLC χρησιμοποιεί τις υπάρχουσες γραμμές ρεύματος για τη μετάδοση δεδομένων, γεγονός που παρέχει μια οικονομικά αποδοτική λύση δικτύωσης χωρίς την ανάγκη πρόσθετης καλωδίωσης. Το PLC είναι ιδανικό για τη μετάδοση δεδομένων εντός κτιρίου και χρησιμοποιείται συχνά για εφαρμογές έξυπνου σπιτιού και αυτοματισμού κτιρίων.

LoRaWAN (Δίκτυο ευρείας περιοχής μεγάλης εμβέλειας)

Το πρωτόκολλο LPWAN LoRaWAN είναι μια ασύρματη τεχνολογία για επικοινωνίες μεγάλης εμβέλειαςμεταδίδοντας δεδομένα σε μεγάλες αποστάσεις με χαμηλή κατανάλωση ενέργειας. Είναι ιδανικό για συσκευές που λειτουργούν με μπαταρία σε μεγάλες περιοχές, όπως στη γεωργία.

Sigfox

Το πρωτόκολλο ασύρματης δικτύωσης Sigfox είναι βελτιστοποιημένο για χαμηλούς ρυθμούς δεδομένων και μεγάλες αποστάσεις. Το Sigfox είναι ενεργειακά αποδοτικό, οικονομικά αποδοτικό και εύκολο στην εγκατάσταση. Εφαρμογές εντοπισμού περιουσιακών στοιχείων και έξυπνης μέτρησης χρησιμοποιούν την τεχνολογία.

Neocortec

Το πρότυπο Neocortec είναι μια στιβαρή και επεκτάσιμη λύση για τη σύνδεση πολλών συσκευών IoT. Επιτρέπει την ευέλικτη και ενεργειακά αποδοτική επικοινωνία για τη βιομηχανία 4.0 και τα έξυπνα κτίρια.

Weightless

Το πρωτόκολλο IoT Weightless είναι κατάλληλο για ασύρματη επικοινωνία σε μεγάλες αποστάσεις με χαμηλούς ρυθμούς δεδομένων. Είναι για εφαρμογές που έχουν περιστασιακές απαιτήσεις μετάδοσης δεδομένων, όπως η έξυπνη μέτρηση ή η παρακολούθηση περιουσιακών στοιχείων.

RFID (Αναγνώριση ραδιοσυχνοτήτων)

Η τεχνολογία RFID χρησιμοποιεί ραδιοκύματα για τον εντοπισμό και την παρακολούθηση αντικειμένων. Είναι γρήγορη, ανθεκτική, ανέπαφη και μπορεί να εφαρμοστεί σε διάφορα περιβάλλοντα, όπως logistics, διαχείριση αποθεμάτων και συστήματα ελέγχου πρόσβασης.

Wi-Fi

Η ευρέως διαδεδομένη τεχνολογία Wi-Fi δημιουργεί τοπικά περιορισμένα δίκτυα. Είναι εύκολη στην ανάπτυξη, συμβατή με πολλές συσκευές και εφαρμογές και παρέχει γρήγορη μετάδοση δεδομένων. Τα έξυπνα σπίτια, τα κτίρια γραφείων και τα δημόσια hotspots χρησιμοποιούν Wi-Fi.

Z-Wave

Το πρωτόκολλο Z-Wave έχει σχεδιαστεί για τον οικιακό αυτοματισμό. Είναι ενεργειακά αποδοτικό, εύκολο στην εγκατάσταση και υποστηρίζει υψηλό επίπεδο διαλειτουργικότητας μεταξύ συσκευών από διαφορετικούς κατασκευαστές. Το Z-Wave χρησιμοποιείται κυρίως σε εφαρμογές έξυπνου σπιτιού, όπως συστήματα φωτισμού, ασφάλειας και παρακολούθησης.

ZigBee

Η τεχνολογία ZigBee είναι ένα πρωτόκολλο IoT χαμηλής κατανάλωσης ενέργειας. Είναι ιδανικό για συσκευές που λειτουργούν με μπαταρία σε εφαρμογές έξυπνου σπιτιού, βιομηχανικού αυτοματισμού και υγειονομικής περίθαλψης.

Πρωτόκολλα εφαρμογών ειδικά για τον κλάδο για το Διαδίκτυο των πραγμάτων

Σημαντικά εξειδικευμένα πρότυπα για εφαρμογές του IoT περιλαμβάνουν:

LTE-M (Long Term Evolution for Machines)

Σχεδιασμένο για συσκευές IoT, το πρότυπο κινητής τηλεφωνίας LTE-M παρέχει χαμηλή καθυστέρηση και υψηλούς ρυθμούς δεδομένων. Το LTE-M επιτρέπει επίσης αξιόπιστες επικοινωνίες και υποστηρίζει κινητές εφαρμογές IoT, όπως η αυτόνομη οδήγηση. Αυτή η τεχνολογία χρησιμοποιείται στην εφοδιαστική, τη διαχείριση στόλου και την τηλεματική.

NB-IoT (Narrowband IoT)

Ως δίκτυο ευρείας περιοχής χαμηλής ισχύος (LPWAN), το NB-IoT είναι βελτιστοποιημένο για συσκευές χαμηλής ισχύος. Το πρωτόκολλο προσφέρει μεγάλη διάρκεια ζωής της μπαταρίας και υψηλή κάλυψη δικτύου σε κτίρια και απομακρυσμένες περιοχές. Το NB-IoT είναι κατάλληλο για έξυπνη μέτρηση, περιβαλλοντική παρακολούθηση και έξυπνες πόλεις.

MIoTy

Ένα άλλο πρωτόκολλο LPWAN που έχει σχεδιαστεί για ανθεκτικότητα και επεκτασιμότητα, είναι το MIoTy. Μεταδίδει μεγάλες ποσότητες δεδομένων με χαμηλή ισχύ και προσφέρει υψηλή ανοσία στις παρεμβολές και εύκολη ενσωμάτωση σε υπάρχοντα συστήματα. Το πρότυπο πρωτοκόλλου χρησιμοποιείται σε βιομηχανικές εφαρμογές έντασης δεδομένων και μεγάλης κλίμακας, σε έξυπνες πόλεις και στη γεωργία.

5G IoT

Το 5G IoT είναι μια ειδική τεχνολογία δικτύου 5G για εφαρμογές IoT. Προσφέρει εξαιρετικά υψηλούς ρυθμούς δεδομένων, χαμηλή καθυστέρηση και συνδέει ένα ευρύ φάσμα συσκευών. Το πρότυπο υποστηρίζει εφαρμογές πραγματικού χρόνου, όπως η αυτόνομη οδήγηση και ο βιομηχανικός αυτοματισμός.

OCPP (πρωτόκολλο ανοικτού σημείου φόρτισης)

Το Open Charge Point Protocol είναι ένα ανοικτό πρωτόκολλο επικοινωνίας για υποδομές φόρτισης ηλεκτρικών οχημάτων και έξυπνων δικτύων. Η τεχνολογία επιτρέπει τη διαλειτουργικότητα μεταξύ διαφορετικών παρόχων φόρτισης και υποστηρίζει την ενσωμάτωση συστημάτων διαχείρισης ενέργειας.

IEC 62056

Το IEC 62056 είναι το πρότυπο πρωτόκολλο για τη μετάδοση δεδομένων έξυπνων μετρητών. Είναι μια αξιόπιστη και τυποποιημένη μέθοδος για την απομακρυσμένη παρακολούθηση των μετρητών ενέργειας.

OBD2/CAN-BUS

Το OBD2 (On-Board Diagnostics) και το CAN-BUS (Controller Area Network) είναι πρωτόκολλα διάγνωσης και επικοινωνίας οχημάτων. Και οι δύο τεχνολογίες παρέχουν μια τυποποιημένη διεπαφή για τη διάγνωση και την παρακολούθηση των δεδομένων του οχήματος για την ευφυή διαχείριση του στόλου.

OPC UA (ενοποιημένη αρχιτεκτονική επικοινωνιών ανοικτής πλατφόρμας)

Το OPC UA χρησιμοποιείται στη βιομηχανία 4.0 ως τυποποιημένο πρωτόκολλο για την ασφαλή ανταλλαγή δεδομένων μεταξύ μηχανών και συστημάτων. Η τεχνολογία παρέχει διαλειτουργικότητα, επεκτασιμότητα και υψηλό επίπεδο ασφάλειας για βιομηχανικές εφαρμογές.

Ασύρματο M-Bus

Το ασύρματο M-Bus επιτρέπει την αξιόπιστη, ασφαλή και ενεργειακά αποδοτική επικοινωνία δεδομένων μετρήσεων σε μεγάλες αποστάσεις. Το πρωτόκολλο IoT χρησιμοποιείται στην ενέργεια, το νερό και την έξυπνη μέτρηση.

Πρωτόκολλα ασφαλείας IoT: Προστασία συνδεδεμένων συσκευών

Τα σημαντικά πρότυπα ασφαλείας για τις συσκευές IoT περιλαμβάνουν:

IPSec (ασφάλεια πρωτοκόλλου διαδικτύου)

Αυτό το πρωτόκολλο είναι υπεύθυνο για την ασφαλή επικοινωνία μέσω του πρωτοκόλλου Διαδικτύου. Το IPSec παρέχει υψηλό επίπεδο ασφάλειας μέσω κρυπτογράφησης και ελέγχου ταυτότητας για την προστασία των δεδομένων κατά τη μεταφορά και την αποτροπή μη εξουσιοδοτημένης πρόσβασης. Το IPSec χρησιμοποιείται συνήθως σε εικονικά ιδιωτικά δίκτυα (VPN) και εφαρμογές του Διαδικτύου των πραγμάτων (IoT) που απαιτούν ασφαλή μετάδοση δεδομένων μέσω του Διαδικτύου.

OpenVPN

Το λογισμικό ανοικτού κώδικα χρησιμοποιεί SSL/TLS για τη διανομή κλειδιών, υποστηρίζει πολλαπλούς αλγόριθμους κρυπτογράφησης και μπορεί να προστατευτεί από τείχη προστασίας. Το OpenVPN υποστηρίζει ασφαλή απομακρυσμένη συνδεσιμότητα για συσκευές IoT που πρέπει να συνδεθούν μέσω δημόσιων δικτύων.

TLS (Ασφάλεια επιπέδου μεταφοράς)

Το πρωτόκολλο TLS κρυπτογραφεί δεδομένα και παρέχει προστασία ακεραιότητας κατά τη μετάδοση δεδομένων. Είναι μια καλή λύση για την εξασφάλιση των καναλιών επικοινωνίας και την προστασία ευαίσθητων δεδομένων. Το TLS χρησιμοποιείται ευρέως για υπηρεσίες ιστού, ηλεκτρονικό ταχυδρομείο και ανταλλαγή μηνυμάτων.

Επιλογή των σωστών πρωτοκόλλων IoT

Η επιλογή των σωστών πρωτοκόλλων IoT είναι κρίσιμη για την επιτυχία ενός έργου IoT. Οι εταιρείες πρέπει να εξετάσουν διάφορους παράγοντες για να διασφαλίσουν ότι τα πρωτόκολλα είναι κατάλληλα για την εφαρμογή τους. Τα σημαντικά κριτήρια περιλαμβάνουν τις απαιτήσεις εύρους ζώνης, τις απαιτήσεις καθυστέρησης, την εμβέλεια, την ενεργειακή απόδοση και τον

αριθμό των συσκευών που πρόκειται να συνδεθούν. Τα πρωτόκολλα που θα επιλεγούν ενδέχεται να πρέπει να είναι συμβατά με τα υπάρχοντα συστήματα και συσκευές, ώστε να διευκολύνεται η ενσωμάτωση και η λειτουργία τους. Ορισμένα πρωτόκολλα είναι επίσης λιγότερο δαπανηρά στην υλοποίηση και τη λειτουργία από άλλα. Αναλύοντας προσεκτικά αυτούς τους παράγοντες, οι οργανισμοί μπορούν να επιλέξουν τα πρωτόκολλα IoT που ανταποκρίνονται στις συγκεκριμένες ανάγκες τους.

2.2 Μεγάλα Δεδομένα (Big Data)

2.2.1 Ορισμός και Ιδιότητες των Μεγάλων Δεδομένων

Τα μεγάλα δεδομένα είναι ένας όρος που αναφέρεται σε οποιοδήποτε μεγάλο σύνολο δεδομένων, αλλά σε επιχειρηματικό πλαίσιο, συνήθως αναφέρεται σε ένα σύνολο δεδομένων που μια εταιρεία μπορεί να αναλύσει για να αποκαλύψει μοτίβα και τάσεις που θα τη βοηθήσουν να προωθήσει την επιχείρησή της. Αυτό θα μπορούσε να είναι δεδομένα αγορών καταναλωτών, αναλυτικά στοιχεία ιστοτόπων ή οποιοδήποτε άλλο σύνολο δεδομένων που μπορεί να παρέχει χρήσιμες πληροφορίες. Τα τέσσερα V των μεγάλων δεδομένων περιγράφουν τις τέσσερις διαστάσεις των μεγάλων δεδομένων, οι οποίες είναι ο όγκος, η ταχύτητα, η ποικιλία και η αληθοφάνεια. Σε αυτό το άρθρο, περιγράφουμε τα τέσσερα V των μεγάλων δεδομένων και τον τρόπο με τον οποίο οι αναλυτές χρησιμοποιούν τα μεγάλα δεδομένα για να καθοδηγήσουν τις επιχειρηματικές στρατηγικές (Rad et al., 2016).

Προτού προσδιορίσετε τα τέσσερα V των μεγάλων δεδομένων, αξίζει να ορίσετε τι πραγματικά αφορούν τα μεγάλα δεδομένα. Τα μεγάλα δεδομένα δεν είναι ένας νέος όρος. Θεωρείται η συντομογραφία για έναν μεγάλο όγκο δεδομένων, όταν χρησιμοποιείται σε αναλυτικό πλαίσιο, τα μεγάλα δεδομένα συνήθως υπονοούν ότι τα δεδομένα έχουν σκόπιμα συγκεντρωθεί και διατηρηθεί από έναν οργανισμό προκειμένου οι αναλυτές δεδομένων να τα αξιολογήσουν και να τα αξιοποιήσουν. Υπάρχουν τέσσερα βασικά χαρακτηριστικά των μεγάλων δεδομένων που χρησιμοποιούν οι επιστήμονες δεδομένων για να ταξινομήσουν ένα σύνολο δεδομένων και να καθορίσουν την αναλυτική του χρησιμότητα. Αυτά είναι τα εξής:

- όγκος
- ποικιλία
- ταχύτητα
- αξιοπιστία

Εάν ένα σύνολο δεδομένων έχει αυτές τις τέσσερις ιδιότητες, μπορεί να ενταχθεί στην ταξινόμηση των «μεγάλων δεδομένων». Ένα πέμπτο V, η αξία, χρησιμοποιείται επίσης μερικές φορές ως καθοριστικό κριτήριο όταν πρόκειται για την επιχειρηματική χρήση ενός συνόλου δεδομένων.

Ένας από τους βασικούς παράγοντες που διαφοροποιεί τα μεγάλα δεδομένα από ένα τυπικό σύνολο δεδομένων είναι το μέγεθός τους. Ο όγκος των δεδομένων σε αυτού του είδους το

σύνολο δεδομένων απαιτεί συνήθως μη παραδοσιακή αποθήκευση και επεξεργασία. Ενώ μια μικρή εταιρεία μπορεί να διατηρεί σύνολα δεδομένων σχετικά με τις πωλήσεις ή τις αλληλεπιδράσεις με τους πελάτες σε ένα υπολογιστικό φύλλο, για παράδειγμα, χρησιμοποιώντας το Microsoft Excel, ένας χρήστης γενικά δεν μπορεί να αποθηκεύσει ή να επεξεργαστεί μεγάλα δεδομένα σε έναν τυπικό φορητό υπολογιστή χρησιμοποιώντας βασικό λογισμικό. Μπορεί να περιλαμβάνουν πολλά terabytes ή ακόμη και petabytes δεδομένων που απαιτούν πιο εξελιγμένη αποθήκευση και επεξεργασία από ένα τυπικό σύνολο δεδομένων.
Υπάρχουν πολλοί τρόποι για να οριστούν τα δεδομένα. Ένας από τους κύριους τρόπους με τους οποίους οι αναλυτές χωρίζουν τα δεδομένα είναι σε τρεις ταξινομήσεις: αδόμητα, ημιδομημένα και δομημένα. Αυτές ποικίλλουν σε τεράστιο βαθμό ως προς το πόσο εύκολο είναι να πραγματοποιηθεί ανάλυση σε αυτές. Τα δομημένα σύνολα δεδομένων μπορεί να περιλαμβάνουν έναν κατάλογο με τον αριθμό των αγορών που πραγματοποιήθηκαν, ο οποίος υπάρχει σε μορφή συνόλου. Αυτά τα στοιχεία μπορούν να χειριστούν μαθηματικά και να διερευνηθούν για να παρέχουν πληροφορίες. Αντίθετα, ένα μη δομημένο σύνολο δεδομένων θα μπορούσε να είναι υλικό βίντεο από ένα δίκτυο καμερών CCTV. για να αναλύσει μη δομημένα δεδομένα, ένας αναλυτής ή μια εφαρμογή τεχνητής νοημοσύνης τα διαμορφώνει πρώτα. Στην περίπτωση των CCTV, αυτό θα μπορούσε να γίνει μέσω λογισμικού που εκτελεί έναν συγκεκριμένο αλγόριθμο πάνω στα δεδομένα για να ανιχνεύσει τον αριθμό των προσώπων που εμφανίζονται και να αποδώσει ένα μοναδικό αναγνωριστικό σε κάθε πρόσωπο. Αυτό το σύνολο δεδομένων παρέχει στη συνέχεια αριθμητικά δεδομένα από τα οποία ένας αναλυτής μπορεί να αντλήσει σημαντικά μοτίβα (Chebotko et al., 2015).
Παράλληλα με την εξέταση του όγκου των δεδομένων που συνθέτουν το σύνολο δεδομένων, ένα άλλο βασικό προσδιοριστικό χαρακτηριστικό των μεγάλων δεδομένων είναι ο όγκος με τον οποίο λαμβάνονται. Ο όρος μεγάλα δεδομένα συχνά σχετίζεται με ταχέως κινούμενα σύνολα δεδομένων που λαμβάνουν τεράστιες ποσότητες εισερχόμενων δεδομένων. Για παράδειγμα, ένα σύνολο δεδομένων που τηρείται από μια αεροπορική αρχή μπορεί να περιλαμβάνει όλους τους επιβάτες που αναχωρούν μια συγκεκριμένη ημέρα. Δεδομένου του αριθμού των αεροδρομίων και του αριθμού των πτήσεων που αναχωρούν κάθε λεπτό, η τιμή αυτή μεταβάλλεται ταχύτατα. ένα άλλο παράδειγμα ταχέως κινούμενου συνόλου δεδομένων είναι τα δεδομένα από μια ροή κοινωνικής δικτύωσης. Για παράδειγμα, μια εταιρεία μπορεί να παρακολουθεί τον αριθμό των ατόμων που την αναφέρουν στο Facebook. Για μια μεγάλη διεθνή μάρκα, αυτό μπορεί να είναι χιλιάδες αναφορές κάθε λεπτό. Αν και οι πλατφόρμες μέσων κοινωνικής δικτύωσης παρέχουν ορισμένα ενσωματωμένα εργαλεία ανάλυσης, για να αναλύσει μια μάρκα με εμπειρογνωμοσύνη αυτά τα δεδομένα, μπορεί να τα εξάγει στα δικά της συστήματα. Η επικαιρότητα είναι ένα άλλο στοιχείο της ανάλυσης μεγάλων δεδομένων. Η πραγματοποίηση ανάλυσης σε δεδομένα κοινωνικών μέσων δικτύωσης πριν από τρεις μήνες

μπορεί να έχει σχετικά μικρή αξία σε σύγκριση με μια γρήγορη ανάλυση δεδομένων από εκείνη την εβδομάδα (Fink et al., 2017).

Η αξιοπιστία αναφέρεται στο πόσο ακριβή είναι τα δεδομένα και πόσο αληθινή και αντιπροσωπευτική είναι η εικόνα που δίνουν. Είναι ουσιαστικά ένα μέτρο της ποιότητας των δεδομένων. Υπάρχουν πολλά πράγματα που την επηρεάζουν, συμπεριλαμβανομένης της προέλευσης του συνόλου δεδομένων. Όσο περισσότερο ελέγχει μια εταιρεία τη διαδικασία συλλογής δεδομένων, τόσο μεγαλύτερη είναι πιθανόν να είναι η ειλικρίνεια. Εάν μια εταιρεία έχει τη διαβεβαίωση ότι ένα σύνολο δεδομένων είναι ακριβές και έγκυρο, αυτό της επιτρέπει να το εμπιστεύεται. Αυτό παρέχει μια πιο στέρεη βάση για τη λήψη σημαντικών επιχειρηματικών αποφάσεων από ό,τι ένα μη επικυρωμένο σύνολο δεδομένων από μια αβέβαιη πηγή. αυτό αποτελεί επίσης σημαντικό στοιχείο σε οποιαδήποτε ανάλυση δεδομένων που αφορά το αίσθημα των καταναλωτών. Μία από τις πιο δημοφιλείς δυνατότητες ανάλυσης κοινωνικών μέσων που χρησιμοποιούνται από μεγάλες εταιρείες είναι η δυνατότητα ανάλυσης του συναισθήματος των καταναλωτών με βάση τις λέξεις-κλειδιά που εμφανίζονται στις αναρτήσεις τους στα μέσα κοινωνικής δικτύωσης. Οι αναλυτές εξετάζουν την αξιοπιστία της συγκεκριμένης πλατφόρμας όταν αποφασίζουν πώς θα πραγματοποιήσουν ανάλυση σε μεγάλα δεδομένα, ώστε να διασφαλίσουν ότι το αποτέλεσμα είναι έγκυρο και χρήσιμο για τον τελικό πελάτη. Για να ενταχθεί κατάλληλα στην ταξινόμηση των μεγάλων δεδομένων, ένα σύνολο δεδομένων απαιτεί τουλάχιστον κάποια ένδειξη ειλικρίνειας.

Παρόλο που τα τέσσερα V καθορίζουν τι κάνει ένα συγκεκριμένο σύνολο δεδομένων να εμπίπτει στην κατηγορία των μεγάλων δεδομένων, υπάρχει μια πρόσθετη θεώρηση που είναι επίσης σημαντική όταν πρόκειται για την πραγματική εκτέλεση ανάλυσης σε μεγάλα δεδομένα. Αυτό το πέμπτο V είναι η αξία ή η χρησιμότητα του συνόλου δεδομένων. Μια εταιρεία μπορεί να έχει πολλαπλές εκδόσεις συνόλων δεδομένων που μπορεί να είναι μεγάλα σε όγκο, να ενημερώνονται τακτικά και να ποικίλουν, αλλά δεν έχει νόημα να διενεργεί ανάλυση σε αυτά τα σύνολα δεδομένων αν δεν υπάρχει αντιληπτή αξία για την εταιρεία. για παράδειγμα, ένας διαδικτυακός λιανοπωλητής μπορεί να αποστέλλει χιλιάδες δέματα ετησίως. Τα μεγάλα δεδομένα που απορρέουν από αυτό έρχονται σε πολλές μορφές, όπως λίστες διευθύνσεων αποστολής, βάρη δεμάτων και συγκεκριμένες παραγγελίες. Φυσικά, ορισμένες από αυτές τις τιμές συνεπάγονται δομημένα δεδομένα, ενώ άλλες όχι. Προτού ένας επιχειρηματικός αναλυτής εξετάσει τον καλύτερο τρόπο ταξινόμησης των δεδομένων, θα πρέπει πρώτα να εξετάσει ποια δεδομένα είναι πραγματικά χρήσιμα για την εταιρεία. Για παράδειγμα, η διενέργεια ανάλυσης του μέσου αριθμού χιλιομέτρων μεταξύ μιας παράδοσης και ενός κέντρου διανομής μπορεί να παρέχει πολύτιμες πληροφορίες για τη λήψη αποφάσεων σχετικά με τη θέση των μελλοντικών σημείων διανομής (Chebotko et al., 2015).

2.2.2 Εφαρμογές Μεγάλων Δεδομένων στο IoT

Το ταχύτατα αναπτυσσόμενο δίκτυο συνδεδεμένων αισθητήρων σε όλο τον κόσμο, το διαδίκτυο των πραγμάτων συμβάλλει σε κολοσσιαίο βαθμό στο σώμα των μεγάλων δεδομένων. Καθώς οι συνδεδεμένες συσκευές ξεφεύγουν από το πεδίο των καταναλωτικών ηλεκτρονικών, μια μεγάλη ποικιλία διαφορετικών ειδών δεδομένων μπαίνει στο παιχνίδι. Οι αισθητήρες IoT μπορούν να συλλέγουν πληροφορίες σχετικά με τη στάθμη του νερού σε αγροκτήματα, τις σεισμικές συνθήκες στο υπέδαφος, τα ζωτικά σημεία στην απομακρυσμένη παρακολούθηση ασθενών και πολλά άλλα. Η ύπαρξη διαφορετικών τύπων δεδομένων σε μια συγκεκριμένη περίπτωση χρήσης βοηθά τους αναλυτές να αποκτήσουν μια πιο ακριβή εικόνα του τι συμβαίνει και αποδίδει βαθύτερες γνώσεις. Για παράδειγμα, σε ένα έξυπνο εργοστάσιο, ο συνδυασμός δεδομένων από τους ανθρώπινους εργαζόμενους και τις συνδεδεμένες μηχανές - και από τα συστήματα της επιχείρησης και το χώρο του εργοστασίου (γεφύρωση του χάσματος OT/IT) - οδηγεί σε μεγαλύτερη αποδοτικότητα και παραγωγικότητα.

Μαζί με την παροχή νέων και διαφορετικών τύπων δεδομένων, οι αισθητήρες IoT παρέχουν επίσης απίστευτο όγκο δεδομένων. Οι εταιρείες που υλοποιούν μια εγκατάσταση συσκευών IoT συχνά δεν προβλέπουν το πλήθος δεδομένων που θα έρθει στο δρόμο τους μόλις ενεργοποιηθούν οι συσκευές. Για να τα αξιοποιήσουν στο έπακρο, θα χρειαστούν συστήματα για την αποθήκευση και την επεξεργασία τους, κατά προτίμηση σε πραγματικό χρόνο (ανάλογα με την περίπτωση χρήσης). Αυτός είναι ο λόγος για τον οποίο ο πολλαπλασιασμός των συσκευών IoT οδηγεί σε μια αυξανόμενη ανάγκη για επεξεργαστές άκρων υψηλής ταχύτητας και ευέλικτα συστήματα αποθήκευσης που βασίζονται στο cloud. Επειδή εξαρτάται από τις μηχανές για τη συλλογή δεδομένων και όχι από τους ανθρώπους, το IoT δημιουργεί σύνολα δεδομένων πολύ γρήγορα - συχνά σε πραγματικό χρόνο. Η επίτευξη αναλύσεων σε πραγματικό χρόνο εξαρτάται από εργαλεία που καθοδηγούνται από μηχανές, όπως η τεχνητή νοημοσύνη, η μηχανική μάθηση και η βαθιά μάθηση. Οι γνώσεις που αποκαλύπτουν μπορούν στη συνέχεια να επηρεάσουν τις συσκευές IoT, τους χρήστες και τις οργανωτικές αποφάσεις. Οι φωτεινοί σηματοδότες και οι κάμερες κυκλοφορίας είναι συνδεδεμένες στο δίκτυο IoT μιας έξυπνης πόλης. Μια κάμερα αποκαλύπτει τακτική κυκλοφοριακή συμφόρηση σε έναν κόμβο γύρω από την ώρα αιχμής κάθε μέρα, με αποτέλεσμα η κυκλοφορία να συσσωρεύεται στον αυτοκινητόδρομο. Εργαλεία τεχνητής νοημοσύνης αναλύουν τα δεδομένα και προτείνουν μια λύση: αύξηση της συχνότητας και της διάρκειας των πράσινων φώτων για την κυκλοφορία που εξέρχεται από τον αυτοκινητόδρομο κατά τη συγκεκριμένη χρονική περίοδο κάθε μέρα. Δεδομένου ότι οι φωτεινοί σηματοδότες είναι επίσης συνδεδεμένοι με το δίκτυο, οι ηγέτες της πόλης μπορούν εύκολα να κάνουν αυτή την προσαρμογή και να προκαλέσουν την ομαλότερη ροή της κυκλοφορίας κατά τις ώρες αιχμής (Sanja et al., 2020).

