

WATERGAME

Raport științific

Etapa 3 – Integrarea, verificarea și validarea sistemului

1	CONSIDERAȚII GENERALE.....	1
2	INTEGRAREA SISTEMULUI.....	3
2.1	Integrarea componentelor de monitorizare și suport decizional	5
2.1.1	Metoda de colectare a datelor și detecție a anomaliilor	7
2.1.2	Metoda de grupare a datelor	16
2.1.3	Metoda de clasificare a datelor.....	17
2.2	Integrarea componentelor de raportare urbană și Blockchain.....	19
2.2.1	Integrarea aplicației de raportare urbană	20
2.2.2	Integrarea componentei Blockchain	21
3	EXPERIMENTAREA SISTEMULUI	26
3.1	Monitorizarea sistemului de colectare a datelor de la senzori	26
3.2	Raportarea anomaliilor în rețeaua de senzori	28
3.2.1	Detecția scurgerilor	29
3.2.2	Detecția senzorilor defecti	29
3.2.3	Notificarea defectiunilor	30
3.2.4	Citirea datelor de la edge device	31
3.2.5	Detecția comportamentului anormal.....	32
3.2.6	Analiza datelor	33
3.3	Vizualizare și suport decizional în rețeaua de senzori	42
3.3.1	Gruparea modelelor de consum	45
3.3.2	Clasificarea evenimentelor de consum.....	47
3.4	Raportarea evenimentelor în infrastructura de apă	48
3.4.1	Arhitectura jocului serios	48
3.4.2	Evaluarea scalabilității	49
3.5	Blockchain.....	51
3.5.1	Setarea pre-rechizitelor pentru experimentarea soluției blockchain	51
3.5.2	Experimentarea soluției blockchain	53
4	VALIDAREA SISTEMULUI ȘI DEMONSTRAREA FUNCȚIONALITĂȚILOR	55
4.1	Verificarea funcționării sistemului integrat	55
4.1.1	Monitorizare în rețeaua de senzori	55
4.1.2	Vizualizare și suport decizional în rețeaua de senzori	62

4.1.3	Raportarea evenimentelor cu suport blockchain	63
4.2	Evaluarea rezultatelor experimentale obținute din funcționarea sistemului.....	69
4.2.1	Monitorizarea consumului de apă.....	69
4.2.2	Monitorizare și suport decizional	72
4.2.3	Raportarea evenimentelor cu suport blockchain	76
5	CONCLUZII.....	79
	BIBLIOGRAFIE	81

1 Considerații generale

Apa reprezintă o resursă fundamentală pentru majoritatea domeniilor de activitate, iar în situația internațională actuală este nevoie de o implicare mult mai mare atât din partea autorităților, cât și a populației în general cu scopul minimizării pierderilor de apă și a utilizării raționale a acesteia.

În acest context, proiectul Watergame a propus realizarea unui sistem cyber-fizico-social pentru gestionarea resurselor de apă în comunități urbane, care să integreze tehnologii informatice avansate pentru a crește colaborarea și participarea cetățenilor la rezolvarea problemelor legate de apă, ca parte a unei strategii guvernamentale inteligente.

În cadrul proiectului s-a urmărit dezvoltarea unor sisteme pilot care să identifice consumurile de apă din interiorul unei gospodării/apartament cu scopul asigurării suportului pentru identificarea rapidă a unor incidente în rețeaua de apă și, de asemenea, cu scopul permiterii predicției cu o precizie ridicată a consumurilor de apă. Aceste predicții pot fi folosite ulterior pentru un management în timp real mai bun al resurselor de apă necesare la diverse momente de timp ale zilei, precum și la calibrarea infrastructurii și planificarea realizării mentenanței acesteia astfel încât să afecteze cât mai puțini utilizatori.

Tot în ceea ce privește infrastructura, este nevoie și de contribuția și implicarea directă a consumatorilor în identificarea rapidă a problemelor ce pot fi ușor observabile de către aceștia, dar mult mai dificil sau lent de determinat de către autoritățile responsabile de administrarea ei. În acest scop, proiectul de față propune un sistem colaborativ de colectare a informațiilor ce are la bază conceptul de Crowdsensing și Serious Gaming.

Întrucât pentru a fi adoptat un sistem este nevoie ca principalii participanți să aibă încredere în acest sistem, este prevăzută o componentă bazată pe Blockchain, în care informațiile sunt reprezentate sub formă de contracte inteligente și tranzacții în Blockchain.

În etapele anterioare ale proiectului au fost identificate scenariile în care ar putea fi utilizat un astfel de sistem cyber-fizico-social, după care au fost construite componentele individuale necesare pentru realizarea sistemului propus. Drept urmare, în acest raport vom prezenta modul în care aceste componente au fost integrate unele cu altele pentru asigurarea funcționalităților sistemului, precum și modul în care a fost validat sistemul integrat și rezultatele experimentale care s-au obținut folosind componentele sistemului informatic.

Astfel, în următorul capitol (Capitolul 2) vom descrie modul în care s-au integrat serviciile de preluare a datelor și de monitorizare a infrastructurii din interiorul casei cu cele de suport decizional, care permit identificarea anomaliilor și clasificarea acestora, precum și a șabloanelor de consum specifice fiecărui consumator și a necesităților de livrare de apă la diferite momente ale zilei. Tot aici, prezentăm modul în care serviciile de crowdsensing sunt folosite pentru identificarea problemelor la infrastructura outdoor și modul în care aceste informații sunt transmise pe Blockchain ca urmare a interconectării celor două componente.

Capitolul 3 prezintă experimentele efectuate după integrarea componentelor sistemului construit în ceea ce privește monitorizarea senzorilor, analiza anomaliilor și clasificarea acestora, vizualizarea diferitelor șabloane de consum (în funcție de consumatori sau de momentul de timp al zilei) în cadrul modulului de suport al deciziilor, raportarea evenimentelor urbane și scalabilitatea acestor raportări

odată cu creșterea numărului de utilizatori / evenimente raportate precum și a tranzacțiilor pe Blockchain.

În Capitolul 4 prezentăm pașii parcurși în scopul validării funcționării sistemului integrat, precum și al evaluării rezultatelor experimentale obținute.

Raportul se încheie cu prezentarea concluziilor finale ale proiectului.

2 Integrarea sistemului

Integrarea sistemului este ilustrată prin diagrama de activități din Figura 1. Componentele sistemului au fost dezvoltate independent și interacționează în contextul unor activități bine definite. Astfel, sistemul WMS – monitorizare a consumului de apă, se integrează cu sistemul DSS – suport decizional, la nivel de date, iar sistemul DSS se integrează cu sistemul Blockchain tot la nivel de date. Sistemul CS – raportare urbană, se integrează cu sistemul Blockchain la nivel de aplicație, cu privire la recompensarea interacțiunilor. Integrarea dintre DSS și CS este indirectă, prin intermediul monedei virtuale emise la nivelul sistemului Blockchain, neexistând altfel interacțiuni directe între cele două componente.

Figura 1 mai evidențiază corespondența dintre componentele sistemului informatic integrat și aplicațiile dezvoltate.

Astfel, aplicația WaterMonitoringSystem – WMS App (UPB) realizează interfața cu utilizatorul, i.e., consumatorul din rețeaua de distribuție a apei, și interacționează cu sistemul de monitorizare (WMS) și de suport decizional (DSS). Aplicația WaterMonitoringSystem preia datele de la sistemul de monitorizare a consumului de apă, și implementează vizualizarea pe hartă a senzorilor instalați în locuințe, precum și consumul de apă în timp real.

Aplicația AutomaticAI – WMS App (UTCN) realizează interfața cu operatorul de date și interacționează cu sistemul de suport decizional (DSS), cu privire la introducerea, procesarea, și validarea datelor. Sistemul DSS presupune un set de algoritmi de procesare a datelor, care sunt apelabili din interfață și implementează scenariile propuse pentru extragerea informațiilor relevante în contextul metodelor de suport decizional.

Aplicația Watergame – Watergame Crowdsensing App (UPB), realizează interfața cu utilizatorul care poate raporta incidente în contextul infrastructurii de apă la nivel urban, și interacționează cu sistemul Blockchain la nivel de aplicație. Integrarea presupune accesarea contului din rețeaua blockchain și realizarea de operațiuni cu moneda virtuală WGT (Watergame Token).

Aplicația Watergame Token App (UPB) prezintă o interfață de control pentru gestionarea contului utilizatorului în rețeaua blockchain, cu funcționalități specifice.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

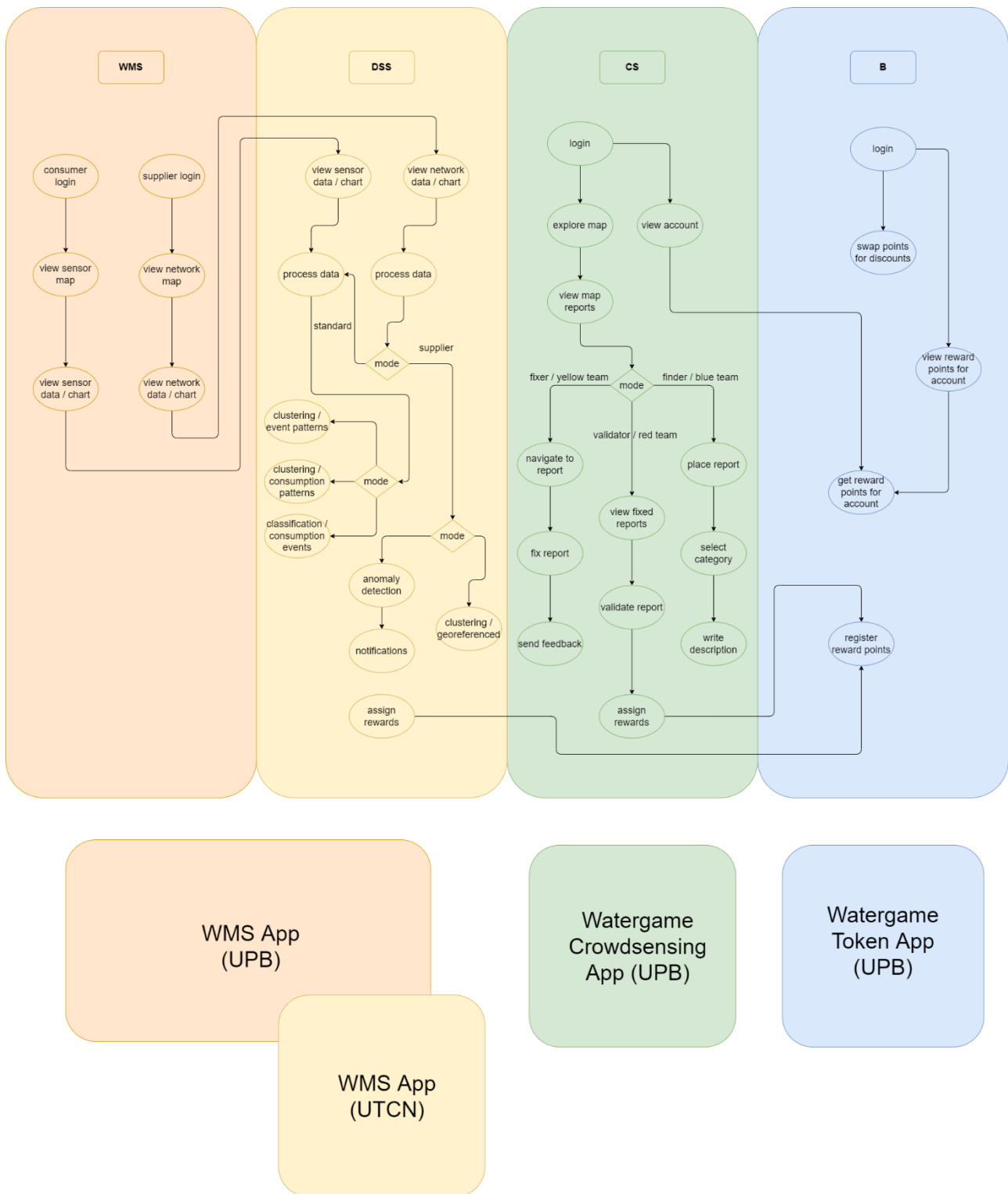


Figura 1 Diagrama de activități a sistemului integrat

2.1 Integrarea componentelor de monitorizare și suport decizional

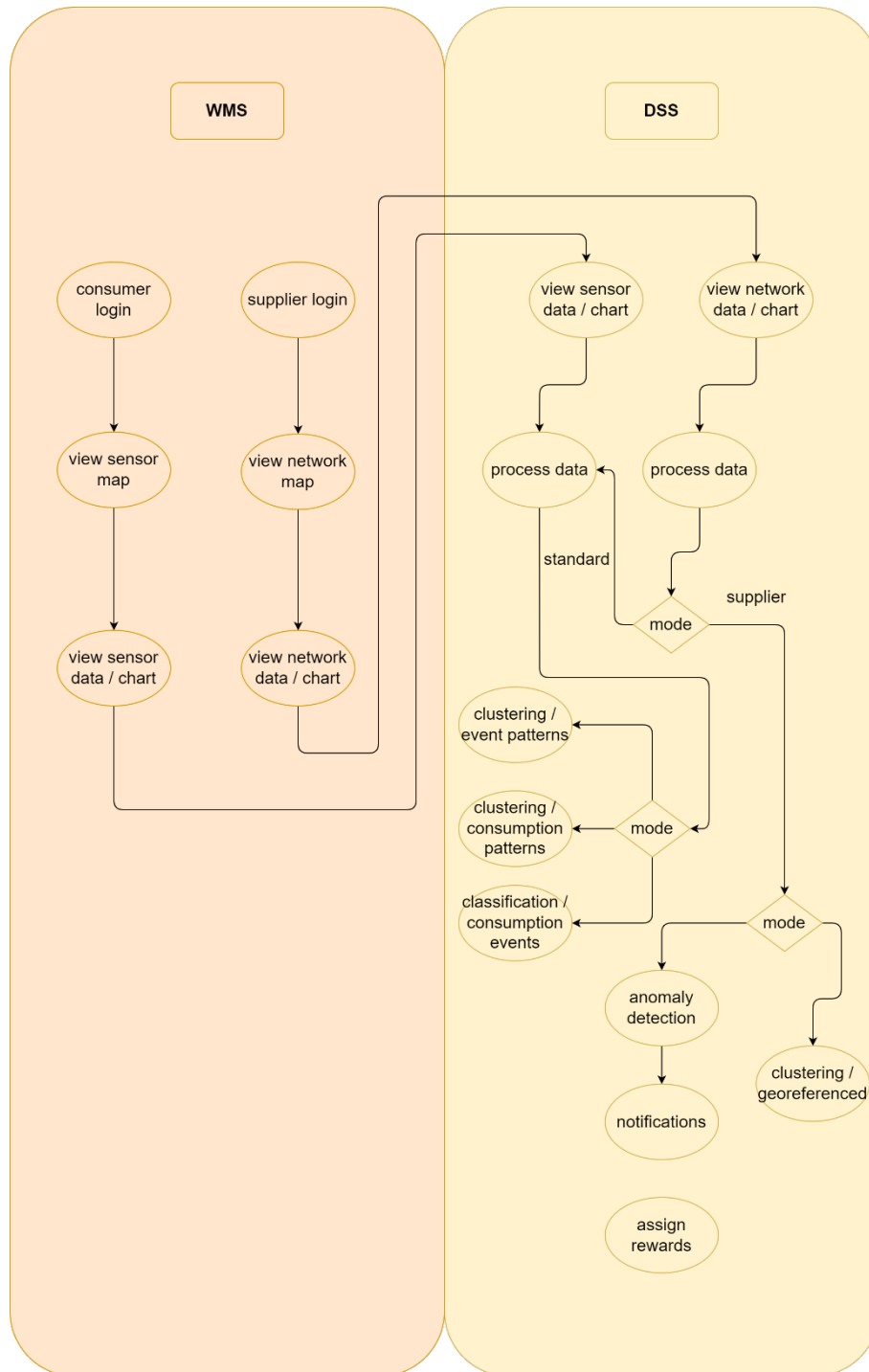


Figura 2 Diagrama de activități a sistemului integrat pentru monitorizare și suport decizional

Integrarea componentelor WMS și DSS este ilustrată în diagrama de activități din Figura 2. În contextul sistemului de monitorizare a consumului de apă (WMS), sunt disponibile metode de

accesare și vizualizare a datelor preluate de la senzorii instalați în locuințe: accesare date în timp real sau stocate, vizualizare pe hartă a senzorilor instalați, vizualizare sub formă de grafice. Sistemul de suport decizional (DSS) este alcătuit dintr-un set de algoritmi de procesare a datelor pentru implementarea diferitelor scenarii propuse: gruparea și clasificarea datelor, detecția anomaliilor, și notificări la nivel de sistem.

Arhitectura sistemului propus este descrisă mai jos, după cum urmează:

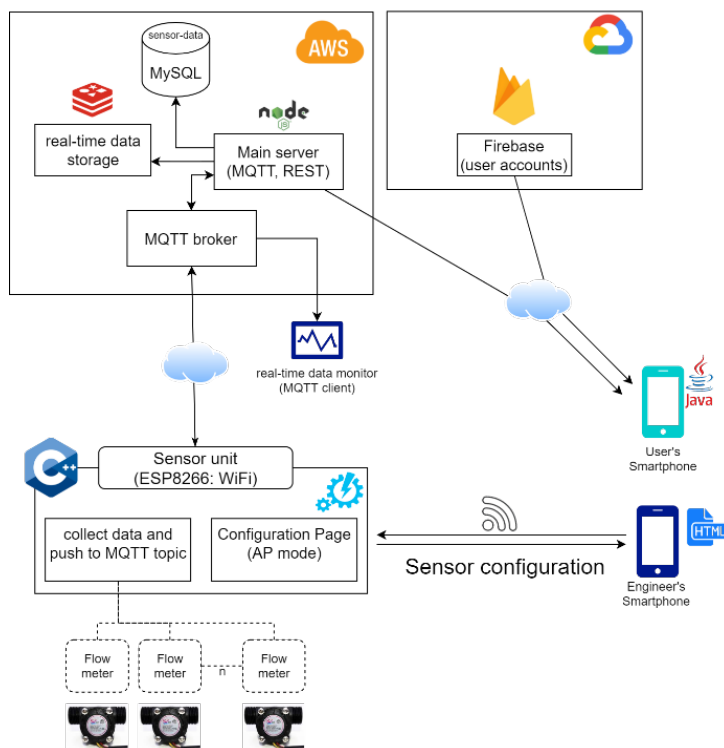


Figura 3 Arhitectura sistemului de monitorizare a consumului de apă

În procesul de colectare a datelor legate de consumul de apă, rețeaua de senzori este integrată într-o arhitectură bazată pe servicii Cloud, așa cum se arată în Figura 3. Colectarea datelor se face cu ajutorul unei plăci de dezvoltare NodeMCU bazată pe microcontrolerul ESP8266. Acesta conține comunicare Wi-Fi, pini GPIO (intrare/ieșire de uz general) și poate fi programat în mediul Arduino (Parihar Y., 2019).

Debitmetrele YF-S201 oferă o ieșire digitală bazată pe un senzor cu efect Hall, care este măsurat folosind temporizatoare și întreruperi, fiind convertit în unități de debit, adică L/h. Debitmetrele sunt instalate în mai multe locații unde se folosesc tuburi flexibile pentru a furniza apă pentru prizele casnice, de exemplu, chiuvetă, duș, toaletă.

Interfața este expusă printr-un server găzduit direct pe microcontrolerul ESP8266 configurat în același timp în modul stație și client Wi-Fi. Colectarea datelor se face printr-un broker de mesaje MQTT, care permite o arhitectură decuplată printr-un protocol de comunicare standard, adaptat domeniului IoT (Mishra B. et al., 2020). Serverul de aplicații se conectează la brokerul MQTT pentru a primi datele trimise de senzori și interfațează cu baza de date MySQL (Bell C., 2016). Datele

colectate sunt salvate periodic în baza de date conform intervalului configurat la nivel de dispozitiv IoT.

Există două tipare pentru transmiterea datelor: date de consum instantanee (monitorizare în timp real, de înaltă frecvență), date de volum (înregistrate în baza de date, cu o frecvență mai mică). Datele pot fi accesate în timp real folosind un client MQTT (de exemplu, aplicație desktop, module IoT), iar conexiunea este securizată pe baza acreditărilor de acces la nivel de broker. Pentru disponibilitatea datelor pentru vizualizare în timp real se adaugă un modul REDIS pentru stocarea temporară a ultimelor măsurători culese pe tema consumului instantaneu (Chen S. et al., 2016). Datele consumatorului și starea senzorului sunt prezentate clientului folosind o aplicație mobilă (Android). Autentificarea și gestionarea contului pentru utilizatori se realizează prin serviciul Firebase, iar datele de consum sunt încărcate prin solicitări HTTP de la serverul principal.

În ceea ce privește front-endul aplicației mobile dezvoltate, acesta este împărțit în două părți, în funcție de tipul de utilizator.

Rolurile disponibile în aplicația mobilă sunt următoarele:

- Furnizori - persoane responsabile cu distribuția resurselor de apă în mediul urban;
- Consumatori - persoane care consumă resursele de apă distribuite.

Front-endul aplicației include diferite funcționalități în funcție de rolul persoanei care s-a autentificat în aplicație.