Οι συσκευές IoT συλλέγουν πλήθος σημείων δεδομένων σε πραγματικό χρόνο και τα δεδομένα αυτά μπορούν να παρέχουν πληροφορίες για διάφορες λειτουργίες, είτε αυτές πραγματοποιούνται αυτόνομα, όπως ο έλεγχος των φωτεινών σηματοδοτών με βάση το IoT, είτε χειροκίνητα, όπως η διαχείριση αεροδρομίων που ανακατευθύνει την κυκλοφορία των πεζών από μια περιοχή με συμφόρηση.

Οι αισθητήρες IoT μπορούν να συλλέγουν πολλαπλούς τύπους δεδομένων, όπως:

- Δεδομένα κατάστασης: ή άλλα ακριβή δεδομένα όπως η θερμοκρασία.
- Δεδομένα τοποθεσίας: Παρακολουθεί την κίνηση ανθρώπων ή αντικειμένων πάνω, πάνω ή κάτω από την επιφάνεια της γης.
- Δεδομένα αυτοματισμού: Μπορούν να χρησιμοποιηθούν για τον έλεγχο των ενεργειών αυτοματοποιημένων λειτουργιών ή συστημάτων, όπως μια αυτόνομη διαδρομή λεωφορείου.

Μόλις οι αισθητήρες συλλέξουν τα δεδομένα, τα στέλνουν σε μια κεντρική τοποθεσία χρησιμοποιώντας ένα πρωτόκολλο δεδομένων. Επειδή οι αισθητήρες IoT συλλέγουν δεδομένα είτε σε πραγματικό είτε σε σχεδόν πραγματικό χρόνο, ο όγκος των πληροφοριών που παράγουν είναι τεράστιος. Αυτό έχει ως αποτέλεσμα τα μεγάλα δεδομένα του IoT - και έχουν τη δυνατότητα να υπερφορτώσουν τα παραδοσιακά εργαλεία επεξεργασίας και διαχείρισης δεδομένων. Συνήθως, τα δεδομένα IoT διοχετεύονται σε μια κεντρική τοποθεσία όπου είναι διαθέσιμα για ανάλυση, ερμηνεία και δράση. Δυστυχώς, αυτό δεν είναι πάντα τόσο εύκολο όσο ακούγεται. Όταν τα σύνολα δεδομένων γίνονται τόσο μεγάλα και περίπλοκα, η εξαγωγή συμπερασμάτων και η πραγματοποίηση βελτιώσεων από αυτά καθίσταται δύσκολη. Τα δεδομένα έχουν ουσιαστικά χάσει τη χρησιμότητά τους για τη μετατροπή των επιχειρήσεων σε επιχειρήσεις με ψηφιακή πρωτοπορία που αξιοποιούν πλήρως τις δυνατότητες της μάθησης με βάση την τεχνητή νοημοσύνη.

Για την πλήρη αξιοποίηση των δεδομένων IoT, απαιτούνται εργαλεία διαχείρισης αποθήκευσης δεδομένων και ανάλυσης που έχουν σχεδιαστεί ειδικά για την υποστήριξη μεγάλων δεδομένων. Η διαχείριση των μεγάλων δεδομένων και του διαδικτύου των πραγμάτων απαιτεί αποθηκευτικά συστήματα που να μπορούν να ανταποκριθούν στις αυξημένες απαιτήσεις. Οι πιο προηγμένες πλατφόρμες μεγάλων δεδομένων όχι μόνο αποθηκεύουν τεράστιες ποσότητες δεδομένων IoT, αλλά υποστηρίζουν και γρήγορη αναζήτηση, ευρετηρίαση και ανάλυση των δεδομένων σε πραγματικό χρόνο. Οι σύγχρονες πλατφόρμες υψηλής απόδοσης είναι σε θέση να εισάγουν δεδομένα με μεγάλη ταχύτητα, κλιμακώνοντας τις δυνατότητές τους ώστε να ανταποκρίνονται στις ανάγκες του οργανισμού. Παράλληλα, διευκολύνουν την αναζήτηση και την ευρετηρίαση των δεδομένων, εξοικονομώντας πολύτιμο χρόνο που διαφορετικά θα δαπανούταν σε ερωτήματα και διαδικασίες ανάλυσης. Με αυτόν τον τρόπο, αυτές οι πλατφόρμες καθιστούν δυνατή τη γρήγορη και αποδοτική αξιοποίηση των δεδομένων, συμβάλλοντας στη λήψη τεκμηριωμένων

αποφάσεων. Όταν επιλέγεται μια πλατφόρμα για τη διαχείριση μεγάλων δεδομένων και του διαδικτύου των πραγμάτων, είναι σημαντικό να προτιμάται μια πλατφόρμα βελτιστοποιημένη για το cloud. Μια τέτοια πλατφόρμα επιτρέπει την εκτέλεση αναλύσεων στο σύννεφο, ενώ ταυτόχρονα προσφέρει τη δυνατότητα ελέγχου της πρόσβασης και των δικαιωμάτων στα δεδομένα που αποθηκεύονται στις εγκαταστάσεις. Αυτή η προσέγγιση βελτιώνει τη συνολική ταχύτητα ανάλυσης των δεδομένων IoT και μειώνει τον χρόνο που απαιτείται για τη λήψη τεκμηριωμένων αποφάσεων. Τέτοιες αποφάσεις είναι απαραίτητες για την αξιοποίηση της επένδυσης σε μια υποδομή IoT και την επίτευξη στρατηγικών στόχων (Damirova et al., 2022).

2.3 Τεχνικές και Εργαλεία για Ανάλυση Δεδομένων

2.3.1 Βιβλιοθήκες Python

Η Pandas είναι μια ισχυρή βιβλιοθήκη Python για την επεξεργασία και ανάλυση δεδομένων. Παρέχει εύχρηστες δομές δεδομένων, όπως τα DataFrames, τα οποία είναι δομές σε μορφή πίνακα που επιτρέπουν να αποθηκεύετε και να χειρίζεστε δεδομένα αποτελεσματικά. Το Pandas προσφέρει ένα ευρύ φάσμα συναρτήσεων και μεθόδων για τον καθαρισμό, την προεπεξεργασία, τη συγχώνευση, την ομαδοποίηση και πολλά άλλα δεδομένα. Χρησιμοποιείται ευρέως για εργασίες επεξεργασίας και εξερεύνησης δεδομένων σε έργα επιστήμης δεδομένων. Διαθέτει πολλά πλεονεκτήματα, τα περισσότερα από τα οποία σχετίζονται με το γεγονός ότι στο pandas, τα δεδομένα έχουν ετικέτες. Ένα άλλο πλεονέκτημα του pandas είναι οι καλές δυνατότητες εισόδου/εξόδου - τα δεδομένα μπορούν εύκολα να τραβηχτούν στο πλαίσιο δεδομένων του pandas (pandas - Python Data Analysis Library).

Η NumPy (Numerical Python) είναι μια θεμελιώδης βιβλιοθήκη για επιστημονικούς υπολογισμούς στην Python. Παρέχει υποστήριξη για μεγάλους, πολυδιάστατους πίνακες και πίνακες, μαζί με μια συλλογή μαθηματικών συναρτήσεων για την αποτελεσματική λειτουργία αυτών των πινάκων. Οι λειτουργίες των πινάκων της NumPy είναι σημαντικά ταχύτερες από τις παραδοσιακές λίστες της Python, γεγονός που την καθιστά δημοφιλή επιλογή για αριθμητικούς υπολογισμούς. Αποτελεί τη βάση για πολλές άλλες βιβλιοθήκες στο επιστημονικό οικοσύστημα της Python. Το Numpy είναι ένα πακέτο επεξεργασίας συστοιχιών γενικού σκοπού. Παρέχει ένα αντικείμενο πολυδιάστατων πινάκων υψηλής απόδοσης και εργαλεία για την εργασία με αυτούς τους πίνακες. Είναι το θεμελιώδες πακέτο για επιστημονικούς υπολογισμούς με την Python. Εκτός από τις προφανείς επιστημονικές χρήσεις του, το Numpy μπορεί επίσης να χρησιμοποιηθεί ως αποδοτικός πολυδιάστατος περιέκτης γενικών δεδομένων (NumPy)

Η βιβλιοθήκη γραφικών παραστάσεων Python του Plotly δημιουργεί διαδραστικά γραφήματα ποιότητας δημοσίευσης. Παραδείγματα για το πώς να φτιάχνετε γραμμικά διαγράμματα, διαγράμματα διασποράς, διαγράμματα περιοχής, ραβδογράμματα, διαγράμματα πλαισίου,

ιστογράμματα ή θερμοχάρτες. Η βιβλιοθήκη Plotly στην Python είναι μια βιβλιοθήκη ανοιχτού κώδικα που μπορεί να χρησιμοποιηθεί για την οπτικοποίηση δεδομένων και την κατανόηση δεδομένων απλά και εύκολα. Το Plotly έχει δυνατότητες που επιτρέπουν να εντοπιστούν τυχόν ακραίες τιμές ή ανωμαλίες σε ένα μεγάλο αριθμό σημείων δεδομένων. Είναι οπτικά ελκυστικό και μπορεί να γίνει αποδεκτό από ένα ευρύ φάσμα κοινού και επιτρέπει την ατελείωτη προσαρμογή των γραφημάτων μας και καθιστά το οικόπεδό μας πιο ουσιαστικό και κατανοητό για τους άλλους (Plotly Python Graphing Library

Η βιβλιοθήκη Dash της Python είναι ένα ισχυρό εργαλείο για τη δημιουργία διαδραστικών web εφαρμογών δεδομένων, ειδικά σχεδιασμένο για επιστήμονες δεδομένων, αναλυτές και προγραμματιστές που επιθυμούν να παρουσιάσουν και να αλληλοεπιδράσουν με τα δεδομένα τους μέσα από ένα γραφικό περιβάλλον. Η Dash βασίζεται στις τεχνολογίες Flask, Plotly και React.js, συνδυάζοντας την ευκολία στη χρήση με τη μεγάλη ευελιξία. Με τη Dash, οι χρήστες μπορούν να κατασκευάσουν web εφαρμογές πλήρως σε Python, χωρίς την ανάγκη γνώσης HTML, CSS ή JavaScript. Η διαμόρφωση της εφαρμογής γίνεται με κώδικα Python, που περιλαμβάνει την οπτικοποίηση δεδομένων, τη διαχείριση αλληλεπιδράσεων και τη δυναμική ενημέρωση των στοιχείων της εφαρμογής. Ένα από τα πιο αξιοσημείωτα χαρακτηριστικά της Dash είναι η υποστήριξη διαδραστικών γραφημάτων και widgets, όπως dropdown menus, sliders και checkboxes, που επιτρέπουν στους χρήστες να αλλάζουν παραμέτρους και να βλέπουν άμεσα τα αποτελέσματα στις οπτικοποιήσεις. Η βιβλιοθήκη χρησιμοποιεί τον Flask για τη διαχείριση των backend λειτουργιών και τον Plotly για την παραγωγή εντυπωσιακών γραφημάτων και οπτικοποιήσεων. Ως αποτέλεσμα, οι εφαρμογές Dash μπορούν να λειτουργούν σε οποιοδήποτε πρόγραμμα περιήγησης και είναι ανεξάρτητες από την πλατφόρμα, παρέχοντας ευελιξία και ευρεία συμβατότητα. Η Dash υποστηρίζει επίσης σύνθετες λειτουργίες, όπως η δυναμική φόρτωση δεδομένων, οι κλήσεις API, η διαχείριση μεγάλων όγκων δεδομένων και η δημιουργία εξατομικευμένων layouts. Είναι ιδιαίτερα κατάλληλη για τη δημιουργία πινάκων ελέγχου και εφαρμογών ανάλυσης δεδομένων σε πραγματικό χρόνο, καθιστώντας την δημοφιλή σε τομείς όπως τα χρηματοοικονομικά, η βιοεπιστήμη, η εκπαίδευση και η τεχνολογία (Dash Documentation & User Guide | Plotly).

2.3.2 Τεχνικές και Εργαλεία για Ανάλυση Δεδομένων

Η ανάλυση δεδομένων είναι η συλλογή, ο καθαρισμός, η ανάλυση, η παρουσίαση και η ερμηνεία δεδομένων για την εξαγωγή συμπερασμάτων. Η διαδικασία αυτή βοηθά στη λήψη αποφάσεων παρέχοντας χρήσιμες πληροφορίες και στατιστικά στοιχεία. Η ιστορία της ανάλυσης δεδομένων χρονολογείται από τη δεκαετία του 1640. Ο Τζον Γκραντ, ένας κατασκευαστής καπέλων, άρχισε να συλλέγει τον αριθμό των θανάτων στο Λονδίνο. Ήταν ο πρώτος άνθρωπος που χρησιμοποίησε την ανάλυση δεδομένων για την επίλυση ενός

προβλήματος. Ο σύγχρονος επιχειρηματικός κόσμος βασίζεται στα γεγονότα και όχι στη διαίσθηση. Η ανάλυση δεδομένων χρησιμεύει ως θεμέλιο για τη λήψη τεκμηριωμένων αποφάσεων. Η ανάλυση δεδομένων είναι, με απλά λόγια, η διαδικασία ανακάλυψης χρήσιμων πληροφοριών μέσω της αξιολόγησης δεδομένων. Αυτό γίνεται μέσω μιας διαδικασίας επιθεώρησης, καθαρισμού, μετασχηματισμού και μοντελοποίησης δεδομένων με τη χρήση αναλυτικών και στατιστικών εργαλείων, τα οποία θα εξερευνήσουμε λεπτομερώς στη συνέχεια αυτού του άρθρου. Η αποτελεσματική ανάλυση δεδομένων βοηθά τους οργανισμούς να λαμβάνουν επιχειρηματικές αποφάσεις. Τα δεδομένα συλλέγονται από τις επιχειρήσεις συνεχώς: μέσω ερευνών, διαδικτυακής παρακολούθησης, ανάλυσης διαδικτυακού μάρκετινγκ, συλλογής δεδομένων συνδρομής και εγγραφής παρακολούθησης των μέσων κοινωνικής δικτύωσης, μεταξύ άλλων μεθόδων (Giorgini et al., 2005).

Οι τεχνικές ανάλυσης δεδομένων έχουν εξελιχθεί σημαντικά, παρέχοντας μια ολοκληρωμένη εργαλειοθήκη για την κατανόηση, την ερμηνεία και την πρόβλεψη μοτίβων δεδομένων. Αυτές οι μέθοδοι είναι ζωτικής σημασίας για την εξαγωγή αξιοποιήσιμων πληροφοριών από τα δεδομένα, επιτρέποντας στους οργανισμούς να λαμβάνουν τεκμηριωμένες αποφάσεις. Η περιγραφική ανάλυση θεωρείται το σημείο εκκίνησης του αναλυτικού ταξιδιού και συχνά προσπαθεί να απαντήσει σε ερωτήματα που σχετίζονται με το τι συνέβη. Η τεχνική αυτή ακολουθεί την ταξινόμηση των παραγόντων, τον χειρισμό και την ερμηνεία ποικίλων δεδομένων από διαφορετικές πηγές και, στη συνέχεια, τη μετατροπή τους σε πολύτιμες γνώσεις. Επιπλέον, η διενέργεια αυτής της ανάλυσης είναι επιβεβλημένη, καθώς επιτρέπει στα άτομα να προβάλλουν τις γνώσεις τους με μια βελτιωμένη μέθοδο. Αυτή η τεχνική δεν επιτρέπει να εκτιμήσετε μελλοντικά αποτελέσματα - όπως ο εντοπισμός συγκεκριμένων αιτιολογήσεων για έναν συγκεκριμένο παράγοντα. Θα διατηρήσει τα δεδομένα εξορθολογισμένα και θα απλοποιήσει τη διεξαγωγή μιας ενδελεχούς αξιολόγησης για περαιτέρω συνθήκες.

Οι τεχνικές ποιοτικής ανάλυσης δεδομένων δεν μπορούν να μετρηθούν άμεσα και, ως εκ τούτου, η τεχνική αυτή χρησιμοποιείται όταν ένας οργανισμός πρέπει να λάβει αποφάσεις βάσει υποκειμενικής ερμηνείας. Για παράδειγμα, τα ποιοτικά δεδομένα μπορεί να περιλαμβάνουν την αξιολόγηση των ανατροφοδοτήσεων των πελατών, τον αντίκτυπο των ερωτήσεων της έρευνας, την αποτελεσματικότητα των αναρτήσεων στα μέσα κοινωνικής δικτύωσης, την ανάλυση συγκεκριμένων αλλαγών ή χαρακτηριστικών ενός προϊόντος και πολλά άλλα. Η εστίαση αυτής της τεχνικής θα πρέπει να είναι ο εντοπισμός ουσιαστικών πληροφοριών ή απαντήσεων από μη δομημένα δεδομένα, όπως απομαγνητοφωνήσεις, φωνητικά σχόλια και άλλα. Επιπλέον, η ποιοτική ανάλυση βοηθά στην οργάνωση των δεδομένων σε θέματα ή κατηγορίες, οι οποίες μπορούν να αυτοματοποιηθούν περαιτέρω. Η ποσοτική ανάλυση δεδομένων αναφέρεται σε μετρήσιμες πληροφορίες, οι οποίες περιλαμβάνουν συγκεκριμένους αριθμούς και ποσότητες. Για παράδειγμα, αριθμοί

πωλήσεων, εκστρατείες ηλεκτρονικού ταχυδρομείου με βάση τα ποσοστά κλικ, επισκέπτες ιστότοπου, ποσοστό απόδοσης εργαζομένων ή ποσοστό για τα παραγόμενα έσοδα και πολλά άλλα. Τα άτομα πρέπει να χρησιμοποιούν τα αποτελέσματα της περιγραφικής ανάλυσης δεδομένων, τις τεχνικές διερευνητικής και διαγνωστικής ανάλυσης και να συνδυάζουν τη μηχανική μάθηση και την τεχνητή νοημοσύνη (Gorton et al., 2015).

Με εύκολα κατανοητές γνώσεις, οι επιχειρήσεις μπορούν να αξιοποιήσουν τις τάσεις, τα κοινά μοτίβα ή τους λόγους για ένα συγκεκριμένο γεγονός, καθιστώντας ευκολότερες τις πρωτοβουλίες ή τις αποφάσεις για περαιτέρω στρατηγικές. Όταν είναι γνωστό το "γιατί" συνέβη κάτι, είναι πιο εύκολο να προσδιοριστεί το "πώς" για τη συγκεκριμένη πτυχή. Για παράδειγμα, με τη διαγνωστική ανάλυση, μπορεί να εντοπιστεί γιατί μειώνονται τα αποτελέσματα των πωλήσεων και να διερευνηθούν οι ακριβείς παράγοντες που οδήγησαν σε αυτή την απώλεια. Επιπλέον, η τεχνική αυτή προσφέρει αξιοποιήσιμες απαντήσεις σε συγκεκριμένα ερωτήματα. Είναι επίσης η πιο συχνά προτιμώμενη μέθοδος στην έρευνα σε ποικίλους τομείς. Ανάλυση παλινδρόμησης είναι η μέθοδος που χρησιμοποιεί ιστορικά δεδομένα για να κατανοήσει τον αντίκτυπο στην τιμή της εξαρτημένης μεταβλητής όταν μία ή περισσότερες ανεξάρτητες μεταβλητές τείνουν να αλλάξουν ή να παραμείνουν ίδιες. Επιπλέον, ο προσδιορισμός της σχέσης κάθε μεταβλητής και της παρελθούσας εξέλιξης ή πρωτοβουλίας επιτρέπει να προβλέψετε πιθανά αποτελέσματα στο μέλλον. Η τεχνική αυτή δίνει τη σωστή πορεία για την αποτελεσματική λήψη τεκμηριωμένων αποφάσεων.

Η ανάλυση Cohort περιλαμβάνει ιστορικά δεδομένα για την ανάλυση και τη σύγκριση συγκεκριμένων τμημάτων στη συμπεριφορά των χρηστών και ομαδοποιεί μερικές πτυχές με άλλα παρόμοια στοιχεία. Αυτή η τεχνική μπορεί να δώσει μια ιδέα για τις εξελισσόμενες ανάγκες των πελατών και του κοινού-στόχου. Επιπλέον, μπορεί να χρησιμοποιηθεί η ανάλυση Cohort για τον προσδιορισμό του αντίκτυπου μιας εκστρατείας μάρκετινγκ σε συγκεκριμένες ομάδες κοινού. Για παράδειγμα, τα χαρακτηριστικά της τεχνικής ανάλυσης Cohort μπορούν να αξιοποιηθούν για την αξιολόγηση δύο τύπων εκστρατειών ηλεκτρονικού ταχυδρομείου - που συχνά αναφέρονται ως δοκιμές A/B με την πάροδο του χρόνου - και για την κατανόηση του ποια παραλλαγή αποδείχθηκε πιο αποδοτική και είχε μεγαλύτερο αντίκτυπο στην απόδοση. Η παραγοντική ανάλυση δεδομένων προσδιορίζει τις μεταβολές με τις παρατηρούμενες σχετικές μεταβλητές με βάση χαμηλότερες μη παρατηρούμενες μεταβλητές που ονομάζονται παράγοντες. Εν ολίγοις, βοηθά στην εξαγωγή ανεξάρτητων μεταβλητών, οι οποίες θεωρούνται ιδανικές για τη βελτιστοποίηση συγκεκριμένων τμημάτων.

Για παράδειγμα, εάν ένα προϊόν συνοδεύεται από τη συλλογή σχολίων πελατών για διάφορους σκοπούς, η συγκεκριμένη τεχνική ανάλυσης βοηθά στη συγκέντρωση της προσοχής σε συγκεκριμένους παράγοντες, όπως οι τρέχουσες τάσεις, η διάταξη, η απόδοση του προϊόντος, τα πιθανά σφάλματα και άλλα. Οι παράγοντες αυτοί μπορεί να διαφέρουν ανάλογα με το τι παρακολουθείται και σε τι επικεντρώνεται η ανάλυση. Επιπλέον, η ανάλυση παραγόντων

διευκολύνει τη σύνοψη των σχετικών παραμέτρων ομαδοποιώντας τους σε παρόμοιες κατηγορίες. Αυτή η τεχνική είναι ιδανική για τον προσδιορισμό του αν μια μεταβλητή άλλαξε κατά τη διάρκεια του διαστήματος αξιολόγησης, πώς εξαρτάται κάθε μεταβλητή και πώς επιτεύχθηκε το αποτέλεσμα για μια συγκεκριμένη πτυχή. Επιπλέον, η ανάλυση χρονοσειρών μπορεί να χρησιμοποιηθεί για τον εντοπισμό των τάσεων και των προτύπων της αγοράς με την πάροδο του χρόνου. Επίσης, αυτή η μέθοδος επιτρέπει την πρόβλεψη μελλοντικών γεγονότων με βάση συγκεκριμένες πληροφορίες που προκύπτουν από τα δεδομένα.

Η ανάλυση συστάδων περιγράφει δεδομένα και εντοπίζει κοινά πρότυπα. Χρησιμοποιείται συχνά όταν τα δεδομένα χρειάζονται πιο εμφανείς ετικέτες ή ιδέες ή έχουν διφορούμενες κατηγορίες. Η διαδικασία αυτή περιλαμβάνει την αναγνώριση παρόμοιων παρατηρήσεων και την ομαδοποίηση αυτών των πτυχών για τη δημιουργία συστάδων, δηλαδή την απόδοση ονομάτων και την κατηγοριοποίηση ομάδων. Επιπλέον, η τεχνική αυτή βοηθά στον εντοπισμό ομοιοτήτων και ανισοτήτων σε βάσεις δεδομένων και στην παρουσίασή τους με μια οπτικά οργανωμένη μέθοδο για την απρόσκοπτη σύγκριση παραγόντων. Η οπτικοποίηση του box plot προτιμάται κυρίως για την παρουσίαση των συστάδων δεδομένων. Κάθε μέθοδος προσφέρει μοναδικά οφέλη και είναι κατάλληλη για διαφορετικούς τύπους προκλήσεων δεδομένων. Η κατανόηση και η εφαρμογή των σωστών τεχνικών ανάλυσης δεδομένων μπορεί να επηρεάσει σημαντικά τη στρατηγική και τις διαδικασίες λήψης αποφάσεων ενός οργανισμού, οδηγώντας σε πιο στοχευμένα, αποδοτικά και αποτελεσματικά αποτελέσματα (Sumit et al., 2017).

2.4 Έξυπνος οικιακός έλεγχος - Πίνακες εργαλείων

Οι πίνακες ελέγχου επιτρέπουν την εμφάνιση πληροφοριών σχετικά με το έξυπνο σπίτι, παρέχοντας μια ισχυρή και προσαρμόσιμη λύση για τη διαχείρισή του μέσω κινητού ή υπολογιστή. Μπορούν να προσαρμοστούν με διάφορους τρόπους, όπως η χρήση διαφορετικών τύπων καρτών για την απεικόνιση δεδομένων και τον έλεγχο συσκευών, η εφαρμογή θεμάτων (ακόμη και ανά κάρτα), καθώς και η παράκαμψη ονομάτων και εικονιδίων οντοτήτων. Επιπλέον, προσφέρεται η δυνατότητα χρήσης προσαρμοσμένων καρτών που έχουν δημιουργηθεί από την ενεργή κοινότητα χρηστών. Ένας πίνακας ελέγχου έξυπνου σπιτιού αποτελεί ένα ισχυρό εργαλείο που μπορεί να συμβάλει στη βελτίωση της άνεσης, της ευκολίας και της ασφάλειας ενός σπιτιού. Είτε χρησιμοποιείται σε μια κεντρική τοποθεσία, είτε σε διάφορα σημεία του σπιτιού, είτε μέσω φορητών συσκευών, παρέχει ένα κεντρικό σημείο διαχείρισης όλων των έξυπνων συσκευών, προσφέροντας ενημερώσεις σε πραγματικό χρόνο και δυνατότητες αυτοματοποίησης που απλοποιούν την καθημερινότητα (Lea et al., 2018).

Ένας πίνακας ελέγχου επιτρέπει τον έλεγχο όλων των έξυπνων συσκευών από μία κεντρική τοποθεσία, όπως ένα smartphone, tablet ή υπολογιστή. Αυτό εξοικονομεί χρόνο και προσπάθεια, ενώ διευκολύνει την παρακολούθηση των συσκευών. Για παράδειγμα, μπορούν

να ενεργοποιηθούν τα φώτα, να κλειδωθούν οι πόρτες ή να ρυθμιστεί ο θερμοστάτης, όλα από ένα σημείο. Οι πίνακες ελέγχου παρέχουν ενημερώσεις σε πραγματικό χρόνο για την κατάσταση του σπιτιού, όπως τη θερμοκρασία, την ασφάλεια και την κατανάλωση ενέργειας. Αυτό βοηθά στη βελτίωση της ενεργειακής απόδοσης, της ασφάλειας και της άνεσης του σπιτιού. Επιπλέον, σε συνδυασμό με μια μηχανή κανόνων, μπορούν να αποστέλλονται ειδοποιήσεις για ασυνήθιστες δραστηριότητες που ανιχνεύονται από το σύστημα ασφαλείας. Μέσω ενός πίνακα ελέγχου, μπορούν να εκτελούνται αυτοματισμοί που απλοποιούν την καθημερινότητα. Για παράδειγμα, με την επιλογή ενός κουμπιού στον πίνακα ελέγχου, μπορούν να χαμηλώσουν τα φώτα, να ρυθμιστεί η θερμοκρασία και να διασφαλιστεί ότι οι πόρτες είναι κλειδωμένες, εξοικονομώντας χρόνο και ενέργεια, ενώ παράλληλα αυξάνεται η ασφάλεια του σπιτιού. Οι πίνακες ελέγχου μπορούν να προσαρμοστούν ώστε να καλύπτουν τις ανάγκες του χρήστη. Είναι δυνατό να προστεθούν widgets για αγαπημένες συσκευές, να αλλάξει ο χρωματικός σχεδιασμός και να ενσωματωθεί προσωπικό περιεχόμενο, όπως φωτογραφίες. Αυτή η προσαρμογή κάνει τον πίνακα ελέγχου πιο ευχάριστο στη χρήση και πιο χρήσιμο, όπως η προσθήκη ενός widget για το σύστημα ασφαλείας, ώστε να εμφανίζεται η κατάσταση του σπιτιού με μία ματιά. Στην εποχή των έξυπνων σπιτιών και του αυτοματισμού, η ύπαρξη ενός κεντρικού συστήματος ελέγχου είναι απαραίτητη για την αποτελεσματική διαχείριση διαφόρων συσκευών και εργασιών. Ένας πίνακα ελέγχου χρησιμεύει ως ένα οπτικό περιβάλλον εργασίας όπου οι χρήστες μπορούν να αλληλοεπιδρούν με τις έξυπνες οικιακές συσκευές και τους αισθητήρες. Ενοποιεί πληροφορίες από διάφορες πηγές και τις παρουσιάζει με σαφή και οργανωμένο τρόπο, διευκολύνοντας τον έλεγχο και την παρακολούθηση διαφόρων πτυχών του σπιτιού. Είτε πρόκειται για τη ρύθμιση του φωτισμού, τον έλεγχο της θερμοκρασίας ή τη διαχείριση των συστημάτων ασφαλείας, ένα καλά σχεδιασμένο πίνακα ελέγχου βελτιώνει την εμπειρία του χρήστη και απλοποιεί τις καθημερινές εργασίες (Lea et al., 2018).

Η υιοθέτηση έξυπνων οικιακών πινάκων ελέγχου προσφέρει πολλά πλεονεκτήματα:

- Απλοποιημένη οικιακή διαχείριση: Η διαχείριση πολλαπλών έξυπνων συσκευών μέσω μιας ενιαίας διεπαφής μειώνει την πολυπλοκότητα και βελτιώνει την εμπειρία του χρήστη.
- Βελτίωση της αξίας του σπιτιού: Τα σπίτια που είναι εξοπλισμένα με προηγμένα έξυπνα οικιακά συστήματα είναι πιο ελκυστικά για τους δυνητικούς αγοραστές, αυξάνοντας την αξία του ακινήτου.
- Προσβασιμότητα: Οι πίνακες ελέγχου του έξυπνου σπιτιού μπορούν να σχεδιαστούν έτσι ώστε να εξυπηρετούν άτομα με αναπηρίες, παρέχοντας μεγαλύτερη ανεξαρτησία και ευκολία στη χρήση.

- Επεκτασιμότητα: Καθώς οι ανάγκες και οι τεχνολογίες εξελίσσονται, οι έξυπνοι οικιακοί πίνακες ελέγχου μπορούν εύκολα να κλιμακωθούν για να ενσωματώσουν νέες συσκευές και λειτουργίες.

Οι έξυπνοι οικιακοί πίνακες ελέγχου δεν αποτελούν απλώς πολυτέλεια αλλά αναγκαιότητα για τη σύγχρονη διαβίωση. Προσφέρουν κεντρικό έλεγχο, ενισχύουν την ασφάλεια, βελτιώνουν την ενεργειακή απόδοση και παρέχουν απαράμιλλη ευκολία. Με τις εξελίξεις στην τεχνητή νοημοσύνη, τη διαλειτουργικότητα και την προσαρμογή, το μέλλον των έξυπνων οικιακών πινάκων ελέγχου είναι λαμπρό, υποσχόμενο ακόμη πιο έξυπνα και φιλικά προς τον χρήστη σπίτια. Η επένδυση σε έναν έξυπνο οικιακό πίνακα ελέγχου σήμερα είναι ένα βήμα προς ένα πιο συνδεδεμένο, αποτελεσματικό και άνετο μέλλον.

- Ευκολία: Ένα από τα κύρια πλεονεκτήματα του οικιακού αυτοματισμού είναι η ευκολία που προσφέρει. Με τον οικιακό αυτοματισμό, μπορούν να ελεγχθούν διάφορες πτυχές του σπιτιού με μερικά μόνο κλικ στο smartphone ή με φωνητικές εντολές στον εικονικό βοηθό. Αυτό περιλαμβάνει τα πάντα, από το σβήσιμο των φώτων μέχρι τη ρύθμιση του θερμοστάτη και το ξεκλείδωμα των θυρών.
- Ενεργειακή απόδοση: Ο οικιακός αυτοματισμός μπορεί επίσης να βοηθήσει να εξοικονομηθεί ενέργεια και να μειωθεί το αποτύπωμα άνθρακα. Για παράδειγμα, μπορεί να ρυθμιστεί ο έξυπνος θερμοστάτη να ρυθμίζει αυτόματα τη θερμοκρασία ανάλογα με το αν κάποιος βρίσκεται στο σπίτι ή αν λείπει, ή να σβήσει τα φώτα σε δωμάτια που δεν χρησιμοποιούνται.
- Αυξημένη ασφάλεια: Ο οικιακός αυτοματισμός μπορεί επίσης να ενισχύσει την ασφάλεια του σπιτιού. Με ένα έξυπνο σύστημα οικιακής ασφάλειας, μπορείτε να παρακολουθεί το σπίτι και να λαμβάνονται ειδοποιήσεις εάν υπάρχει οποιαδήποτε ύποπτη δραστηριότητα. Μπορούν επίσης να ελεγχθούν οι κάμερες ασφαλείας και το σύστημα συναγερμού μέσω ενός smartphone.
- Εξοικονόμηση κόστους: Τέλος, η αυτοματοποίηση του σπιτιού μπορεί επίσης να εξοικονομήσει χρήματα μακροπρόθεσμα. Μειώνοντας την κατανάλωση ενέργειας και αυτοματοποιώντας την ασφάλεια του σπιτιού, μπορεί να μειωθούν οι μηνιαίοι λογαριασμοί κοινής ωφέλειας.

2.5 Αποτελεσματική διαχείριση ενέργειας – Στρατηγικές

Ένα σύστημα οικιακής ενεργειακής διαχείρισης βελτιστοποιεί και ελέγχει την παραγωγή, την αποθήκευση και τη χρήση ενέργειας στο νοικοκυριό. Με την ενσωμάτωση έξυπνων συσκευών και ενεργειακών δεδομένων από διάφορες πηγές παρέχει πληροφορίες και εργαλεία διαχείρισης σε πραγματικό χρόνο για τη βελτίωση της ενεργειακής απόδοσης και τη μείωση του κόστους. Παρακολουθεί τα πρότυπα κατανάλωσης, ενσωματώνεται με ανανεώσιμες πηγές ενέργειας, όπως οι ηλιακοί συλλέκτες, και διαχειρίζεται τα οικιακά συστήματα αποθήκευσης

ενέργειας και αυτοματοποιεί τη χρήση ενέργειας βάσει προκαθορισμένων ρυθμίσεων ή συνθηκών πραγματικού χρόνου, προσαρμόζοντας θερμοστάτες, φωτισμό, φορτιστές EV και συσκευές για την ελαχιστοποίηση του κόστους και την ανταπόκριση στα δυναμικά ενεργειακά τιμολόγια.

Υπάρχουν μερικοί βασικοί παράγοντες γύρω από τη διαχείριση της ενέργειας που απαντώνται στα σύγχρονα σπίτια:

- Ηλιακή ενέργεια στο σπίτι.
- Τύπος θέρμανσης και ψύξης: θέρμανση πόλης (ζεστό νερό), φυσικό αέριο, αντλία θερμότητας ή ηλεκτρική/ακτινοβολία.
- Μεγάλοι καταναλωτές ενέργειας και η ρυθμιστική τους ικανότητα.
- Διαθεσιμότητα και χρήση ηλεκτρικών οχημάτων.
- Διαθεσιμότητα και χρήση οικιακής αποθήκευσης ενέργειας, όπως οικιακές μπαταρίες.
- Τύπος σύμβασης ενέργειας: σταθερή, μεταβλητή ή δυναμική.

Η ηλιακή ενέργεια αποτελεί έναν ολοένα και πιο δημοφιλή και αυξανόμενο παράγοντα στα συστήματα διαχείρισης ενέργειας για το σπίτι. Η εγκατάσταση ηλιακών συλλεκτών επιτρέπει την αξιοποίηση ανανεώσιμων πηγών ενέργειας για την τροφοδοσία του νοικοκυριού, παρέχοντας τη δυνατότητα εξοικονόμησης χρημάτων από τους λογαριασμούς ηλεκτρικού ρεύματος και μείωσης του αποτυπώματος άνθρακα. Για την αποτελεσματική διαχείριση της ηλιακής ενέργειας, είναι σημαντική η παρακολούθηση της κατανάλωσης και της παραγωγής ενέργειας, ώστε να διασφαλίζεται η βέλτιστη χρήση της, ιδανικά κατά την περίοδο παραγωγής. Η θέρμανση και η ψύξη αποτελούν συνήθως τις μεγαλύτερες πηγές κατανάλωσης ενέργειας στα νοικοκυριά. Υπάρχουν πολλές διαθέσιμες επιλογές θέρμανσης, καθεμία με τα δικά της χαρακτηριστικά. Σε γενικές γραμμές, οι αντλίες θερμότητας θεωρούνται η πιο αποδοτική επιλογή θέρμανσης που βασίζεται στην ηλεκτρική ενέργεια. Συχνά συνδυάζονται με ηλεκτρικές θερμάστρες ακτινοβολίας ή θερμάστρες χώρου για την κάλυψη των αναγκών συγκεκριμένων δωματίων, ένας ιδιαίτερα δημοφιλής συνδυασμός σε νέα, χωρίς αέριο σπίτια. Αντίθετα, λύσεις που βασίζονται στη θέρμανση πόλης, στο ζεστό νερό ή στο αέριο είναι από τη φύση τους λιγότερο ευέλικτες. Καθώς αυτές οι μορφές ενέργειας δεν μπορούν να παραχθούν από τους ίδιους τους καταναλωτές και οι τιμές τους είναι πιο σταθερές σε μακροπρόθεσμη βάση, προσφέρουν περιορισμένο έλεγχο στο ενεργειακό κόστος. Γενικά, όσο πιο ηλεκτρικό είναι το σύστημα θέρμανσης, τόσο μεγαλύτερη είναι η δυνατότητα διαχείρισης και αξιοποίησης της ενέργειας μέσω αποτελεσματικών πρακτικών διαχείρισης. Τέλος, η εξοικονόμηση που προκύπτει από τη μείωση της κατανάλωσης ενέργειας αποτελεί έναν ελκυστικό στόχο για όλους τους τύπους συστημάτων θέρμανσης (Carlos et al., 2015).

Η απογραφή των μεγαλύτερων καταναλωτών ενέργειας ενός σπιτιού αποτελεί σημαντικό βήμα για την κατανόηση και τη βελτιστοποίηση της ενεργειακής διαχείρισης. Εκτός από τη θέρμανση, αυτοί οι καταναλωτές μπορεί να περιλαμβάνουν πλυντήρια ρούχων, στεγνωτήρια,

πλυντήρια πιάτων, σάουνες, πισίνες, ηλεκτρικές ή επαγωγικές σόμπες, φούρνους και καταψύκτες. Ορισμένες από αυτές τις συσκευές διαθέτουν ρυθμιστική ικανότητα, πράγμα που σημαίνει ότι μπορούν να αποθηκεύουν ενέργεια, παρέχοντας σημαντικές δυνατότητες διαχείρισης. Για παράδειγμα, οι καταψύκτες μπορούν να λειτουργούν σε εύρος θερμοκρασιών, όπως μεταξύ -18 και -30 βαθμών Κελσίου, ενώ τα δωμάτια θέρμανσης ή ψύξης μπορούν να διατηρούν θερμοκρασίες μεταξύ 19 και 22 βαθμών Κελσίου. Πισίνες με ελαφρώς μεταβαλλόμενες θερμοκρασίες, δεξαμενές απομόνωσης αντλιών θερμότητας που μπορούν να προσαρμόζουν τη θερμοκρασία τους, καθώς και ηλεκτρικά ρεύματα και μπαταρίες που φορτίζονται σε ποσοστά μεταξύ 60% και 95%, προσφέρουν δυνατότητες εξοικονόμησης και ευελιξίας. Αυτοί οι ρυθμιστικοί μηχανισμοί επιτρέπουν τη διαχείριση και τον προγραμματισμό της κατανάλωσης ενέργειας σε ευνοϊκές χρονικές περιόδους, όπως όταν παράγεται ηλιακή ενέργεια ή όταν οι τιμές των δυναμικών τιμολογίων είναι χαμηλές.

Η χρήση ηλεκτρικών οχημάτων και οικιακών μπαταριών συμβάλλει επίσης στην εξοικονόμηση ενέργειας. Τα ηλεκτρικά οχήματα καταναλώνουν μεγάλες ποσότητες ηλεκτρικής ενέργειας, αλλά μπορούν να λειτουργήσουν ως απομονωτές ενέργειας. Η σύνδεση του ηλεκτρικού οχήματος ή του φορτιστή EV με το σύστημα διαχείρισης ενέργειας του σπιτιού προσφέρει τη δυνατότητα αξιοποίησης της αποθηκευτικής τους ικανότητας. Το ίδιο ισχύει και για τις οικιακές μπαταρίες, οι οποίες επιτρέπουν την αποθήκευση ενέργειας και τη χρήση της όταν η ζήτηση είναι υψηλή ή οι τιμές αυξημένες. Από την άποψη της ενεργειακής διαχείρισης, τα ηλεκτρικά οχήματα λειτουργούν ουσιαστικά ως κινητές μπαταρίες, προσφέροντας ευελιξία και δυνατότητες εξοικονόμησης. Με αυτόν τον τρόπο, οι ρυθμιστικοί αυτοί μηχανισμοί δίνουν τη δυνατότητα αποτελεσματικής διαχείρισης της ενέργειας, με στόχο τη μείωση του κόστους και τη βελτίωση της ενεργειακής αποδοτικότητας (Khajenasiri et al., 2017).

Ο τύπος του ενεργειακού συμβολαίου διαδραματίζει καθοριστικό ρόλο στο σύστημα διαχείρισης ενέργειας ενός σπιτιού. Με την άνοδο των ανανεώσιμων πηγών ενέργειας, οι τιμές της ενέργειας παρουσιάζουν σημαντικές διακυμάνσεις, ακόμη και μέσα σε μία μόνο ημέρα. Ορισμένα ενεργειακά συμβόλαια, γνωστά ως δυναμικά τιμολόγια, λαμβάνουν υπόψη αυτές τις διακυμάνσεις, χρεώνοντας διαφορετικές τιμές ανά ώρα, οι οποίες συνήθως δημοσιοποιούνται 24 ώρες πριν. Με τη σωστή διαχείριση της ενέργειας, αυτός ο τύπος συμβολαίου μπορεί να αποδειχθεί σημαντικά οικονομικότερος από τα παραδοσιακά συμβόλαια σταθερής ή μεταβλητής τιμής. Παράλληλα, περισσότερες ευρωπαϊκές χώρες υιοθετούν ταχέως αυτού του είδους τις συμβάσεις, υποστηρίζοντας την ενεργειακή μετάβαση. Για όσους έχουν συμβόλαια σταθερού ή μεταβλητού τιμολογίου, η διαχείριση της ενέργειας επικεντρώνεται κυρίως στη μείωση της κατανάλωσης. Αντίθετα, με τα δυναμικά τιμολόγια, η διαχείριση περιλαμβάνει επίσης τον προγραμματισμό ή τη μετατόπιση της κατανάλωσης ενέργειας σε ώρες με χαμηλότερες τιμές. Αυτό επιτρέπει στους καταναλωτές να πληρώνουν λιγότερα για την ίδια

ποσότητα ενέργειας που καταναλώνουν, εκμεταλλευόμενοι στο έπακρο τις χαμηλές τιμές ενέργειας που προσφέρουν αυτού του είδους τα συμβόλαια (Alavi et al., 2018).

Ένα έξυπνο σύστημα διαχείρισης ενέργειας στο σπίτι (SHEMS) είναι μια τεχνολογία που επιτρέπει στους ιδιοκτήτες σπιτιού να διαχειρίζονται αποτελεσματικά την ενεργειακή τους κατανάλωση. Το σύστημα χρησιμοποιεί έξυπνη τεχνολογία για την παρακολούθηση και τον έλεγχο της ποσότητας ενέργειας που χρησιμοποιείται σε ένα σπίτι, παρέχοντας μια πιο βιώσιμη και οικονομικά αποδοτική ενεργειακή λύση. Το SHEMS περιλαμβάνει διάφορα στοιχεία, όπως έξυπνους θερμοστάτες, έξυπνες πρίζες, ενεργειακά αποδοτικές συσκευές και έξυπνα συστήματα φωτισμού. Αυτά τα εξαρτήματα επικοινωνούν μεταξύ τους μέσω ενός κεντρικού κόμβου ή μιας εφαρμογής, παρέχοντας δεδομένα σε πραγματικό χρόνο σχετικά με τη χρήση ενέργειας και βοηθώντας τους ιδιοκτήτες σπιτιού να λαμβάνουν τεκμηριωμένες αποφάσεις σχετικά με την κατανάλωση ενέργειας. Το σύστημα χρησιμοποιεί προηγμένους αλγορίθμους για την ανάλυση ενεργειακών δεδομένων, τον εντοπισμό μοτίβων και τη βελτιστοποίηση της χρήσης ενέργειας. Μπορεί να ρυθμίζει αυτόματα τη θερμοκρασία του σπιτιού με βάση τις εξωτερικές καιρικές συνθήκες ή τα πρότυπα χρήσης, να απενεργοποιεί τις συσκευές όταν δεν χρησιμοποιούνται και να μεταβαίνει σε ανανεώσιμες πηγές ενέργειας όταν είναι διαθέσιμες (Nadezhda, 2024).

Το SHEMS επιτρέπει επίσης στους ιδιοκτήτες σπιτιού να ορίζουν προσαρμοσμένα χρονοδιαγράμματα εξοικονόμησης ενέργειας και να λαμβάνουν ειδοποιήσεις όταν η χρήση ενέργειας υπερβαίνει ένα καθορισμένο όριο. Αυτή η λειτουργία συμβάλλει στην πρόληψη της σπατάλης και διασφαλίζει ότι οι λογαριασμοί ενέργειας είναι πάντα υπό έλεγχο.

- Αυξημένη υιοθέτηση: Τα έξυπνα συστήματα ενεργειακής διαχείρισης σπιτιού θα γίνουν πιο διαδεδομένα καθώς η τεχνολογία συνεχίζει να βελτιώνεται και περισσότεροι άνθρωποι συνειδητοποιούν τα οφέλη των ενεργειακά αποδοτικών σπιτιών.
- Ενσωμάτωση της τεχνητής νοημοσύνης (AI): Τα συστήματα έξυπνης οικιακής ενεργειακής διαχείρισης με τεχνητή νοημοσύνη θα γίνουν πιο δημοφιλή, καθώς θα επιτρέπουν στους ιδιοκτήτες σπιτιών να βελτιστοποιούν την ενεργειακή τους κατανάλωση και να εξοικονομούν χρήματα.
- Φωνητικός έλεγχος: Τα συστήματα διαχείρισης ενέργειας έξυπνων σπιτιών θα ενσωματωθούν με φωνητικούς βοηθούς, όπως το Amazon Alexa και το Google Home, διευκολύνοντας τους ιδιοκτήτες σπιτιών να ελέγχουν την ενεργειακή τους κατανάλωση.
- Ενσωμάτωση του Διαδικτύου των Πραγμάτων (IoT): Τα έξυπνα συστήματα διαχείρισης ενέργειας στο σπίτι θα ενσωματωθούν περισσότερο με άλλες συσκευές IoT, όπως έξυπνοι θερμοστάτες, έξυπνος φωτισμός και έξυπνες συσκευές.
- Ενσωμάτωση των ανανεώσιμων πηγών ενέργειας: Τα έξυπνα συστήματα διαχείρισης ενέργειας στο σπίτι θα χρησιμοποιούν πηγές ενέργειας, όπως ηλιακούς συλλέκτες και

ανεμογεννήτριες, επιτρέποντας στους ιδιοκτήτες σπιτιού να παράγουν τη δική τους καθαρή ενέργεια και να μειώσουν την εξάρτησή τους από τις παραδοσιακές πηγές ενέργειας.

- Ανάλυση δεδομένων: Τα έξυπνα συστήματα ενεργειακής διαχείρισης σπιτιού θα είναι σχεδιασμένα για να συλλέγουν και να αναλύουν δεδομένα σχετικά με τα πρότυπα κατανάλωσης ενέργειας, επιτρέποντας στους ιδιοκτήτες σπιτιού να εντοπίζουν περιοχές όπου μπορούν να μειώσουν την κατανάλωση ενέργειας και να εξοικονομήσουν χρήματα.
- Αποθήκευση ενέργειας: Τα έξυπνα συστήματα ενεργειακής διαχείρισης του σπιτιού θα σχεδιάζονται έτσι ώστε να ενσωματώνουν λύσεις αποθήκευσης ενέργειας, όπως μπαταρίες, επιτρέποντας στους ιδιοκτήτες σπιτιού να αποθηκεύουν την περίσσεια ενέργειας και να τη χρησιμοποιούν σε περιόδους υψηλής ζήτησης ή όταν οι παραδοσιακές πηγές ενέργειας δεν είναι διαθέσιμες.
- Ενσωμάτωση στο δίκτυο: Τα έξυπνα συστήματα διαχείρισης ενέργειας στο σπίτι θα σχεδιαστούν για να ενσωματώνονται στο ευρύτερο ενεργειακό δίκτυο, επιτρέποντας στους ιδιοκτήτες σπιτιού να πωλούν την πλεονάζουσα ενέργεια πίσω στο δίκτυο και να κερδίζουν χρήματα για την παραγωγή ενέργειας.
- Διαχείριση ενέργειας ως υπηρεσία: Τα έξυπνα συστήματα ενεργειακής διαχείρισης σπιτιού θα προσφέρονται ως υπηρεσία, επιτρέποντας στους ιδιοκτήτες σπιτιού να πληρώνουν μηνιαία αμοιβή για την εγκατάσταση, τη συντήρηση και την παρακολούθηση του συστήματος ενεργειακής διαχείρισης.
- Εξατομίκευση: Τα έξυπνα συστήματα ενεργειακής διαχείρισης σπιτιού θα είναι σχεδιασμένα ώστε να εξατομικεύουν την κατανάλωση ενέργειας με βάση τις ατομικές προτιμήσεις και συνήθειες, επιτρέποντας στους ιδιοκτήτες σπιτιού να εξοικονομούν χρήματα, διατηρώντας παράλληλα το επιθυμητό επίπεδο άνεσης.

2.6 Ασφάλεια έξυπνων σπιτιών - Κίνδυνοι και λύσεις

Τα έξυπνα σπίτια, που κάποτε αποτελούσαν πεδίο επιστημονικής φαντασίας, έχουν γίνει αναπόσπαστο μέρος του σύγχρονου τρόπου ζωής. Αυτές οι τεχνολογικά προηγμένες κατοικίες διαθέτουν διασυνδεδεμένες συσκευές και συστήματα που έχουν σχεδιαστεί για να ενισχύουν την ευκολία, την άνεση και την ασφάλεια. Από θερμοστάτες και ελέγχους φωτισμού μέχρι κάμερες ασφαλείας και βοηθούς που ενεργοποιούνται με φωνή, η τεχνολογία έξυπνων σπιτιών προσφέρει πρωτοφανή επίπεδα αυτοματισμού και ελέγχου. Ωστόσο, παράλληλα με την ευκολία και την καινοτομία, υπάρχουν και σημαντικές ανησυχίες για την ασφάλεια στον κυβερνοχώρο. Καθώς οι έξυπνες οικιακές συσκευές γίνονται όλο και πιο διαδεδομένες, γίνονται επίσης πιο ελκυστικοί στόχοι για κυβερνοεπιθέσεις. Οι κακόβουλοι φορείς εκμεταλλεύονται τα τρωτά σημεία αυτών των διασυνδεδεμένων συστημάτων για να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση, να θέσουν σε κίνδυνο προσωπικά δεδομένα και ακόμη και να χειραγωγήσουν φυσικές συσκευές μέσα στο σπίτι (Gatsis et al., 2017).

Τα έξυπνα σπίτια ορίζονται ως βολικές οικιακές ρυθμίσεις όπου οι συσκευές και οι συσκευές μπορούν να ελέγχονται αυτόματα από απόσταση από οπουδήποτε με σύνδεση στο διαδίκτυο χρησιμοποιώντας ένα κινητό ή άλλη δικτυωμένη συσκευή. Οι συσκευές σε ένα έξυπνο σπίτι συνδέονται μεταξύ τους μέσω του διαδικτύου, επιτρέποντας στον χρήστη να ελέγχει λειτουργίες όπως η πρόσβαση ασφαλείας στο σπίτι, η θερμοκρασία, ο φωτισμός και ένα οικιακό σινεμά από απόσταση. Αυτό σημαίνει ότι σε αντίθεση με τα παραδοσιακά σπίτια, όπου τα φυσικά εμπόδια και οι κλειδαριές αποτελούν το κύριο μέσο προστασίας, τα έξυπνα σπίτια βασίζονται σε ψηφιακά μέτρα ασφαλείας για την άμυνα κατά των απειλών στον κυβερνοχώρο. Στην πλειονότητα των οικιακών συστημάτων ασφαλείας, όλα τα εξαρτήματα συνδέονται σε έναν σταθμό βάσης και στην συνέχεια σε μια εφαρμογή για κινητά, επιτρέποντάς την παρακολούθηση και τον έλεγχο από απόσταση.

Οι έξυπνες οικιακές συσκευές, όπως οι έξυπνες κλειδαριές, οι κάμερες και οι θερμοστάτες, είναι ευάλωτες σε διάφορες απειλές στον κυβερνοχώρο, όπως κακόβουλο λογισμικό, πειρατεία και μη εξουσιοδοτημένη πρόσβαση. Οι κατασκευαστές πρέπει να εφαρμόζουν ισχυρά μέτρα ασφαλείας, όπως κρυπτογράφηση, έλεγχο ταυτότητας και ενημερώσεις υλικού και λογισμικού, για να μετριάσουν αυτούς τους κινδύνους. Η υποδομή δικτύου που συνδέει τις συσκευές έξυπνου σπιτιού διαδραματίζει κρίσιμο ρόλο στη συνολική ασφάλεια. Τα δίκτυα WiFi, οι δρομολογητές και οι πύλες πρέπει να είναι κατάλληλα διαμορφωμένα και ασφαλισμένα ώστε να αποτρέπεται η μη εξουσιοδοτημένη πρόσβαση και η υποκλοπή δεδομένων. Η εφαρμογή ισχυρών κωδικών πρόσβασης, η ενεργοποίηση της κρυπτογράφησης δικτύου και η τακτική ενημέρωση του υλικό και λογισμικό αποτελούν βασικές πρακτικές για τη διατήρηση της ασφάλειας του δικτύου. Οι έξυπνες οικιακές συσκευές συλλέγουν και μεταδίδουν τεράστιες ποσότητες ευαίσθητων δεδομένων, συμπεριλαμβανομένων των προτιμήσεων του χρήστη, των προτύπων συμπεριφοράς και των στατιστικών στοιχείων χρήσης της συσκευής. Η προστασία αυτών των δεδομένων από μη εξουσιοδοτημένη πρόσβαση και εκμετάλλευση είναι υψίστης σημασίας για τη διατήρηση της ιδιωτικότητας των χρηστών. Οι κατασκευαστές θα πρέπει να εφαρμόζουν ισχυρή κρυπτογράφηση δεδομένων, ασφαλείς πρακτικές αποθήκευσης δεδομένων και διαφανείς πολιτικές απορρήτου για την προστασία των πληροφοριών των χρηστών. Η κατανόηση αυτών των θεμελιωδών πτυχών της ασφάλειας του έξυπνου σπιτιού είναι απαραίτητη τόσο για τους ιδιοκτήτες σπιτιού όσο και για τους κατασκευαστές. Θέτοντας ως προτεραιότητα την ασφάλεια των συσκευών, την ασφάλεια του δικτύου και το απόρρητο των δεδομένων, τα άτομα μπορούν να απολαμβάνουν τα οφέλη της τεχνολογίας έξυπνου σπιτιού, ελαχιστοποιώντας παράλληλα τους σχετικούς κινδύνους (Weber, 2010).