În contextul furnizorilor, funcționalitățile aplicației includ:

- monitorizarea tuturor senzorilor de rețea;
- adăugarea unui nou senzor;
- asocierea unui senzor unui consumator (procesul de conectare - unul sau mai mulți senzori sunt alocați pe baza codului unic al consumatorului);
- primirea notificărilor de la clienți cu privire la anumite probleme care pot apărea în rețeaua de apă.

Consumatorii beneficiază de funcționalitățile aplicației astfel:

- monitorizarea propriilor senzori;
- vizualizarea și actualizarea datelor personale și de autentificare;
- vizualizarea rapoartelor lunare privind consumul de apă;
- raportarea problemelor către furnizor prin notificări.

În ceea ce privește profilurile de consumator, aceștia pot fi împărțiți în trei clase principale: singur, familial, multifamilial. În plus, există o varietate de profiluri de consumatori în funcție de vârste și ocupații diferite. De exemplu: copii, persoane în vârstă, adulți care lucrează full-time și astfel consumă apă doar dimineața și seara sau adulți care au joburi remote și petrec mai mult timp acasă, aceștia din urmă având activități consumatoare de apă pe toată perioada zilei.

2.1.1 Metoda de colectare a datelor și detecție a anomaliilor

Anomaliile care pot apărea în infrastructurile de apă variază de la probleme de calitate a apei la probleme de infrastructură, cum ar fi funcționarea defectuoasă a contoarelor, senzorilor sau scurgerile de apă, până la perturbarea tiparelor de consum ale utilizatorilor. Aceste anomalii pot fi

detectate prin analiza datelor colectate de la contoare inteligente, de la senzori care măsoară diferiți parametri legați de calitatea apei (cum ar fi valoarea pH, conductivitate, turbiditate etc.) și de la alte tipuri de senzori instalați de-a lungul rețelei de distribuție a apei.

Sistemul de colectare, analiză și raportare a anomaliilor este structurat pe trei straturi: Monitoring, Edge și Cloud.

La baza arhitecturii se află nivelul de Monitorizare, care include punctele de colectare a datelor (contoare inteligente) instalate la locația consumatorului și măsoară consumul instant de apă. Contoarele inteligente au fost implementate cu debitmetre de apă YF-S201 (optimusdigital.ro, 2022) atașate la plăcile NodeMCU (Nodemcu.readthedocs.io, 2022) echipate cu microcipuri wifi ESP8266 (espressif.com, 2022) (așa cum se vede în Figura 4). Mai multe astfel de puncte de colectare a datelor pot fi instalate într-o singură locație. Datele neprocesate sunt trimise periodic, în pachete, către dispozitivele Edge.

Nivelul Edge cuprinde dispozitive cu capacitate redusă de stocare și procesare, care aplică algoritmi de preprocesare asupra datelor primite de la contoarele inteligente. Datele preprocesate sunt apoi trimise către serviciul de colectare a datelor din Cloud pentru stocare pe termen lung și procesare offline. La Edge, datele sunt stocate într-o fereastră de timp configurată în funcție de capacitatea de stocare a dispozitivului utilizat. Pentru experimentele noastre inițiale, am folosit un dispozitiv Nvidia Jetson Nano (developer.nvidia.com, 2022). Acest dispozitiv are un CPU de tip Quad-core ARM A57 cu o frecvență de 1.43 GHz, cu o memorie de 4GB LPDDR4, spațiu de stocare folosind un microSD de 64GB și un GPU de tip Maxwell 128-core.

La nivel de Cloud, există un serviciu de colectare a datelor și un serviciu avansat de procesare a datelor (offline). Serviciul de colectare a datelor primește date preprocesate de la dispozitivele Edge și le stochează într-o bază de date standard OGC SensorThings (fraunhoferiosb.github.io, 2022). Serviciul de prelucrare a datelor poate aplica proceduri complexe de preprocesare și analiză, folosind algoritmi de învățare automată asupra datelor stocate de la serviciul de colectare a datelor pentru consumatori sau pe seturi de date din alte surse. Serviciul de procesare a datelor are atașată o interfață web care permite utilizatorului să încarce seturi de date, să configureze și să ruleze mai mulți algoritmi de procesare a datelor și să vizualizeze rezultatele analizei datelor. Serviciul de analiză a datelor poate suporta sarcini periodice de analiză a datelor care emit notificări de anomalii către sisteme externe, cum ar fi alte dispozitive sau aplicații mobile.

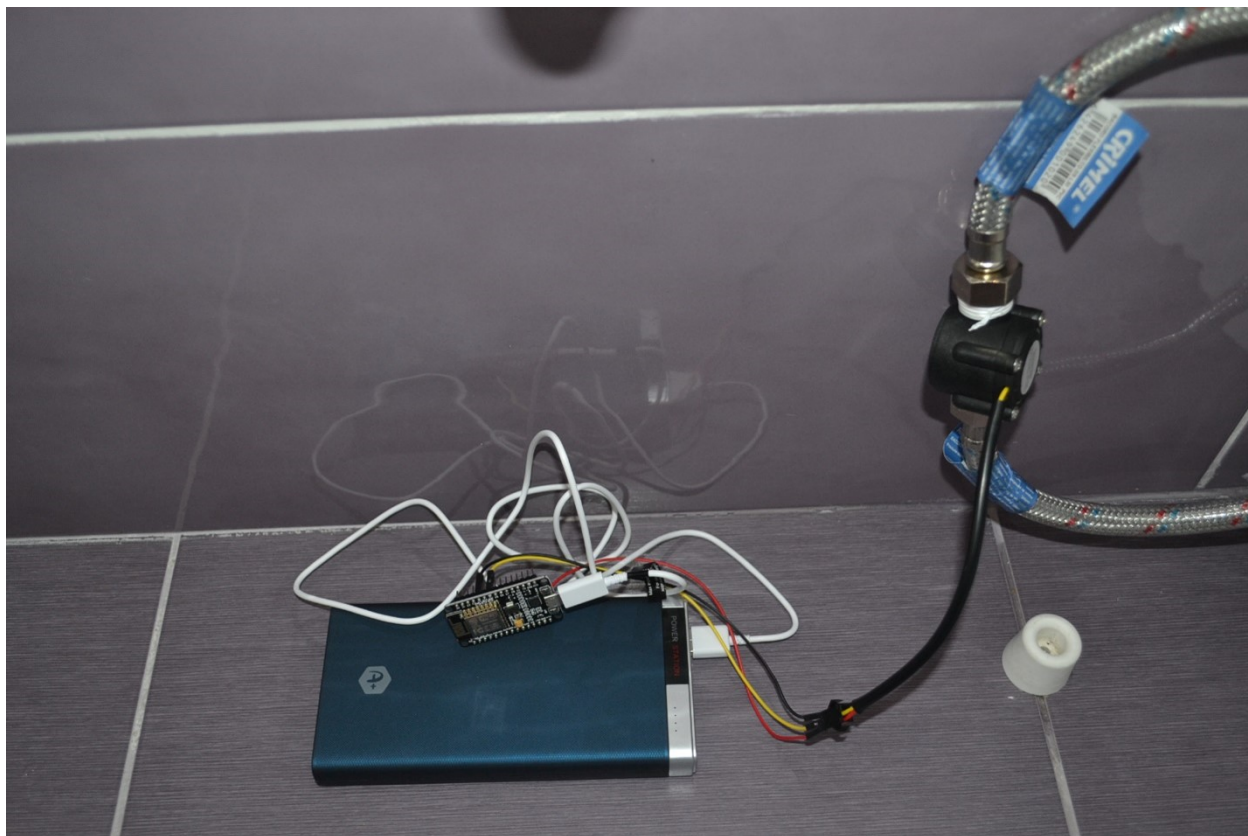


Figura 4 Contor inteligent de consum de apă

Figura 5 reprezintă o viziune de ansamblu de nivel înalt a sistemului descris mai sus.

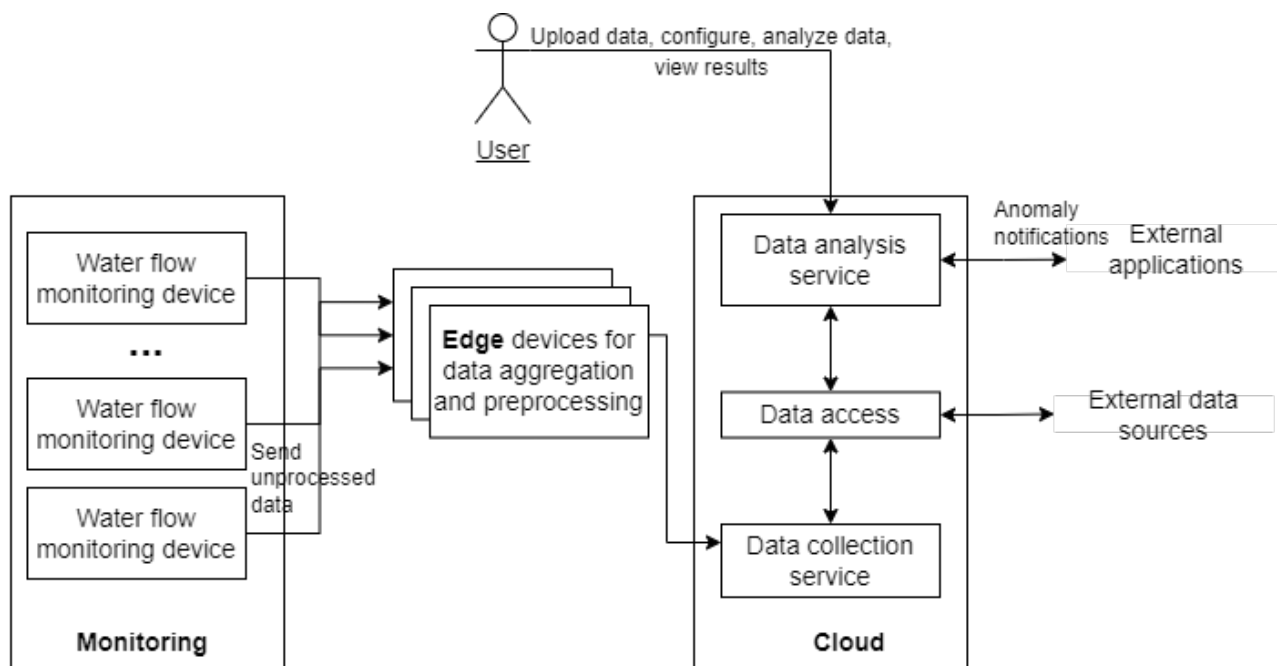


Figura 5 Arhitectura sistemului de monitorizare a consumului de apă și de detectare a anomaliilor

Colectarea și stocarea datelor

Serviciul de colectare a datelor are două componente (vezi Figura 6):

1. API de colectare a datelor (API-ul OGC SensorThings)
2. Baza de date pentru stocarea datelor

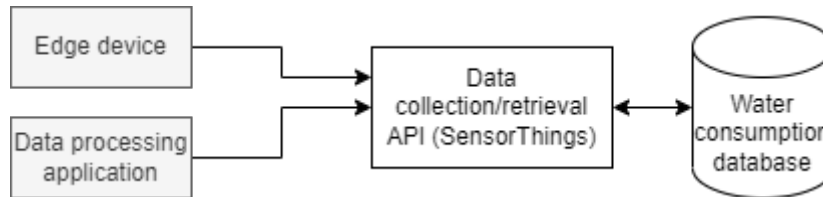


Figura 6 Serviciul de colectare a datelor

API-ul poate fi accesat de dispozitivele Edge și alte aplicații care procesează date prin solicitări HTTP sau MQTT. Solicitățile trimise către API și răspunsurile acestuia utilizează date împachetate în format JSON care respectă modelul de date OGC SensorThings (Figura 7).

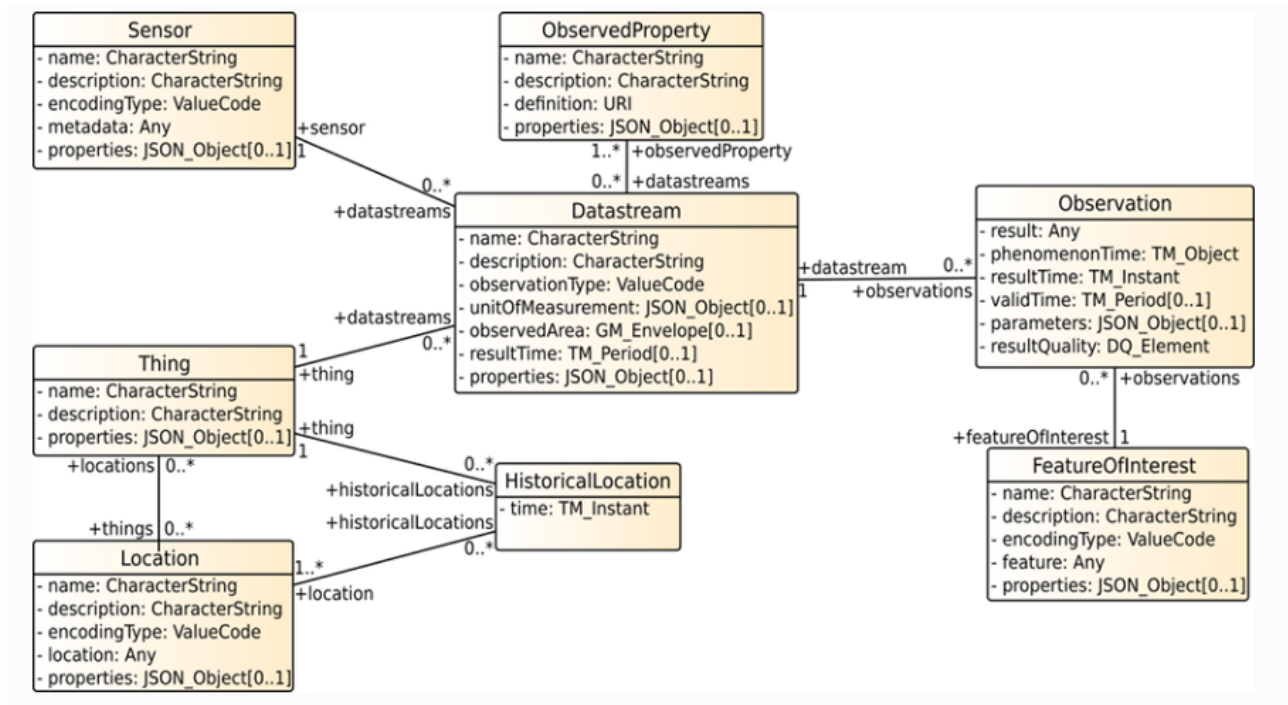


Figura 7 Diagrama bazei de date OGC SensorThings

Un dispozitiv sau un sistem IoT este modelat ca un lucru (Thing). Un lucru are un număr arbitrar de locații și un număr arbitrar de fluxuri de date (Datastream). Fiecare flux de date observă o proprietate observată (ObservedProperty) cu un senzor (Sensor) și are mai multe observații (Observations) colectate de senzor. În cazul nostru, am mapat o gospodărie la un lucru, în timp ce diferitele puncte de monitorizare (contoare inteligente) din interiorul unei gospodării sunt reprezentate ca fluxuri de

date care conțin toate observațiile corespunzătoare (valorile de consum colectate). Este responsabilitatea dispozitivului Edge să organizeze datele primite de la contoarele inteligente în formatul specificat de modelul de date SensorThings și să le transmită către API-ul de colectare a datelor pentru a fi stocate.

Acces la date externe

Pentru a putea analiza datele colectate de terți în cadrul sistemului nostru, am proiectat o componentă de acces la date care este capabilă să citească date din servicii de date externe sau din baze de date. Datele pot fi trimise către sistemul nostru în mai multe moduri:

1. Încărcarea fișierelor de date - acceptăm mai multe extensii de fișiere: .csv, .xlsx, .json, .xml, .npy, .h5, .mat, .arff;
2. Furnizarea unui șir de conexiune (connection string) - utilizatorul poate furniza un șir de conexiune și numele tabelului care conține datele, astfel sistemul nostru poate fi conectat direct la un sistem extern;
3. Prin solicitări HTTP - este disponibil un API care poate fi folosit pentru a încărca date în sistemul nostru;
4. Comunicare MQTT pentru date în timp real.

Această componentă comunică și cu serviciul nostru de colectare a datelor, care este responsabil de colectarea datelor de la diferiți senzori și dispozitive de Edge conectate.

Detecția anomaliilor

Pentru a găsi un algoritm de prognoză a consumului de apă cu performanță bună, am efectuat mai multe experimente pe setul de date numit Helix (Chatzigeorgakidis, 2018). Setul de date conține serii de timp privind consumul de apă pentru o perioadă de doi ani și pentru un total de 822 de gospodării. Fiecare element din setul de date constă dintr-un identificator anonim a unei case, latitudinea și longitudinea geografică a locației sale și o serie de timp medie de $24 \times 7 = 168$ de valori pentru fiecare oră din săptămână. Seria de timp medie este produsă prin medierea datelor orare de-a lungul întregii perioade.

Pentru fiecare serie de timp din setul de date am antrenat și testat mai mulți algoritmi:

1. Moving Average (Nakano, Takahashi, & Takahashi, 2021)
2. Auto-regression (Zhou & Li, 2019)
3. Exponential Smoothing (Raiyn & Toledo, 2014)
4. Auto-ARIMA (Tim, 2015)
5. XGBRegressor (Zhagparov et al., 2021)
6. LSTM (Yu et al., 2019)
7. Facebook's Prophet (Toni et al., 2021)

După rularea acestor algoritmi pe datele fiecărei gospodării, am calculat scorurile medii de evaluare utilizând RMSE (Eroare pătratică medie) și MSE (eroare pătrată medie). Valorile obținute pot fi văzute în Tabel 1.

Tabel 1 Metrici de evaluare

Algoritm	MSE	RMSE
Moving Average	11.349	2.9
Auto-regression	13.917	3.371
Exponential Smoothing	81.805	7.792
Auto-ARIMA	10.493	2.539
XGBRegressor	29.67	5.136
LSTM	38.538	5.686
Facebook's Prophet	84.964	8.872

Aceste rezultate pot fi, de asemenea, vizualizate în Figura 8. pentru MSE și Figura 9. pentru RMSE.

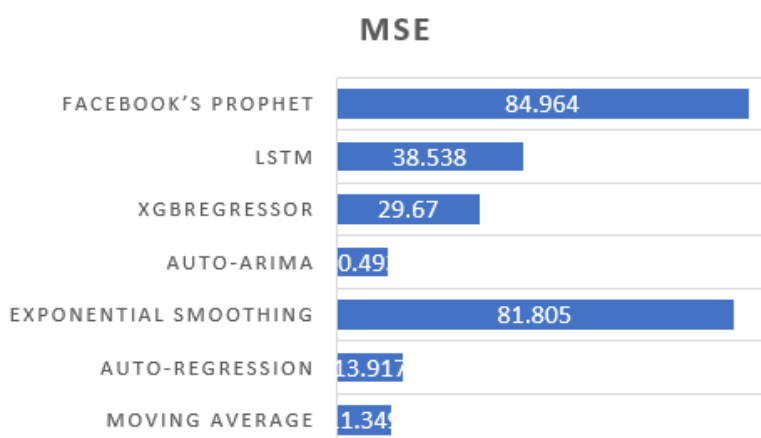


Figura 8 Valorile MSE

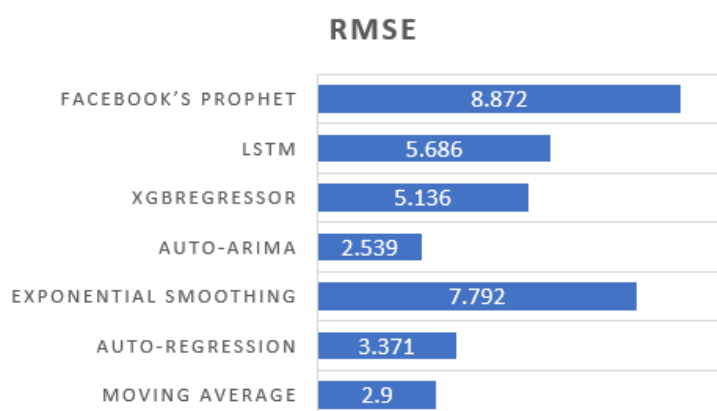


Figura 9 Valorile RMSE

După cum putem vedea în Tabel 1 și în Figura 8 și Figura 9, cel mai bun rezultat a fost obținut utilizând Auto-ARIMA cu un scor RMSE sub 3, ceea ce este un rezultat impresionant, predicția fiind aproape identică cu datele de test. Un exemplu de prognoză Auto-ARIMA poate fi văzut în Figura 10.