Τα έξυπνα σπίτια, εξοπλισμένα με διασυνδεδεμένες συσκευές και συστήματα, μπορούν να κάνουν την καθημερινότητα ευκολότερη, αλλά ταυτόχρονα εξακολουθούν να είναι ευάλωτα σε διάφορους κινδύνους και τρωτά σημεία κυβερνοασφάλειας, ιδίως όσον αφορά την προστασία

των δεδομένων. Η κατανόηση αυτών των απειλών είναι ζωτικής σημασίας τόσο για τους ιδιοκτήτες σπιτιών όσο και για τους κατασκευαστές, ώστε να μετριάσουν τις πιθανές παραβιάσεις. Οι έξυπνες οικιακές συσκευές συχνά δεν διαθέτουν ισχυρά χαρακτηριστικά ασφαλείας, με αποτέλεσμα να είναι ευάλωτες στην εκμετάλλευση από εγκληματίες του κυβερνοχώρου. Οι αδύναμοι μηχανισμοί ελέγχου ταυτότητας, όπως οι προεπιλεγμένοι κωδικοί πρόσβασης ή καθόλου έλεγχος ταυτότητας, δημιουργούν σημεία εισόδου για μη εξουσιοδοτημένη πρόσβαση. Επιπλέον, τα μη κρυπτογραφημένα πρωτόκολλα επικοινωνίας που χρησιμοποιούνται από αυτές τις συσκευές εκθέτουν ευαίσθητα δεδομένα σε υποκλοπή και χειραγώγηση. Το παρωχημένο υλικολογισμικό είναι ένας άλλος σημαντικός κίνδυνος, καθώς μπορεί να περιέχει γνωστές ευπάθειες που οι επιτιθέμενοι μπορούν να εκμεταλλευτούν για να θέσουν σε κίνδυνο τις συσκευές. Τα οικοσυστήματα έξυπνου σπιτιού αποτελούνται από ένα ευρύ φάσμα συσκευών από διαφορετικούς κατασκευαστές, το καθένα με τα δικά του πρότυπα και πρωτόκολλα ασφαλείας. Αυτή η ποικιλομορφία μπορεί να οδηγήσει σε προκλήσεις ενσωμάτωσης, καθώς οι ανεπαρκώς ενσωματωμένες συσκευές μπορεί να εισάγουν ευπάθειες ασφαλείας. Επιπλέον, οι ασυνεπείς πρακτικές ασφάλειας μεταξύ των κατασκευαστών μπορεί να οδηγήσουν σε συσκευές με διαφορετικά επίπεδα προστασίας, καθιστώντας ολόκληρο το οικοσύστημα ευάλωτο σε παραβιάσεις.

Αρκετές αξιοσημείωτες παραβιάσεις ασφαλείας σε έξυπνα σπίτια έχουν αναδείξει τη σοβαρότητα αυτών των ευπαθειών. Περιστατικά όπου χάκερς απέκτησαν πρόσβαση σε έξυπνες κλειδαριές θυρών ή κάμερες παρακολούθησης και τις αλλοίωσαν, εγείρουν ανησυχίες σχετικά με την ιδιωτικότητα και τη φυσική ασφάλεια. Ομοίως, επιθέσεις με στόχο έξυπνους οικιακούς κόμβους ή βοηθούς που ελέγχονται με φωνή έχουν επίσης καταδείξει τη δυνατότητα εκτεταμένης διαταραχής και παραβίασης δεδομένων. Τα τυπικά έξυπνα σπίτια είναι ευάλωτα σε μεγάλο αριθμό επιθέσεων δεδομένων. Οι αναφερθείσες περιπτώσεις επιθέσεων σε έξυπνα σπίτια περιλαμβάνουν χάκερς που ελέγχουν εξ αποστάσεως έξυπνα φώτα και έξυπνες τηλεοράσεις, ξεκλειδώνουν πόρτες με δυνατότητα IoT και ενεργοποιούν εξ αποστάσεως και μεταδίδουν βίντεο από έξυπνες κάμερες. Η αντιμετώπιση αυτών των κινδύνων και των τρωτών σημείων απαιτεί προληπτικά μέτρα ασφαλείας, όπως τακτικές ενημερώσεις λογισμικού, ισχυρούς μηχανισμούς ελέγχου ταυτότητας και τμηματοποίηση του δικτύου. Με την εφαρμογή αυτών και άλλων μέτρων ασφαλείας, οι ιδιοκτήτες σπιτιού και οι κατασκευαστές μπορούν να ενισχύσουν την ασφάλεια των έξυπνων οικιακών περιβαλλόντων και να προστατευθούν από απειλές στον κυβερνοχώρο. Οι ανησυχίες για την προστασία της ιδιωτικής ζωής είναι μεγάλες στο πεδίο των έξυπνων οικιακών συσκευών, εγείροντας θεμελιώδη ερωτήματα σχετικά με τη συλλογή, τη χρήση και την προστασία των δεδομένων. Ένα από τα πρωταρχικά ζητήματα περιστρέφεται γύρω από τις εκτεταμένες πρακτικές συλλογής δεδομένων που χρησιμοποιούν οι πάροχοι τεχνολογίας έξυπνου σπιτιού. Οι συσκευές αυτές συγκεντρώνουν πληθώρα ευαίσθητων πληροφοριών, όπως ηχογραφήσεις και βιντεοσκοπήσεις, μοτίβα χρήσης της

συσκευής και προσωπικά στοιχεία, συχνά χωρίς τη ρητή συγκατάθεση του χρήστη (Heer et al., 2011).

Ο διάχυτος χαρακτήρας της συλλογής δεδομένων εγείρει σημαντικές επιπτώσεις στην προστασία της ιδιωτικής ζωής. Οι χρήστες μπορεί να αισθάνονται άβολα γνωρίζοντας ότι οι πιο προσωπικές τους στιγμές και δραστηριότητες μέσα στο σπίτι τους καταγράφονται και αναλύονται από έξυπνες συσκευές. Επιπλέον, ο κίνδυνος παραβίασης δεδομένων και μη εξουσιοδοτημένης πρόσβασης σε αυτές τις πληροφορίες αποτελεί σοβαρή απειλή για την ιδιωτικότητα και την ασφάλεια των ατόμων. Σύμφωνα με έρευνες ένα σημαντικό ποσοστό των καταναλωτών εξέφρασε ανησυχία ή έντονη ανησυχία σχετικά με την ασφάλεια των προσωπικών τους δεδομένων που συλλέγονται και μεταδίδονται από έξυπνες οικιακές συσκευές. Ομοίως, ανησυχούν εξίσου για την πιθανή μη εξουσιοδοτημένη πρόσβαση ή τον έλεγχο των έξυπνων συσκευών χωρίς την άδειά τους. Ενώ οι κατασκευαστές έξυπνων οικιακών συσκευών και πλατφορμών συλλέγουν δεδομένα των καταναλωτών για να προσαρμόσουν καλύτερα τα προϊόντα τους ή να προσφέρουν νέες και βελτιωμένες υπηρεσίες στους πελάτες, η εμπιστοσύνη και η διαφάνεια είναι ζωτικής σημασίας για τους κατασκευαστές που επιθυμούν να κερδίσουν νέους πελάτες. Οι παραβιάσεις του απορρήτου σε περιβάλλοντα έξυπνου σπιτιού μπορεί να έχουν εκτεταμένες συνέπειες. Η μη εξουσιοδοτημένη πρόσβαση σε καταγραφές ήχου και βίντεο μπορεί να θέσει σε κίνδυνο το προσωπικό απόρρητο και να οδηγήσει σε κλοπή ταυτότητας ή εκβιασμό. Επιπλέον, η έκθεση ευαίσθητων δεδομένων σε κακόβουλους φορείς μπορεί να οδηγήσει σε ζημία φήμης, οικονομική απώλεια και συναισθηματική οδύνη για τα επηρεαζόμενα άτομα. Επιπλέον, ο διασυνδεδεμένος χαρακτήρας των οικοσυστημάτων έξυπνου σπιτιού ενισχύει τις ανησυχίες για την προστασία της ιδιωτικής ζωής. Τα δεδομένα που συλλέγονται από μια συσκευή ενδέχεται να κοινοποιούνται σε άλλες συσκευές ή υπηρεσίες τρίτων, αυξάνοντας τις πιθανότητες για κατάχρηση ή εκμετάλλευση δεδομένων. Αυτή η έλλειψη διαφάνειας και ελέγχου των πρακτικών ανταλλαγής δεδομένων επιδεινώνει τους κινδύνους για την προστασία της ιδιωτικής ζωής και υπονομεύει την εμπιστοσύνη στις τεχνολογίες έξυπνου σπιτιού (Kothmayr et al., 2013).

Η αντιμετώπιση των ανησυχιών για την προστασία της ιδιωτικής ζωής σε περιβάλλοντα έξυπνου σπιτιού απαιτεί μια πολύπλευρη προσέγγιση. Περιλαμβάνει την εφαρμογή ισχυρών μέτρων προστασίας δεδομένων, όπως η κρυπτογράφηση και οι έλεγχοι πρόσβασης, για τη διασφάλιση των ευαίσθητων πληροφοριών. Επιπλέον, οι διαφανείς πολιτικές προστασίας της ιδιωτικής ζωής και οι μηχανισμοί συναίνεσης των χρηστών είναι ουσιώδεις για να διασφαλιστεί ότι τα άτομα έχουν πλήρη ορατότητα και έλεγχο των δεδομένων τους. Δίνοντας προτεραιότητα στην προστασία της ιδιωτικής ζωής παράλληλα με τη λειτουργικότητα και την ευκολία, οι κατασκευαστές έξυπνων κατοικιών μπορούν να καλλιεργήσουν μεγαλύτερη εμπιστοσύνη και σιγουριά μεταξύ των καταναλωτών. Η διασφάλιση των συσκευών και των δικτύων έξυπνου σπιτιού είναι υψίστης σημασίας για τον μετριασμό των απειλών κυβερνοασφάλειας στο ολοένα

και πιο συνδεδεμένο τοπίο των σύγχρονων σπιτιών. Η εφαρμογή βέλτιστων πρακτικών μπορεί να βελτιώσει σημαντικά την κατάσταση ασφαλείας των περιβαλλόντων έξυπνων σπιτιών. Μια τέτοια πρακτική είναι η διασφάλιση ότι όλες οι συσκευές είναι εξοπλισμένες με τις πιο πρόσφατες ενημερώσεις και επιδιορθώσεις ασφαλείας για την αποτελεσματική αντιμετώπιση των γνωστών ευπαθειών. Επιπλέον, οι χρήστες θα πρέπει να χρησιμοποιούν ισχυρούς, μοναδικούς κωδικούς πρόσβασης για κάθε έξυπνη συσκευή και δίκτυο Wi-Fi για την αποτροπή μη εξουσιοδοτημένης πρόσβασης. Η χρήση του ελέγχου ταυτότητας πολλαπλών παραγόντων (MFA) προσθέτει ένα επιπλέον επίπεδο ασφάλειας απαιτώντας πρόσθετη επαλήθευση πέραν του κωδικού πρόσβασης, ενισχύοντας περαιτέρω τις άμυνες του έξυπνου σπιτιού.

Η τακτική παρακολούθηση της κυκλοφορίας του δικτύου και της δραστηριότητας των συσκευών μπορεί επίσης να βοηθήσει στον εντοπισμό ύποπτης συμπεριφοράς που υποδηλώνει πιθανή παραβίαση της ασφάλειας. Τα συστήματα ανίχνευσης εισβολών (IDS) και τα συστήματα πρόληψης εισβολών (IPS) μπορούν να εντοπίζουν και να ανταποκρίνονται αυτόματα σε ανώμαλες δραστηριότητες, ενισχύοντας τη συνολική κατάσταση ασφαλείας των έξυπνων οικιακών δικτύων. Επιπλέον, οι χρήστες θα πρέπει να είναι προσεκτικοί όταν χορηγούν δικαιώματα και μοιράζονται δεδομένα με συσκευές και εφαρμογές έξυπνου σπιτιού. Ο περιορισμός των περιττών αδειών και της κοινής χρήσης δεδομένων μπορεί να συμβάλει στην ελαχιστοποίηση του κινδύνου μη εξουσιοδοτημένης πρόσβασης και έκθεσης δεδομένων. Κατά την επιλογή προϊόντων έξυπνου σπιτιού, οι καταναλωτές θα πρέπει να δίνουν προτεραιότητα στα χαρακτηριστικά ασφαλείας και να επιλέγουν αξιόπιστες μάρκες με ιστορικό προτεραιότητας στην ασφάλεια. Επιπλέον, η διαμόρφωση των συσκευών ώστε να λειτουργούν με ασφαλή τρόπο, όπως η απενεργοποίηση περιττών λειτουργιών και υπηρεσιών, μπορεί να μειώσει περαιτέρω την επιφάνεια επίθεσης και να ενισχύσει τη συνολική ασφάλεια (Olivier et al., 2015).

3. Οπτικοποίηση των δεδομένων από το IoT

3.1 Οπτικοποίηση δεδομένων για το IoT: Μετασχηματισμός σε αξιοποιήσιμες γνώσεις

Η διαδικασία μετατροπής τεράστιων όγκων δεδομένων IoT σε γραφικές απεικονίσεις, όπως πίνακες οργάνων, γραφήματα, διαγράμματα και χάρτες, είναι γνωστή ως οπτικοποίηση IoT. Μέσω της οπτικοποίησης IoT, οι ενδιαφερόμενοι μπορούν να παρακολουθούν την απόδοση του συστήματος, να εντοπίζουν ανωμαλίες, να αποκτούν διορατικές πληροφορίες και να λαμβάνουν αποφάσεις βάσει δεδομένων. Η ανάπτυξη του IoT έχει οδηγήσει σε πληθώρα δεδομένων, δημιουργώντας προκλήσεις στη διαχείριση, επεξεργασία και κατανόησή τους. Σε αντίθεση με τις παραδοσιακές πηγές δεδομένων, τα δεδομένα IoT είναι συνήθως μεγάλου όγκου, σε πραγματικό χρόνο και ποικίλης φύσης. Περιλαμβάνουν τα πάντα, από μετρήσεις

θερμοκρασίας και αισθητήρες κίνησης έως τροφοδοσίες βίντεο και δεδομένα γεωγραφικού εντοπισμού. Για να δώσουν νόημα σε αυτά τα δεδομένα, οι οργανισμοί πρέπει να τα οργανώσουν σε μορφές που είναι εύκολα ερμηνεύσιμες και αξιοποιήσιμες.

Η οπτικοποίηση δεδομένων στο IoT είναι κάτι περισσότερο από την απλή παρουσίαση πληροφοριών - πρόκειται για την αποκάλυψη μοτίβων, τάσεων και ακραίων τιμών που οδηγούν στη λήψη αποφάσεων. Είτε πρόκειται για τη βελτιστοποίηση των λειτουργιών, την παρακολούθηση των περιουσιακών στοιχείων ή την πρόβλεψη των αναγκών συντήρησης, η οπτικοποίηση των δεδομένων IoT επιτρέπει στις εταιρείες να μετατρέψουν τους ακατέργαστους αριθμούς σε στρατηγικές γνώσεις. Πραγματοποιείται η κατανόηση σύνθετων δεδομένων από διάφορες πηγές, όπως αισθητήρες, συσκευές και δίκτυα. Η οπτικοποίηση βοηθά στην κατανόηση αυτών των σύνθετων δεδομένων παρέχοντας πληροφορίες και μοτίβα που μπορεί να μην είναι εμφανή από τα ακατέργαστα δεδομένα. Οι εφαρμογές IoT συχνά απαιτούν παρακολούθηση των ροών δεδομένων σε πραγματικό χρόνο. Οι πίνακες οπτικοποίησης επιτρέπουν στους χρήστες να παρακολουθούν ταυτόχρονα πολλαπλές παραμέτρους και να εντοπίζουν γρήγορα τυχόν ανωμαλίες ή τάσεις. Η οπτική αναπαράσταση των δεδομένων IoT επιτρέπει στους ενδιαφερόμενους να λαμβάνουν γρήγορα τεκμηριωμένες αποφάσεις. Είτε πρόκειται για τη βελτιστοποίηση των λειτουργιών, την πρόβλεψη των αναγκών συντήρησης ή τη βελτίωση της αποδοτικότητας, η οπτικοποίηση βοηθά στην ερμηνεία των δεδομένων για τη λήψη καλύτερων αποφάσεων. Τα εργαλεία οπτικοποίησης επιτρέπουν στους χρήστες να εντοπίζουν μοτίβα, τάσεις και συσχετίσεις εντός των δεδομένων IoT. Αυτό μπορεί να είναι πολύτιμο για την προγνωστική ανάλυση, την ανίχνευση ανωμαλιών και την πρόβλεψη. Η οπτικοποίηση παρέχει μια κοινή γλώσσα για την επικοινωνία των πληροφοριών και των ευρημάτων σε διαφορετικές ομάδες εντός ενός οργανισμού. Βοηθά στη γεφύρωση του χάσματος μεταξύ τεχνικών και μη τεχνικών ενδιαφερομένων με την παρουσίαση πληροφοριών σε διαισθητική και προσιτή μορφή. Οι διαδραστικές οπτικοποιήσεις εμπλέκουν τους χρήστες πιο αποτελεσματικά από ό,τι οι στατικές αναφορές ή τα λογιστικά φύλλα. Δίνουν τη δυνατότητα στους χρήστες να εξερευνήσουν τα δεδομένα, να εμβαθύνουν σε λεπτομέρειες και να αποκτήσουν βαθύτερες γνώσεις, οδηγώντας σε καλύτερη δέσμευση και κατανόηση. Με την οπτικοποίηση δεδομένων IoT που σχετίζονται με τη χρήση πόρων, όπως η κατανάλωση ενέργειας, η χρήση νερού ή η χρήση του εξοπλισμού, οι οργανισμοί μπορούν να εντοπίσουν ανεπάρκειες και ευκαιρίες βελτιστοποίησης, οδηγώντας σε εξοικονόμηση κόστους και οφέλη από τη βιωσιμότητα (Sumit at al., 2017).

3.2 Πηγές δεδομένων για οπτικοποίηση στο IoT

Τα δεδομένα IoT προέρχονται από πολλές διαφορετικές πηγές. Οι αισθητήρες και οι συσκευές του Διαδικτύου των Πραγμάτων (IoT) καταγράφουν ποικίλα δεδομένα, όπως θερμοκρασία, υγρασία, πίεση, ταχύτητα και τοποθεσία. Αυτοί οι αισθητήρες ενσωματώνονται σε πολλά

διαφορετικά προϊόντα, συμπεριλαμβανομένης της φορητής τεχνολογίας, του βιομηχανικού εξοπλισμού και των έξυπνων οικιακών συσκευών. Δεδομένα σχετικά με τη συνδεσιμότητα, την ισχύ του σήματος, τις ταχύτητες μεταφοράς δεδομένων και την απόδοση του δικτύου παράγονται από τα δίκτυα του διαδικτύου των πραγμάτων. Κάνοντας αυτά τα δεδομένα οπτικά ελκυστικά, μπορεί να βελτιωθεί η απόδοση του δικτύου και να εντοπιστούν τυχόν σημεία συμφόρησης ή περιοχές με κακή κάλυψη. Για την αποθήκευση και την ανάλυση δεδομένων, οι πλατφόρμες νέφους χρησιμοποιούνται σε πολλές υλοποιήσεις του Διαδικτύου των πραγμάτων. Οι χρήστες μπορούν να αναλύουν και να οπτικοποιούν δεδομένα σε πραγματικό χρόνο χρησιμοποιώντας εργαλεία οπτικοποίησης που συνδυάζονται με πλατφόρμες cloud, δίνοντάς τους μια λεπτομερή κατανόηση των επιδόσεων των συσκευών και των μοτίβων δεδομένων. Συχνά παραδίδονται ροές δεδομένων IoT σε πραγματικό χρόνο και η θέαση αυτών των δεδομένων τη στιγμή που συμβαίνουν μπορεί να προσφέρει γρήγορες πληροφορίες. Οι ζωντανοί πίνακες οργάνων, για παράδειγμα, μπορούν να δείχνουν τους τρέχοντες αριθμούς, να ειδοποιούν τους χρήστες για ανωμαλίες και να διευκολύνουν τη γρήγορη λήψη αποφάσεων (Mahantesh et al., 2017).

3.3 Βασικά οφέλη της οπτικοποίησης δεδομένων για το IoT

Οι οπτικές αναπαραστάσεις των δεδομένων βοηθούν τους υπεύθυνους λήψης αποφάσεων να κατανοήσουν γρήγορα τις πολύπλοκες πληροφορίες του IoT, διευκολύνοντας τον εντοπισμό τάσεων και συσχετίσεων. Για παράδειγμα, οι διαχειριστές μπορούν να παρακολουθούν μετρήσεις σε πραγματικό χρόνο από αισθητήρες IoT σε ένα πίνακα ελέγχου, επιτρέποντας τη λήψη γρήγορων, τεκμηριωμένων αποφάσεων σχετικά με τους πόρους και τις λειτουργίες. Η οπτικοποίηση δεδομένων βοηθά στον εντοπισμό της αναποτελεσματικότητας. Στην κατασκευή, για παράδειγμα, τα οπτικοποιημένα δεδομένα IoT από αισθητήρες μηχανημάτων μπορούν να βοηθήσουν στον εντοπισμό σημείων συμφόρησης ή πιθανών βλαβών πριν από την εμφάνισή τους, οδηγώντας σε βελτιωμένη παραγωγικότητα και μειωμένο χρόνο διακοπής λειτουργίας. Με την εφαρμογή προγνωστικών αναλύσεων και την εμφάνιση τάσεων στην υγεία του εξοπλισμού, τα εργαλεία οπτικοποίησης δεδομένων μπορούν να προβλέψουν πότε τα μηχανήματα είναι πιθανό να αποτύχουν. Αυτή η προληπτική προσέγγιση επιτρέπει στους οργανισμούς να εκτελούν συντήρηση μόνο όταν χρειάζεται, εξοικονομώντας τόσο χρόνο όσο και χρήματα. Τα εργαλεία οπτικοποίησης δεδομένων IoT μπορούν να παρέχουν ζωντανή παρακολούθηση των περιουσιακών στοιχείων, των πόρων και των περιβαλλοντικών συνθηκών. Οι πίνακες ελέγχου σε πραγματικό χρόνο επιτρέπουν στις εταιρείες να παρακολουθούν τις μετρήσεις καθώς συμβαίνουν, επιτρέποντας ταχύτερες αντιδράσεις σε πιθανά προβλήματα. Στις εφαρμογές IoT που απευθύνονται στον πελάτη - όπως τα έξυπνα οικιακά συστήματα ή οι συσκευές παρακολούθησης της υγείας- τα οπτικά δεδομένα βοηθούν τους τελικούς χρήστες να κατανοήσουν την κατάσταση και τη

δραστηριότητα της συσκευής τους. Παρουσιάζοντας τα δεδομένα με σαφή, προσιτό τρόπο, οι οργανισμοί μπορούν να παρέχουν καλύτερες εμπειρίες στους χρήστες και να βελτιώνουν τη δέσμευση.

3.4 Τεχνικές οπτικοποίησης IoT

Διάφορες μέθοδοι οπτικοποίησης χρησιμοποιούνται συνήθως για την ανάλυση και την κατανόηση των δεδομένων του Διαδικτύου των Πραγμάτων (IoT). Η οπτικοποίηση χρονοσειρών που περιλαμβάνει τη χρήση θερμικών χαρτών, διαγραμμάτων γραμμών και διαγραμμάτων περιοχής για την ανάλυση μοτίβων δεδομένων με την πάροδο του χρόνου. Για παράδειγμα, ένα γραμμικό διάγραμμα μπορεί να χρησιμοποιηθεί για την παρακολούθηση των μεταβολών της θερμοκρασίας με την πάροδο του χρόνου για τον εντοπισμό τάσεων ή ανωμαλιών. Η γεωχωρική οπτικοποίηση περιλαμβάνει την απεικόνιση δεδομένων IoT σε χάρτες για την απόκτηση πληροφοριών με βάση την τοποθεσία. Εργαλεία όπως η χαρτογράφηση GIS, οι χάρτες choropleth και οι χάρτες θερμότητας μπορούν να χρησιμοποιηθούν για την απεικόνιση της κάλυψης δικτύου, της πυκνότητας συσκευών και της κατανομής δεδομένων σε διάφορες τοποθεσίες. Οι πίνακες ελέγχου παρέχουν μια ενοποιημένη προβολή για την παρακολούθηση σημαντικών μετρήσεων και βασικών δεικτών απόδοσης. Συνδυάζουν πολλαπλές απεικονίσεις σε μια ενιαία οθόνη, διευκολύνοντας την παρακολούθηση των δεδομένων IoT και τη λήψη τεκμηριωμένων αποφάσεων. Η προσθήκη διαδραστικότητας στην οπτικοποίηση δεδομένων IoT επιτρέπει στους χρήστες να εξερευνήσουν τα δεδομένα με μεγαλύτερη λεπτομέρεια. Χαρακτηριστικά όπως οι δυνατότητες διάτρησης, οι επιλογές φιλτραρίσματος και οι πληροφορίες με το πέρασμα από πάνω επιτρέπουν στους χρήστες να αποκαλύπτουν κρυμμένα μοτίβα ή συσχετίσεις. Η αποτελεσματική οπτικοποίηση δεδομένων IoT απαιτεί ενδελεχή κατανόηση των πηγών δεδομένων, του περιβάλλοντος στο οποίο παράγονται και των συγκεκριμένων πληροφοριών που πρέπει να εξαχθούν. Με την επιλογή των σωστών προσεγγίσεων και εργαλείων οπτικοποίησης, οι οργανισμοί μπορούν να μεγιστοποιήσουν το δυναμικό των δεδομένων IoT και να επιτύχουν αυξημένη επιχειρησιακή αποδοτικότητα, δημιουργική επίλυση προβλημάτων και βελτιωμένες δυνατότητες λήψης αποφάσεων (Zanella et al., 2014).

3.5 Εργαλεία για την οπτικοποίηση δεδομένων IoT

Ακολουθεί μια λίστα με τα εργαλεία και τεχνολογίες για την οπτικοποίηση δεδομένων IoT.

1. **Tableau:** Ένα ευρέως χρησιμοποιούμενο εργαλείο οπτικοποίησης δεδομένων που παρέχει εκτεταμένες δυνατότητες για την οπτικοποίηση δεδομένων IoT, συμπεριλαμβανομένων διαδραστικών πινάκων ελέγχου και χαρτών.
2. **Power BI:** Ολοκληρωμένες δυνατότητες ανάλυσης και οπτικοποίησης δεδομένων προσφέρει το Microsoft Power BI, το οποίο ενσωματώνεται απρόσκοπτα με άλλα προϊόντα και υπηρεσίες της Microsoft.

3. **ThingSpeak:** Το ThingSpeak είναι μια πλατφόρμα ανάλυσης IoT που επιτρέπει στους χρήστες να συγκεντρώνουν, να προβάλλουν και να αξιολογούν δεδομένα σε πραγματικό χρόνο από αισθητήρες και συσκευές.
4. **InfluxDB:** Μια βάση δεδομένων χρονοσειρών ανοιχτού κώδικα που ονομάζεται InfluxDB έχει ενσωματωμένες δυνατότητες οπτικοποίησης δεδομένων, ειδικά φτιαγμένη για δεδομένα πραγματικού χρόνου και δεδομένα του Διαδικτύου των πραγμάτων.
5. **Grafana:** Μια ισχυρή πλατφόρμα ανάλυσης και οπτικοποίησης με διαμορφώσιμους πίνακες ελέγχου που λειτουργεί με μια ποικιλία πηγών δεδομένων, συμπεριλαμβανομένων των συσκευών IoT.

3.6 Εφαρμογές της οπτικοποίησης IoT

Οι εφαρμογές για την οπτικοποίηση IoT είναι πολλές και καλύπτουν διάφορους κλάδους. Αναλύοντας δεδομένα από συνδεδεμένους αισθητήρες και συσκευές σε όλη την υποδομή της πόλης, η οπτικοποίηση μπορεί να βοηθήσει τους διαχειριστές και τους σχεδιαστές των πόλεων να βελτιώσουν τη διαχείριση των αποβλήτων, να κατανοήσουν τη ροή της κυκλοφορίας και να βελτιώσουν τις δημοτικές υπηρεσίες. Η οπτικοποίηση του IoT βοηθά τους κατασκευαστές να παρακολουθούν τις διαδικασίες παραγωγής τους, να εντοπίζουν τα σημεία συμφόρησης και να μεγιστοποιούν την παραγωγικότητα. Η προληπτική συντήρηση μπορεί να καταστεί δυνατή με την οπτικοποίηση δεδομένων αισθητήρων σε μηχανήματα, γεγονός που θα μειώσει τον χρόνο διακοπής λειτουργίας. Παρακολουθώντας την υγεία των ασθενών, εντοπίζοντας ανωμαλίες και προσφέροντας επιλογές απομακρυσμένης παρακολούθησης, οι φορητές τεχνολογίες και ο ιατρικός εξοπλισμός παρέχουν δεδομένα που μπορούν να προβληθούν για να βελτιώσουν τη φροντίδα των ασθενών και να διευκολύνουν την έγκαιρη δράση. Δεδομένα σχετικά με την υγεία των καλλιεργειών, τον καιρό και την υγρασία του εδάφους παρέχονται μέσω αισθητήρων του Διαδικτύου των πραγμάτων που είναι εγκατεστημένοι σε αγροκτήματα και θερμοκήπια. Οι αγρότες μπορούν να μεγιστοποιήσουν τις αποδόσεις των καλλιεργειών, να χρησιμοποιούν λιγότερο νερό και να κάνουν επιλογές βάσει δεδομένων με την οπτικοποίηση αυτών των δεδομένων. Αναλύοντας τη συμπεριφορά των καταναλωτών, βελτιστοποιώντας την τοποθέτηση προϊόντων και βελτιώνοντας τη διαχείριση των αποθεμάτων, οι φυσικοί λιανοπωλητές μπορούν να χρησιμοποιήσουν την οπτικοποίηση δεδομένων IoT για να βελτιώσουν ολόκληρη την εμπειρία αγορών. Η οπτικοποίηση δεδομένων από έξυπνες συσκευές μπορεί να βοηθήσει τους ιδιοκτήτες σπιτιών να παρακολουθούν τη χρήση ενέργειας, να εντοπίζουν παραβιάσεις ασφαλείας και να δημιουργούν προσαρμοσμένα χρονοδιαγράμματα αυτοματισμού.

3.7 Προκλήσεις και προβληματισμοί στην οπτικοποίηση δεδομένων IoT

Μεταξύ των δυσκολιών και των παραγόντων που πρέπει να ληφθούν υπόψη κατά την οπτικοποίηση δεδομένων IoT είναι ο όγκος και η ταχύτητα των δεδομένων. Η διαχείριση του ρυθμού με τον οποίο δημιουργούνται τα δεδομένα IoT και ο χειρισμός τεράστιου όγκου δεδομένων μπορεί να είναι δύσκολα καθήκοντα που απαιτούν κλιμακούμενες και αποτελεσματικές λύσεις επεξεργασίας δεδομένων. Είναι απαραίτητη η προστασία της ασφάλειας και του απορρήτου των δεδομένων του Διαδικτύου των πραγμάτων, ιδίως όταν γίνεται χειρισμός ευαίσθητων δεδομένων. Πρέπει να θεσπιστούν επαρκείς διασφαλίσεις για τα δεδομένα. Τα δεδομένα από συσκευές IoT ενδέχεται να είναι ακανόνιστα και θορυβώδη. Απαιτούνται τεχνικές για τον καθαρισμό και την επικύρωση των δεδομένων για την παροχή ακριβών και αξιόπιστων απεικονίσεων. Οι αποτελεσματικές δεξιότητες ροής και επεξεργασίας δεδομένων είναι απαραίτητες για οπτικοποιήσεις που διαχειρίζονται αλλαγές δεδομένων σε πραγματικό χρόνο. Προκειμένου να βοηθηθούν οι χρήστες στην ακριβή ερμηνεία των δεδομένων και να αποφευχθούν προκαταλήψεις ή παρερμηνείες, οι οπτικοποιήσεις θα πρέπει να παρέχουν το πλαίσιο των δεδομένων.

4. Μεθοδολογία για την ανάπτυξη Πίνακα Ελέγχου οπτικοποίησης δεδομένων IoT

Στο πλαίσιο της παρούσας εργασίας, σχεδιάστηκαν και αναπτύχθηκαν dashboards τα οποία καλύπτουν ένα φάσμα λειτουργιών, από την παρακολούθηση περιβαλλοντικών δεδομένων και την κατανάλωση ενέργειας, έως την ανάλυση δεδομένων ασφάλειας και υγείας. Από το κεντρικό dashboard δίνεται η δυνατότητα στους χρήστες να μεταβαίνουν στα dashboard που οπτικοποιούν τις διαφορετικές κατηγορίες δεδομένων. Οι κύριες τεχνολογίες που χρησιμοποιήθηκαν περιλαμβάνουν την Dash για την κατασκευή του front-end, το Plotly για τη δημιουργία οπτικοποιήσεων δεδομένων, καθώς και τη χρήση του Flask ως backend για τη διαχείριση της διαδραστικότητας. Η προσομοίωση δεδομένων σε πραγματικό χρόνο επιτεύχθηκε μέσω της βιβλιοθήκης random και της χρήσης multithreading για την ταυτόχρονη επεξεργασία δεδομένων.

Στο πλαίσιο της μεθοδολογίας που ακολουθήθηκε το πρώτο βήμα ήταν η ανάλυση απαιτήσεων. Προσδιορίστηκαν τα δεδομένα που επρόκειτο να παρακολουθηθούν και χωρίστηκαν σε οκτώ κατηγορίες: περιβαλλοντικά δεδομένα, κατανάλωση ενέργειας, ασφάλεια, συνδεσιμότητα, δεδομένα κατοίκησης, παρακολούθηση συσκευών, υγεία και ευεξία, και γενική επισκόπηση. Κάθε κατηγορία περιλάμβανε συγκεκριμένες μεταβλητές, όπως η θερμοκρασία, η υγρασία, η κατανάλωση ηλεκτρικής ενέργειας και η καρδιακή συχνότητα, που θα

προβάλλονταν με γραφικά στοιχεία σε πραγματικό χρόνο. Για την επίτευξη αυτών των στόχων, καθορίστηκαν λειτουργικές απαιτήσεις που περιλάμβαναν τη συνεχή ενημέρωση των δεδομένων, τη χρήση οπτικοποιήσεων για την απεικόνιση τάσεων, τη διαδραστικότητα μέσω επιλογών και φιλτραρίσματος, καθώς και την ανάδειξη κρίσιμων καταστάσεων μέσω συναγερμών. Εκτός από τις λειτουργικές απαιτήσεις, προσδιορίστηκαν και τεχνικές ανάγκες, όπως η υποστήριξη συνεχούς ροής δεδομένων και η χρήση πολυνηματικής αρχιτεκτονικής για την ταυτόχρονη επεξεργασία και παρουσίαση των πληροφοριών. Επιπλέον, τέθηκαν μη λειτουργικές απαιτήσεις όπως η επεκτασιμότητα του συστήματος, η γρήγορη απόκριση της εφαρμογής και η σαφήνεια της διεπαφής χρήστη.

Στο πλαίσιο τις διερεύνησης των δεδομένων που θα παρακολουθούνται στους πίνακες ελέγχου σχεδιάστηκαν πίνακες ελέγχου για την παρακολούθηση της κατανάλωσης ενέργειας, των περιβαλλοντικών δεδομένων, των δεδομένων ασφάλειας, την παρακολούθηση συσκευών, την παρακολούθηση δεδομένων πληρότητας, την παρακολούθηση δεδομένων υγείας, την παρακολούθηση δεδομένων συσκευών, την παρακολούθηση δεδομένων συνδεσιμότητας και δικτύου και τέλος την παρακολούθηση των ειδοποιήσεων (alarms) από το σύστημα. Για τον πίνακα ελέγχου κατανάλωσης ενέργειας αποφασίστηκε να εμφανίζει σε πραγματικό χρόνο δεδομένα κατανάλωσης και παραγωγής ενέργειας για τέσσερις μετρήσεις: χρήση ηλεκτρικής ενέργειας, χρήση φυσικού αερίου, χρήση νερού και παραγωγή ηλιακής ενέργειας.

Αντίστοιχα ο πίνακας περιβαλλοντικών δεδομένων παρέχει παρακολούθηση και απεικόνιση σε πραγματικό χρόνο τεσσάρων βασικών περιβαλλοντικών παραμέτρων: θερμοκρασία, υγρασία, ποιότητα αέρα και ένταση φωτός ενώ ο πίνακας δεδομένων ασφαλείας παρέχει παρακολούθηση σε πραγματικό χρόνο σχετικά με ανίχνευση κίνησης, κατάσταση θύρας, δραστηριότητα τροφοδοσίας βίντεο και αρχεία καταγραφής πρόσβασης. Στην συνέχεια για τον πίνακα ελέγχου συσκευών κρίθηκε χρήσιμο να παρέχει πληροφορίες σε πραγματικό χρόνο για την απόδοση και την κατάσταση διαφόρων συσκευών, όπως το πλυντήριο ρούχων, το ψυγείο, ο φούρνος, το πλυντήριο πιάτων και ο φούρνος μικροκυμάτων ενώ ο πίνακας ελέγχου παρακολούθησης δεδομένων πληρότητας παρέχει πληροφορίες σε πραγματικό χρόνο σχετικά τα επίπεδα δραστηριότητας και τη θερμοκρασία σε διάφορα δωμάτια ενός ακινήτου, όπως το σαλόνι, το υπνοδωμάτιο, η κουζίνα, το μπάνιο και το γραφείο. Για τον πίνακα ελέγχου παρακολούθησης δεδομένων υγείας κρίθηκε σκόπιμο να εμφανίζει πληροφορίες σε πραγματικό χρόνο σχετικά με τον καρδιακό ρυθμό, την θερμοκρασία σώματος, τα επίπεδα δραστηριότητας και την ποιότητα ύπνου.

Στην συνέχεια καθορίστηκαν τα δεδομένα για τον πίνακα ελέγχου δεδομένων συσκευών. Αυτός ο πίνακας παρέχει παρακολούθηση σε πραγματικό χρόνο και απεικόνιση βασικών παραμέτρων που σχετίζονται με διάφορες συσκευές, όπως φωτεινότητα φωτισμού, θερμοκρασία HVAC, επίπεδα ήχου και κατάσταση λειτουργίας της συσκευής. Επίσης ο

πίνακας ελέγχου συνδεσιμότητας και επιδόσεων δικτύου εμφανίζει δεδομένα σε πραγματικό χρόνο σχετικά με τις συσκευές σε ένα δίκτυο, εστιάζοντας σε μετρήσεις όπως η χρήση εύρους ζώνης, η ισχύς του σήματος και η κατάσταση online/offline και παρουσιάζεται μια επισκόπηση της τρέχουσας κατάστασης όλων των συσκευών, συμπεριλαμβανομένης της χρήσης εύρους ζώνης, της ισχύος σήματος και της κατάστασης σε απευθείας σύνδεση με τον πίνακα να ενημερώνεται δυναμικά για να αντικατοπτρίζει τα πιο πρόσφατα δεδομένα, προσφέροντας ένα στιγμιότυπο της δραστηριότητας του δικτύου. Τέλος θεωρήθηκε εξαιρετικά σημαντικός να υπάρχει ένας πίνακας ελέγχου με τις συνολικές ειδοποιήσεις από όλους τους επιμέρους πίνακες ελέγχου παρέχει μια ολοκληρωμένη σύνοψη της κατάστασης του συστήματος και των επιπέδων συναγερμού σε πολλαπλές κατηγορίες, συμπεριλαμβανομένων των περιβαλλοντικών δεδομένων, της κατανάλωσης ενέργειας, της παρακολούθησης συσκευών, της συνδεσιμότητας, της παρακολούθησης συσκευών, της παρακολούθησης της υγείας, της ασφάλειας και της πληρότητας.

Ο σχεδιασμός της αρχιτεκτονικής του συστήματος βασίστηκε στην αρχή της αρθρωτής σχεδίασης, όπου κάθε dashboard αναπτύχθηκε ως ανεξάρτητη μονάδα, αλλά όλες οι μονάδες συνδέθηκαν σε ένα κεντρικό dashboard. Αυτή η προσέγγιση εξασφάλισε ότι κάθε επιμέρους dashboard λειτουργούσε αυτόνομα, ενώ παράλληλα μπορούσε να ενσωματωθεί αρμονικά με τα υπόλοιπα. Κάθε dashboard σχεδιάστηκε για να εξυπηρετεί συγκεκριμένες ανάγκες, όπως η παρακολούθηση περιβαλλοντικών δεδομένων, κατανάλωσης ενέργειας ή ασφάλειας. Η διεπαφή χρήστη σχεδιάστηκε με τρόπο που να εξασφαλίζει την εύκολη πλοήγηση και την αμεσότητα στις πληροφορίες. Χρησιμοποιήθηκαν widgets όπως dropdown menus, γραφήματα και πίνακες, ενώ το σύστημα οπτικοποίησης βασίστηκε στη βιβλιοθήκη Plotly για τη δημιουργία δυναμικών γραφημάτων που ανταποκρίνονται σε πραγματικό χρόνο στις αλλαγές δεδομένων. Το κεντρικό dashboard αναπτύχθηκε ως πύλη προς τα επιμέρους dashboards, επιτρέποντας στους χρήστες να επιλέγουν την κατηγορία δεδομένων που θέλουν να εξετάσουν. Η σύνδεση μεταξύ του κεντρικού dashboard και των επιμέρους πραγματοποιήθηκε μέσω στατικής χαρτογράφησης των dashboards σε συγκεκριμένες θύρες του συστήματος. Κάθε dashboard μπορούσε να ξεκινήσει και να λειτουργήσει ανεξάρτητα, εξασφαλίζοντας την αυτονομία του συστήματος.

Η Dash της Python επιλέχθηκε ως η κύρια τεχνολογία για την κατασκευή της διεπαφής χρήστη, χάρη στη δυνατότητά της να δημιουργεί διαδραστικές web εφαρμογές απευθείας με Python. Η Dash παρέχει ένα εύχρηστο πλαίσιο και σε συνδυασμό με το Flask για το backend, το React.js για το frontend και το Plotly για την οπτικοποίηση δεδομένων δημιουργήθηκε ένα ολοκληρωμένο dashboard. Για την προσομοίωση δεδομένων, χρησιμοποιήθηκαν οι βιβλιοθήκες random και threading. Η random παρείχε τα απαραίτητα εργαλεία για τη δημιουργία τυχαίων δεδομένων, ενώ η threading χρησιμοποιήθηκε για την υλοποίηση πολυνηματικής επεξεργασίας. Με αυτή την προσέγγιση, κατέστη δυνατή η ταυτόχρονη

παραγωγή δεδομένων και η προβολή τους στις διεπαφές χρηστών, εξασφαλίζοντας ομαλή λειτουργία σε πραγματικό χρόνο.

Η λειτουργικότητα του συστήματος επικεντρώθηκε στην υλοποίηση μιας πολυεπίπεδη προσέγγισης που περιλάμβανε τη διαχείριση δεδομένων, τη δημιουργία μηχανισμών για τη συνεχή ενημέρωσή τους, την ανάπτυξη γραφικών στοιχείων και την ενσωμάτωση διαδραστικότητας. Η διαδραστικότητα του συστήματος επιτεύχθηκε μέσω του μηχανισμού callbacks που παρέχει η Dash. Τα callbacks επιτρέπουν την αυτόματη ενημέρωση των στοιχείων της διεπαφής κάθε φορά που μεταβάλλονται τα δεδομένα ή ο χρήστης εκτελεί κάποια ενέργεια. Για παράδειγμα, οι επιλογές σε ένα dropdown menu πυροδοτούν την ανανέωση του γραφήματος με τις αντίστοιχες πληροφορίες. Αυτός ο μηχανισμός εξασφάλισε μια ομαλή εμπειρία χρήσης, όπου οι πληροφορίες ενημερώνονται δυναμικά χωρίς να απαιτείται επανεκκίνηση της εφαρμογής.

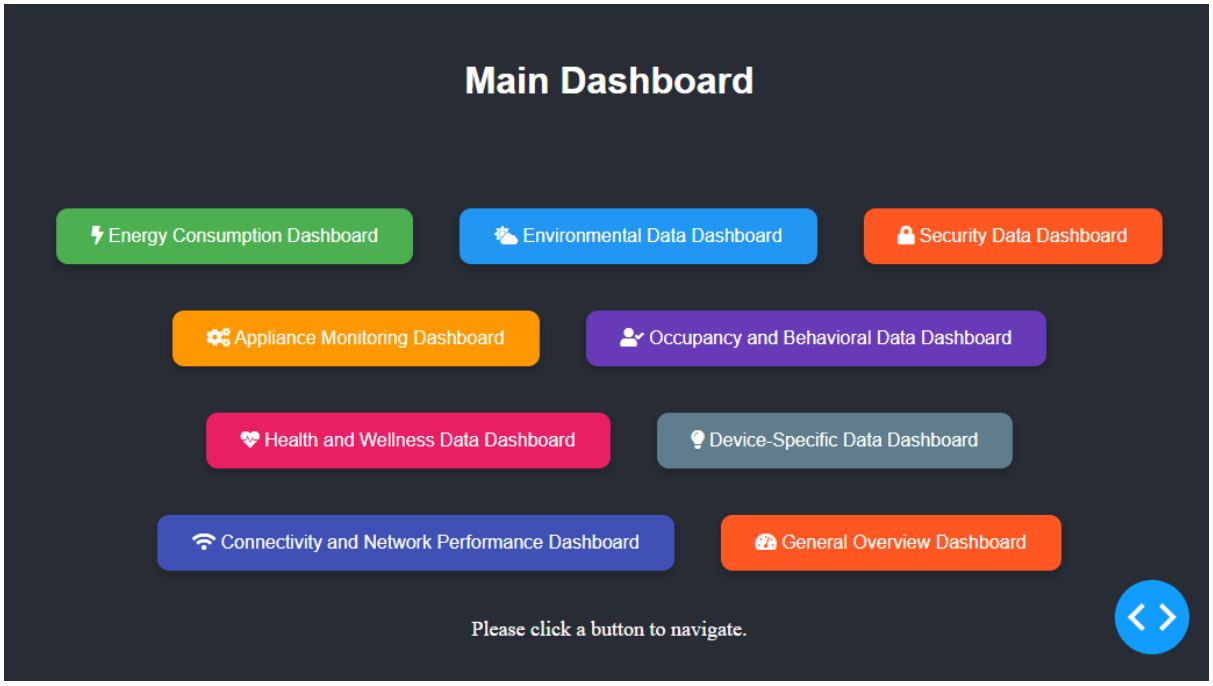
5. Υλοποίηση Πίνακα Ελέγχου οπτικοποίησης δεδομένων IoT

Η εφαρμογή που αναπτύχθηκε χρησιμεύσει σαν μια ολοκληρωμένη πλατφόρμα για την παρακολούθηση την ανάλυση και την διαχείριση των διασυνδεδεμένων συσκευών ενός έξυπνου σπιτιού ή κατ' επέκταση ενός οικοσυστήματος με στόχο να δίνει στους χρήστες την δυνατότητα να παρακολουθούν σε πραγματικό χρόνο τα παραγόμενα δεδομένα και να ενημερώνονται για την αποτελεσματική λειτουργεία όλων των πτυχών του οικοσυστήματος. Η εφαρμογή αποτελείται από έναν κεντρικό πίνακα ελέγχου στον οποίο υπάρχουν κουμπιά μέσα από τα οποία ο χρήστης μπορεί να ανακατευθυνθεί στους υπόλοιπους πίνακες ελέγχου της εφαρμογής.

Οι διαθέσιμοι πίνακες ελέγχου είναι:

- Πίνακας ελέγχου κατανάλωσης ενέργειας
- Πίνακας ελέγχου περιβαλλοντικών δεδομένων
- Πίνακας ελέγχου δεδομένων ασφάλειας
- Πίνακας ελέγχου παρακολούθησης συσκευών
- Πίνακας ελέγχου παρακολούθησης δεδομένων πληρότητας
- Πίνακας ελέγχου παρακολούθησης δεδομένων υγείας
- Πίνακας ελέγχου παρακολούθησης δεδομένων συσκευών
- Πίνακας ελέγχου παρακολούθησης δεδομένων συνδεσιμότητας και δικτύου
- Πίνακας ελέγχου με δεδομένα ειδοποιήσεων (alarms).

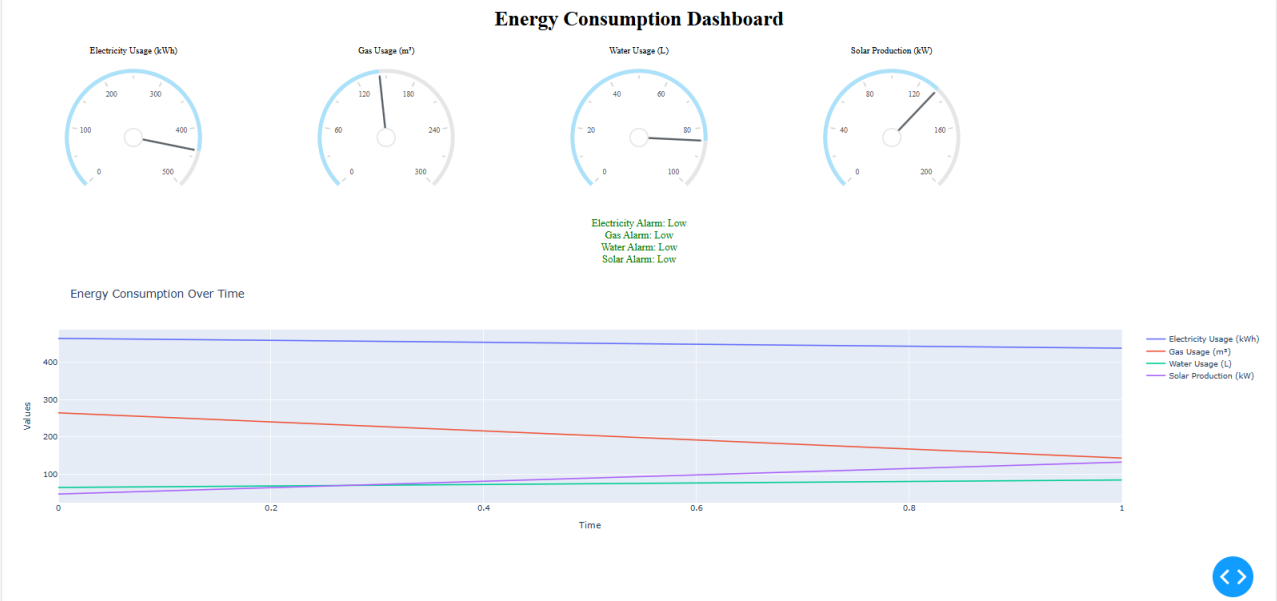
Ο μηχανισμός ενημέρωσης δεδομένων σε πραγματικό χρόνο στην εφαρμογή βασίζεται στο γεγονός ότι τα δεδομένα είτε προσομοιώνονται χρησιμοποιώντας τις δυνατότητες παραγωγής τυχαίων αριθμών της Python.



Εικόνα 1: Κεντρικός Πίνακας ελέγχου

5.1 Πίνακας ελέγχου κατανάλωσης ενέργειας

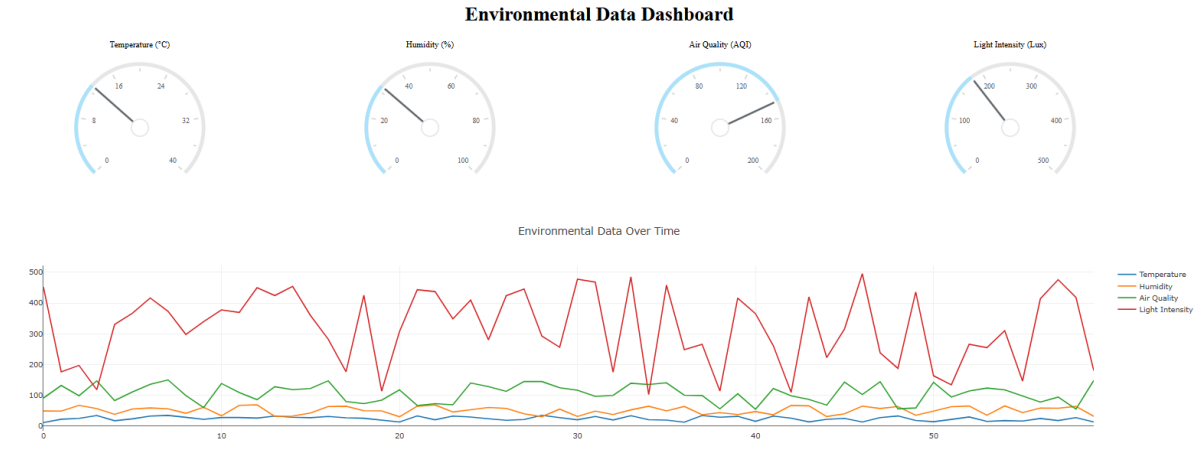
Ο πίνακας ελέγχου κατανάλωσης ενέργειας εμφανίζει σε πραγματικό χρόνο δεδομένα κατανάλωσης και παραγωγής ενέργειας για τέσσερις μετρήσεις: χρήση ηλεκτρικής ενέργειας, χρήση φυσικού αερίου, χρήση νερού και παραγωγή ηλιακής ενέργειας. Για κάθε μετρική, η τρέχουσα τιμή απεικονίζεται με τη χρήση ενός στοιχείου μετρητή που αναπαριστά τα δεδομένα σε μια κλίμακα που αντιστοιχεί στο καθορισμένο εύρος της. Ο πίνακας ελέγχου περιλαμβάνει επίσης ένα γραμμικό γράφημα χρονοσειρών που απεικονίζει τις ιστορικές τάσεις και των τεσσάρων μετρήσεων με την πάροδο του χρόνου. Ο πίνακας οργάνων περιλαμβάνει επίσης μηνύματα κειμένου κάτω από κάθε μετρητή, με το κατώφλι να καθορίζεται από τη σύγκριση της τρέχουσας τιμής με προκαθορισμένα όρια.



Εικόνα 2: Πίνακας ελέγχου κατανάλωσης ενέργειας

5.2 Πίνακας ελέγχου περιβαλλοντικών δεδομένων

Ο πίνακας περιβαλλοντικών δεδομένων παρέχει παρακολούθηση και απεικόνιση σε πραγματικό χρόνο τεσσάρων βασικών περιβαλλοντικών παραμέτρων: θερμοκρασία, υγρασία, ποιότητα αέρα και ένταση φωτός. Κάθε μία από αυτές τις παραμέτρους εμφανίζεται με τη χρήση ενός διαγράμματος μετρητή, το οποίο παρέχει μια άμεση οπτική αναπαράσταση της τρέχουσας τιμής εντός προκαθορισμένων περιοχών. Εκτός από τους μετρητές, ο πίνακας ελέγχου περιλαμβάνει ένα γραμμικό γράφημα που απεικονίζει τα ιστορικά δεδομένα και για τις τέσσερις παραμέτρους με την πάροδο του χρόνου.



Εικόνα 3: Πίνακας ελέγχου περιβαλλοντικών δεδομένων

5.3 Πίνακας ελέγχου δεδομένων ασφάλειας

Ο πίνακας δεδομένων ασφαλείας παρέχει παρακολούθηση σε πραγματικό χρόνο σχετικά με ανίχνευση κίνησης, κατάσταση θύρας, δραστηριότητα τροφοδοσίας βίντεο και αρχεία καταγραφής πρόσβασης. Για τις πληροφορίες αυτές δημιουργούνται τέσσερα γραμμικά γραφήματα, καθένα από τα οποία ενημερώνεται δυναμικά καθώς προσομοιώνονται νέα δεδομένα. Το γράφημα ανίχνευσης κίνησης εμφανίζει δυαδικά δεδομένα (0 ή 1), όπου η τιμή 1 υποδηλώνει ανιχνευμένη κίνηση και το 0 υποδηλώνει μηδενική κίνηση, ομοίως, το γράφημα κατάστασης θύρας χρησιμοποιεί επίσης δυαδικά δεδομένα για να δείξει αν οι θύρες ή τα παράθυρα είναι ανοικτά (1) ή κλειστά (0). Το γράφημα δραστηριότητας τροφοδοσίας βίντεο αναπαριστά εικονικούς δείκτες τροφοδοσίας βίντεο, παρουσιάζοντας διακυμάνσεις ή δραστηριότητα σε πολλαπλές τροφοδοσίες και μπορεί να βοηθήσει στον εντοπισμό των τροφοδοσιών που είναι ενεργές ή παρουσιάζουν αλλαγές. Επίσης το γράφημα καταγραφής πρόσβασης απεικονίζει προσομοιωμένη δραστηριότητα πρόσβασης χρηστών με αριθμητικά αναγνωριστικά για συμβάντα πρόσβασης.