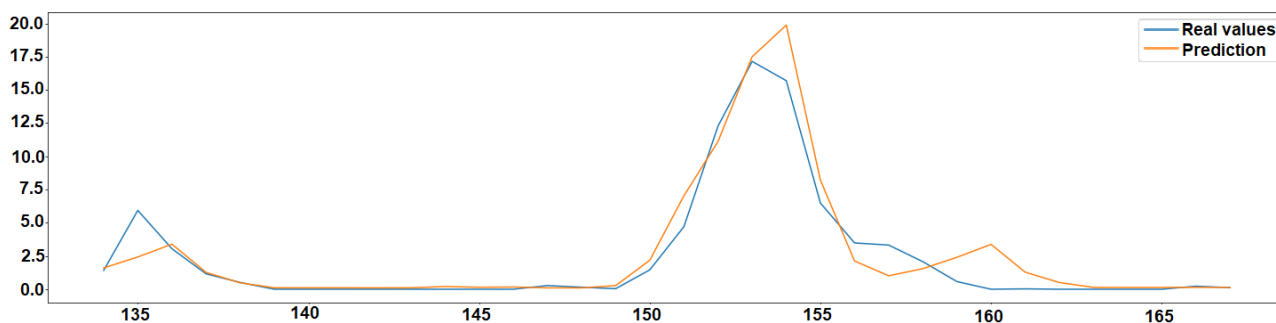


Figura 10 Exemplu de prognoză folosind Auto-ARIMA

Spre surprinderea noastră, modelul LSTM ne-a oferit un scor de eroare mult mai mare decât ceilalți algoritmi, ceea ce se poate explica prin faptul că, în cazul LSTM, am folosit un singur model pentru a prognoza valorile pentru fiecare gospodărie din setul de date, în timp ce în cazul celorlalți algoritmi am antrenat modele separate pentru fiecare serie de timp. Fiecare gospodărie poate avea un comportament de consum foarte diferit și de aceea modelul LSTM s-a încurcat. O posibilă soluție pentru a depăși această problemă ar fi gruparea în primul rând a datelor de consum în mai multe clustere și pentru fiecare cluster a antrena un model LSTM diferit. Acest lucru ar crește precizia prognozei, dar ar crește complexitatea soluției.

Pentru a vizualiza comportamentul tuturor algoritmilor antrenați, am ales aleatoriu câteva date de uz casnic și pentru fiecare algoritm am generat o diagramă care conține valorile reale cu albastru și valorile prezise cu portocaliu (Figura 11).

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

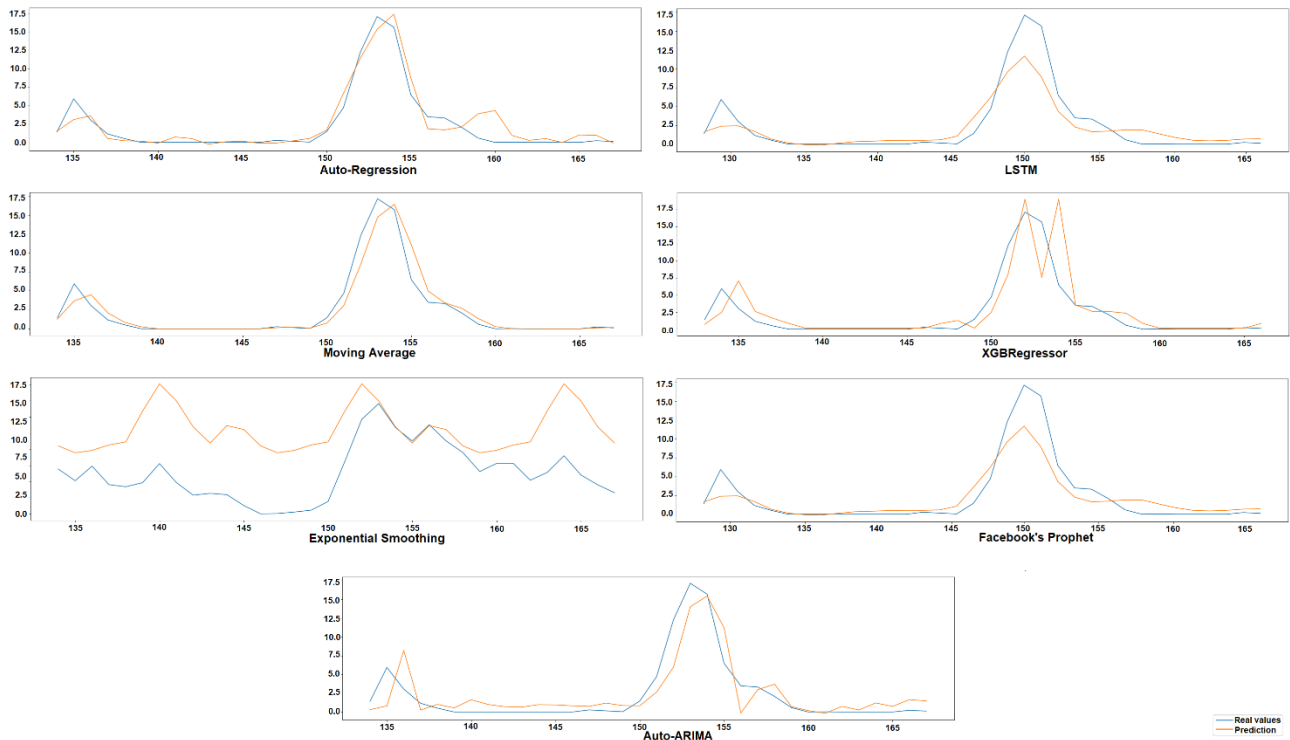


Figura 11 Exemple de prognoză pentru fiecare algoritm

Am folosit setul de date (Chatzigeorgakidis, 2018) pentru a antrena și selecta cel mai bun algoritm, dar scopul final este de a folosi datele primite din surse externe (senzori, dispozitive Edge, API, fișiere încărcate etc.) prin stratul de acces la date.

Pentru detectarea anomaliilor am reutilizat acest model antrenat și am detectat anomaliile pe baza diferenței dintre valorile prezise de model și valorile reale din setul de date. Primul pas a fost calcularea nivelului de incertitudine, care este diferența dintre intervalele de încredere superior și inferior. Un exemplu al acestui nivel de incertitudine poate fi văzut în Figura 12.

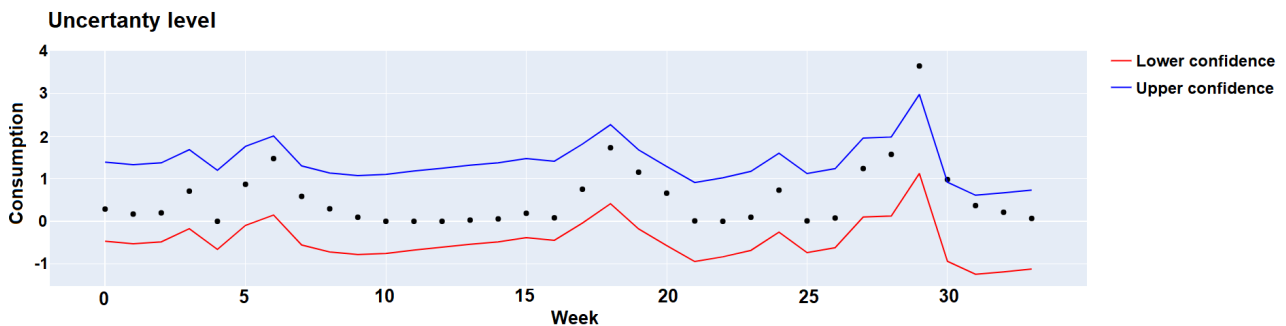


Figura 12 Exemplu de bandă de incertitudine

Eroarea poate fi negativă sau pozitivă, prin urmare, luăm valoarea ei absolută și verificăm că, dacă este mai mare decât pragul de incertitudine, cel mai probabil este o valoare anormală. Aceste puncte vor fi clasificate drept anomalii. Un exemplu de detecție a anomaliilor pe baza valorilor prezise și a nivelului de incertitudine poate fi văzut în Figura 13. În această figură punctele negre reprezintă valorile normale, în timp ce punctele roșii reprezintă anomalii.

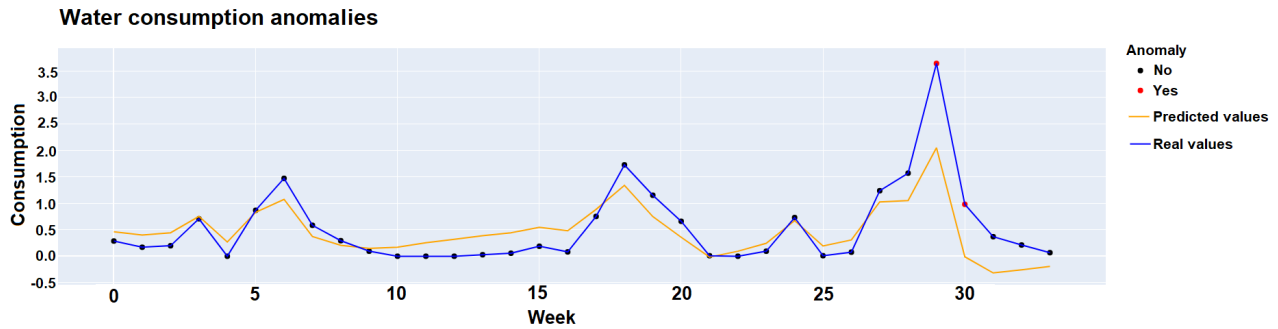


Figura 13 Exemplu de detectare a anomaliilor în consumul de apă

Ca punct de plecare am folosit setul de date (Chatzigeorgakidis, 2018) pentru a crea modelul inițial de detectare a anomaliilor în consumul de apă, dar pe viitor sistemul va prelua datele din surse externe prin componenta Acces la date și va reantrena periodic algoritmul Auto-ARIMA pentru a avea actualizate atât prognoza cât și detectarea anomaliilor. Pentru fiecare utilizator vom avea un model separat, deoarece fiecare gospodărie poate avea un stil de consum foarte diferit. Pe baza anomaliilor detectate, sistemul va trimite notificări utilizatorului. O anomalie poate avea semnificații diferite în funcție de context. De exemplu, dacă utilizatorul este în vacanță (deci nu este acasă) este normal ca consumul să scadă la 0 sau aproape 0, caz în care totuși vom detecta o anomalie și vom trimite o notificare, dar asta nu înseamnă că există o problemă în sistemul de apă. În cazul invers, dacă utilizatorul este acasă și folosește apa în mod regulat, un avertisment generat de sistemul nostru poate sugera o posibilă problemă în sistemul de apă, cum ar fi scurgeri, țevi sparte sau modificări în comportamentul de consum al utilizatorului. Pe baza acestor notificări, utilizatorul poate urmări consumul și poate lua măsuri pentru a maximiza eficiența consumului de apă.

Sistemul de notificări

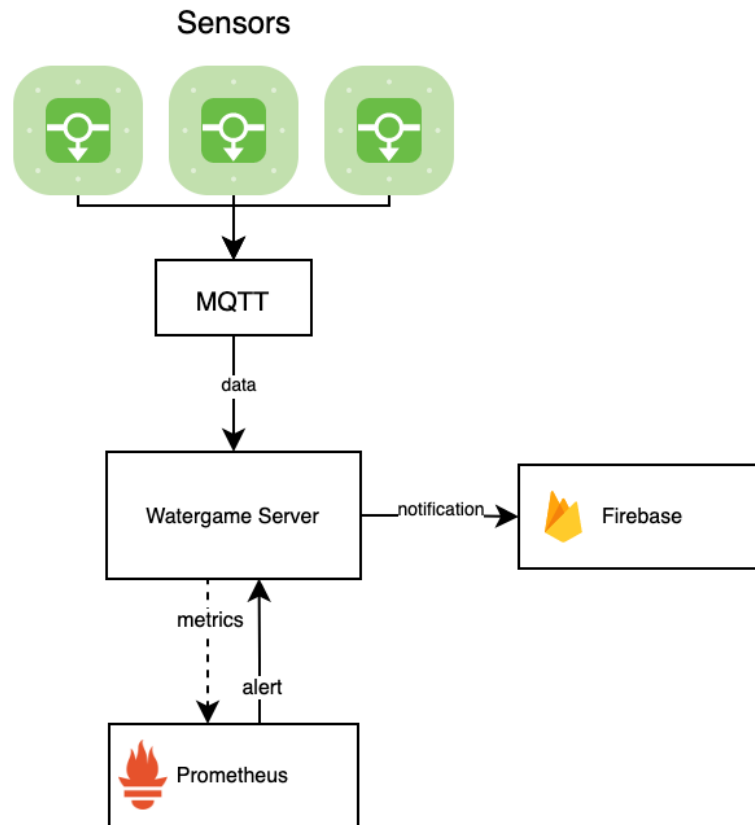


Figura 14 Sistemul de notificări

Sistemul de notificări, reprezentat în este integrat în server-ul aplicației de colectare a datelor de la senzori și permite notificarea utilizatorilor aplicației WaterMonitoringSystem cu privire la starea de funcționare a senzorilor.

Prometheus îndeplinește funcția de monitorizare și alertare, folosindu-se de metricile puse la dispoziție de server. Alertele sunt tratate de server salvând notificări corespunzătoare în baza de date Firebase.

2.1.2 Metoda de grupare a datelor

În continuare vom analiza diferite tipuri de informații care pot fi extrase pe baza datelor colectate de la diverse gospodării în ceea ce privește consumul zilnic de apă pe o perioadă de două săptămâni. Astfel, se poate oferi o imagine de ansamblu asupra consumului de apă în contextul realizării un DSS (Decision Support System) pentru un management durabil al resurselor de apă în viitor.

Un prim tip de informație ce poate fi generat este modul în care se grupează consumatorii de apă în funcție de consumul lor zilnic. Astfel, pentru a împărți setul de date în grupuri cu consum similar de apă, am folosit metoda de grupare K-Means (Syakur M. et al., 2018). Mai mult, pentru a determina numărul optim de clustere, au fost folosite două metode:

- Prima metodă utilizată este metoda Elbow deoarece rezultatele sunt promițătoare pentru multiple aplicații în diverse domenii (Poyrazoglu G., 2021). Această analiză se bazează pe WCSS (Within-Cluster Sum of Squares), unde se calculează suma distanței pătrate dintre fiecare membru al clusterului și centroidul clusterului (Nair A., 2022).
- A doua metodă utilizează analiza siluetei clusterelor (Rousseeuw P., 1987), care evaluează separarea dintre clustere și permite atât inspecția vizuală, cât și evaluarea numerică folosind scorul siluetei.

Prin urmare, cele două metode utilizate reprezintă un suport pentru luarea deciziilor privind alegerea numărului optim de clustere. În plus, datele au fost procesate folosind Python, pachetul scikit-learn (Pedregosa F., 2011) și NumPy (Harris C., 2020) pentru a normaliza datele.

2.1.3 Metoda de clasificare a datelor

Această subsecțiune prezintă principalele metode experimentale concepute pentru determinarea sursei unui eveniment de consum de apă, care se caracterizează prin durată (măsurată în minute) și volumul de apă consumat (măsurat în litri).

Au fost folosiți patru algoritmi de clasificare, doi cu o abordare de învățare automată: Decision Tree și Random Forest, și doi cu deep learning: Dense Neural Network și Recurrent Neural Network.

Arborele decizional

Învățarea automată este o ramură importantă a inteligenței artificiale. Este împărțită în diferite clase de algoritmi, inclusiv clasificarea/regresia pentru învățarea supravegheată și gruparea (clusteringul) pentru învățarea nesupravegheată. În ceea ce privește modelele de clasificare, acestea pot lua forma unei structuri de decizie printr-un clasificator de tip arbore de decizie.

Pașii pentru construirea unui arbore de decizie sunt următorii:

- Setul de date este împărțit în subseturi mai mici;
- Se asociază un nod atributului care are cea mai bună diviziune;
- Pentru fiecare nod, se adaugă atâtea ramuri câte valori distincte are atributul.

Pentru ca nodul să devină o frunză, majoritatea observațiilor dintr-un subset trebuie să aibă aceeași clasă. Seturile de date liniare folosesc adesea arbori de decizie deoarece sunt eficienți și rapizi. Prin urmare, am ales să folosim arbori de decizie și în studiul nostru.

Modelul Random Forest

Pentru a obține o mai bună acuratețe a rezultatelor, s-a ales și testarea algoritmul Random Forest pe setul de date. Acest algoritm își propune să combine mai mulți arbori de decizie pentru a obține un rezultat mai precis. Având în vedere că cei doi algoritmi de învățare automată prezentați (Decision Tree și Random Forest) folosesc date categorice, s-a folosit o reprezentare întreagă pentru maparea valorilor țintă. Modelele de învățare profundă, pe de altă parte, necesită codificare one-hot, folosind o funcție de pierdere adecvată pentru clasificarea multi-clasă.

Rețeaua neuronală densă

Un eșantion de intrare pentru analiza propusă este reprezentat de un vector unidimensional având două caracteristici (volum și durată). Având în vedere această proprietate, cea mai comună soluție de deep learning pentru problema actuală de clasificare este o arhitectură Dense Neural Network. Blocul de bază al unui astfel de model constă dintr-un strat liniar, urmat de o funcție de activare. Termenul „dens” folosit în numele modelului provine din proprietatea matematică a unui strat liniar de mapare a fiecărui element de intrare la toate elementele de ieșire printr-o combinație liniară. Această proprietate este în opoziție cu alte funcții parametrice aplicate în mod obișnuit în învățarea profundă care funcționează pe porțiuni ale intrării (de exemplu, convoluția). Pentru soluția de față, modelul dens este descris de trei blocuri intermediare de acest fel (cu ReLU ca funcție de activare) și un strat liniar final asociat cu o funcție softmax pentru îndeplinirea obiectivului de clasificare. Deoarece datele sunt extrase din relativ puține surse, după fiecare bloc dens intermediar a fost inserat și un modul de regularizare a abandonului pentru a preveni supra-adaptarea (overfitting).

Rețeaua neuronală recurentă (RNN)

Al doilea model neuronal care a fost utilizat se bazează pe un bloc LSTM (Long Short-Term Memory) (Hochreiter S. et al., 1997) ca strat de bază. În această configurație, există trei porți: intrare, ieșire și uitare. Modelul primește intrarea curentă (x_t), starea ascunsă anterioară (h_{t-1}) și starea anterioară a memoriei (c_{t-1}) la fiecare pas de timp. Ieșirile constau din starea ascunsă curentă (h_t) și starea curentă a memoriei (c_t).

Expresiile pentru fiecare poartă (adică intrarea i_t , ieșirea o_t și uitarea f_t) și, precum și cele pentru starea ascunsă curentă (h_t) și starea de memorie curentă (c_t) sunt următoarele:

$$\begin{aligned} i_t &= \sigma_s(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_s(W_o x_t + U_o h_{t-1} + b_o) \\ f_t &= \sigma_s(W_f x_t + U_f h_{t-1} + b_f) \\ \tilde{c}_t &= \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t * \tilde{c}_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned}$$

unde:

- x_t este termenul de intrare la pasul de timp t ;
- h_t și h_{t-1} sunt următoarea stare ascunsă, respectiv starea anterioară ascunsă;
- $\tilde{c}_t$ este vectorul de activare a intrării celulei;
- c_t este starea curentă a memoriei, iar c_{t-1} este starea anterioară a memoriei;
- W_i , W_o și W_f sunt ponderile pentru starea curentă de intrare a fiecărei porți, în timp ce U_i , U_o și U_f sunt ponderile pentru starea anterioară ascunsă;
- b_i , b_o , b_f și b_c sunt vectorii de polarizare;
- σ_s este funcția de activare a sigmoidului;
- σ_h este funcția de activare a tangentei hiperbolice;
- $\odot$ operator este produsul Hadamard.

2.2 Integrarea componentelor de raportare urbană și Blockchain

În contextul aplicației Watergame, factorul uman este principalul motor pentru raportarea incidentelor în infrastructura de apă. Aplicația permite plasarea de indicatori pe hartă în mod interactiv, pe bază de geo-localizare și elemente de joc pentru stimularea participării.

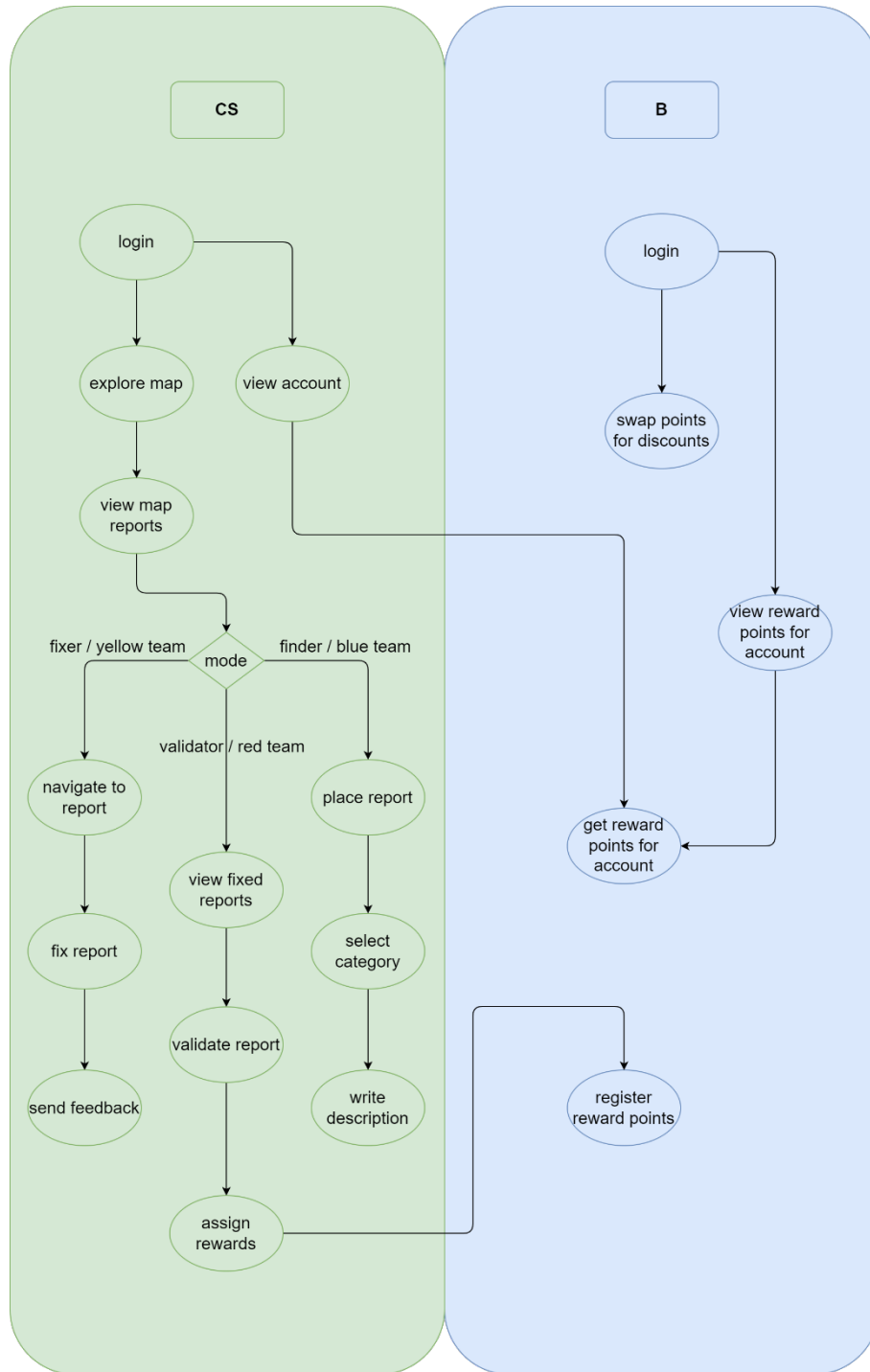


Figura 15 Diagrama de activități a sistemului integrat pentru raportare și recompensare

Integrarea componentelor CS și B este ilustrată în diagrama de activități din Figura 15. În contextul sistemului de raportare a incidentelor în rețeaua de distribuție a apei la nivel urban (CS), sunt disponibile metode interactive cu elemente de gamificare: explorare și vizualizare raportări pe hartă, plasare și editare a raportărilor pe hartă, validare raportări în mod interactiv, vizualizare nivel de reputație, magazin virtual cu recompense. Sistemul de suport Blockchain (B) constituie mecanismul de validare a recompenselor, prin intermediul monedei virtuale WGT (Watergame Token), urmărind scenariile de valorificare a punctelor: vizualizarea tranzacțiilor și a balanței contului în rețeaua blockchain, preschimbarea WGT în oferte avantajoase sau discount-uri în furnizarea serviciilor în rețeaua de distribuție a apei, înregistrarea punctelor asociate WGT în contextul recompensării utilizatorilor.

2.2.1 Integrarea aplicației de raportare urbană

Arhitectura componentei de raportare urbană (Figura 16) are la bază o infrastructură de tip Cloud pentru suportul serviciilor de date. Infrastructura Cloud este găzduită pe AWS (Amazon Web Services) reprezentată printr-un server de aplicație care gestionează cererile HTTP precum și logica de acces la baza de date și la sistemul de stocare a datelor multimedia. Sistemul operează cu o bază de date MySQL ce asigură persistența datelor și o structură bine definită pentru operabilitate. Pentru scalabilitate, sistemul încorporează și o memorie temporară de tip REDIS. Datele multimedia sunt stocate într-un serviciu de stocare cloud (Amazon S3).

Interfața este realizată sub formă de aplicație web interactivă, cu extensie pentru dispozitive mobile. Aplicația web hibridă se ocupă de motorul de joc, interfața cu utilizatorul și interacțiunile cu serviciile externe. Principalele caracteristici gravitează în jurul componentei Map, care se bazează pe biblioteca Google Maps. Aplicația a fost proiectată folosind un framework hibrid care permite o flexibilitate sporită și suport cross-platform. Mai multe componente native sunt utilizate pentru a valorifica capabilitățile dispozitivelor mobile, de exemplu, geo-localizarea, notificările push, vibrațiile și camera. Implementarea funcționalităților presupune integrări cu servicii externe precum: Firebase și Google Maps. Arhitectura jocului este reprezentată de cele 3 scenarii specifice fiecărui tip de utilizator (Finder, Fixer, Validator) și are la bază harta interactivă configurată pentru a integra elementele de joc propuse.

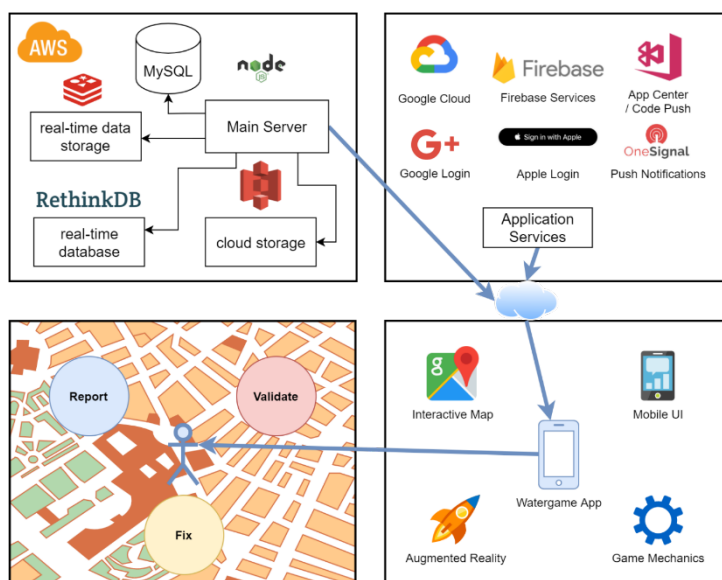


Figura 16 Arhitectura sistemului de raportare a incidentelor pe bază de crowdsensing

Back end-ul aplicației este construit pe Node.js, folosind framework-ul Nest.js¹ și conține logica de aplicație pe partea de server. Acesta are la bază o arhitectură pe 3 niveluri ce implică o modularitate nativă și mentenabilitate crescută prin separarea controlerelor (rutelor) de stratul de servicii (Business Logic) și de stratul de acces la date (Object Data Manager) (NestJS, 2022).

Nivelul de stocare a datelor este compus dintr-o bază de date relațională MySQL, și un depozit de date REDIS în memorie. Subsistemul REDIS² a fost adăugat pentru a îmbunătăți performanța și scalabilitatea sistemului atunci când se lucrează cu cantități mari de date și se gestionează mai multe solicitări. Sistemele de stocare în memorie au fost utilizate pe scară largă pentru aplicații cu randament ridicat pentru memorarea în cache a bazelor de date și mesageria în timp real, asigurând o bună scalabilitate și fiabilitate a datelor pentru majoritatea scenariilor de utilizare a sistemelor cloud (Chen, 2016).

2.2.2 Integrarea componentei Blockchain

Pentru realizarea integrării cu componenta Blockchain, acest modul a fost supus unor transformări pentru facilitarea procesului de interconectare cu celelalte sisteme.

Un prim pas a constat în modificarea contractelor inteligente specifice blockchain, acestea fiind împărțite după cum urmează:

- WaterGameToken – contract ce administrează token în sine cu toate funcționalitățile specifice (Figura 17). Pentru implementarea acestei funcționalități, în scopul optimizării și eficientizării codului existent s-a introdus biblioteca openzeppelin³, bibliotecă gratuită

¹ <https://nestjs.com/>

² <https://redis.io/>

³ <https://github.com/OpenZeppelin>

specifică limbajului Solidity, ce conferă o securitate mai ridicată și mentenanță în timp real din partea comunității de contribuitori.

- WaterGameTokenSale – contract ce administrează zona de ICO (oferta inițială de vânzare a token-ului) și de achiziționare token (Figura 18). Această funcționalitate permite alocarea în cadrul contractului a unei cantități de token specificată la momentul deploymentului în rețea. În scopul oferirii de control asupra acestui proces, au fost implementate metode care permit: pornirea campaniei de vânzare, oprirea temporară a campaniei de vânzare și închiderea campaniei, prin returnarea cantității de monedă rămasă nevândută în contul administratorului soluției.
- WaterGameSpender – contract prin intermediul căruia se va permite cumpărarea unui produs prin criptomonedă (Figura 19). Această funcționalitate a fost dezvoltată pentru a putea fi apelată direct în soluția de Crowdsensing. Pentru îmbunătățirea zonei de administrare a fost implementată și funcționalitatea de restricționare a cheltuirii tokenilor, funcționalitate activabilă la cerere.
- WaterGameRewarder – contract prin intermediul căruia se va permite oferirea de recompense către un utilizator în momentul în care acesta efectuează o acțiune în sistemele integrate ca, de exemplu încărcarea unui eveniment în soluția de Crowdsensing (Figura 20). Pentru îmbunătățirea zonei de administrare a fost implementată și funcționalitatea de restricționare a primirii recompenselor, funcționalitate activabilă la cerere.

Următoarea etapă pentru facilitarea integrării a fost reprezentată de generarea fișierelor JSON caracteristice contractelor inteligente și modificarea zonei de front-end a aplicației deja existente astfel încât metodele caracteristice să fie ușor integrabile în orice modul front-end al unuia dintre celelalte module ale sistemului.

În final, a rezultat o aplicație de sine stătătoare ce se conectează la rețeaua de test Ethereum (Goerli) și comunică direct cu infrastructura blockchain prin intermediul web3.js și Metamask, fără necesitatea unei zone de back-end. Datorită modularizării acestei aplicații, componentele acesteia, au fost ușor integrabile în aplicația de Crowdsensing.

În scopul integrării componentei și a utilizării ei la nivel public, contractele inteligente au fost înregistrate în rețeaua publică Goerli. Informațiile despre aceste contracte pot fi vizualizate pe site-ul oficial etherscan⁴. În următoarele figuri este prezentat modul de vizualizare a contractelor din etherscan.

⁴ <https://goerli.etherscan.io/>

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0x657a9a00b35ac95255... at bin 0xc804938a81b539ed0a...

Token Tracker: WaterGameToken (WTO)

Transactions

Latest 2 from a total of 2 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x355b4861e4fae86304...	buy	7745573	2 days 21 hrs ago	0x657a9a00b35ac95255...	0xf3696ccc3cced64c20a...	0 Ether	0.00113604
0xc804938a81b539ed0a...	Contract Creation	7745541	2 days 21 hrs ago	0x657a9a00b35ac95255...	Contract Creation	0 Ether	0.02073707

Figura 17 Contractul „WaterGameToken”
(0xf3696Ccc3cCEd64c20a97b555cd524522C9D0943)

Contract Overview

Balance: 0.0002 Ether

Token: \$0.00

More Info

My Name Tag: Not Available

Contract Creator: 0x657a9a00b35ac95255... at bin 0x29f04e3489611ee0230...

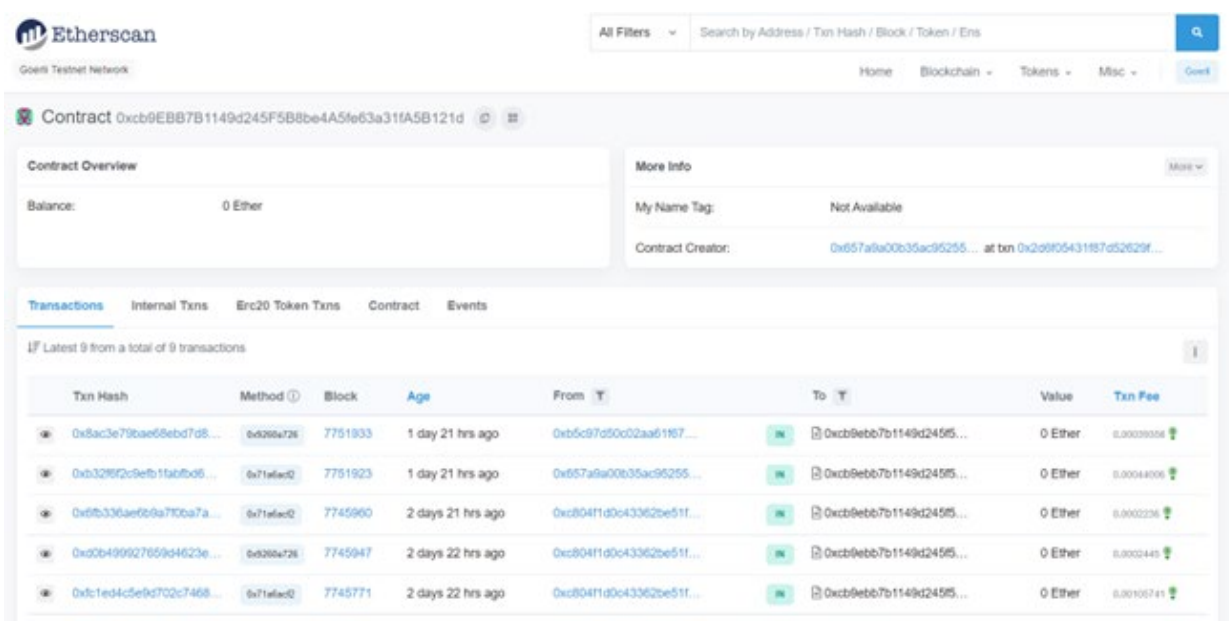
Transactions

Latest 4 from a total of 4 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x50ae892c3140780f80...	Buy Tokens	7745958	2 days 21 hrs ago	0xc804f1d0c43362be51f...	0x5287305c69f20c33a8...	0.0001 Ether	0.0002736
0x155a5026fc57133b72...	Buy Tokens	7745776	2 days 22 hrs ago	0xc804f1d0c43362be51f...	0x5287305c69f20c33a8...	0.0001 Ether	0.00114353
0x11c3b04b20420232bb...	Start ICO	7745575	2 days 23 hrs ago	0x657a9a00b35ac95255...	0x5287305c69f20c33a8...	0 Ether	0.00064335
0x29f04e3489611ee0230...	Contract Creation	7745552	2 days 23 hrs ago	0x657a9a00b35ac95255...	Contract Creation	0 Ether	0.01315085

Figura 18 Contractul „WaterGameTokenSale”
(0x5287305c69f20c33a8Da6912BB757E978647Ff16)

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului



Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

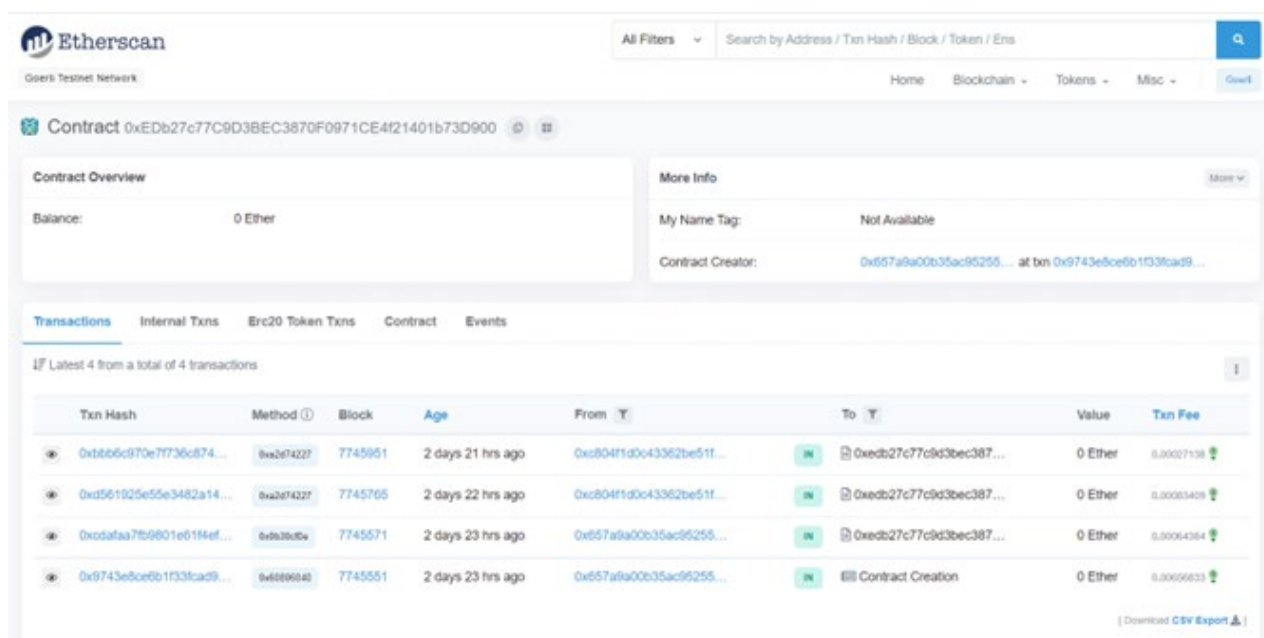
Contract Creator: 0x657a9a00b35ac95255... at tx 0x2a990543187d52629f...

Transactions

17 Latest 9 from a total of 9 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x8ac3e79bae68bd7d8...	0x260e726	7751933	1 day 21 hrs ago	0xb5c97d50c02aa61967...	0xc9b9ebb7b1149d2455...	0 Ether	0.00029006
0xb328f2c9efb1fab5d6...	0x71fae02	7751923	1 day 21 hrs ago	0xb57a9a00b35ac95255...	0xc9b9ebb7b1149d2455...	0 Ether	0.00044006
0xd6b336ae69a70ca7a...	0x71fae02	7745960	2 days 21 hrs ago	0xc804f1d0c43362be51f...	0xc9b9ebb7b1149d2455...	0 Ether	0.0002236
0xd0b499927659d4623e...	0x260e726	7745947	2 days 22 hrs ago	0xc804f1d0c43362be51f...	0xc9b9ebb7b1149d2455...	0 Ether	0.0002445
0xf1ed4c5e9d702c7468...	0x71fae02	7745771	2 days 22 hrs ago	0xc804f1d0c43362be51f...	0xc9b9ebb7b1149d2455...	0 Ether	0.00105741

Figura 19 Contractul „WaterGameTokenRewarder”
(0xc9b9EBB7B1149d245F5B8be4A5fe63a31fA5B121d)



Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0xb57a9a00b35ac95255... at tx 0x9743e6ce6b1f33cad9...

Transactions

17 Latest 4 from a total of 4 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0xb6e6c970e7736c874...	0xa2674227	7745951	2 days 21 hrs ago	0xc804f1d0c43362be51f...	0xedb27c77cd93bec387...	0 Ether	0.00027138
0xd561925e55e3482a14...	0xa2674227	7745765	2 days 22 hrs ago	0xc804f1d0c43362be51f...	0xedb27c77cd93bec387...	0 Ether	0.0003409
0xd0afaa7b0601e0194ef...	0xd93b05e	7745571	2 days 23 hrs ago	0xb57a9a00b35ac95255...	0xedb27c77cd93bec387...	0 Ether	0.00064364
0x9743e6ce6b1f33cad9...	0x68896840	7745551	2 days 23 hrs ago	0xb57a9a00b35ac95255...	Contract Creation	0 Ether	0.00066823

Figura 20 Contractul „WaterGameTokenSpender”
(0xEDb27c77C9D3BEC3870F0971CE4f21401b73D900)

Tranzacțiile prin intermediul cărora au fost activate funcționalitățile descrise mai sus sunt următoarele:

- Activare rewarding: „0xf74ecbfd97245f5cd6b6f4aefc3576435568fca7fc094250c9fbafeb5536914f”
- Activare donații: „0x5fe71e65934528866ed5c672aa807edfaf259dad506c7ae03d6c97439cea1089”

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

- Activare utilizare token: „0xcdafaa7fb9801e61f4efb51097681671071dd9fa1f3ef16620046a6d07e4f200”
- Lichiditate contract sale: „0x355b4861e4fae8630478752e95b6712fb64fc8039d209f422e8eb2027f263f09”
- Cumparare token ICO:
„0x11c3b04b20420232bb4da693049e6cdeeb926598ecee8d70ddc562e338d188d0”

În ceea ce privește zona de front-end a aplicației, aceasta a fost modularizată astfel încât fiecare funcționalitate să aibă o metodă independentă, ușor integrabilă în cadrul oricărei alte soluții front-end.

- *getReward: function()*
- *spendOnItem: function()*
- *donateToken: function()*
- *buyToken: function()*

3 Experimentarea sistemului

În această etapă, a fost definit cadrul pentru dezvoltare și operare în contextul sistemului integrat. Sunt prezentate instrumentele, metodele, și interacțiunile la nivel de sistem, precum și rezultatele experimentale obținute în contextul scenariilor definite în etapa de proiectare.

3.1 Monitorizarea sistemului de colectare a datelor de la senzori

Primul pas în monitorizarea sistemului este citirea și prelucrarea datelor. Platforma de analiză de date se conectează la MQTT pentru a citi datele de la senzori în timp real. Acest proces este prezentat în diagrama de secvență din Figura 21.