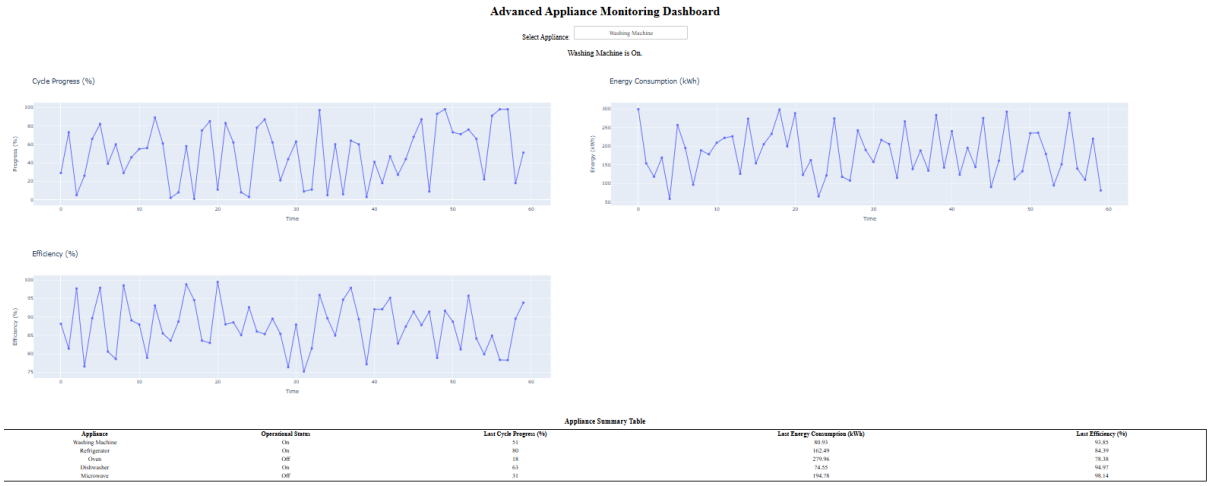
Εικόνα 4: Πίνακας ελέγχου δεδομένων ασφάλειας

5.4 Πίνακας ελέγχου παρακολούθησης συσκευών

Ο πίνακας ελέγχου συσκευών παρέχει πληροφορίες σε πραγματικό χρόνο για την απόδοση και την κατάσταση διαφόρων συσκευών, όπως το πλυντήριο ρούχων, το ψυγείο, ο φούρνος, το πλυντήριο πιάτων και ο φούρνος μικροκυμάτων. Εμφανίζει τέσσερις βασικές παραμέτρους για κάθε συσκευή: κατάσταση λειτουργίας, πρόοδος κύκλου, κατανάλωση ενέργειας και απόδοση. Οι χρήστες μπορούν να επιλέξουν μια συσκευή από ένα αναπτυσσόμενο μενού και ο πίνακας οργάνων ενημερώνεται για να εμφανίσει δεδομένα ειδικά για την επιλεγμένη

συσκευή. Στον πίνακα ελέγχου περιλαμβάνεται ένα μήνυμα κειμένου που υποδεικνύει αν η επιλεγμένη συσκευή είναι ενεργοποιημένη ή απενεργοποιημένη.

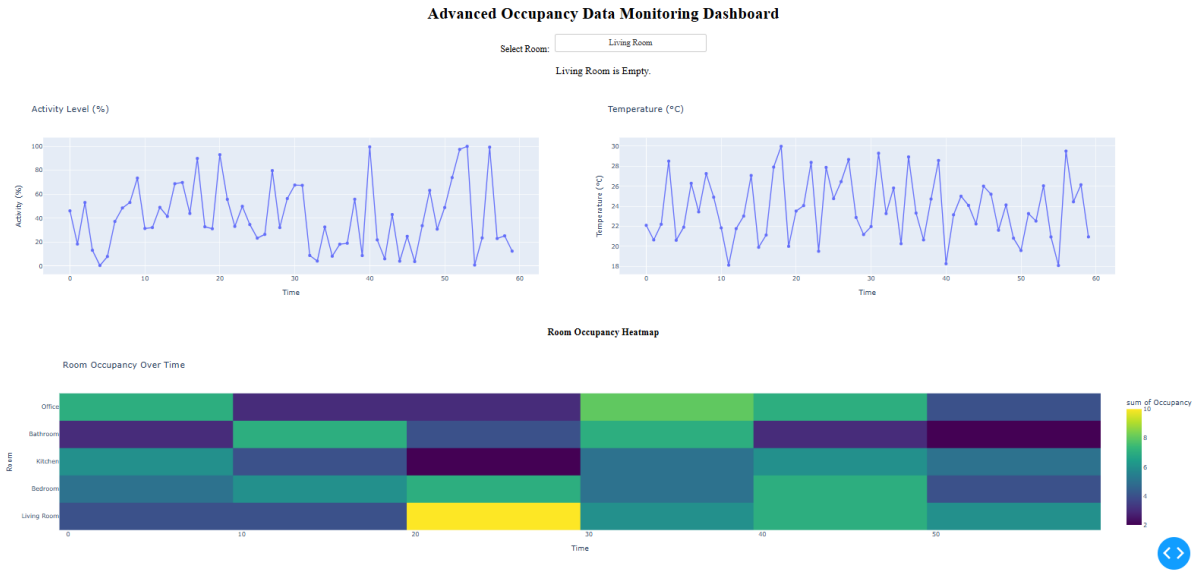
Επίσης διαθέτει τρία γραμμικά γραφήματα με το πρώτο γράφημα δείχνει την πρόοδο του κύκλου ως ποσοστό με την πάροδο του χρόνου, επιτρέποντας στους χρήστες να παρακολουθούν πόσο μακριά βρίσκεται η συσκευή στον τρέχοντα λειτουργικό της κύκλο, το δεύτερο γράφημα απεικονίζει την κατανάλωση ενέργειας σε κιλοβατώρες, απεικονίζοντας τη χρήση ενέργειας της συσκευής με την πάροδο του χρόνου και το τρίτο γράφημα απεικονίζει την αποδοτικότητα ως ποσοστό, παρέχοντας ένα συνεχές μέτρο του πόσο αποτελεσματικά λειτουργεί η συσκευή σε σχέση με τις μέγιστες δυνατότητές της.



Εικόνα 5: Πίνακας ελέγχου παρακολούθησης συσκευών

5.5 Πίνακας ελέγχου παρακολούθησης δεδομένων πληρότητας

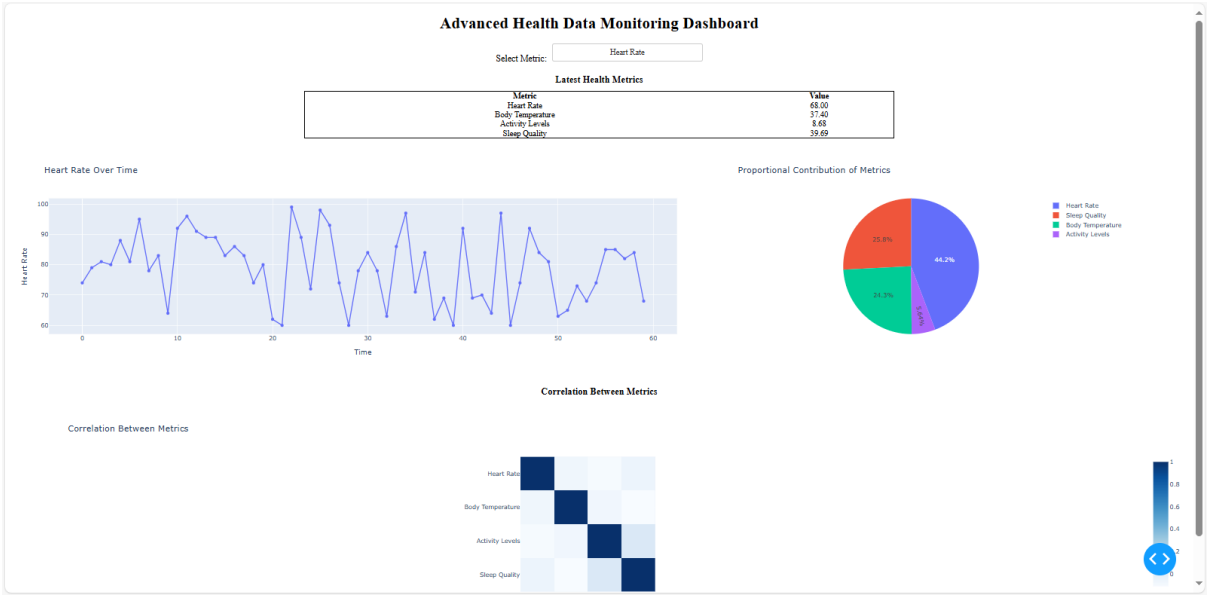
Ο πίνακας ελέγχου παρακολούθησης δεδομένων πληρότητας παρέχει πληροφορίες σε πραγματικό χρόνο σχετικά τα επίπεδα δραστηριότητας και τη θερμοκρασία σε διάφορα δωμάτια ενός ακινήτου, όπως το σαλόνι, το υπνοδωμάτιο, η κουζίνα, το μπάνιο και το γραφείο. Ο χρήστης μπορεί να επιλέξει ένα συγκεκριμένο δωμάτιο από ένα αναπτυσσόμενο μενού και ο πίνακας ελέγχου ενημερώνεται δυναμικά για να εμφανίσει τα δεδομένα για το συγκεκριμένο δωμάτιο. Για το επιλεγμένο δωμάτιο, ο πίνακας ελέγχου περιλαμβάνει μια ένδειξη κειμένου που δείχνει αν το δωμάτιο είναι επί του παρόντος κατειλημμένο ή άδειο. Επίσης διαθέτει δύο γραμμικά γραφήματα με το πρώτο γράφημα απεικονίζει τα επίπεδα δραστηριότητας στο δωμάτιο ως ποσοστό με την πάροδο του χρόνου, επιτρέποντας στους χρήστες να παρατηρούν μοτίβα χρήσης ή κίνησης και το δεύτερο γράφημα απεικονίζει τη θερμοκρασία εντός του δωματίου, δείχνοντας τις διακυμάνσεις σε βαθμούς Κελσίου κατά τη διάρκεια της περιόδου παρακολούθησης.



Εικόνα 6: Πίνακας ελέγχου παρακολούθησης δεδομένων πληρότητας

5.6 Πίνακας ελέγχου παρακολούθησης δεδομένων υγείας

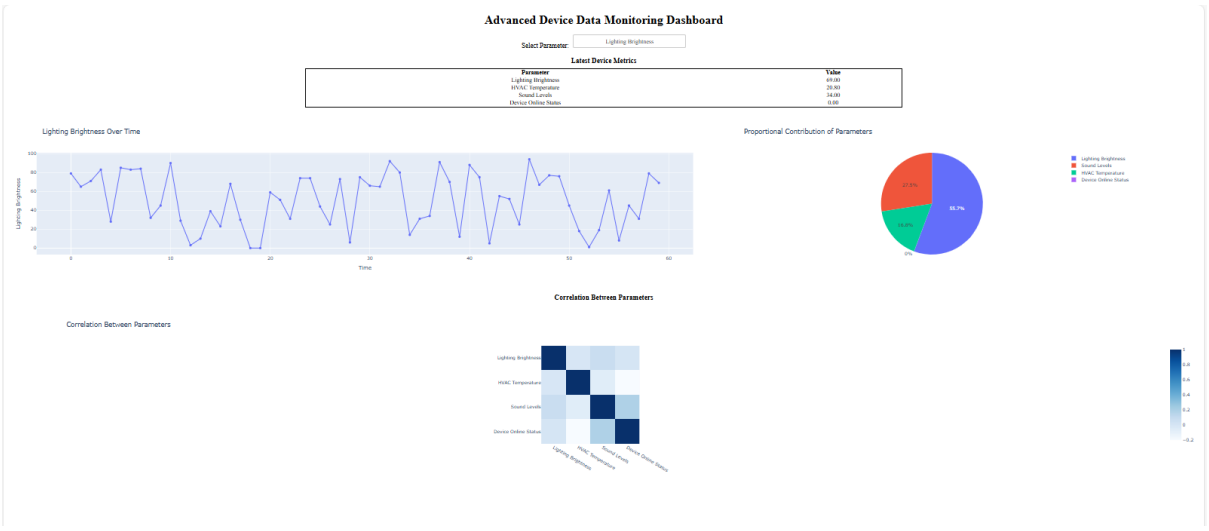
Ο πίνακας ελέγχου παρακολούθησης δεδομένων υγείας εμφανίζει πληροφορίες σε πραγματικό χρόνο σχετικά με τον καρδιακό ρυθμό, την θερμοκρασία σώματος, τα επίπεδα δραστηριότητας και την ποιότητα ύπνου. Οι χρήστες μπορούν να επιλέξουν μια συγκεκριμένη μέτρηση από ένα αναπτυσσόμενο μενού και ο πίνακας οργάνων ενημερώνεται δυναμικά για να εμφανίσει λεπτομερείς απεικονίσεις και δεδομένα για την επιλεγμένη μέτρηση. Για την επιλεγμένη μέτρηση, ένα γραμμικό γράφημα απεικονίζει την τάση της με την πάροδο του χρόνου, επιτρέποντας στους χρήστες να παρατηρήσουν διακυμάνσεις και μοτίβα στα δεδομένα. Επιπλέον, ένα διάγραμμα πίτας παρέχει μια κατανομή των πιο πρόσφατων τιμών σε όλες τις μετρήσεις. Για την περαιτέρω ανάλυση των σχέσεων μεταξύ των μετρήσεων, ο πίνακας οργάνων περιλαμβάνει έναν χάρτη θερμότητας συσχέτισης ο οποίος αναπαριστά οπτικά τη δύναμη και την κατεύθυνση των συσχετίσεων μεταξύ των τεσσάρων μετρήσεων, βοηθώντας τους χρήστες να εντοπίσουν πιθανές συνδέσεις, όπως το πώς τα επίπεδα δραστηριότητας μπορεί να σχετίζονται με τον καρδιακό ρυθμό ή την ποιότητα του ύπνου.



Εικόνα 7: Πίνακας ελέγχου παρακολούθησης δεδομένων υγείας

5.7 Πίνακας ελέγχου παρακολούθησης δεδομένων συσκευών

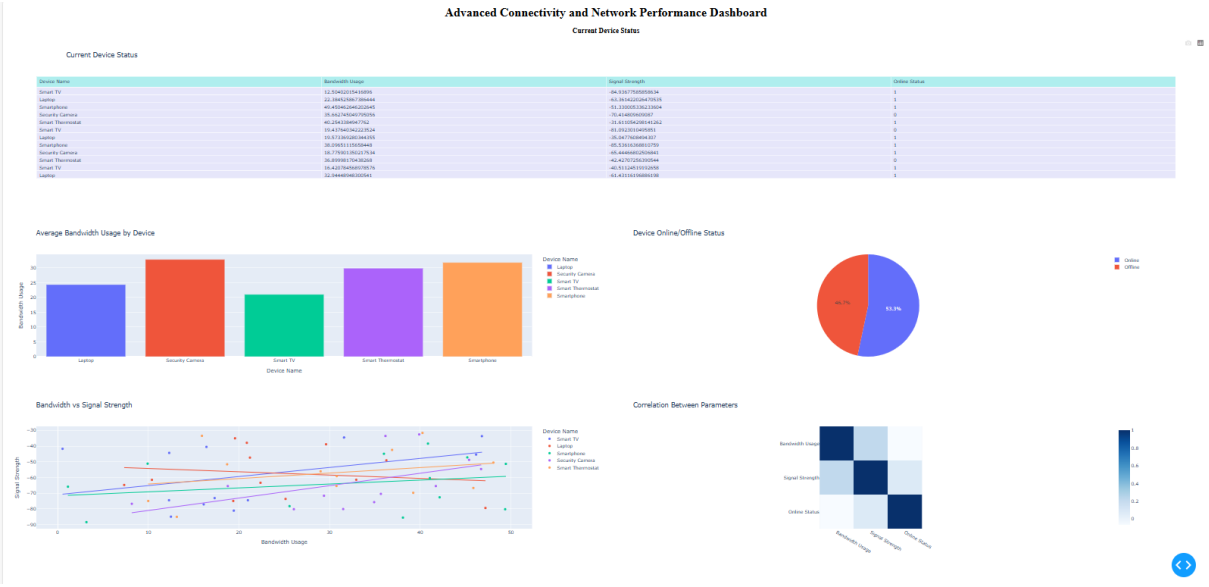
Ο πίνακας ελέγχου δεδομένων συσκευής παρέχει παρακολούθηση σε πραγματικό χρόνο και απεικόνιση βασικών παραμέτρων που σχετίζονται με διάφορες συσκευές, όπως φωτεινότητα φωτισμού, θερμοκρασία HVAC, επίπεδα ήχου και κατάσταση λειτουργίας της συσκευής. Οι χρήστες μπορούν να επιλέξουν μια συγκεκριμένη παράμετρο από ένα αναπτυσσόμενο μενού και ο πίνακας οργάνων ενημερώνεται δυναμικά για να εμφανίσει λεπτομερή δεδομένα και απεικονίσεις για την επιλεγμένη παράμετρο. Για την επιλεγμένη παράμετρο, ένα γραμμικό γράφημα απεικονίζει την τάση της με την πάροδο του χρόνου, παρέχοντας μια σαφή εικόνα του τρόπου με τον οποίο η παράμετρος έχει μεταβληθεί κατά τη διάρκεια της περιόδου παρακολούθησης. Επιπλέον, ένα διάγραμμα πίτας παρέχει μια κατανομή των πιο πρόσφατων τιμών σε όλες τις μετρήσεις.



Εικόνα 8: Πίνακας ελέγχου παρακολούθησης δεδομένων συσκευών

5.8 Πίνακας ελέγχου παρακολούθησης δεδομένων συνδεσιμότητας και δικτύου

Ο πίνακας ελέγχου συνδεσιμότητας και επιδόσεων δικτύου εμφανίζει δεδομένα σε πραγματικό χρόνο σχετικά με τις συσκευές σε ένα δίκτυο, εστιάζοντας σε μετρήσεις όπως η χρήση εύρους ζώνης, η ισχύς του σήματος και η κατάσταση online/offline. Παρουσιάζεται μια επισκόπηση της τρέχουσας κατάστασης όλων των συσκευών, συμπεριλαμβανομένης της χρήσης εύρους ζώνης, της ισχύος σήματος και της κατάστασης σε απευθείας σύνδεση με τον πίνακα να ενημερώνεται δυναμικά για να αντικατοπτρίζει τα πιο πρόσφατα δεδομένα, προσφέροντας ένα στιγμιότυπο της δραστηριότητας του δικτύου. Για την οπτικοποίηση της χρήσης εύρους ζώνης, ένα ραβδόγραμμα δείχνει τη μέση κατανάλωση εύρους ζώνης κάθε συσκευής και ένα διάγραμμα πίτας απεικονίζει το ποσοστό των συσκευών που είναι συνδεδεμένες σε σχέση με τις συσκευές που είναι εκτός σύνδεσης, δίνοντας μια γρήγορη ματιά στην κατάσταση συνδεσιμότητας του δικτύου. Επίσης παρουσιάζεται ένα διάγραμμα διασποράς για την εξέταση της σχέσης μεταξύ της χρήσης εύρους ζώνης και της ισχύος του σήματος, με τις επιμέρους συσκευές να διακρίνονται με χρώμα. Το διάγραμμα περιλαμβάνει επίσης μια γραμμή τάσης για την ανάδειξη γενικών μοτίβων στα δεδομένα. Τέλος, ένας θερμικός χάρτης συσχέτισης απεικονίζει τις στατιστικές σχέσεις μεταξύ των μετρήσεων της χρήσης εύρους ζώνης, της ισχύος σήματος και της κατάστασης σύνδεσης, βοηθώντας τους χρήστες να κατανοήσουν πώς αλληλοεπιδρούν αυτές οι μεταβλητές.



Εικόνα 9: Πίνακας ελέγχου παρακολούθησης δεδομένων συνδεσιμότητας και δικτύου

5.9 Πίνακας ελέγχου με δεδομένα ειδοποιήσεων (alarms).

Ο πίνακας ελέγχου με τις συνολικές ειδοποιήσεις από όλους τους επιμέρους πίνακες ελέγχου παρέχει μια ολοκληρωμένη σύνοψη της κατάστασης του συστήματος και των επιπέδων

συναγερμού σε πολλαπλές κατηγορίες, συμπεριλαμβανομένων των περιβαλλοντικών δεδομένων, της κατανάλωσης ενέργειας, της παρακολούθησης συσκευών, της συνδεσιμότητας, της παρακολούθησης συσκευών, της παρακολούθησης της υγείας, της ασφάλειας και της πληρότητας. Ένα ραβδόγραμμα χρησιμοποιείται για να απεικονίσει τη σοβαρότητα των ειδοποιήσεων για κάθε κατηγορία κατηγοριοποιώντας τις ειδοποιήσεις σε κρίσιμες, μεσαίου επιπέδου και χαμηλού επίπεδου, καθιστώντας εύκολη την απεικόνιση της κατανομής και της σοβαρότητας των προβλημάτων σε διάφορα στοιχεία του συστήματος. Ένας λεπτομερής πίνακας δεδομένων συνοδεύει το ραβδόγραμμα, παραθέτοντας τις ειδοποιήσεις και τις σχετικές συμβουλές για κάθε κατηγορία παρέχοντας ακριβή αριθμητικά δεδομένα και συστάσεις για να καθοδηγήσει τους χρήστες στην αντιμετώπιση πιθανών προβλημάτων. Επιπλέον, ο πίνακας οργάνων περιλαμβάνει ένα διάγραμμα πίτας που αναπαριστά τη συνολική κατανομή των επιπέδων ειδοποιήσεων σε όλο το σύστημα.

.



Εικόνα 10: Πίνακας ελέγχου με δεδομένα ειδοποιήσεων (alarm).

6. Συμπεράσματα

Το διαδίκτυο των πραγμάτων αναφέρεται σε ένα δίκτυο φυσικών συσκευών, οχημάτων, συσκευών και άλλων φυσικών αντικειμένων που είναι ενσωματωμένα με αισθητήρες, λογισμικό και συνδεσιμότητα δικτύου, επιτρέποντάς τους να συλλέγουν και να μοιράζονται δεδομένα. Οι συσκευές IoT χρησιμοποιούν αισθητήρες για τη συλλογή δεδομένων του πραγματικού κόσμου και μέσω τεχνικών ανάλυσης δεδομένων εξάγουν πολύτιμες πληροφορίες από τα δεδομένα, επιτρέποντας τη λήψη τεκμηριωμένων αποφάσεων και τη βελτιστοποίηση. Επιπλέον, η δυναμική φύση του IoT φέρνει επανάσταση στις αλληλεπιδράσεις με την τεχνολογία και υπόσχεται ένα μέλλον όπου τα αντικείμενα αλληλοεπιδρούν έξυπνα, ενισχύοντας την ευκολία, την αποδοτικότητα και τη συνδεσιμότητα. Χρησιμοποιεί πολλαπλά ενσωματωμένα συστήματα για τη συλλογή δεδομένων σχετικά με την κίνηση, τη θερμοκρασία και άλλες πτυχές του περιβάλλοντος του χρήστη και είναι μια εξαιρετική επιλογή για τις επιχειρήσεις που θέλουν να προσφέρουν καλύτερη εξυπηρέτηση πελατών.

Στο πλαίσιο αυτής της εργασίας αναπτύχθηκε μια εφαρμογή η οποία λειτουργεί σαν μια ολοκληρωμένη πλατφόρμα για την παρακολούθηση την ανάλυση και την διαχείριση των διασυνδεδεμένων συσκευών ενός έξυπνου σπιτιού ή κατ’ επέκταση ενός οικοσυστήματος με στόχο να δίνει στους χρήστες την δυνατότητα να παρακολουθούν σε πραγματικό χρόνο τα παραγόμενα δεδομένα και να ενημερώνονται για την αποτελεσματική λειτουργεία όλων των πτυχών του οικοσυστήματος. Η εφαρμογή αποτελείται από έναν κεντρικό πίνακα ελέγχου στον οποίο υπάρχουν κουμπιά μέσα από τα οποία ο χρήστης μπορεί να ανακατευθυνθεί στους υπόλοιπους πίνακες ελέγχου της εφαρμογής. Ο μηχανισμός ενημέρωσης δεδομένων σε πραγματικό χρόνο στην εφαρμογή βασίζεται στο γεγονός ότι τα δεδομένα είτε προσομοιώνονται χρησιμοποιώντας τις δυνατότητες παραγωγής τυχαίων αριθμών της Python. Ο συνδυασμός μετρητών, γραμμικών γραφημάτων και μηνυμάτων ειδοποιήσεων δημιουργεί μια ολοκληρωμένη απεικόνιση σε πραγματικό χρόνο που υποστηρίζει την παρακολούθηση, την ανάλυση και τη λήψη αποφάσεων σχετικά με τις παραμέτρους που παρακολουθούνται από την πλατφόρμα η οποία παρέχει στους χρήστες μια διαισθητική και διαδραστική διεπαφή για την αξιολόγηση της τρέχουσας κατάστασης και των πρόσφατων τάσεων διευκολύνοντας την έγκαιρη λήψη αποφάσεων και την κατανόηση.

Η πλατφόρμα αποδεικνύεται ιδιαίτερα χρήσιμη για την παρακολούθηση, ανάλυση και διαχείριση διασυνδεδεμένων συστημάτων σε πραγματικό χρόνο και επιτρέπει την απρόσκοπτη ενσωμάτωση διαφόρων τομέων, όπως η κατανάλωση ενέργειας, οι περιβαλλοντικές συνθήκες, η απόδοση των συσκευών, η συνδεσιμότητα και άλλα, σε μια ενιαία διεπαφή παρακολούθησης. Οι χρήστες επιβλέπουν αποτελεσματικά πολλαπλές πτυχές των συστημάτων τους χωρίς να χρειάζεται να εναλλάσσονται μεταξύ διαφορετικών εργαλείων

ή πινάκων οργάνων. Η επεξεργασία δεδομένων σε πραγματικό χρόνο και οι δυνατότητες δυναμικής απεικόνισης της πλατφόρμας παρέχουν άμεσες πληροφορίες σχετικά με τη συμπεριφορά του συστήματος, βοηθώντας τους χρήστες να εντοπίζουν ανωμαλίες, αναποτελεσματικότητα ή πιθανούς κινδύνους κατά την εμφάνισή τους. Προσφέροντας διαισθητικές απεικονίσεις, όπως γραμμικά γραφήματα, ραβδογράμματα, διαγράμματα πίτας, heatmaps και πίνακες, η πλατφόρμα εξυπηρετεί τόσο λεπτομερή ανάλυση όσο και γρήγορες επισκοπήσεις. Αυτή η ευελιξία την καθιστά κατάλληλη για ένα ευρύ φάσμα χρηστών, από τεχνικούς εμπειρογνώμονες έως μη ειδικούς.

Μέσω λειτουργιών όπως οι ειδοποιήσεις, η ανάλυση συσχετίσεων και οι εξατομικευμένες συστάσεις, η πλατφόρμα όχι μόνο εντοπίζει τα ζητήματα αλλά και καθοδηγεί τους χρήστες σχετικά με τον τρόπο αποτελεσματικής αντιμετώπισής τους. Αυτή η προληπτική προσέγγιση ενισχύει τη λήψη αποφάσεων και τη βελτιστοποίηση των πόρων. Η επεκτασιμότητα και η προσαρμοστικότητα της πλατφόρμας αυξάνουν περαιτέρω τη χρησιμότητά της. Είναι ικανή να φιλοξενήσει πρόσθετες μονάδες παρακολούθησης, υποστηρίζοντας ένα αυξανόμενο δίκτυο συσκευών και συστημάτων καθώς εξελίσσονται οι ανάγκες των χρηστών. Ο φιλικός προς το χρήστη σχεδιασμός της εξασφαλίζει την προσβασιμότητα, επιτρέποντας στους χρήστες να αλληλεπιδρούν με τα δεδομένα και να κάνουν προσαρμογές χωρίς κόπο. Οι οπτικές αναπαραστάσεις των δεδομένων βοηθούν τους υπεύθυνους να κατανοήσουν γρήγορα τις πληροφορίες του IoT, διευκολύνοντας τον εντοπισμό τάσεων, συσχετίσεων και στην λήψη αποφάσεων. Οι διαχειριστές μπορούν να παρακολουθούν μετρήσεις σε πραγματικό χρόνο από αισθητήρες IoT σε ένα πίνακα ελέγχου, επιτρέποντας τη λήψη γρήγορων, τεκμηριωμένων αποφάσεων σχετικά με τους πόρους και τις λειτουργίες αλλά και στον εντοπισμό της αναποτελεσματικότητας του συστήματος σε συγκεκριμένες καταστάσεις. Οι πίνακες ελέγχου σε πραγματικό χρόνο επιτρέπουν στους οργανισμούς να παρακολουθούν τις μετρήσεις καθώς συμβαίνουν, επιτρέποντας ταχύτερες αντιδράσεις σε πιθανά προβλήματα.