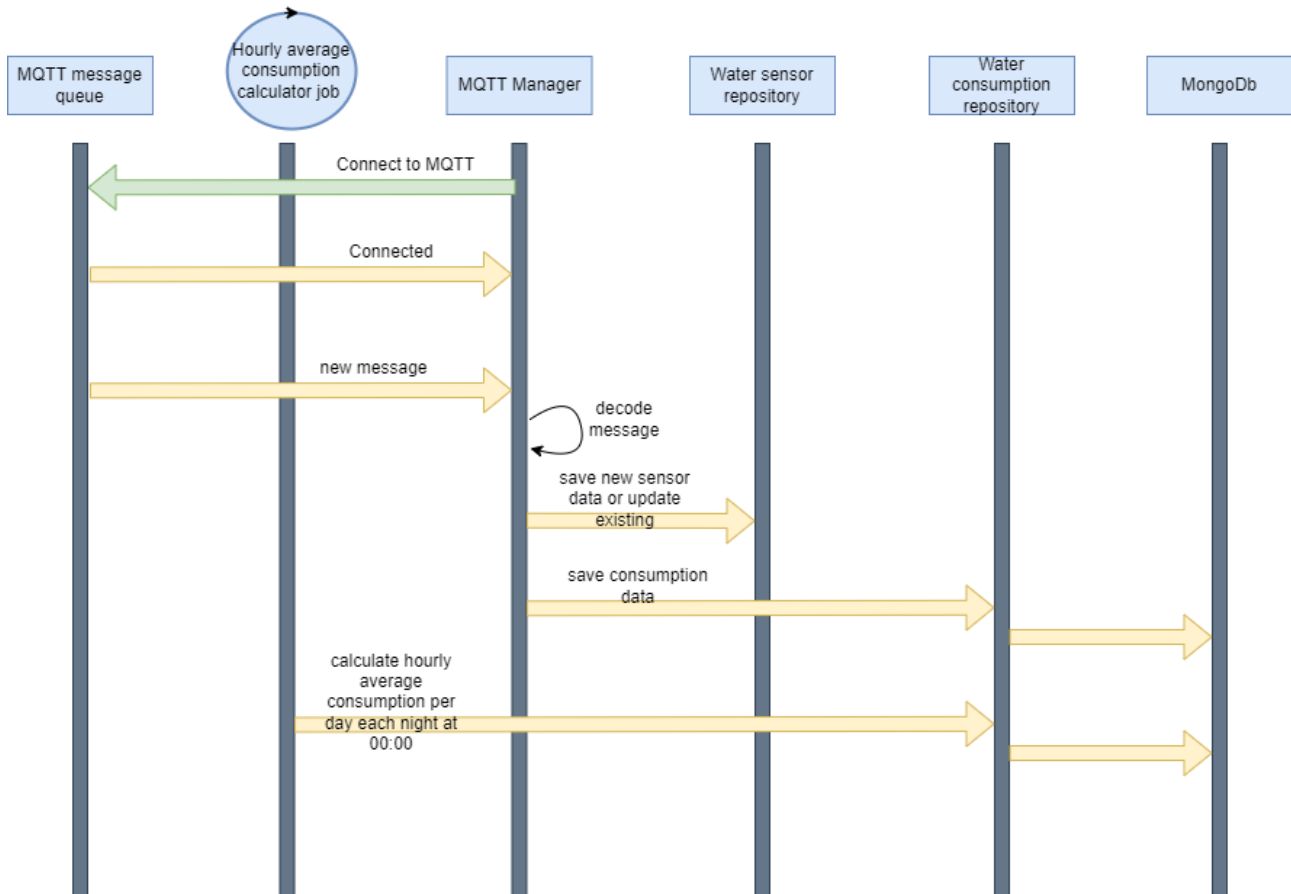


Figura 21 Diagrama de secvență pentru citirea datelor de la senzori prin MQTT

Aceste date vor fi salvate în baza de date a platformei de analiză de date pentru procesare ulterioară, offline. Pe baza datelor salvate, la sfârșitul fiecărei zi se rulează un job pentru a calcula consumul pe oră care va fi folosit de modelele ARIMA pentru predicție.

Pe baza consumului calculat pe oră, în fiecare noapte se antrenează modele noi de ARIMA pentru fiecare senzor în parte. Prin reantrenarea modelului încercăm să avem o învățare continuă; cu trecerea timpului vom putea prezice din ce în ce mai bine comportamentul utilizatorului. Figura 22 prezintă modul în care se analizează comportamentul consumatorului și se generează un mesaj în caz de anomalie. Orice decizie luată de platforma de analiză, de exemplu dacă se detectează un

senzor stricat, o scurgere sau un consum anormal, va fi comunicată către restul sistemului prin mesaje publicate pe o coadă de tip RabbitMQ.

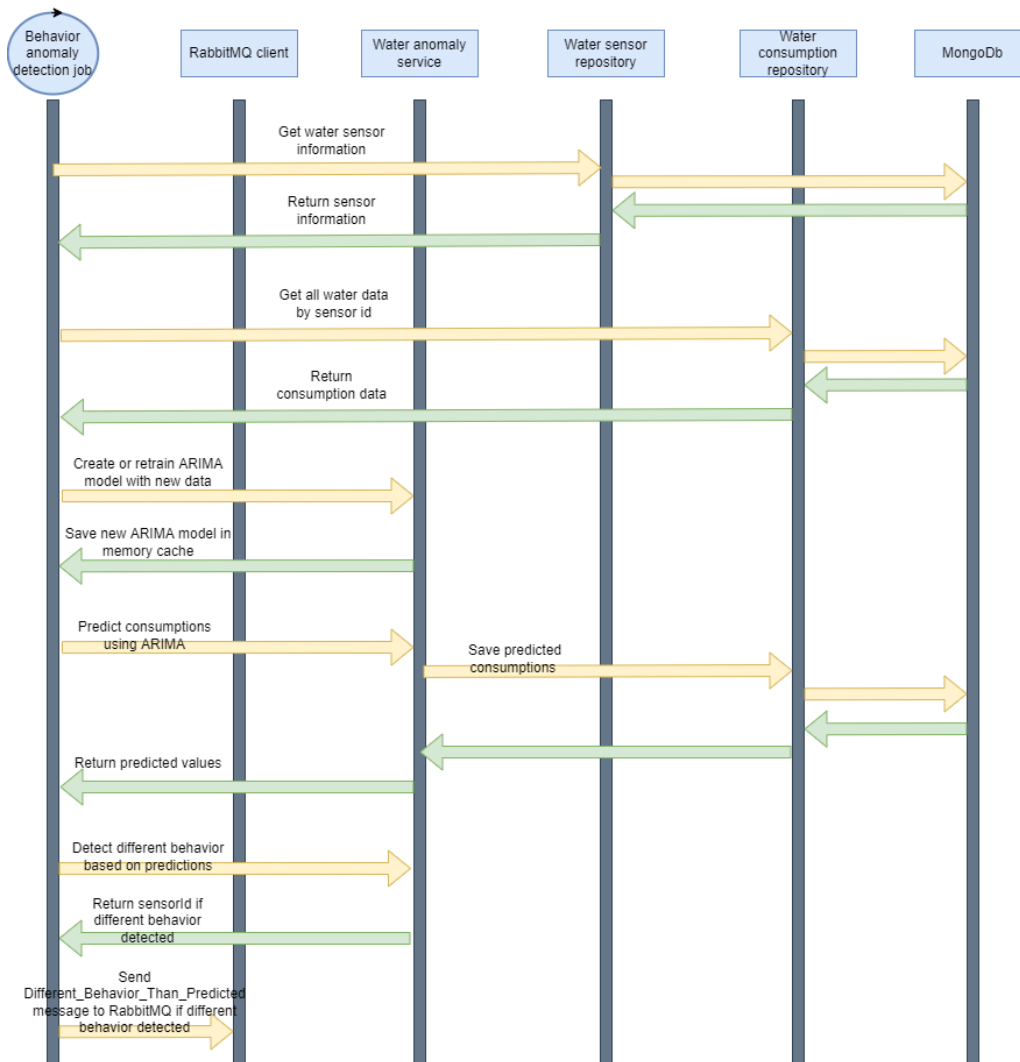


Figura 22 Diagrama de secvență pentru detecția comportamentului anormal

În afară de conectarea pe coada de MQTT avem și un API prin care datele pot fi încărcate în sistem pentru procesare și o interfață grafică de tip web prin care utilizatorul poate să interacționeze cu serviciile noastre de analiză de date. Un exemplu de antrenare al unui model ARIMA se poate vedea în Figura 23. După încărcarea setului de date de antrenare și selecția coloanei având valorile țintă (așa cum am prezentat în capitolul 3.2.5) putem folosi funcționalitatea numită dynamic pipeline pentru a antrena un model ARIMA și pentru a vizualiza rezultatele. Linia albastră reprezintă valorile reale, citite de la senzori, iar culoarea roșie marchează valorile prezise.

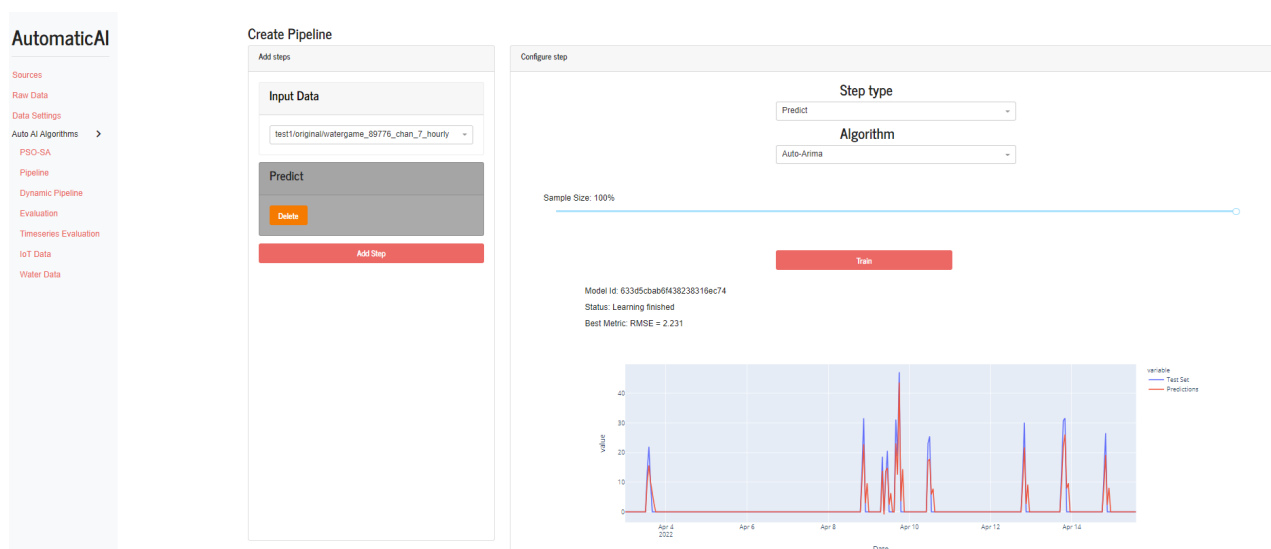


Figura 23 Antrenarea unui model ARIMA pe date reale

3.2 Raportarea anomaliilor în rețeaua de senzori

Anomaliile detectate vor fi publicate pe cozi de mesaje de tip RabbitMQ. Pe lângă anomaliile comportamentale care detectează un consum diferit de consumul obișnuit, vom detecta scurgeri și defecțiuni în sistem și vom avea și o coadă care va conține consumul prezis pentru ziua următoare. Aceste cozi pot fi văzute în Figura 24.

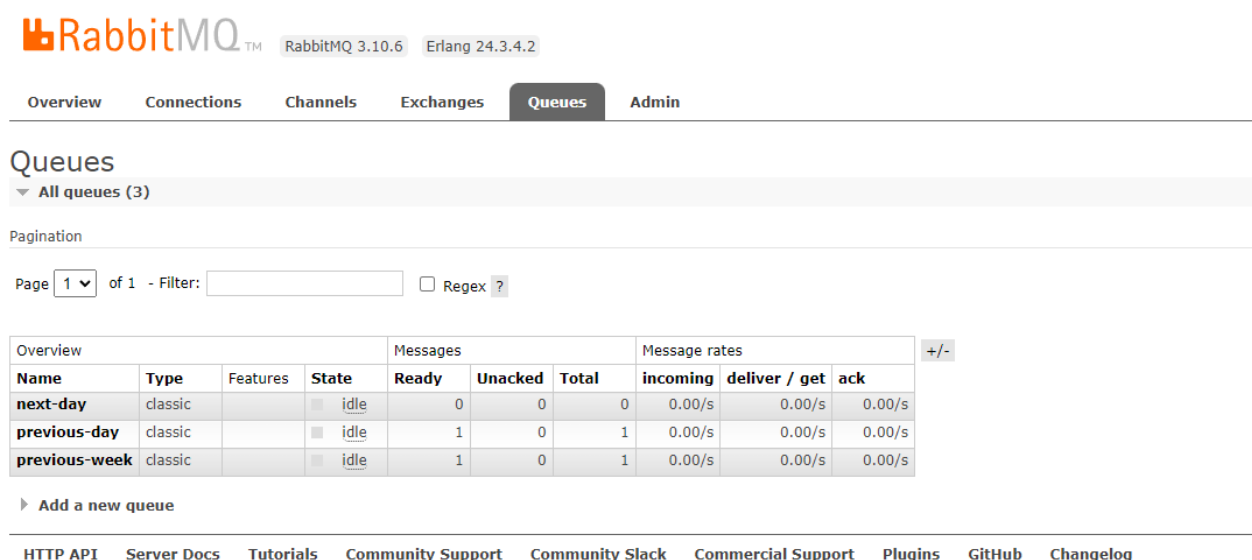


Figura 24 Cozi de tip RabbitMQ

Fiecare coadă va avea un rol diferit:

1. **next-day** - conține predicții de consum orare pentru ziua următoare;
2. **previous-day** - conține anomalii pentru ziua anterioară;
3. **previous-week** - conține anomalii care au fost detectate săptămâna anterioară.

Pentru cele trei tipuri de anomalii avem trei tipuri de mesaje:

1. **Broken_Sensor** - dacă mai mult de 2 zile consecutive nu primim valori mai mari de 0 de la un senzor, atunci adăugăm un mesaj de acest tip, adică presupunem că senzorul respectiv nu funcționează corect;
2. **Leak** - dacă niciodată nu primim 0 sau dacă în timpul nopții avem consum continuu atunci generăm această anomalie considerând că avem undeva scurgere;
3. **Different_Behavior_Than_Predicted** - dacă între consumul prezis și consumul măsurat este o diferență mare, atunci generăm aceasta anomalie, considerând că ceva s-a întâmplat, consumul actual diferă de cel obișnuit.

3.2.1 Detecția scurgerilor

Pentru detecția scurgerilor am implementat un job recurent, care se declanșează în fiecare noapte. În primul pas se citește lista de senzori existenți în sistem, iar apoi pentru fiecare senzor se iau datele de consum pentru ziua curentă și pentru două zile anterioare. Dacă se detectează consum continuu, fără valori de zero sau dacă se detectează consum continuu de mai multe ore în timpul nopții atunci se generează un mesaj de tip Leak, adică scurgere și se pune pe coada RabbitMQ identificatorul senzorului și tipul anomaliei. Procesul acesta poate fi văzut în Figura 25.

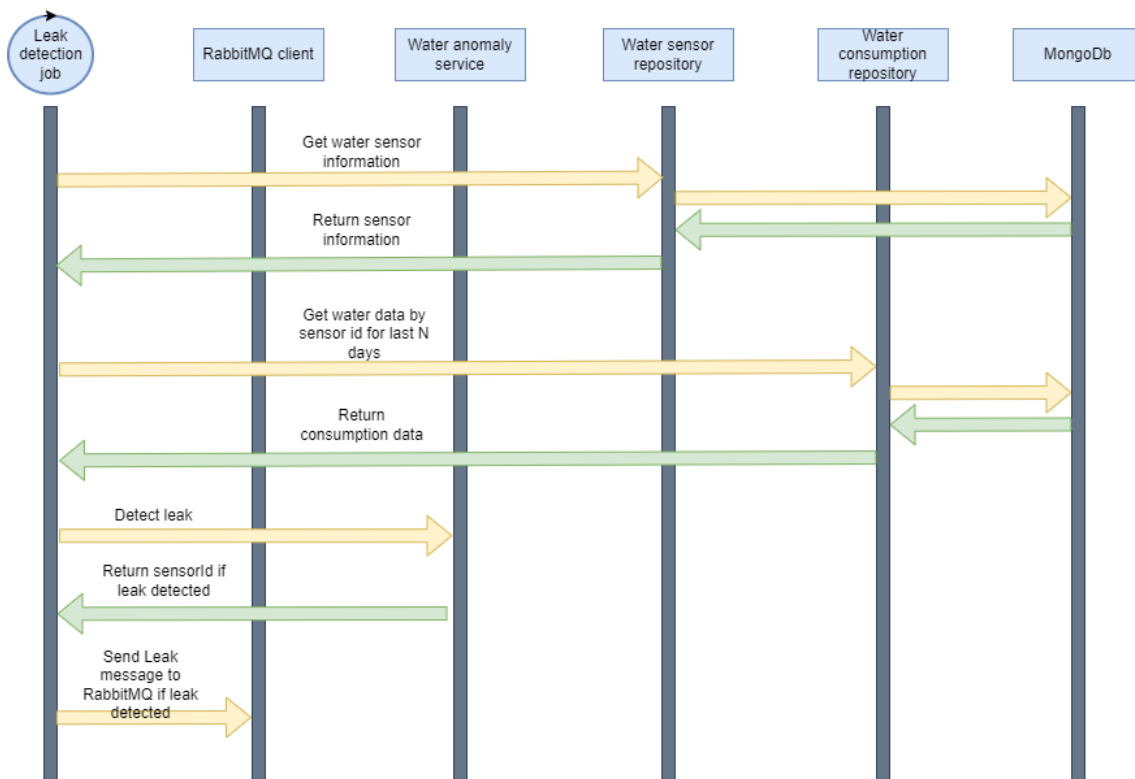


Figura 25 Diagrama de secvență pentru detecția scurgerilor

3.2.2 Detecția senzorilor defecți

Pentru detecția senzorilor defecți am implementat un job recurent, care se declanșează în fiecare noapte. În primul pas se citește lista de senzori existenți în sistem, iar apoi pentru fiecare senzor se

iau datele de consum pentru ziua curentă și pentru două zile anterioare. Dacă pentru două sau mai multe zile consecutive nu avem consum, adică avem doar valori de 0, atunci se generează un mesaj de tip Broken_Sensor, adică senzor defect și se pune pe coada RabbitMQ identificatorul senzorului și tipul anomaliei. Mai multe detalii despre detecția senzorilor defecti pot fi văzute în Figura 26.

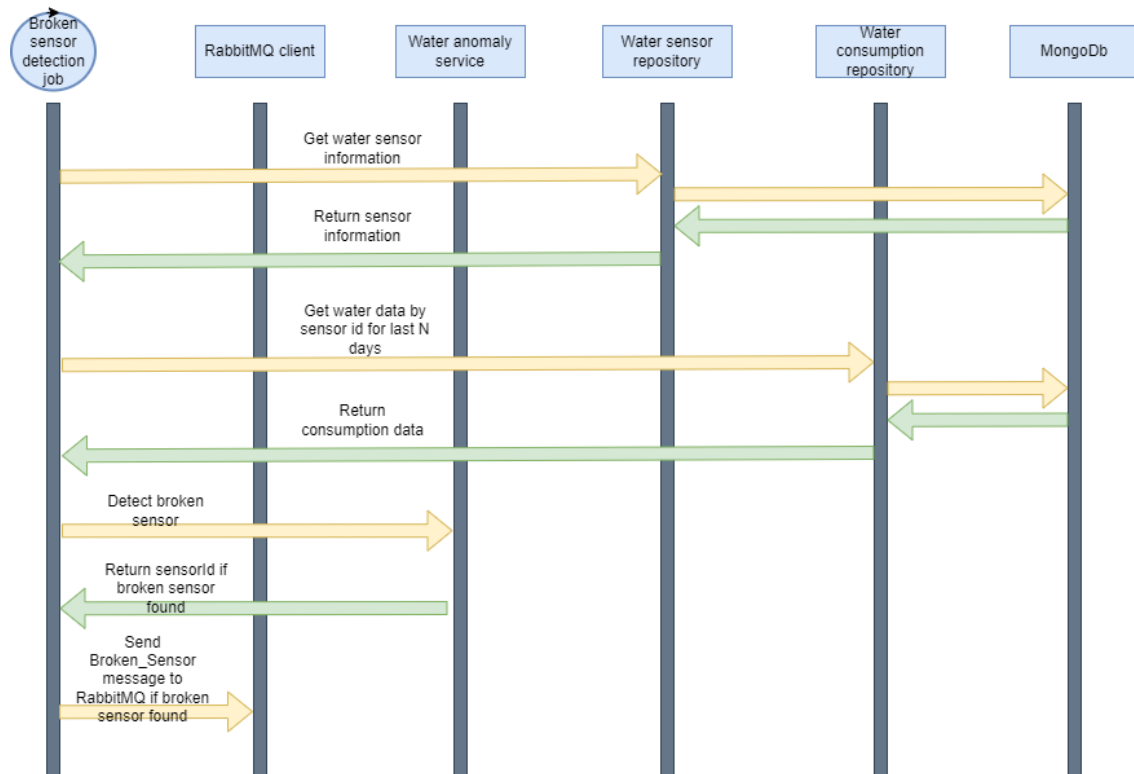


Figura 26 Diagrama de secvență pentru detecția senzorilor defecti

3.2.3 Notificarea defecțiunilor

Fluxul de lucru al sistemului de notificări este evidențiat în Figura 27.