6. Βιβλιογραφικές Αναφορές

1. L. Atzori, A. Iera, and G. Morabito, (2010), “The Internet of Things: A Survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805; www.sciencedirect.com/science/article/pii/S1389128610001568.
2. Gatsis K, Pappas GJ., (2017), Wireless control for the IoT: power spectrum and security challenges. In: Proc. 2017 IEEE/ACM second international conference on internet-of-things design and implementation (IoTDI), Pittsburg, PA, USA, 18–21. INSPEC Accession Number: 16964293.
3. Khajenasiri I, Estebsari A, Verhelst M, Gielen G., (2017), A review on internet of things for intelligent energy control in buildings for smart city applications. Energy Procedia.111:770–9.
4. Zanella A, Bui N, Castellani A, Vangelista L, Zorgi M., (2014), Internet of things for smart cities. IEEE IoT-J.1(1):22–32.
5. Khajenasiri I, Estebsari A, Verhelst M, Gielen G., (2017), A review on internet of things for intelligent energy control in buildings for smart city applications. Energy Procedia.111:770–9.
6. Liu T, Yuan R, Chang H., (2012), Research on the internet of things in the automotive industry. In: ICMeCG 2012 international conference on management of e-commerce and e-Government, Beijing, China. 20–21. p. 230–3.
7. Alavi AH, Jiao P, Buttlar WG, Lajnef N., (2018), Internet of things-enabled smart cities: state-of-the-art and future trends. Measurement. 129:589–606.
8. Weber RH., (2010), Internet of things-new security and privacy challenges. Comput Law Secur Rev.26(1):23–30.
9. Heer T, Garcia-Morchon O, Hummen R, Keoh SL, Kumar SS, Wehrle K., (2011), Security challenges in the IP based internet of things. Wirel Pers Commun. 61(3):527–42.
10. Kothmayr T, Schmitt C, Hu W, Brunig M, Carle G., (2013), DTLS based security and two-way authentication for the internet of things. Ad Hoc Netw.11:2710–23.
11. Li Y, et al., (2019), IoT-CANE: a unified knowledge management system for data centric internet of things application systems. J Parallel Distrib Comput.131:161–72.
12. Olivier F, Carlos G, Florent N., (2015), New security architecture for IoT network. In: International workshop on big data and data mining challenges on IoT and pervasive systems (BigD2M 2015), procedia computer science, vol. 52. p. 1028–33.
13. Luk M, Mezzour G, Perrig A, Gligor V., (2007), MiniSec: a secure sensor netowrk communication architecture. In: Proc: 6th international symposium on information processing in sensor networks, Cambridge, MA, USA, 25–27.

14. Karlof C, Sastry N, Wagner D., (2004), TinySec: a link layer security architecture for wireless sensor networks. In: Proceedings of the second ACM conference on embedded networked sensor systems (SenSys 2004).
15. Yan Z, Zhang P, Vasilakos AV., (2014), A survey on trust management for internet of things. J Netw Comput Appl.42:120–34.
16. Bao F, Chen I-R, Guo J., (2013), Scalable, adaptive and survivable trust management for community of interest based internet of things systems. In: Proc. IEEE 11th international symposium on autonomous decentralized systems (ISADS). p. 1–7.
17. Noura M, Atiquazzaman M, Gaedke M., (2019), Interoperability in internet of things: taxonomies and open challenges. Mob Netw Appl.24(3):796–809.
18. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M., (2015), Internet of things: a survey, on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor. 2347–76.
19. Palattella MR, Dohler M, Grieco A, Rizzo G, Torsner J, Engel T, Ladid L., (2016), Internet of things in the 5G era: enablers, architecture and business models. IEEE J Sel Areas Commun. 34(3):510–27.
20. Pereira C, Aguiar A., (2014), Towards efficient mobile M2M communications: survey and open challenges. Sensors.14(10):19582–608.
21. Kim NS, Lee K, Ryu JH., (2015), Study on IoT based wild vegetation community ecological monitoring system. In: Proc. 2015 7th international conference on ubiquitous and future networks, Sapporo, Japan. IEEE.
22. Wang JY, Cao Y, Yu GP, Yuan M., (2014), Research on applications of IoT in domestic waste treatment and disposal. In: Proc. 11th World congress on intelligent control and automation, Shenyang, China,. IEEE.
23. Qiu T, Xiao H, Zhou P., (2013), Framework and case studies of intelligent monitoring platform in facility agriculture ecosystem. In: Proc. 2013 second international conference on agro-geoinformatics (agro-geoinformatics), Fairfax, VA, USA. IEEE.
24. Fang S, et al., (2014), An integrated system for regional environmental monitoring and management based on internet of things. IEEE Trans Ind Inf.10(2):1596–605.
25. Cheng Y, et al., (2014), AirCloud: a cloud based air-quality monitoring system for everyone. In: Proceedings of the 12th ACM conference on embedded network sensor systems, ACM, Memphis, Tennessee, p. 251–65.
26. Temglit N, Chibani A, Djouani K, Nacer MA., (2018), A distributed agent-based approach for optimal QoS selection in web of object choreography. IEEE Syst J.12(2):1655–66.
27. Talavera JM, et al., (2017), Review of IoT applications in agro-industrial and environmental fields. Comput Electron Agric.142(7):283–97.

28. Gubbi J, Buyya R, Marusic S, Palaniswami M., (2013), Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener Comput Syst. 2013;29(7):1645–60.
29. Sebastian S, Ray PP., (2015), Development of IoT invasive architecture for complying with health of home. In: Proc: I3CS, Shillong; 2015. p. 79–83.
30. B. B. Rad, N. Akbarzadeh, P. Ataei and Y. Khakbiz,, (2016), "Security and privacy challenges in big data era", Int. J. Control Theory Appl., vol. 9, no. 43, pp. 437-448.
31. P. Ataei and A. T. Litchfield, (2020), "Big data reference architectures a systematic literature review", Proc. ACIS, pp. 1-11.
32. B. B. Rada, P. Ataeib, Y. Khakbizc and N. Akbarzadehd, (2017), "The hype of emerging technologies: Big data as a service", Int. J. Control Theory Appl., vol. 9, no. 43, pp. 1-18,.
33. Big Data and AI Executive Survey, (2021), Boston, MA, USA, 2021.
34. B. B. Rad and P. Ataei, (2017), "The big data ecosystem and its environs", Int. J. Comput. Sci. Netw. Secur., vol. 17, no. 3, pp. 38.
35. Gorton and J. Klein, (2015), "Distribution data deployment", STC, vol. 32, no. 3, pp. 78.
36. R. Cloutier, G. Müller, D. Verma, R. Nilchiani, E. Hole and M. Bone, 2010, "The concept of reference architectures", Syst. Eng., vol. 13, no. 1, pp. 14-27.
37. M. Derras, L. Deruelle, J. M. Douin, N. Levy, F. Losavio, Y. Pollet, et al., (2018), "Reference architecture design: A practical approach", Proc. 13th Int. Conf. Softw. Technol., pp. 633-640.
38. Dangat, Prof M. T., (2024) , "Industrial Internet of Things (IIOT)." International Journal for Research in Applied Science and Engineering Technology 12, no. 3, 2721–26. http://dx.doi.org/10.22214/ijraset.2024.59103.
39. Sharma, Shivani Ramesh., (2019),"Internet of Things IoT: IoT in Healthcare." International Journal of Trend in Scientific Research and Development Volume-3, Issue-4: 980–82.
40. M. Al-Joboury, Istabraq, and Emad H. Al-Hemiary., (2017), "Internet of Things (IoT): Readme." Qalaai Zanist Scientific Journal 2, no. 2: 343–58. http://dx.doi.org/10.25212/lfu.qzj.2.2.35.
41. Birje, Mahantesh N., Arun A. Kumbi, and Ashok V. Sutagundar., (2017), "Internet of Things." International Journal of Hyperconnectivity and the Internet of Things 1, no. 2, 45–71. http://dx.doi.org/10.4018/ijhiot.2017070104.
42. Milivojević, Sanja, and Elizabeth Radulski., (2020), "The 'Future Internet' and crime: Towards a criminology of the Internet of Things." Crimen 11, no. 3, 255–71. http://dx.doi.org/10.5937/crimen2003255m.
43. Eltayeb, Mohamed A., (2017), "Internet of Things." International Journal of Hyperconnectivity and the Internet of Things 1, no. 1, 1–18. http://dx.doi.org/10.4018/ijhiot.2017010101.

44. Javida Damirova, Javida Damirova, and Laman Musayeva Laman Musayeva., (2022), "INTERNET OF THINGS." PAHTEI-Procedings of Azerbaijan High Technical Educational Institutions 13, no. 02, 33–43. http://dx.doi.org/10.36962/pahtei13022022-33.
45. Salukhe, Adarsh., (2023), "Internet of Things." International Journal for Research in Applied Science and Engineering Technology 11, no. 9, 1062–69. http://dx.doi.org/10.22214/ijraset.2023.55710.
46. Kumar, Sumit, and Zahid Raza., (2017), "Internet of Things." International Journal of Systems and Service-Oriented Engineering 7, no. 3, 32–52. http://dx.doi.org/10.4018/ijssoe.2017070103.
47. Okumura, Brandon M., (2017), "IoTA: Internet of Things Assistant." DigitalCommons@CalPoly, 2017. https://digitalcommons.calpoly.edu/theses/1769.
48. "The Enterprise Lacks a Big Data Strategy for IoT Transformation", (2017), Harvard Business Review, pp. 1-12.
49. S. Lavalle, M. S. Hopkins, E. Lesser, R. Schockley and N. Krushcwitz, (2010), "Analytics: The New Path to Value", MITSloan Management Review.,.
50. S. Viaene and A. Van den Bunder, (2011), "The secrets to managing business analytics projects", MITSloan Manag. Rev., pp. 65-69.
51. A. Chebotko, A. Kashlev and S. Lu, (2015) "A Big Data Modeling Methodology for Apache Cassandra", Proc. of IEEE Int. Congress on Big Data, pp. 238-245.
52. A. Fink, R. Guzzo and S. Roberts, (2017), "Big Data at Work: Lessons from the Field", Society for Industrial and Organizational Psychology.
53. S. Nalchigar and E. Yu, (2018), "Business-driven data analytics: a conceptual modeling framework", Data Knowledge Engineering, pp. 1-14.
54. J. Horkoff, D. Barone, L. Jiang, E. Yu, D. Amyot, A. Borgida, et al., (2014), "Strategic Business Modeling: Representation and Reasoning", Software Systems Modeling (13:3), pp. 1015-1041.
55. P. Giorgini, S. Rizzi and M. Garzetti, (2005), "Goal-oriented requirement analysis for data warehouse design", Proc. of the 8th ACM Int. Workshop on Data Warehousing and OLAP.
56. Pallavi Sethi and Smruti R. Sarangi, (2017), Internet of Things: Architectures, Protocols, and Applications,Journal of Electrical and Computer Engineering,Volume 2017, Article ID 9324035, 25 pages https://doi.org/10.1155/2017/9324035.
57. B.R. Lea et al., (2018), Data visualization for assessing the biofuel commercialization potential within the business intelligence framework, Journal of Cleaner Production.
58. Nadezhda I., (2024), Understanding the effectiveness and implications of Smart Home Energy Management Systems (SHEMS) for electricity consumption optimization in Finnish residential sector, https://urn.fi/URN:NBN:fi:aalto-202405263557.
59. pandas - Python Data Analysis Library

60. NumPy
61. Plotly Python Graphing Library
62. Dash Documentation & User Guide | Plotly

ΠΑΡΑΡΤΗΜΑ Α. Κώδικας της εφαρμογής

Αρχείο MainDashboard.py

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
from dash import Dash, html, dcc, Output, Input, callback_context
import webbrowser
from threading import Timer
import subprocess
# Initialize the main dashboard app
app = Dash(__name__, external_stylesheets=["https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"])

# Define the layout for the main dashboard
app.layout = html.Div(
    [
        html.H1(
            "Main Dashboard",
            style={
                "textAlign": "center",
                "color": "white",
                "fontFamily": "Arial, sans-serif",
                "paddingTop": "20px",
            },
        ),
        html.Div(
            [
                html.Button(
                    [
                        html.I(className="fas fa-bolt"),  # Icon for Energy
                        " Energy Consumption Dashboard",
                    ],
                    id="energy-btn",
                    style={
                        "margin": "20px",
                        "padding": "15px 30px",
                        "fontSize": "16px",
                        "color": "white",
                        "backgroundColor": "#4CAF50",
                        "border": "none",
                        "borderRadius": "10px",
                        "cursor": "pointer",
                        "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
                        "transition": "0.3s",
                    },
                ),
                html.Button(
                    [
                        html.I(className="fas fa-cloud-sun"),  # Icon for Environmental
                        " Environmental Data Dashboard",
                    ],
                    id="env-btn",
                    style={
```

```
            "margin": "20px",
            "padding": "15px 30px",
            "fontSize": "16px",
            "color": "white",
            "backgroundColor": "#2196F3",
            "border": "none",
            "borderRadius": "10px",
            "cursor": "pointer",
            "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
            "transition": "0.3s",
        },
    ),
    html.Button(
        [
            html.I(className="fas fa-lock"),  # Icon for Security
            " Security Data Dashboard",
        ],
        id="security-btn",
        style={
            "margin": "20px",
            "padding": "15px 30px",
            "fontSize": "16px",
            "color": "white",
            "backgroundColor": "#FF5722",
            "border": "none",
            "borderRadius": "10px",
            "cursor": "pointer",
            "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
            "transition": "0.3s",
        },
    ),
    html.Button(
        [
            html.I(className="fas fa-cogs"),  # Icon for Appliance Monitoring
            " Appliance Monitoring Dashboard",
        ],
        id="appliance-btn",
        style={
            "margin": "20px",
            "padding": "15px 30px",
            "fontSize": "16px",
            "color": "white",
            "backgroundColor": "#FF9800",
            "border": "none",
            "borderRadius": "10px",
            "cursor": "pointer",
            "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
            "transition": "0.3s",
        },
    ),
    html.Button(
        [
            html.I(className="fas fa-user-check"),  # Icon for Occupancy
            " Occupancy and Behavioral Data Dashboard",
        ],
```

```
        id="occupancy-btn",
        style={
            "margin": "20px",
            "padding": "15px 30px",
            "fontSize": "16px",
            "color": "white",
            "backgroundColor": "#673AB7",
            "border": "none",
            "borderRadius": "10px",
            "cursor": "pointer",
            "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
            "transition": "0.3s",
        },
    ),
    html.Button(
        [
            html.I(className="fas fa-heartbeat"),  # Icon for Health
            " Health and Wellness Data Dashboard",
        ],
        id="health-btn",
        style={
            "margin": "20px",
            "padding": "15px 30px",
            "fontSize": "16px",
            "color": "white",
            "backgroundColor": "#E91E63",
            "border": "none",
            "borderRadius": "10px",
            "cursor": "pointer",
            "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
            "transition": "0.3s",
        },
    ),
    html.Button(
        [
            html.I(className="fas fa-lightbulb"),  # Icon for Device-Specific
            " Device-Specific Data Dashboard",
        ],
        id="device-btn",
        style={
            "margin": "20px",
            "padding": "15px 30px",
            "fontSize": "16px",
            "color": "white",
            "backgroundColor": "#607D8B",
            "border": "none",
            "borderRadius": "10px",
            "cursor": "pointer",
            "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
            "transition": "0.3s",
        },
    ),
    html.Button(
        [
            html.I(className="fas fa-wifi"),  # Icon for Connectivity
```

```
                " Connectivity and Network Performance Dashboard",
            ],
            id="connectivity-btn",
            style={
                "margin": "20px",
                "padding": "15px 30px",
                "fontSize": "16px",
                "color": "white",
                "backgroundColor": "#3F51B5",
                "border": "none",
                "borderRadius": "10px",
                "cursor": "pointer",
                "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
                "transition": "0.3s",
            },
        ),
        html.Button(
            [
                html.I(className="fas fa-tachometer-alt"),  # Icon for General Overview
                " General Overview Dashboard",
            ],
            id="general-overview-btn",
            style={
                "margin": "20px",
                "padding": "15px 30px",
                "fontSize": "16px",
                "color": "white",
                "backgroundColor": "#FF5722",
                "border": "none",
                "borderRadius": "10px",
                "cursor": "pointer",
                "boxShadow": "0 4px 8px 0 rgba(0,0,0,0.2)",
                "transition": "0.3s",
            },
        ),
    ],
    style={"textAlign": "center", "paddingTop": "50px"},
),
html.Div(
    id="placeholder",
    style={
        "textAlign": "center",
        "marginTop": "20px",
        "color": "white",
        "fontSize": "18px",
    },
),
],
style={
    "backgroundColor": "#282C34",
    "height": "100vh",
    "display": "flex",
    "flexDirection": "column",
    "justifyContent": "center",
    "alignItems": "center",
```

```
    },
)

@app.callback(
    Output("placeholder", "children"),
    [
        Input("energy-btn", "n_clicks"),
        Input("env-btn", "n_clicks"),
        Input("security-btn", "n_clicks"),
        Input("appliance-btn", "n_clicks"),
        Input("occupancy-btn", "n_clicks"),
        Input("health-btn", "n_clicks"),
        Input("device-btn", "n_clicks"),
        Input("connectivity-btn", "n_clicks"),
        Input("general-overview-btn", "n_clicks"),
    ],
)
def navigate_to_dashboard(
    energy_clicks,
    env_clicks,
    security_clicks,
    appliance_clicks,
    occupancy_clicks,
    health_clicks,
    device_clicks,
    connectivity_clicks,
    general_overview_clicks,
):
    ctx = callback_context
    if not ctx.triggered:
        return "Please click a button to navigate."

    button_id = ctx.triggered[0]["prop_id"].split(".")[0]
    port_mapping = {
        "energy-btn": 8051,
        "env-btn": 8052,
        "security-btn": 8053,
        "appliance-btn": 8054,
        "occupancy-btn": 8055,
        "health-btn": 8056,
        "device-btn": 8057,
        "connectivity-btn": 8059,
        "general-overview-btn": 8060,  # Port for General Overview
    }
    webbrowser.open(f"http://127.0.0.1:{port_mapping[button_id]}")
    return f"Navigating to {button_id.replace('-btn', '').capitalize()} Dashboard..."

# Start dashboards
def start_dashboards():
    subprocess.run(["jupyter", "nbconvert", "--to", "script", "*.ipynb"], check=True)
    ports = [8051, 8052, 8053, 8054, 8055, 8056, 8057, 8059, 8060]
    scripts = [
        "EnergyConsumptionDashboard.py",
        "EnvironmentalData.py",
        "SecurityData.py",
```

```
        "ApplianceMonitoring.py",
        "OccupancyData.py",
        "AdvancedHealthMonitoring.py",
        "DeviceDataMonitoring.py",
        "ConnectivityDataMonitoring.py",
        "GeneralOverviewDashboard.py",
    ]
    for script, port in zip(scripts, ports):
        subprocess.Popen(["python", script])

    Timer(1, lambda: webbrowser.open("http://127.0.0.1:8050"))
    app.run_server(debug=True, port=8050)

if __name__ == "__main__":
    start_dashboards()

# In[ ]:
```

Αρχείο SecurityData.py

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:
import random
import time
from dash import Dash, dcc, html, Output, Input
from threading import Thread
from collections import deque
import plotly.graph_objs as go

# Global Variables
max_data_points = 60
interval = 1  # 1 second for simulation during testing

data = {
    "motion": deque(maxlen=max_data_points),
    "door_status": deque(maxlen=max_data_points),
    "video_feeds": deque(maxlen=max_data_points),
    "access_logs": deque(maxlen=max_data_points)
}

def simulate_security_data():
    """Simulate security-related data every second."""
    while True:
        data["motion"].append(random.choice([0, 1]))  # Binary motion detection
        data["door_status"].append(random.choice([0, 1]))  # 0: closed, 1: open
        data["video_feeds"].append(random.uniform(1, 5))  # Dummy feed index
        data["access_logs"].append(random.randint(0, 5))  # Dummy user access ID
        time.sleep(interval)

# Initialize Dash App
app = Dash(__name__)

# Layout for the app
```

```
app.layout = html.Div([
    html.H1("Security Data Dashboard", style={"textAlign": "center"}),
    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),

    html.Div([
        dcc.Graph(id="motion-graph", style={"width": "45%", "display": "inline-block"}),
        dcc.Graph(id="door-graph", style={"width": "45%", "display": "inline-block"}),
    ]),

    html.Div([
        dcc.Graph(id="video-graph", style={"width": "45%", "display": "inline-block"}),
        dcc.Graph(id="access-graph", style={"width": "45%", "display": "inline-block"}),
    ])
])

@app.callback(
    [Output("motion-graph", "figure"),
     Output("door-graph", "figure"),
     Output("video-graph", "figure"),
     Output("access-graph", "figure")],
    [Input("update-interval", "n_intervals")]
)
def update_dashboard(n):
    motion_fig = go.Figure(
        data=[go.Scatter(y=list(data["motion"]), mode="lines+markers", name="Motion
Detection")],
        layout={"title": "Motion Detection"}
    )

    door_fig = go.Figure(
        data=[go.Scatter(y=list(data["door_status"]), mode="lines+markers",
name="Door/Window Status")],
        layout={"title": "Door/Window Status (1: Open, 0: Closed)"}
    )

    video_fig = go.Figure(
        data=[go.Scatter(y=list(data["video_feeds"]), mode="lines+markers", name="Video
Feeds")],
        layout={"title": "Video Feed Activity"}
    )

    access_fig = go.Figure(
        data=[go.Scatter(y=list(data["access_logs"]), mode="lines+markers", name="Access
Logs")],
        layout={"title": "Access Logs"}
    )

    return motion_fig, door_fig, video_fig, access_fig

# Main Execution
if __name__ == "__main__":
    # Ensure simulation thread runs only once
    thread = Thread(target=simulate_security_data, daemon=True,
name="SecuritySimulationThread")
    thread.start()
```

```
    # Run server
    app.run_server(debug=True, port=8053)

# In[ ]:
```

Αρχείο OccupancyData.py

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:
import random
import time
from dash import Dash, dcc, html, Output, Input
from threading import Thread
from collections import deque
import plotly.graph_objs as go
import pandas as pd
import plotly.express as px

# Global Variables
max_data_points = 60
interval = 1  # 1 second for simulation during testing

# Rooms being monitored
rooms = ["Living Room", "Bedroom", "Kitchen", "Bathroom", "Office"]

data = {
    room: {
        "presence": deque(maxlen=max_data_points),  # 1: Occupied, 0: Empty
        "activity": deque(maxlen=max_data_points),  # Random activity levels
        "temperature": deque(maxlen=max_data_points)  # Temperature in °C
    }
    for room in rooms
}

def simulate_occupancy_data():
    """Simulate data for room occupancy and activity."""
    while True:
        for room in rooms:
            data[room]["presence"].append(random.choice([0, 1]))
            data[room]["activity"].append(random.uniform(0, 100))  # Activity level (0-100)
            data[room]["temperature"].append(random.uniform(18, 30))  # Temperature in °C
        time.sleep(interval)

# Initialize Dash App
app = Dash(__name__)

# Layout for the app
app.layout = html.Div([
    html.H1("Advanced Occupancy Data Monitoring Dashboard", style={"textAlign": "center",
"marginBottom": "20px"}),

    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),

    # Room Dropdown
```

```
    html.Div([
        html.Label("Select Room:", style={"fontSize": "18px", "marginRight": "10px"}),
        dcc.Dropdown(
            id="room-dropdown",
            options=[{"label": room, "value": room} for room in rooms],
            value=rooms[0],
            style={"width": "300px", "display": "inline-block"}
        )
    ], style={"textAlign": "center", "marginBottom": "20px"}),

    # Room Presence and Activity
    html.Div([
        html.Div(id="room-presence", style={"textAlign": "center", "fontSize": "20px",
"marginBottom": "20px"}),
    ]),

    # Graphs
    html.Div([
        dcc.Graph(id="activity-graph", style={"width": "48%", "display": "inline-block"}),
        dcc.Graph(id="temperature-graph", style={"width": "48%", "display": "inline-block"}),
    ]),

    # Heatmap
    html.Div([
        html.H3("Room Occupancy Heatmap", style={"textAlign": "center", "marginBottom":
"10px"}),
        dcc.Graph(id="occupancy-heatmap")
    ], style={"marginTop": "20px"})
])

@app.callback(
    [Output("room-presence", "children"),
     Output("activity-graph", "figure"),
     Output("temperature-graph", "figure"),
     Output("occupancy-heatmap", "figure")],
    [Input("update-interval", "n_intervals"),
     Input("room-dropdown", "value")]
)
def update_dashboard(n, selected_room):
    # Get data for the selected room
    presence = data[selected_room]["presence"][-1] if data[selected_room]["presence"] else 0
    activity = list(data[selected_room]["activity"])
    temperature = list(data[selected_room]["temperature"])

    # Presence Message
    presence_message = f"{selected_room} is {'Occupied' if presence else 'Empty'}."

    # Activity Graph
    activity_fig = go.Figure(
        data=[go.Scatter(y=activity, mode="lines+markers", name="Activity Level")],
        layout={"title": "Activity Level (%)", "xaxis_title": "Time", "yaxis_title": "Activity (%)"}
    )

    # Temperature Graph
    temperature_fig = go.Figure(
```

```
        data=[go.Scatter(y=temperature, mode="lines+markers", name="Temperature")],
        layout={"title": "Temperature (°C)", "xaxis_title": "Time", "yaxis_title": "Temperature
(°C)"}
    )

    # Heatmap Data Preparation
    heatmap_data = {
        "Room": [],
        "Time": [],
        "Occupancy": []
    }

    for room in rooms:
        heatmap_data["Room"].extend([room] * len(data[room]["presence"]))
        heatmap_data["Time"].extend(list(range(len(data[room]["presence"]))))
        heatmap_data["Occupancy"].extend(data[room]["presence"])

    heatmap_df = pd.DataFrame(heatmap_data)

    # Heatmap Figure
    heatmap_fig = px.density_heatmap(
        heatmap_df, x="Time", y="Room", z="Occupancy", color_continuous_scale="Viridis",
        title="Room Occupancy Over Time"
    )

    return presence_message, activity_fig, temperature_fig, heatmap_fig

# Main Execution
if __name__ == "__main__":
    # Ensure simulation thread runs only once
    thread = Thread(target=simulate_occupancy_data, daemon=True,
name="OccupancySimulationThread")
    thread.start()

    # Run server
    app.run_server(debug=True, port=8055)
# In[ ]:
```

Αρχείο GeneralOverviewDashboard.py

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import random
from dash import Dash, dcc, html, Output, Input
from threading import Thread
import pandas as pd
import plotly.express as px

# Sample data for alarms from different dashboards
categories = [
    "Environmental", "Energy Consumption", "Device Monitoring",
```

```
    "Connectivity", "Appliance Monitoring", "Health Monitoring",
    "Security", "Occupancy"
]

# Simulated summary alarm data
alarm_data = pd.DataFrame({
    "Category": categories,
    "Critical Alarms": [random.randint(1, 5) for _ in categories],
    "Medium Alarms": [random.randint(3, 10) for _ in categories],
    "Low Alarms": [random.randint(5, 15) for _ in categories],
    "Advice": [
        "Check air quality levels",
        "Optimize energy usage",
        "Ensure devices are functioning properly",
        "Improve network signal strength",
        "Inspect appliance statuses",
        "Monitor health data anomalies",
        "Review security logs",
        "Analyze occupancy trends"
    ]
})

# Initialize Dash App
app = Dash(__name__)

# Layout for the General Dashboard
app.layout = html.Div([
    html.H1("General Overview Dashboard", style={"textAlign": "center", "marginBottom":
"20px"}),

    html.Div([
        html.H3("System Alarm Summary", style={"textAlign": "center"}),
        dcc.Graph(
            id="alarm-summary-bar-chart",
            style={"width": "80%", "margin": "auto"}
        ),
    ], style={"marginBottom": "30px"}),

    html.Div([
        html.H3("Critical Alarms Table", style={"textAlign": "center"}),
        html.Div(id="alarm-table", style={"textAlign": "center", "marginBottom": "30px"})
    ]),

    html.Div([
        html.H3("Overall System Health", style={"textAlign": "center"}),
        dcc.Graph(
            id="overall-health-pie-chart",
            style={"width": "60%", "margin": "auto"}
        )
    ])
])

@app.callback(
    [Output("alarm-summary-bar-chart", "figure"),
     Output("alarm-table", "children"),
```

```
     Output("overall-health-pie-chart", "figure")],
    [Input("alarm-summary-bar-chart", "id")]  # Trigger on load
)
def update_general_dashboard(_):
    # Bar Chart for Alarm Summary
    bar_chart_fig = px.bar(
        alarm_data.melt(id_vars=["Category", "Advice"], value_vars=["Critical Alarms", "Medium
Alarms", "Low Alarms"]),
        x="Category",
        y="value",
        color="variable",
        title="Alarm Severity by Category",
        labels={"value": "Number of Alarms", "variable": "Severity Level"}
    )

    # Data Table for Critical Alarms
    critical_alarms_table = html.Table([
        html.Tr([html.Th(col) for col in alarm_data.columns])  # Table Header
    ] + [
        html.Tr([html.Td(alarm_data.iloc[i][col]) for col in alarm_data.columns])
        for i in range(len(alarm_data))
    ], style={"margin": "auto", "border": "1px solid black", "borderCollapse": "collapse", "width":
"80%"})

    # Pie Chart for Overall System Health
    total_alarms = alarm_data[["Critical Alarms", "Medium Alarms", "Low
Alarms"]].sum().to_dict()
    pie_chart_fig = px.pie(
        names=list(total_alarms.keys()),
        values=list(total_alarms.values()),
        title="System Health Distribution by Alarm Level"
    )

    return bar_chart_fig, critical_alarms_table, pie_chart_fig

# Main Execution
if __name__ == "__main__":
    app.run_server(debug=True, port=8060)


# In[ ]:
```