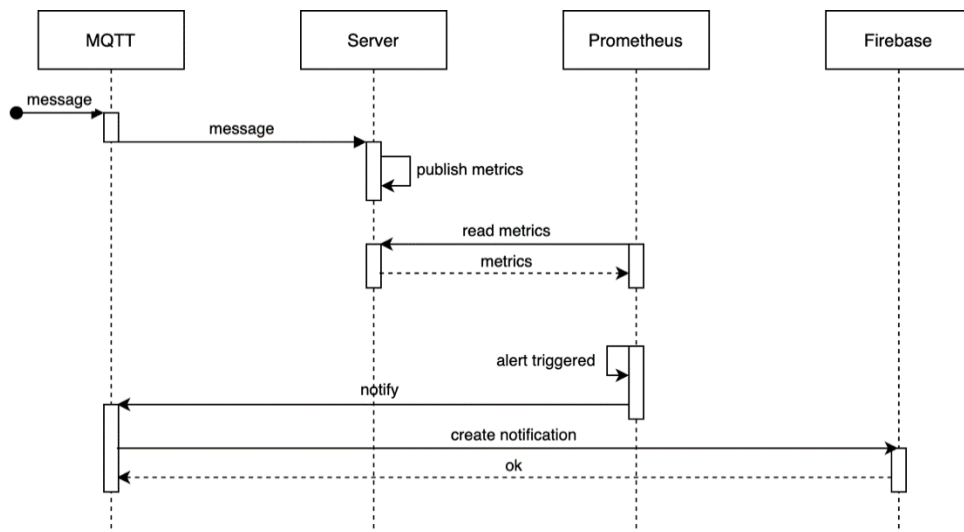


Figura 27 Diagrama de secvență pentru notificarea defecțiunilor

Server-ul primește mesaje de tipul heartbeat de la senzori prin intermediul MQTT. Pentru fiecare mesaj, serverul publica o metrică ce conține id-ul senzorului și are valoarea corespunzătoare timpului de la momentul primirii mesajului în format linux epoch. Toate metricile înregistrate pe server sunt disponibile prin accesarea endpoint-ului /metrics. Prometheus citește periodic metricile publicate de server, comunicarea fiind una de tipul pull-based.

Pentru alertare, în cadrul Prometheus au fost definite reguli pentru agregarea metricilor existente. Cu ajutorul acestora se obține o nouă metrică care capturează timpul scurs de la ultimul heartbeat al unui senzor. O alertă va fi declanșată dacă au trecut mai mult de 5 minute de la ultimul mesaj heartbeat.

La declanșarea unei alerte, se apelează un endpoint specific de pe server care are rolul de a trata alerta și a genera un mesaj de notificare ce este stocat în baza de date Firebase. Din acest moment notificarea poate fi vizualizată de către supplier.

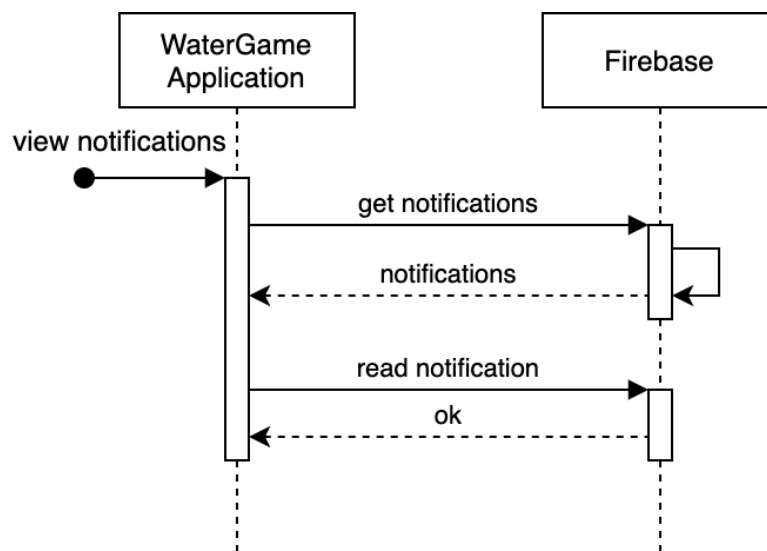


Figura 28 Diagrama de secvență pentru vizualizarea notificărilor

Diagrama de secvență corespunzătoare pentru vizualizarea notificărilor se regăsește în Figura 28. Prin accesarea de către supplier a ecranului pentru notificări, baza de date este interogată pentru listarea tuturor notificărilor. Atunci când notificarea este citită, aceasta este marcată corespunzător.

3.2.4 Citirea datelor de la edge device

Pentru a citi valorile de la dispozitivele Edge, am creat un MQTT manager, care se ocupă de conexiune și de consumarea mesajelor de pe coada de mesaje MQTT. În momentul în care se pune un mesaj pe coadă, managerul MQTT primește mesajul și îl decodifică. Mesajele vin ca un șir de caractere, iar valorile sunt delimitate cu virgulă, astfel: “tip nod, id senzor, tip senzor, debit 1, volum 1, debit 2, volum 2, debit 3, volum 3, debit 4, volum 4”. După decodificare, datele vor fi salvate în baza de date de tip MongoDB prin intermediul “Water consumption repository”. Dacă primim un identificator de senzor care nu apare în baza noastră de date, atunci se adaugă un obiect nou în colecția de senzori din MongoDB prin intermediul “Water sensor repository” (vezi Figura 29).

În afară de a citi valorile de la dispozitive Edge, am creat și un job care rulează în fiecare noapte și calculează valorile de consum mediu pe oră pentru ziua respectivă, astfel pregătind datele pentru modelele de inteligență artificială care vor face predicție de consum pe oră pentru ziua următoare.

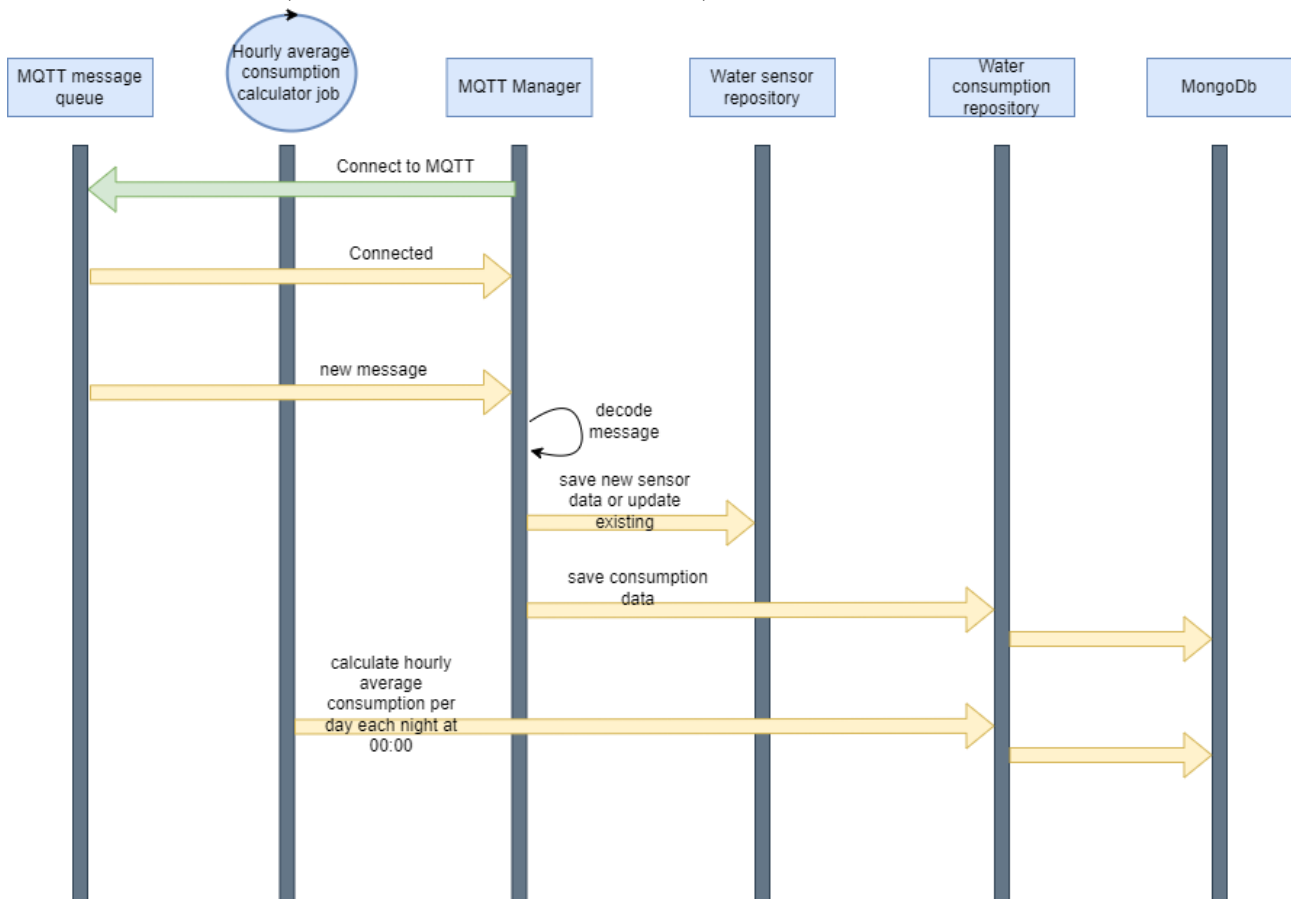


Figura 29 Diagrama de secvență pentru citirea datelor de la Edge device

3.2.5 Detecția comportamentului anormal

Algoritmul de detecție a anomaliilor în comportamentul utilizatorilor este unul mai complex și necesită mai mulți pași. În fiecare noapte se rulează un job care prezice consumul pentru ziua următoare. Pentru a detecta anomalii comportamentale folosim predicțiile făcute de modelul ARIMA. Am ales ARIMA, deoarece, așa cum se vede și în experimente, acest algoritm ne-a dat rezultatele cele mai bune, iar modelul salvat nu ocupă mult spațiu, deci poate fi stocat într-un cache pentru performanță sau poate fi utilizat chiar și pe dispozitivul Edge.

Pentru fiecare utilizator creăm modele ARIMA separate, astfel încercăm să personalizăm interacțiunea utilizatorului cu sistemul nostru. Pentru a detecta anomaliile comportamentale, vom calcula suma erorilor între consumul prezis și consumul real, iar dacă această sumă depășește o limită setată, atunci generăm mesaj de tip `Different_Behavior_Than_Predicted` și îl trimitem pe RabbitMQ, de unde poate fi consumat de terți (de exemplu de aplicații mobile, de alte dispozitive Edge etc.). Figura 30 prezintă diagrama de secvență pentru detecția comportamentului anormal.

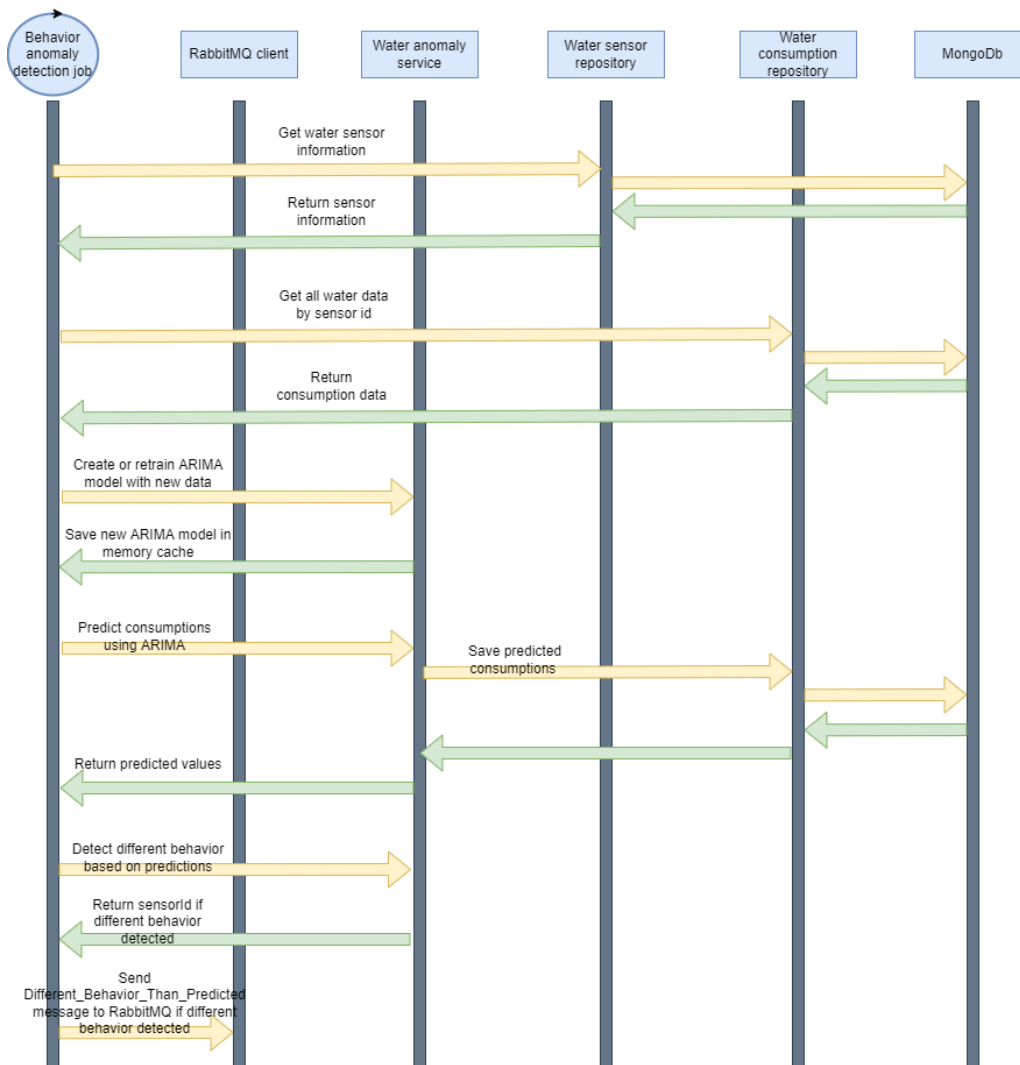


Figura 30 Diagrama de secvență pentru detecția comportamentului anormal

3.2.6 Analiza datelor

Pentru analiza datelor am creat o interfață grafică de tip web, prin care utilizatorii pot să încarce seturi de date în sistemul nostru, să vizualizeze și să configureze datele încărcate, să antreneze diferiți algoritmi de inteligență artificială pentru predicție sau clasificare și să vizualizeze rezultatele. Pe lângă analiza datelor încărcate, platforma oferă posibilitatea de a vizualiza datele obținute de la diferiți senzori.

Așa cum se vede în Figura 3131, utilizatorul poate să încarce fișiere în sistemul nostru sau poate să specifice un connection string și un nume de tabel și astfel sistemul nostru poate fi conectat direct la sisteme externe.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

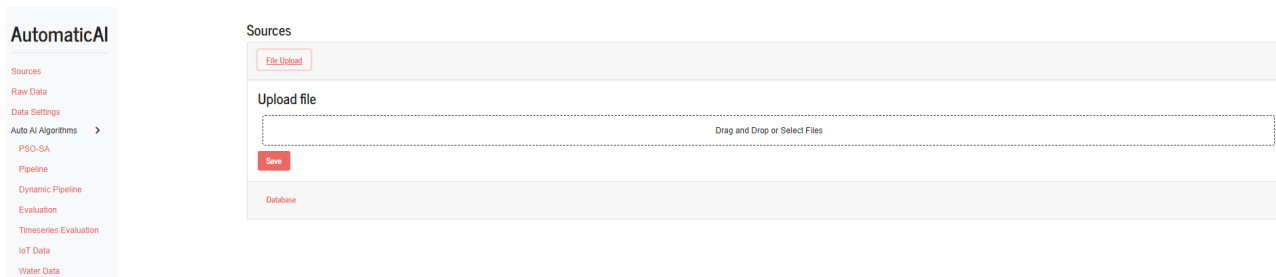


Figura 31 Încărcarea datelor în sistem

Lista de fișiere din sistem poate fi vizualizată accesând meniul Raw data (vezi Figura 32).

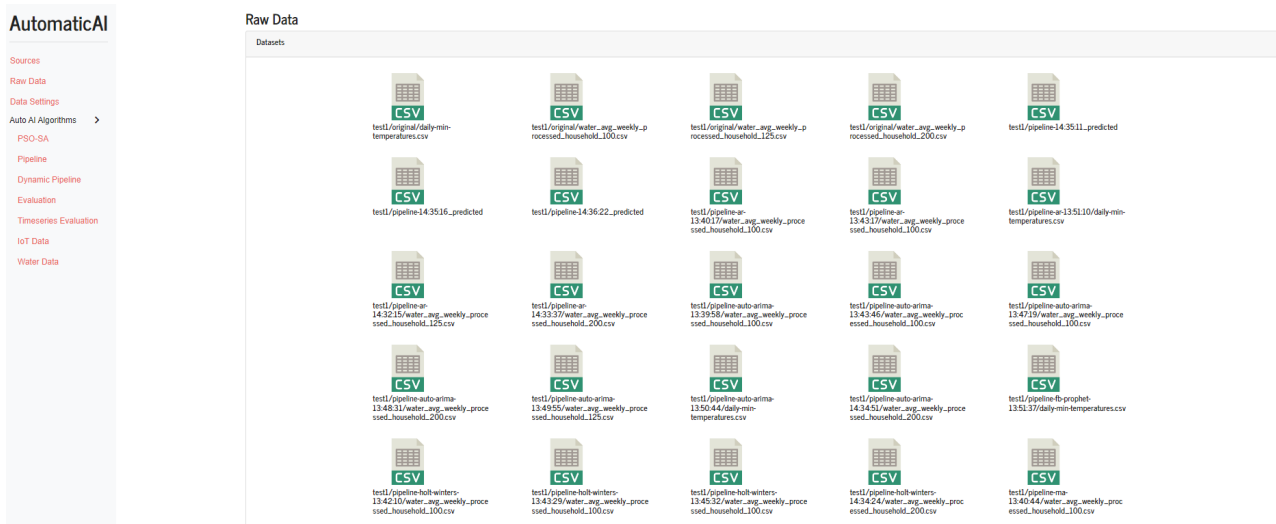


Figura 32 Vizualizarea fișierelor încărcate

Dacă selectăm un fișier, acesta se va deschide și vom putea vizualiza datele salvate în forma tabelară (vezi Figura 33).

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

AutomaticAI		Date	Value
Sources		26/07/2015	0.1
Raw Data		02/08/2015	0.111111111
Data Settings		09/08/2015	0
Auto AI Algorithms		16/08/2015	0
PSO-SA		23/08/2015	0
Pipeline		30/08/2015	0
Dynamic Pipeline		06/09/2015	1.380952381
Evaluation		13/09/2015	4.771428571
Timeseries Evaluation		20/09/2015	12.81904762
IoT Data		27/09/2015	19.44761905
Water Data		04/10/2015	23.43809524
		11/10/2015	17.80952381
		18/10/2015	6.247619048
		25/10/2015	0.754716981
		01/11/2015	2.68952381
		08/11/2015	0.828571429
		15/11/2015	0.561904762
		22/11/2015	0.075471698
		29/11/2015	0.259615385
		06/12/2015	0
		13/12/2015	0
		20/12/2015	0
		27/12/2015	0
		03/01/2016	0
		10/01/2016	0
		17/01/2016	0
		24/01/2016	0
		31/01/2016	0
		07/02/2016	0
		14/02/2016	0.116504854
		21/02/2016	2.349056084
		28/02/2016	5.952830189
		06/03/2016	10.8490566
		13/03/2016	20.1588785
		20/03/2016	17.51428571

Figura 33 Vizualizarea datelor din fișierul selectat

Înainte de a aplica algoritmi de inteligență artificială asupra datelor, utilizatorul trebuie să selecteze coloana care conține clasele sau valorile dorite în caz de serii de timp. Mai mult, pe această pagină utilizatorul poate să elimine coloanele nedorite folosind butonul Delete (vezi Figura 34 și Figura 35). Fiecare pătrat reprezintă o coloană din fișierul încărcat.

AutomaticAI		Data Settings	
Sources		Configure Active Dataset	
Raw Data		test1/original/water_avg_weekly_processed_household_100	
Data Settings			
Auto AI Algorithms			
PSO-SA			
Pipeline			
Dynamic Pipeline			
Evaluation			
Timeseries Evaluation			
IoT Data			
Water Data			

Date	Value
X	Set as target
X	Unset

Figura 34 Configurarea setului de date

AutomaticAI		Data Settings	
Sources		Configure Active Dataset	
Raw Data		test1/original/water_x_train_sample_2	
Data Settings			
Auto AI Algorithms			
PSO-SA			
Pipeline			
Dynamic Pipeline			
Evaluation			
Timeseries Evaluation			
IoT Data			
Water Data			

TP	Cl	pH	Redox	Leit
X	X	X	X	X
Trueb	CL_2	Fm	Fm_2	EVENT
X	X	X	X	X
				Unset

Figura 35 Configurarea setului de date cu mai multe coloane

Dacă utilizatorul alege meniul numit PSO-SA, va avea posibilitatea de a configura un pipeline static în care avem trei pași de preprocesare (pașii care sunt folosiți de obicei la orice problemă de clasificare), iar apoi poate să ruleze algoritmul nostru de selecție și optimizare automată, care alege în mod automat, fără intervenție umană, algoritmul cel mai potrivit pentru datele de intrare (vezi Figura 36).