Αρχείο EnvironmentalData.py

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:


import random
import time
from dash import Dash, html, dcc, Output, Input
import threading  # Ensure threading is imported
import dash_daq as daq
from collections import deque
```

```
# Global Variables
max_data_points = 60
interval = 1  # Shortened for testing

data = {
    "temperature": deque(maxlen=max_data_points),
    "humidity": deque(maxlen=max_data_points),
    "air_quality": deque(maxlen=max_data_points),
    "light": deque(maxlen=max_data_points)
}

def simulate_data():
    """Simulate Environmental Data."""
    while True:
        data["temperature"].append(random.uniform(10, 35))
        data["humidity"].append(random.uniform(30, 70))
        data["air_quality"].append(random.uniform(50, 150))
        data["light"].append(random.uniform(100, 500))
        time.sleep(interval)

# Initialize Dash App
app = Dash(__name__)

# Layout
app.layout = html.Div([
    html.H1("Environmental Data Dashboard", style={"textAlign": "center"}),
    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),
    html.Div([
        html.Div([daq.Gauge(id="temperature-gauge", min=0, max=40, value=0,
label="Temperature (°C)")],
                 style={"display": "inline-block", "width": "24%"}),
        html.Div([daq.Gauge(id="humidity-gauge", min=0, max=100, value=0, label="Humidity
(%)")],
                 style={"display": "inline-block", "width": "24%"}),
        html.Div([daq.Gauge(id="air-quality-gauge", min=0, max=200, value=0, label="Air
Quality (AQI)")],
                 style={"display": "inline-block", "width": "24%"}),
        html.Div([daq.Gauge(id="light-gauge", min=0, max=500, value=0, label="Light Intensity
(Lux)")],
                 style={"display": "inline-block", "width": "24%"}),
    ]),
    dcc.Graph(id="env-graph")
])

@app.callback(
    [Output("temperature-gauge", "value"),
     Output("humidity-gauge", "value"),
     Output("air-quality-gauge", "value"),
     Output("light-gauge", "value"),
     Output("env-graph", "figure")],
    [Input("update-interval", "n_intervals")]
)
def update_dashboard(n):
    temp = data["temperature"][-1] if data["temperature"] else 0
```

```
    humid = data["humidity"][-1] if data["humidity"] else 0
    air_q = data["air_quality"][-1] if data["air_quality"] else 0
    light = data["light"][-1] if data["light"] else 0

    fig = {
        "data": [
            {"x": list(range(len(data["temperature"]))), "y": list(data["temperature"]), "type": "line",
"name": "Temperature"},
            {"x": list(range(len(data["humidity"]))), "y": list(data["humidity"]), "type": "line", "name":
"Humidity"},
            {"x": list(range(len(data["air_quality"]))), "y": list(data["air_quality"]), "type": "line",
"name": "Air Quality"},
            {"x": list(range(len(data["light"]))), "y": list(data["light"]), "type": "line", "name": "Light
Intensity"},
        ],
        "layout": {"title": "Environmental Data Over Time"}
    }

    return temp, humid, air_q, light, fig

if __name__ == "__main__":
    if not any(thread.name == "DataSimulationThread" for thread in threading.enumerate()):
        thread = threading.Thread(target=simulate_data, daemon=True,
name="DataSimulationThread")
        thread.start()
    app.run_server(debug=True, port=8052)
```

Αρχείο EnergyConsumptionDashboard.py

```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:


import random
import time
from dash import Dash, dcc, html, Output, Input
import dash_daq as daq
import plotly.graph_objs as go
from threading import Thread
from collections import deque

# Global Variables
max_data_points = 60
interval = 60  # Update interval in seconds

data = {
    "electricity": deque(maxlen=max_data_points),
    "gas": deque(maxlen=max_data_points),
    "water": deque(maxlen=max_data_points),
    "solar": deque(maxlen=max_data_points)
}

alarms = {
    "electricity": (0, 500),   # Example range for electricity usage
```

```
    "gas": (0, 300),          # Example range for gas usage
    "water": (0, 100),         # Example range for water usage
    "solar": (0, 200)         # Example range for solar production
}

alarm_levels = {
    "low": "green",
    "medium": "orange",
    "high": "red"
}

def simulate_data():
    """Simulate Energy Consumption data every minute."""
    while True:
        data["electricity"].append(random.uniform(0, 500))
        data["gas"].append(random.uniform(0, 300))
        data["water"].append(random.uniform(0, 100))
        data["solar"].append(random.uniform(0, 200))
        time.sleep(interval)

def determine_alarm(value, range_limits):
    """Determine alarm level based on value and range."""
    if value < range_limits[0] or value > range_limits[1]:
        deviation = abs(value - range_limits[0]) if value < range_limits[0] else abs(value -
range_limits[1])
        if deviation > 10:
            return "high"
        else:
            return "medium"
    return "low"

# Dash App
app = Dash(__name__)
app.layout = html.Div([
    html.H1("Energy Consumption Dashboard", style={"textAlign": "center"}),
    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),

    html.Div([
        html.Div([daq.Gauge(id="electricity-gauge", min=0, max=500, value=0,
label="Electricity Usage (kWh)")],
                 style={"display": "inline-block", "width": "20%"}),
        html.Div([daq.Gauge(id="gas-gauge", min=0, max=300, value=0, label="Gas Usage
(m³)")],
                 style={"display": "inline-block", "width": "20%"}),
        html.Div([daq.Gauge(id="water-gauge", min=0, max=100, value=0, label="Water Usage
(L)")],
                 style={"display": "inline-block", "width": "20%"}),
        html.Div([daq.Gauge(id="solar-gauge", min=0, max=200, value=0, label="Solar
Production (kW)")],
                 style={"display": "inline-block", "width": "20%"}),
    ]),

    html.Div([
        html.Div(id="electricity-alarm", style={"textAlign": "center"}),
        html.Div(id="gas-alarm", style={"textAlign": "center"}),
```

```
        html.Div(id="water-alarm", style={"textAlign": "center"}),
        html.Div(id="solar-alarm", style={"textAlign": "center"}),
    ]),

    dcc.Graph(id="energy-data-graph")
])

@app.callback(
    [Output("electricity-gauge", "value"),
     Output("gas-gauge", "value"),
     Output("water-gauge", "value"),
     Output("solar-gauge", "value"),
     Output("energy-data-graph", "figure"),
     Output("electricity-alarm", "children"),
     Output("gas-alarm", "children"),
     Output("water-alarm", "children"),
     Output("solar-alarm", "children")],
    [Input("update-interval", "n_intervals")]
)
def update_dashboard(n):
    electricity = data["electricity"][-1] if data["electricity"] else 0
    gas = data["gas"][-1] if data["gas"] else 0
    water = data["water"][-1] if data["water"] else 0
    solar = data["solar"][-1] if data["solar"] else 0

    # Determine alarms
    electricity_alarm = determine_alarm(electricity, alarms["electricity"])
    gas_alarm = determine_alarm(gas, alarms["gas"])
    water_alarm = determine_alarm(water, alarms["water"])
    solar_alarm = determine_alarm(solar, alarms["solar"])

    # Graph
    fig = go.Figure()
    fig.add_trace(go.Scatter(y=list(data["electricity"]), mode="lines", name="Electricity Usage
(kWh)"))
    fig.add_trace(go.Scatter(y=list(data["gas"]), mode="lines", name="Gas Usage (m³)"))
    fig.add_trace(go.Scatter(y=list(data["water"]), mode="lines", name="Water Usage (L)"))
    fig.add_trace(go.Scatter(y=list(data["solar"]), mode="lines", name="Solar Production
(kW)"))
    fig.update_layout(title="Energy Consumption Over Time", xaxis_title="Time",
yaxis_title="Values")

    return (
        electricity,
        gas,
        water,
        solar,
        fig,
        html.Div(f"Electricity Alarm: {electricity_alarm.capitalize()}", style={"color":
alarm_levels[electricity_alarm]}),
        html.Div(f"Gas Alarm: {gas_alarm.capitalize()}", style={"color":
alarm_levels[gas_alarm]}),
        html.Div(f"Water Alarm: {water_alarm.capitalize()}", style={"color":
alarm_levels[water_alarm]}),
```

```
        html.Div(f"Solar Alarm: {solar_alarm.capitalize()}", style={"color":
alarm_levels[solar_alarm]}),
    )

if __name__ == "__main__":
    # Start data simulation in a separate thread
    thread = Thread(target=simulate_data, daemon=True, name="DataSimulationThread")
    thread.start()
    app.run_server(debug=True, port=8051)


# In[ ]:
```

Αρχείο DeviceDataMonitoring.py

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
import random
import time
from dash import Dash, dcc, html, Output, Input
from threading import Thread
from collections import deque
import pandas as pd
import plotly.graph_objs as go
import plotly.express as px

# Global Variables
max_data_points = 60
interval = 1  # Simulate data every second for testing

# Data categories
data = {
    "Lighting Brightness": deque(maxlen=max_data_points),
    "HVAC Temperature": deque(maxlen=max_data_points),
    "Sound Levels": deque(maxlen=max_data_points),
    "Device Online Status": deque(maxlen=max_data_points)
}

# Simulate real-time device data
def simulate_device_data():
    while True:
        data["Lighting Brightness"].append(random.randint(0, 100))  # Percentage
        data["HVAC Temperature"].append(round(random.uniform(18.0, 25.0), 1))  # °C
        data["Sound Levels"].append(random.randint(30, 100))  # Decibels
        data["Device Online Status"].append(random.randint(0, 1))  # Binary status (1: online,
0: offline)
        time.sleep(interval)

# Initialize Dash App
app = Dash(__name__)

# Layout for the app
app.layout = html.Div([
    html.H1("Advanced Device Data Monitoring Dashboard", style={"textAlign": "center",
"marginBottom": "20px"}),
```

```
    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),

    # Device Metric Selector
    html.Div([
        html.Label("Select Parameter:", style={"fontSize": "18px", "marginRight": "10px"}),
        dcc.Dropdown(
            id="parameter-dropdown",
            options=[{"label": param, "value": param} for param in data.keys()],
            value="Lighting Brightness",
            style={"width": "300px", "display": "inline-block"}
        )
    ], style={"textAlign": "center", "marginBottom": "20px"}),

    # Real-Time Data Table
    html.Div([
        html.H3("Latest Device Metrics", style={"textAlign": "center", "marginBottom": "10px"}),
        html.Div(id="data-table", style={"textAlign": "center", "fontSize": "16px"})
    ], style={"marginBottom": "20px"}),

    # Graphs
    html.Div([
        dcc.Graph(id="parameter-graph", style={"width": "60%", "display": "inline-block"}),
        dcc.Graph(id="parameter-pie-chart", style={"width": "35%", "display": "inline-block"}),
    ]),

    # Correlation Heatmap
    html.Div([
        html.H3("Correlation Between Parameters", style={"textAlign": "center", "marginTop":
"20px"}),
        dcc.Graph(id="correlation-heatmap")
    ]),
])

@app.callback(
    [Output("data-table", "children"),
     Output("parameter-graph", "figure"),
     Output("parameter-pie-chart", "figure"),
     Output("correlation-heatmap", "figure")],
    [Input("update-interval", "n_intervals"),
     Input("parameter-dropdown", "value")]
)
def update_dashboard(n, selected_parameter):
    # Create a DataFrame for device metrics
    if any(len(data[key]) == 0 for key in data.keys()):
        # Default empty response
        return html.Div("No data available yet."), go.Figure(), go.Figure(), go.Figure()

    df = pd.DataFrame({metric: list(data[metric]) for metric in data.keys()})

    # Real-Time Data Table
    latest_data = df.iloc[-1].to_dict() if not df.empty else {}
    latest_data_table = html.Table(
        [
            html.Tr([html.Th("Parameter"), html.Th("Value")])
```

```
        ] + [
            html.Tr([html.Td(metric), html.Td(f"{value:.2f}" if isinstance(value, float) else value)])
            for metric, value in latest_data.items()
        ],
        style={
            "margin": "auto",
            "border": "1px solid black",
            "borderCollapse": "collapse",
            "width": "50%",
            "textAlign": "center"
        }
    )

    # Line Graph for Selected Parameter
    line_fig = go.Figure(
        data=[go.Scatter(y=df[selected_parameter], mode="lines+markers",
name=selected_parameter)],
        layout={"title": f"{selected_parameter} Over Time", "xaxis_title": "Time", "yaxis_title":
selected_parameter}
    )

    # Pie Chart for Latest Metric Breakdown
    pie_data = df.iloc[-1] if not df.empty else {}
    pie_fig = px.pie(
        names=pie_data.index, values=pie_data.values,
        title="Proportional Contribution of Parameters"
    ) if not df.empty else go.Figure()

    # Correlation Heatmap
    correlation_matrix = df.corr() if not df.empty else pd.DataFrame()
    heatmap_fig = px.imshow(
        correlation_matrix, x=correlation_matrix.columns, y=correlation_matrix.index,
        color_continuous_scale="Blues", title="Correlation Between Parameters"
    ) if not correlation_matrix.empty else go.Figure()

    return latest_data_table, line_fig, pie_fig, heatmap_fig

# Main Execution
if __name__ == "__main__":
    # Ensure simulation thread runs only once
    thread = Thread(target=simulate_device_data, daemon=True,
name="DeviceSimulationThread")
    thread.start()
    # Run server
    app.run_server(debug=True, port=8057)
# In[ ]:
```

Αρχείο ConnectivityDataMonitoring.py

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
import random
import time
from dash import Dash, dcc, html, Output, Input
from threading import Thread
from collections import deque
```

```
import pandas as pd
import plotly.graph_objs as go
import plotly.express as px

# Global Variables
max_data_points = 60
interval = 1  # Simulate data every second for testing

# Data categories
data = {
    "Device Name": deque(maxlen=max_data_points),
    "Bandwidth Usage": deque(maxlen=max_data_points),  # Mbps
    "Signal Strength": deque(maxlen=max_data_points),  # dBm
    "Online Status": deque(maxlen=max_data_points)  # 1: online, 0: offline
}

devices = ["Smart TV", "Laptop", "Smartphone", "Security Camera", "Smart Thermostat"]

# Simulate real-time connectivity data
def simulate_network_data():
    while True:
        for device in devices:
            data["Device Name"].append(device)
            data["Bandwidth Usage"].append(random.uniform(0.5, 50.0))  # Mbps
            data["Signal Strength"].append(random.uniform(-90.0, -30.0))  # dBm
            data["Online Status"].append(random.choice([0, 1]))  # 1: online, 0: offline
        time.sleep(interval)

# Initialize Dash App
app = Dash(__name__)

# Layout for the app
app.layout = html.Div([
    html.H1("Advanced Connectivity and Network Performance Dashboard",
style={"textAlign": "center", "marginBottom": "20px"}),

    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),

    # Real-Time Data Table
    html.Div([
        html.H3("Current Device Status", style={"textAlign": "center", "marginBottom": "10px"}),
        dcc.Graph(id="data-table")  # Use a plotly table instead of raw HTML
    ], style={"marginBottom": "20px"}),

    # Graphs
    html.Div([
        dcc.Graph(id="bandwidth-bar-graph", style={"width": "50%", "display": "inline-block"}),
        dcc.Graph(id="online-status-pie-chart", style={"width": "45%", "display": "inline-block"}),
    ]),

    html.Div([
        dcc.Graph(id="scatter-plot-graph", style={"width": "50%", "display": "inline-block"}),  #
Scatter plot
        dcc.Graph(id="correlation-heatmap", style={"width": "45%", "display": "inline-block"})
    ]),
```

```
])

@app.callback(
    [Output("data-table", "figure"),
     Output("bandwidth-bar-graph", "figure"),
     Output("online-status-pie-chart", "figure"),
     Output("scatter-plot-graph", "figure"),
     Output("correlation-heatmap", "figure")],
    [Input("update-interval", "n_intervals")]
)
def update_dashboard(n):
    # Ensure data is populated
    if any(len(data[key]) == 0 for key in data.keys()):
        return go.Figure(), go.Figure(), go.Figure(), go.Figure(), go.Figure()

    # Create a DataFrame for connectivity metrics
    df = pd.DataFrame({
        "Device Name": list(data["Device Name"]),
        "Bandwidth Usage": list(data["Bandwidth Usage"]),
        "Signal Strength": list(data["Signal Strength"]),
        "Online Status": list(data["Online Status"]),
    })

    # Real-Time Data Table
    table_fig = go.Figure(
        data=[go.Table(
            header=dict(values=list(df.columns),
                        fill_color='paleturquoise',
                        align='left'),
            cells=dict(values=[df[col] for col in df.columns],
                       fill_color='lavender',
                       align='left'))
        ]
    )
    table_fig.update_layout(title="Current Device Status")

    # Bandwidth Bar Graph
    bandwidth_fig = px.bar(
        df.groupby("Device Name")["Bandwidth Usage"].mean().reset_index(),
        x="Device Name",
        y="Bandwidth Usage",
        title="Average Bandwidth Usage by Device",
        color="Device Name"
    )

    # Online/Offline Status Pie Chart
    online_status_count = df["Online Status"].value_counts()
    pie_fig = px.pie(
        names=["Online", "Offline"], values=online_status_count,
        title="Device Online/Offline Status"
    )

    # Scatter Plot with Trendline
    scatter_fig = px.scatter(
        df,
```

```
        x="Bandwidth Usage",
        y="Signal Strength",
        title="Bandwidth vs Signal Strength",
        color="Device Name",
        trendline="ols"  # Add a trendline for better visualization
    )

    # Correlation Heatmap
    correlation_matrix = df[["Bandwidth Usage", "Signal Strength", "Online Status"]].corr()
    heatmap_fig = px.imshow(
        correlation_matrix, x=correlation_matrix.columns, y=correlation_matrix.index,
        color_continuous_scale="Blues", title="Correlation Between Parameters"
    )

    return table_fig, bandwidth_fig, pie_fig, scatter_fig, heatmap_fig

# Main Execution
if __name__ == "__main__":
    # Ensure simulation thread runs only once
    thread = Thread(target=simulate_network_data, daemon=True,
name="NetworkSimulationThread")
    thread.start()

    # Run server
    app.run_server(debug=True, port=8059)

# In[ ]:
```

Αρχείο ApplianceMonitoring.py

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import random
import time
from dash import Dash, dcc, html, Output, Input
from threading import Thread
from collections import deque
import plotly.graph_objs as go
import pandas as pd
# Global Variables
max_data_points = 60
interval = 1  # 1 second for simulation during testing
# Appliance Data
appliances = ["Washing Machine", "Refrigerator", "Oven", "Dishwasher", "Microwave"]
data = {
    appliance: {
        "operational_status": deque(maxlen=max_data_points),
        "cycle_progress": deque(maxlen=max_data_points),
        "energy_consumption": deque(maxlen=max_data_points),
        "efficiency": deque(maxlen=max_data_points)
    }
    for appliance in appliances
```

```
}
def simulate_appliance_data():
    """Simulate data for all appliances."""
    while True:
        for appliance in appliances:
            data[appliance]["operational_status"].append(random.choice([0, 1]))  # 0: Off, 1: On
            data[appliance]["cycle_progress"].append(random.randint(0, 100))  # % of progress
            data[appliance]["energy_consumption"].append(random.uniform(50, 300))  # kWh
            data[appliance]["efficiency"].append(random.uniform(75, 100))  # %
        time.sleep(interval)
# Initialize Dash App
app = Dash(__name__)
# Layout for the app
app.layout = html.Div([
    html.H1("Advanced Appliance Monitoring Dashboard", style={"textAlign": "center",
"marginBottom": "20px"}),
    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),
    # Appliance Dropdown
    html.Div([
        html.Label("Select Appliance:", style={"fontSize": "18px", "marginRight": "10px"}),
        dcc.Dropdown(
            id="appliance-dropdown",
            options=[{"label": appliance, "value": appliance} for appliance in appliances],
            value=appliances[0],
            style={"width": "300px", "display": "inline-block"}
        )
    ], style={"textAlign": "center", "marginBottom": "20px"}),
    # Appliance Status
    html.Div([
        html.Div(id="appliance-status", style={"textAlign": "center", "fontSize": "20px",
"marginBottom": "20px"}),
    ]),
    # Graphs
    html.Div([
        dcc.Graph(id="cycle-progress-graph", style={"width": "48%", "display": "inline-block"}),
        dcc.Graph(id="energy-consumption-graph", style={"width": "48%", "display": "inline-
block"}),
    ]),
    html.Div([
        dcc.Graph(id="efficiency-graph", style={"width": "48%", "display": "inline-block"})
    ]),
    # Data Table
    html.Div([
        html.H3("Appliance Summary Table", style={"textAlign": "center", "marginBottom":
"10px"}),
        html.Div(id="appliance-table")
    ], style={"marginTop": "20px"})
])
@app.callback(
    [Output("appliance-status", "children"),
     Output("cycle-progress-graph", "figure"),
     Output("energy-consumption-graph", "figure"),
     Output("efficiency-graph", "figure"),
     Output("appliance-table", "children")],
    [Input("update-interval", "n_intervals"),
```

```
    Input("appliance-dropdown", "value")]
)
def update_dashboard(n, selected_appliance):
    # Get data for the selected appliance
    operational_status = data[selected_appliance]["operational_status"][-1] if
data[selected_appliance]["operational_status"] else 0
    cycle_progress = list(data[selected_appliance]["cycle_progress"])
    energy_consumption = list(data[selected_appliance]["energy_consumption"])
    efficiency = list(data[selected_appliance]["efficiency"])
    # Status Message
    status_message = f"{selected_appliance} is {'On' if operational_status else 'Off'}."
    # Cycle Progress Graph
    cycle_progress_fig = go.Figure(
        data=[go.Scatter(y=cycle_progress, mode="lines+markers", name="Cycle Progress")],
        layout={"title": "Cycle Progress (%)", "xaxis_title": "Time", "yaxis_title": "Progress (%)"}
    )
    # Energy Consumption Graph
    energy_consumption_fig = go.Figure(
        data=[go.Scatter(y=energy_consumption, mode="lines+markers", name="Energy
Consumption")],
        layout={"title": "Energy Consumption (kWh)", "xaxis_title": "Time", "yaxis_title": "Energy
(kWh)"}
    )

    # Efficiency Graph
    efficiency_fig = go.Figure(
        data=[go.Scatter(y=efficiency, mode="lines+markers", name="Efficiency")],
        layout={"title": "Efficiency (%)", "xaxis_title": "Time", "yaxis_title": "Efficiency (%)"}
    )
    # Data Table
    table_data = []
    for appliance in appliances:
        table_data.append({
            "Appliance": appliance,
            "Operational Status": "On" if data[appliance]["operational_status"][-1] else "Off",
            "Last Cycle Progress (%)": data[appliance]["cycle_progress"][-1] if
data[appliance]["cycle_progress"] else 0,
            "Last Energy Consumption (kWh)": round(data[appliance]["energy_consumption"][-1],
2) if data[appliance]["energy_consumption"] else 0,
            "Last Efficiency (%)": round(data[appliance]["efficiency"][-1], 2) if
data[appliance]["efficiency"] else 0
        })
    df = pd.DataFrame(table_data)
    table = html.Table([
        html.Thead(html.Tr([html.Th(col) for col in df.columns])),
        html.Tbody([
            html.Tr([html.Td(df.iloc[i][col]) for col in df.columns]) for i in range(len(df))
        ])
    ], style={"width": "100%", "margin": "auto", "border": "1px solid black", "textAlign":
"center"})
    return status_message, cycle_progress_fig, energy_consumption_fig, efficiency_fig, table
# Main Execution
if __name__ == "__main__":
    # Ensure simulation thread runs only once
```

```
    thread = Thread(target=simulate_appliance_data, daemon=True,
name="ApplianceSimulationThread")
    thread.start()
    # Run server
    app.run_server(debug=True, port=8054)
# In[ ]:
```

Αρχείο AdvancedHealthMonitoring.py

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:
import random
import time
from dash import Dash, dcc, html, Output, Input
from threading import Thread
from collections import deque
import pandas as pd
import plotly.graph_objs as go
import plotly.express as px
# Global Variables
max_data_points = 60
interval = 1  # Simulate data every second for testing
# Data categories
data = {
    "Heart Rate": deque(maxlen=max_data_points),
    "Body Temperature": deque(maxlen=max_data_points),
    "Activity Levels": deque(maxlen=max_data_points),
    "Sleep Quality": deque(maxlen=max_data_points),
}

# Simulate real-time health data
def simulate_health_data():
    while True:
        data["Heart Rate"].append(random.randint(60, 100))  # Beats per minute
        data["Body Temperature"].append(round(random.uniform(36.0, 37.5), 1))  # °C
        data["Activity Levels"].append(random.uniform(0, 100))  # Scale 0-100
        data["Sleep Quality"].append(random.uniform(0, 100))  # Scale 0-100
        time.sleep(interval)
# Initialize Dash App
app = Dash(__name__)
# Layout for the app
app.layout = html.Div([
    html.H1("Advanced Health Data Monitoring Dashboard", style={"textAlign": "center",
"marginBottom": "20px"}),
    dcc.Interval(id="update-interval", interval=1000, n_intervals=0),
    # Health Metric Selector
    html.Div([
        html.Label("Select Metric:", style={"fontSize": "18px", "marginRight": "10px"}),
        dcc.Dropdown(
            id="metric-dropdown",
            options=[{"label": metric, "value": metric} for metric in data.keys()],
            value="Heart Rate",
            style={"width": "300px", "display": "inline-block"}
        )
    ], style={"textAlign": "center", "marginBottom": "20px"}),
    # Real-Time Data Table
```

```
    html.Div([
        html.H3("Latest Health Metrics", style={"textAlign": "center", "marginBottom": "10px"}),
        html.Div(id="data-table", style={"textAlign": "center", "fontSize": "16px"})
    ], style={"marginBottom": "20px"}),
    # Graphs
    html.Div([
        dcc.Graph(id="metric-graph", style={"width": "60%", "display": "inline-block"}),
        dcc.Graph(id="metric-pie-chart", style={"width": "35%", "display": "inline-block"}),
    ]),
    # Correlation Heatmap
    html.Div([
        html.H3("Correlation Between Metrics", style={"textAlign": "center", "marginTop":
"20px"}),
        dcc.Graph(id="correlation-heatmap")
    ]),
])
@app.callback(
    [Output("data-table", "children"),
     Output("metric-graph", "figure"),
     Output("metric-pie-chart", "figure"),
     Output("correlation-heatmap", "figure")],
    [Input("update-interval", "n_intervals"),
     Input("metric-dropdown", "value")]
)
def update_dashboard(n, selected_metric):
    # Create a DataFrame for health metrics
    if any(len(data[key]) == 0 for key in data.keys()):
        # Default empty response
        return html.Div("No data available yet."), go.Figure(), go.Figure(), go.Figure()
    df = pd.DataFrame({metric: list(data[metric]) for metric in data.keys()})
    # Real-Time Data Table
    latest_data = df.iloc[-1].to_dict() if not df.empty else {}
    latest_data_table = html.Table(
        [
            html.Tr([html.Th("Metric"), html.Th("Value")])
        ] + [
            html.Tr([html.Td(metric), html.Td(f"{value:.2f}" if isinstance(value, float) else value)])
            for metric, value in latest_data.items()
        ],
        style={
            "margin": "auto",
            "border": "1px solid black",
            "borderCollapse": "collapse",
            "width": "50%",
            "textAlign": "center"
        }
    )
    # Line Graph for Selected Metric
    line_fig = go.Figure(
        data=[go.Scatter(y=df[selected_metric], mode="lines+markers",
name=selected_metric)],
        layout={"title": f"{selected_metric} Over Time", "xaxis_title": "Time", "yaxis_title":
selected_metric}
    )
    # Pie Chart for Latest Metric Breakdown
```

```
    pie_data = df.iloc[-1] if not df.empty else {}
    pie_fig = px.pie(
        names=pie_data.index, values=pie_data.values,
        title="Proportional Contribution of Metrics"
    ) if not df.empty else go.Figure()
    # Correlation Heatmap
    correlation_matrix = df.corr() if not df.empty else pd.DataFrame()
    heatmap_fig = px.imshow(
        correlation_matrix, x=correlation_matrix.columns, y=correlation_matrix.index,
        color_continuous_scale="Blues", title="Correlation Between Metrics"
    ) if not correlation_matrix.empty else go.Figure()
    return latest_data_table, line_fig, pie_fig, heatmap_fig

# Main Execution
if __name__ == "__main__":
    # Ensure simulation thread runs only once
    thread = Thread(target=simulate_health_data, daemon=True,
name="HealthSimulationThread")
    thread.start()
    # Run server
    app.run_server(debug=True, port=8056)
# In[ ]:
```