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

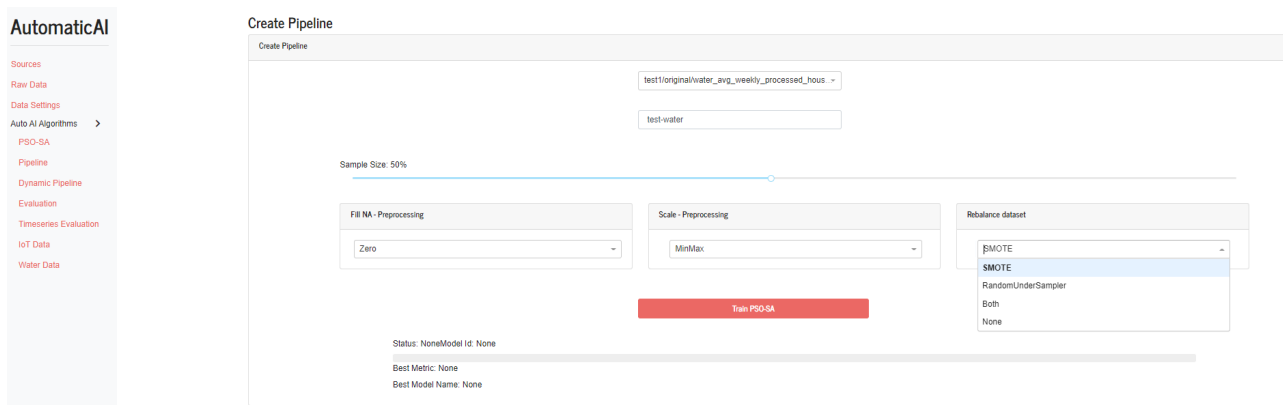


Figura 36 Pipeline static și rularea algoritmului PSO-SA

Dacă utilizatorul dorește un control mai fin asupra datelor, poate să aleagă meniul Dynamic pipeline. Pe această pagină se poate adăuga un număr oarecare de pași, de aceea l-am denumit dynamic pipeline (spre deosebire de pagina PSO-SA, unde am avut un număr fix de pași). După configurarea pipeline-ului și selecția algoritmului final (de exemplu, Moving Average e algoritmul final, așa cum se vede în Figura 37, utilizatorul poate să pornească o sesiune de antrenare, iar la finalul sesiunii vom avea salvat în baza noastră de date modelul antrenat pe datele de intrare. Modelul găsit și antrenat va fi salvat în baza de date, de unde poate fi descărcat pentru utilizare ulterioară. În partea de jos a paginii sunt afișate rezultatele modelului antrenat, cu albastru am marcat datele reale și cu roșu datele generate de model.

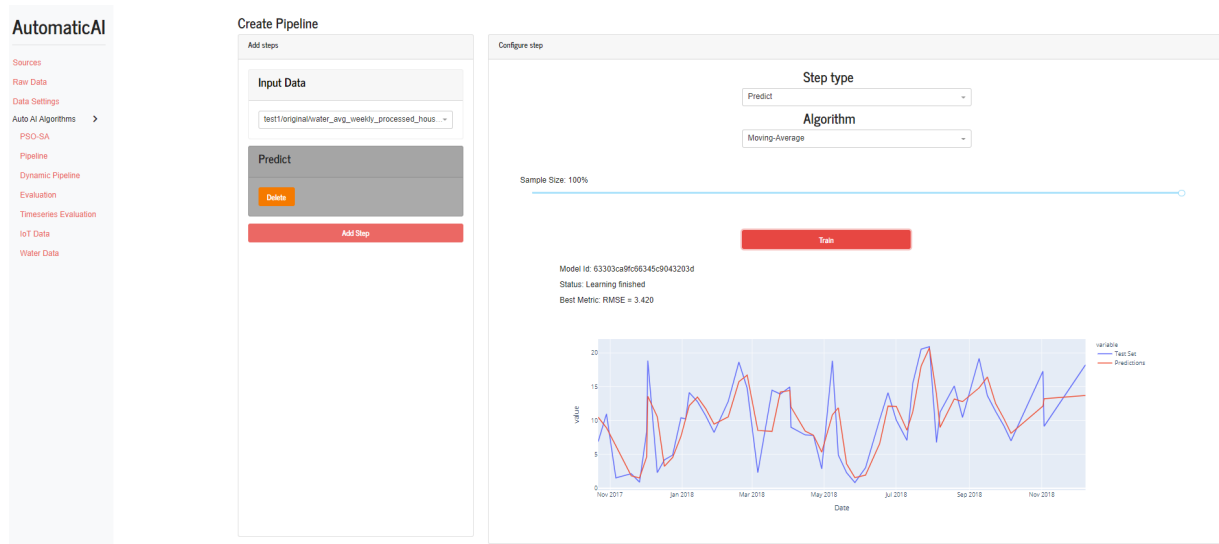


Figura 37 Crearea și rularea pipeline-ului dinamic

Figura 38 ilustrează un caz real, am folosit platforma pentru a antrena un model de inteligență artificială pentru a prezice valori de consum real primite de la senzorul 55891.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

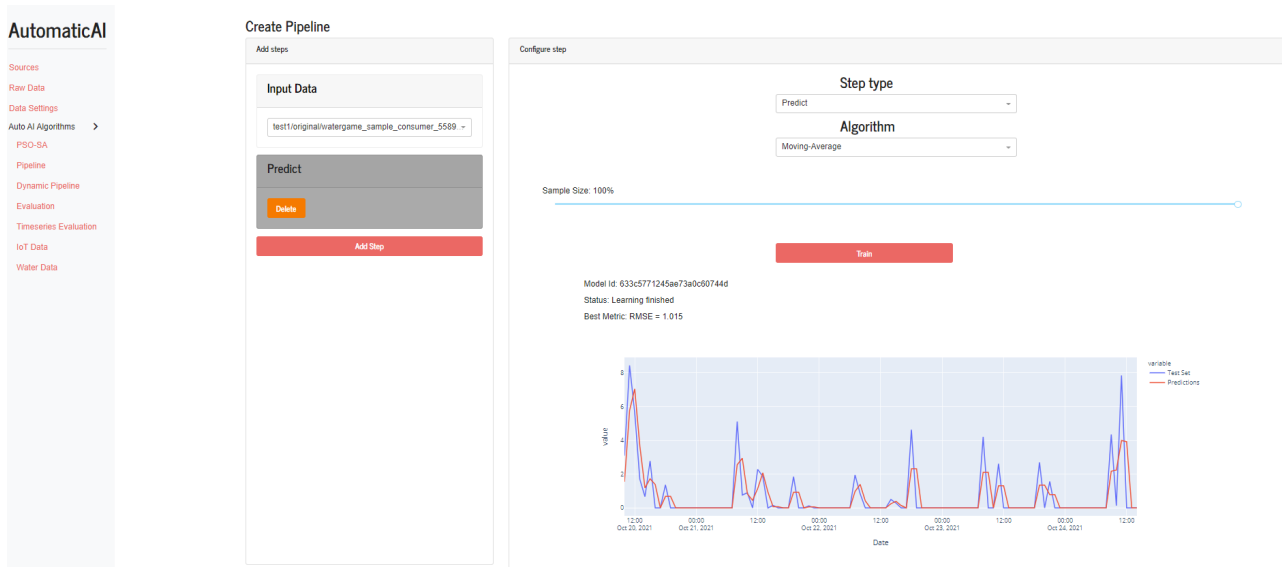


Figura 38 Analiza datelor primite de la senzorul 55891

În cazul algoritmilor de clasificare, putem evalua rezultatele modelului antrenat folosind pagina numită Evaluation (vezi Figura 39), iar în cazul seriilor de timp, rezultatele obținute pot fi validate pe pagina numită Timeseries evaluation.

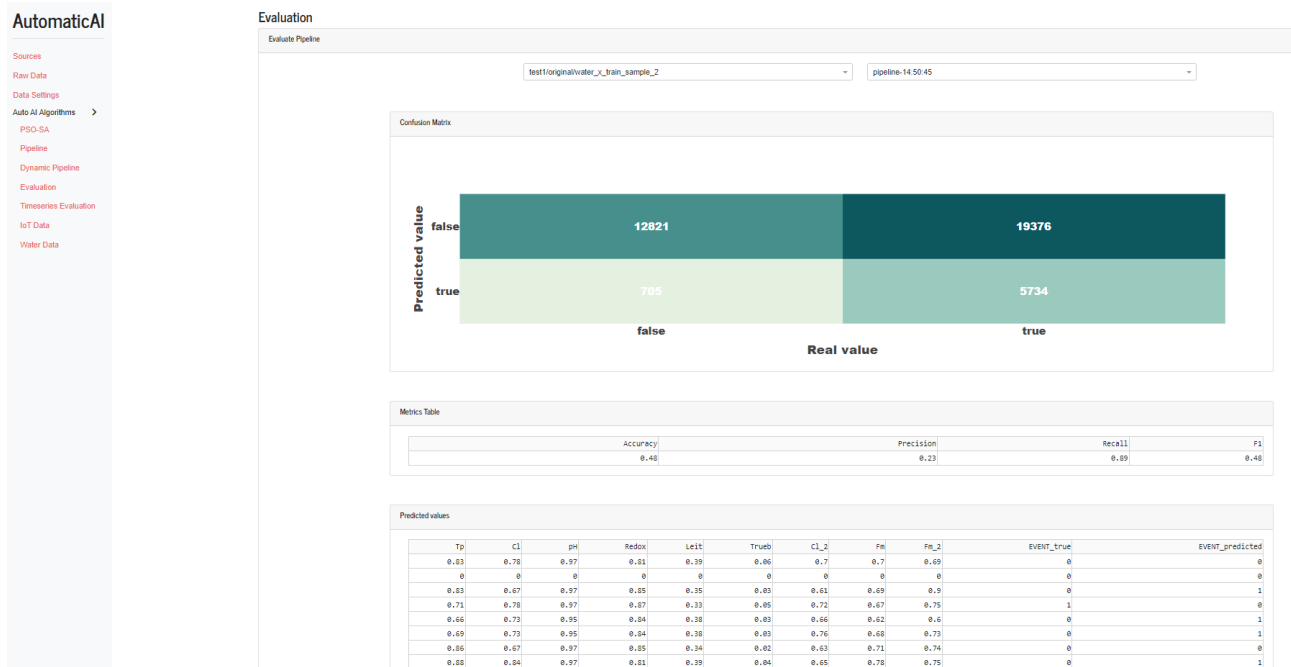


Figura 39 Evaluarea algoritmilor de clasificare

Datele citite de la dispozitive Edge pot fi vizualizate pe pagina IoT Data, unde putem selecta dispozitivul dorit folosind dropdown-ul de sus și putem filtra datele folosind un interval de timp. Valorile seriei de timp vor fi afișate ca o diagramă de linii și locația senzorului va fi afișată pe hartă (vezi Figura 40).

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

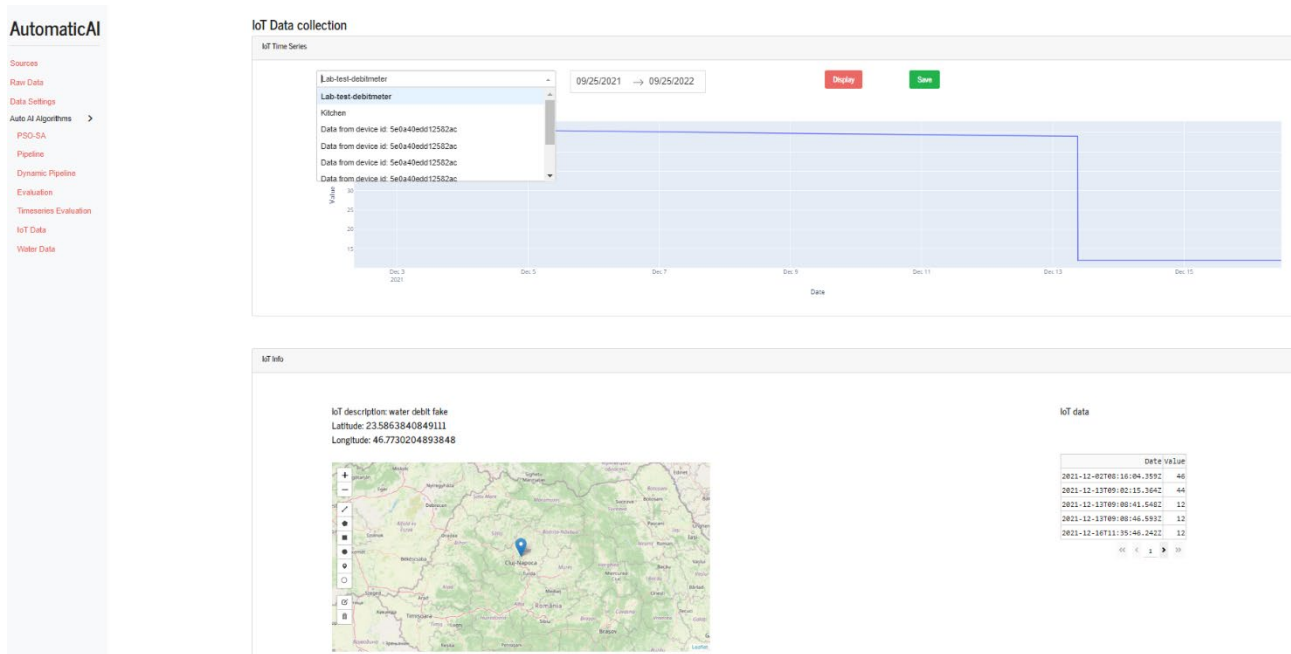


Figura 40 Citirea datelor de la dispozitive Edge

Datele citite de la senzorii de apă pot fi vizualizate pe pagina Water Data, unde putem selecta senzorul dorit folosind dropdown-ul de sus și putem filtra datele folosind un interval de timp. Valorile seriei de timp vor fi afișate ca o diagramă de linii și locația senzorului va fi afișat pe hartă (vezi Figura 41).

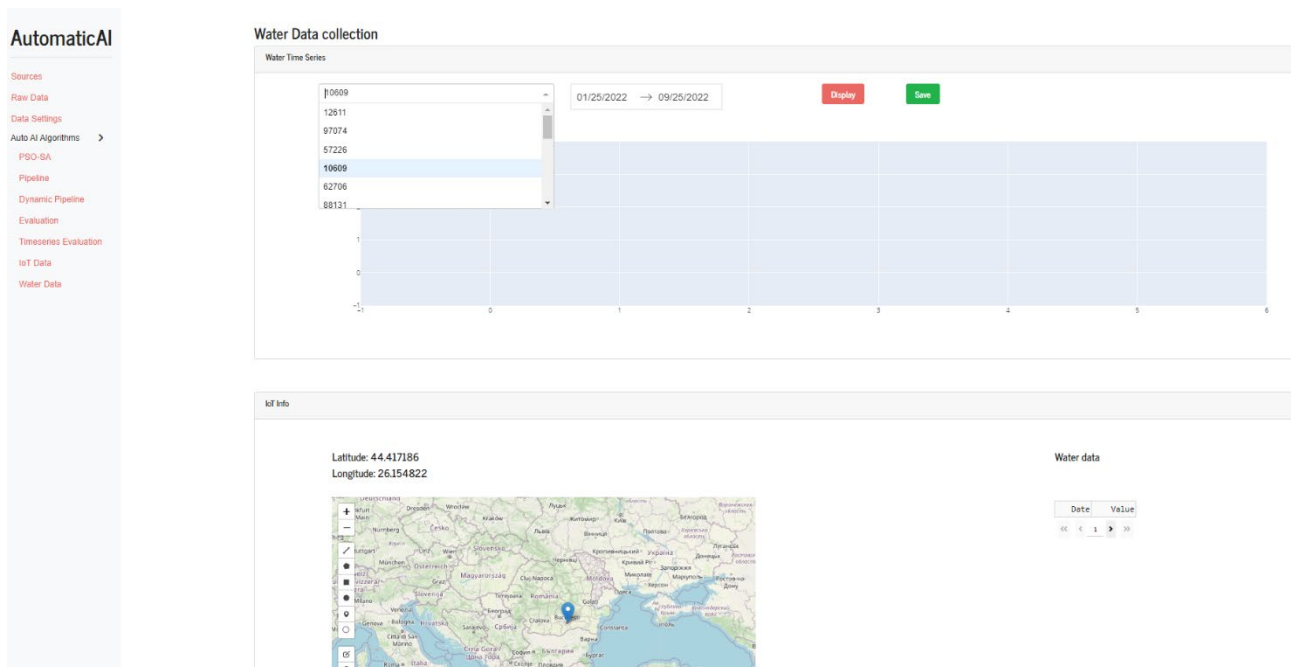


Figura 41 Lista senzorilor existenți

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

După selecția senzorului și filtrarea valorilor, datele pot fi salvate în baza noastră de date pentru procesare și analiză (vezi Figura 42 și Figura 43).

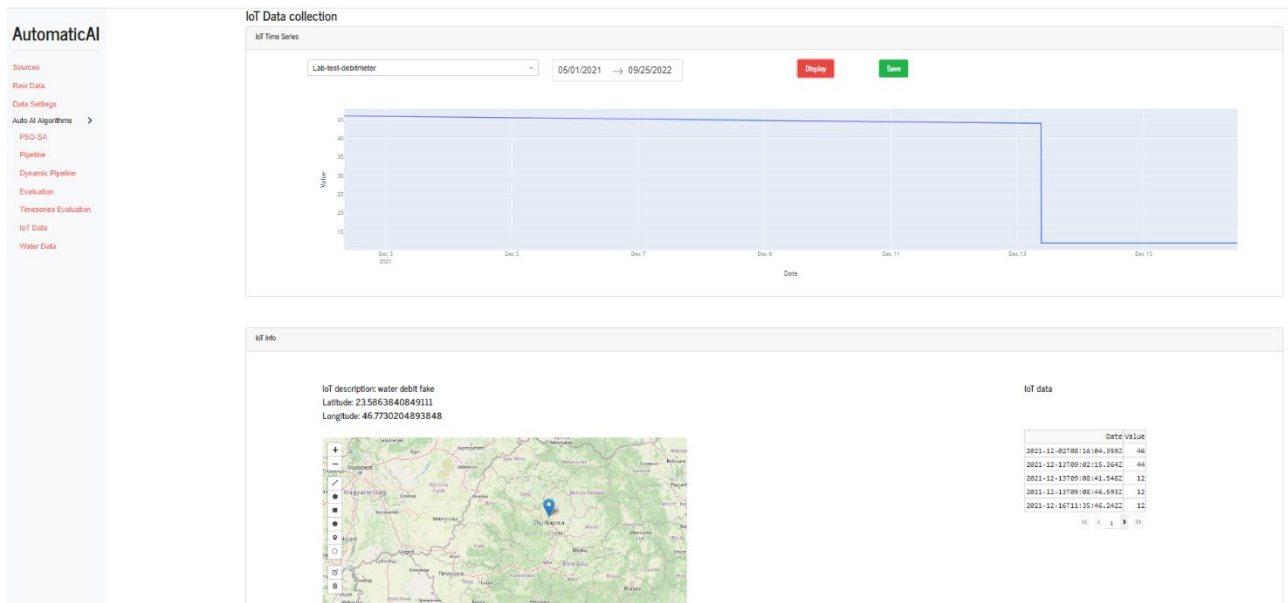


Figura 42 Citirea datelor de la senzorii de consum de apă

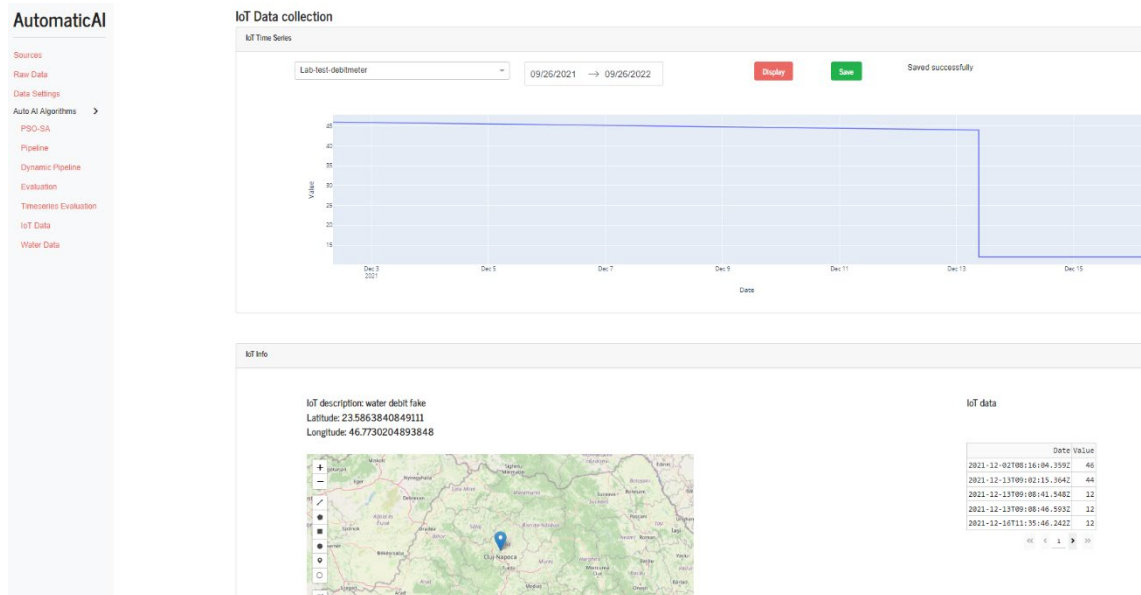


Figura 43 Salvarea datelor citite în baza de date

Datele salvate de la senzori pot fi vizualizate sub meniul Raw data (vezi Figura 44) și pot fi folosite ca orice set de date încărcat de utilizator.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

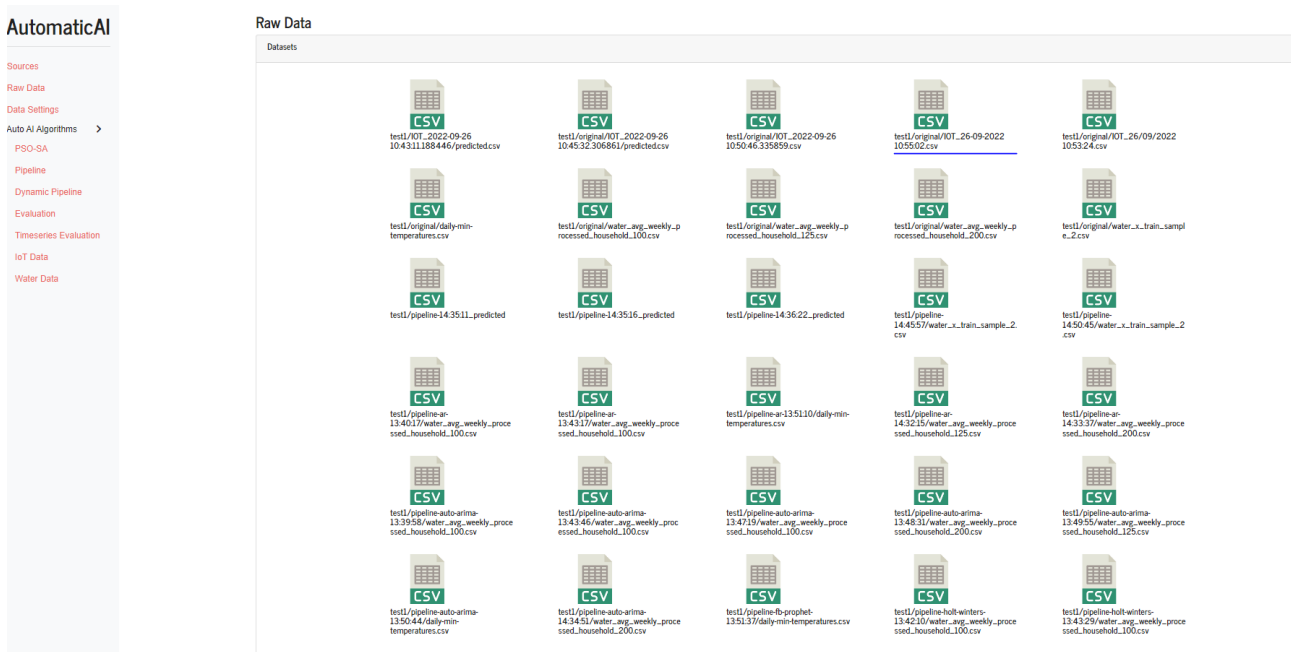


Figura 44 Setul de date salvat

Așa cum am menționat în secțiunile de mai sus, datele de tip real time sunt consumate și salvate de sistemul nostru folosind un client MQTT (vezi Figura 45).

```
pika.adapters.blocking_connection INFO Created channel=1  
creating new instance  
connecting to broker  
subsribing to mqtt topics  
client: <paho.mqtt.client.Client object at 0x0000023C5E19E5C8> connected  
apscheduler.scheduler INFO Adding job tentatively -- it will be properly scheduled when the scheduler starts  
apscheduler.scheduler INFO Added job "fetch_data_from_api" to job store "default"  
apscheduler.scheduler INFO Scheduler started  
apscheduler.scheduler INFO Adding job tentatively -- it will be properly scheduled when the scheduler starts  
apscheduler.scheduler INFO Added job "delete_old_data" to job store "default"  
apscheduler.scheduler INFO Scheduler started  
apscheduler.scheduler INFO Adding job tentatively -- it will be properly scheduled when the scheduler starts  
apscheduler.scheduler INFO Added job "train_ARIMA" to job store "default"  
apscheduler.scheduler INFO Scheduler started  
ENV VAR -----  
None  
mongodb://host.docker.internal:27018/automatic-ai  
6380  
  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: This is a development server. Do not use it in a production deployment.  
  Use a production WSGI server instead.  
* Debug mode: on  
werkzeug INFO * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figura 45 Conectarea si citirea datelor de la MQTT

În Figura 46 putem vedea managerul de RabbitMQ și anomaliile publicate pe coada previous-week, adică pentru săptămâna trecută.

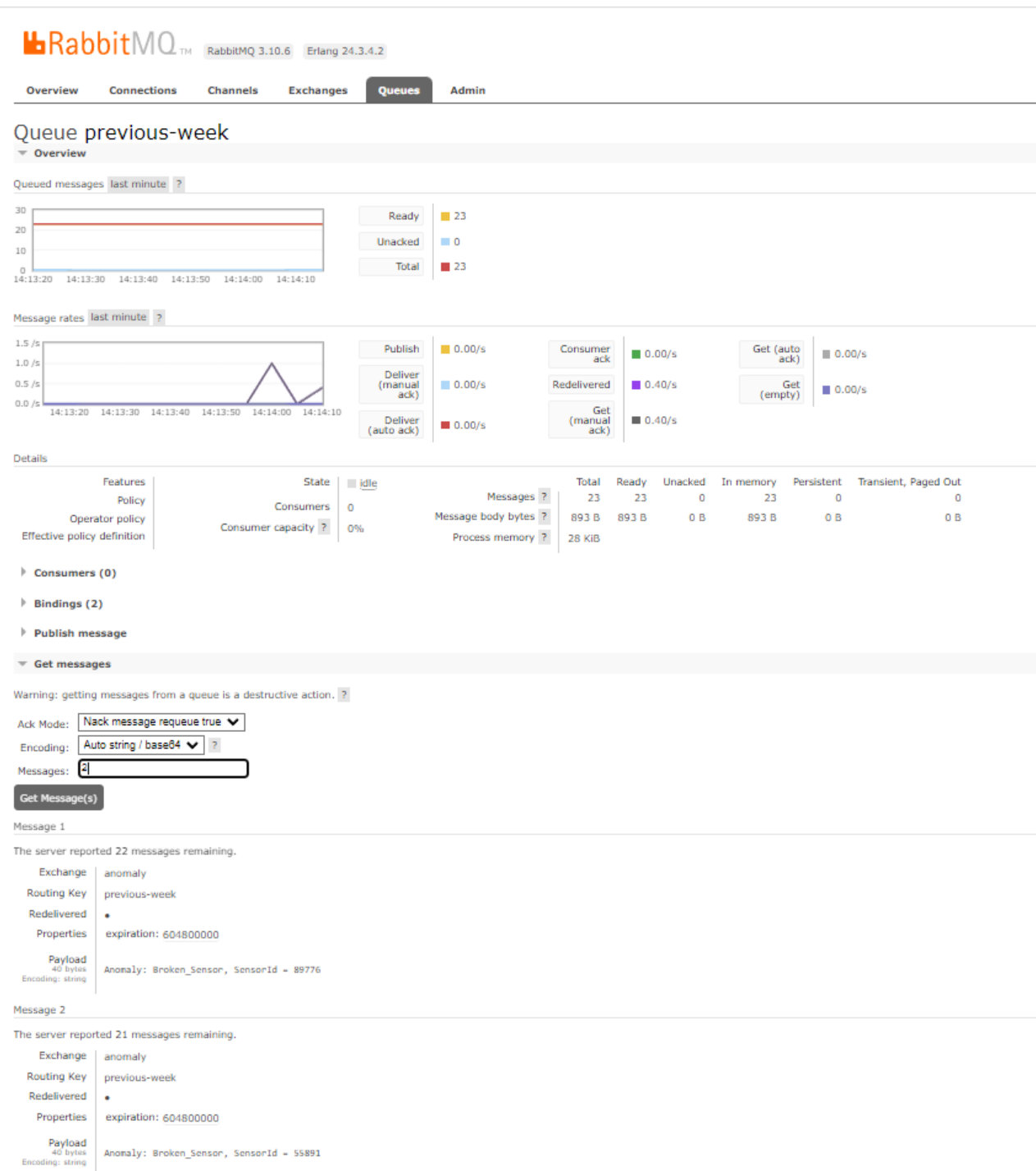


Figura 46 Coda de mesaj de anomalii pentru săptămâna anterioară

Figura 47 prezintă anomaliile generate pentru ziua anterioară.

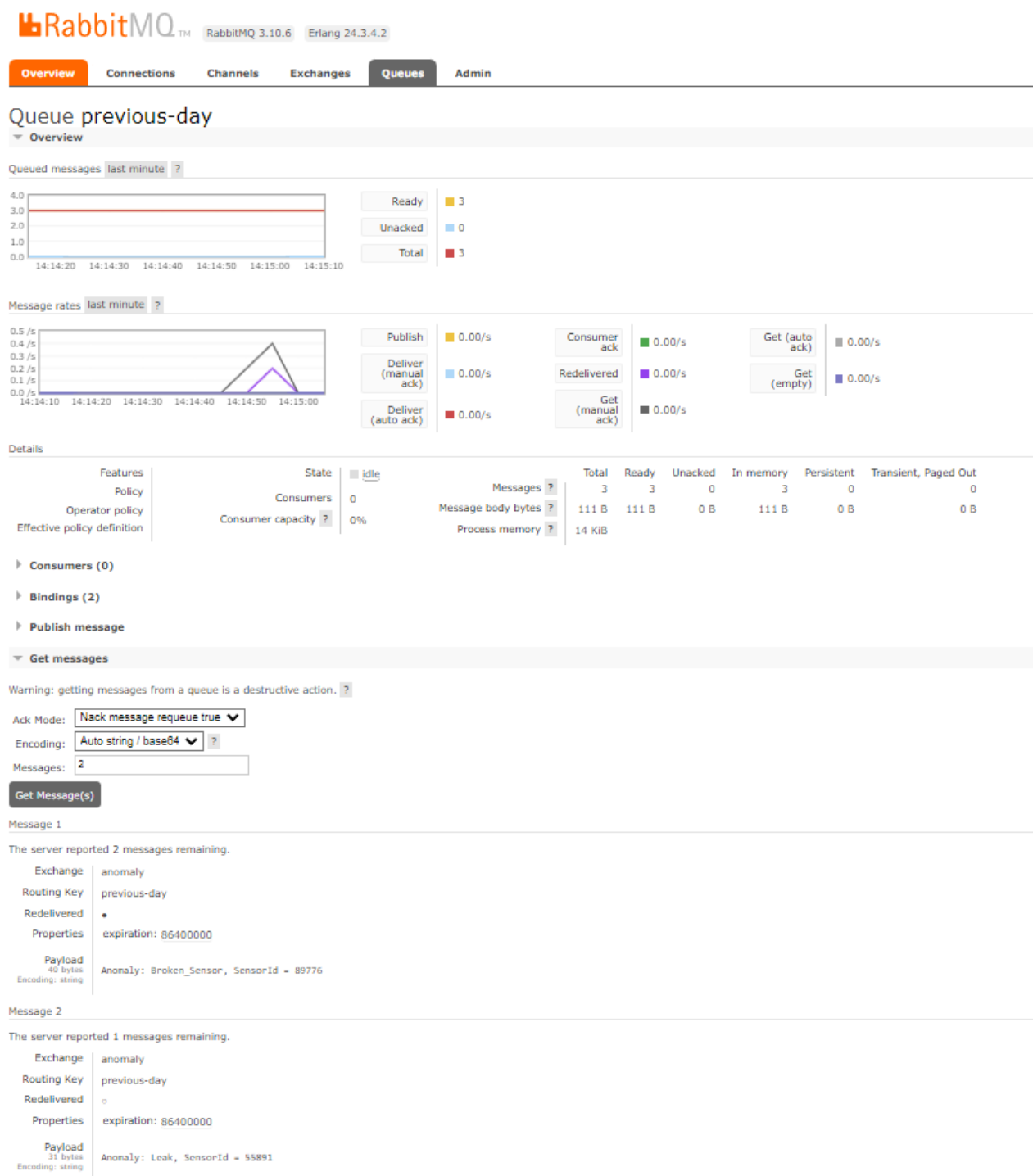


Figura 47 Coada de mesaj de anomalii pentru ziua anterioară

3.3 Vizualizare și suport decizional în rețeaua de senzori

În această secțiune prezentăm rezultatele experimentale obținute din datele colectate de sistemul de monitorizare a consumului de apă.

Datele sunt colectate folosind un timp fix de eșantionare de 60 de secunde pentru a capta un nivel ridicat de detaliu în ceea ce privește evenimentele de consum de apă. Datele brute de consum sunt prezentate în Figura 48, colectate de la unul dintre nodurile de măsurare instalate pe durata experimentului (două săptămâni).

Datele arată consumul pentru mai mulți senzori: chiuvetă (apă caldă), chiuvetă (apă rece), duș și toaletă.

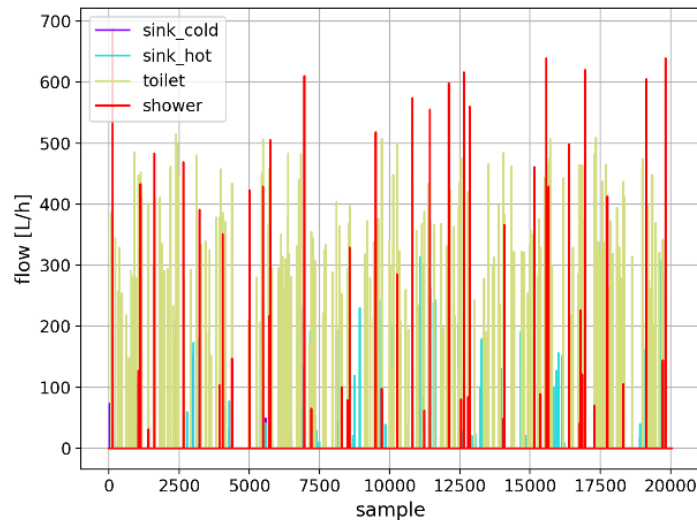


Figura 48 Date brute de consum

Pentru a vizualiza mai bine tiparele de consum și volumul total, am folosit un filtru low-pass pe setul de date, rezultatele fiind prezentate în Figura 49. Această perspectivă dezvăluie o analiză comparativă a diferitelor aparate. Se constată că toaleta reprezintă cel mai mare consum de apă, în comparație cu apa de la robinet (caldă/rece) și duș.

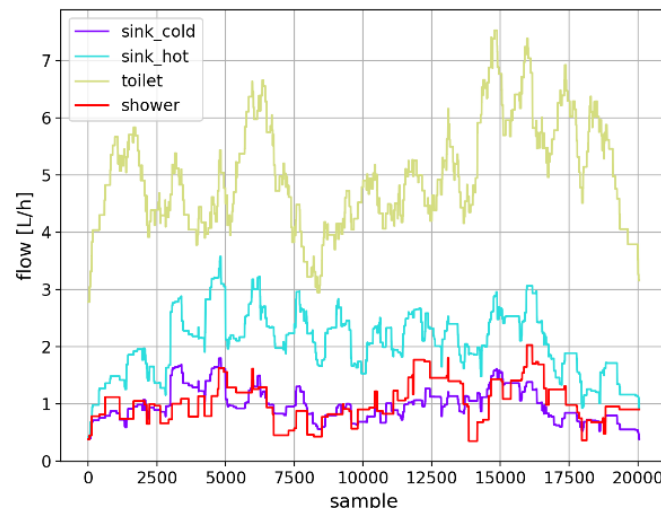


Figura 49 Date de consum filtrate

Apoi, am aplicat metoda de grupare K-Means pe datele din seria temporală de la toate nodurile de măsurare, cu etapa de filtrare low-pass. Numărul optim de clustere s-a obținut utilizând metoda

cotului, așa cum se arată în Figura 51, care a evidențiat un punct de inflexiune la 3 cluster. Astfel, au fost selectate 3 cluster, iar modelele rezultate sunt prezentate în Figura 50. Rezultatele reflectă faptul că gruparea poate oferi doar o înțelegere de bază a comportamentului consumului de apă, deoarece există un nivel mai ridicat de detaliu care trebuie dezvăluit din datele disponibile.

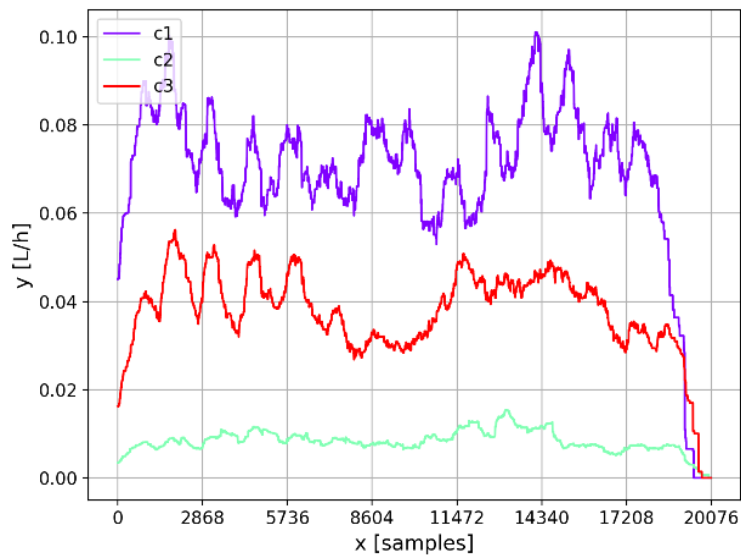


Figura 50 Gruparea seriilor de timp

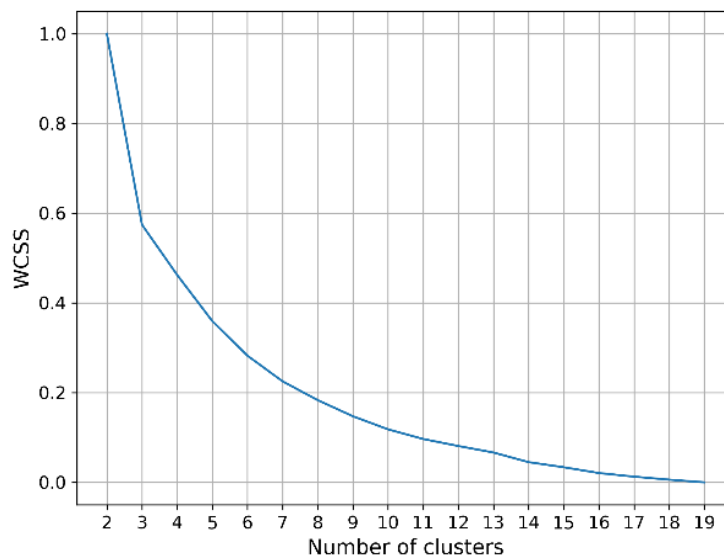


Figura 51 Metoda cotului

În continuare, datele de la chiuvetă au fost combinate (apă caldă și apă rece) pentru a oferi observații relevante cu privire la modelele de consum de apă.

3.3.1 Gruparea modelelor de consum

Datele brute de consum pentru un nod de consum sunt prezentate în Figura 52, pentru un interval de timp de două săptămâni.

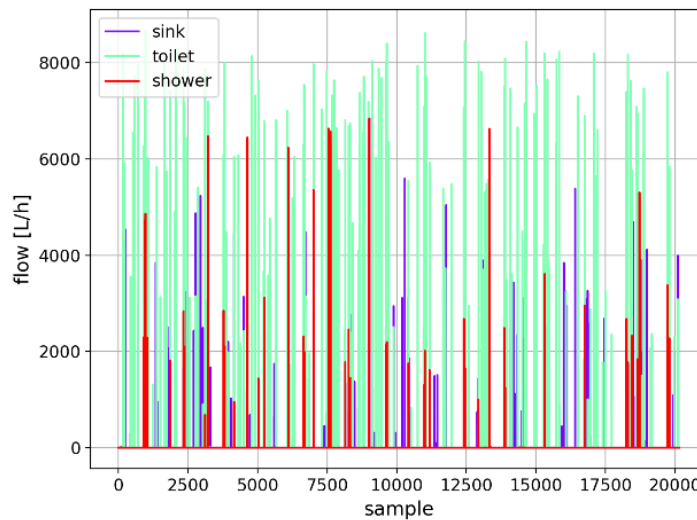


Figura 52 Date brute de consum

Graficul dezvăluie câteva observații generale despre evenimentele de consum din diferite puncte de colectare:

- Cele mai frecvente evenimente pot fi extrase din datele legate de consumul de la toaletă.
- Evenimentele de consum de la toaletă se caracterizează printr-un volum constant, care poate fi explicat prin capacitatea fixă a rezervorului de apă, consumând aproximativ aceeași cantitate de apă pentru fiecare eveniment.
- Evenimentele de duș se caracterizează printr-o variabilitate ridicată, care poate fi atribuită unei game mai largi de modele de consum, de exemplu, de la dușuri rapide până la baie.
- Evenimentele de la chiuvetă (apă caldă/rece) sunt mai consistente decât evenimentele de duș, fără prea multe vârfuri.

Pentru a evalua în continuare evenimentele de consum pe baza orei din zi la care au fost făcute, au fost extrase măsurătorile zilnice pentru fiecare punct de consum, relevând modele diferite, după cum se arată în Figura 53.

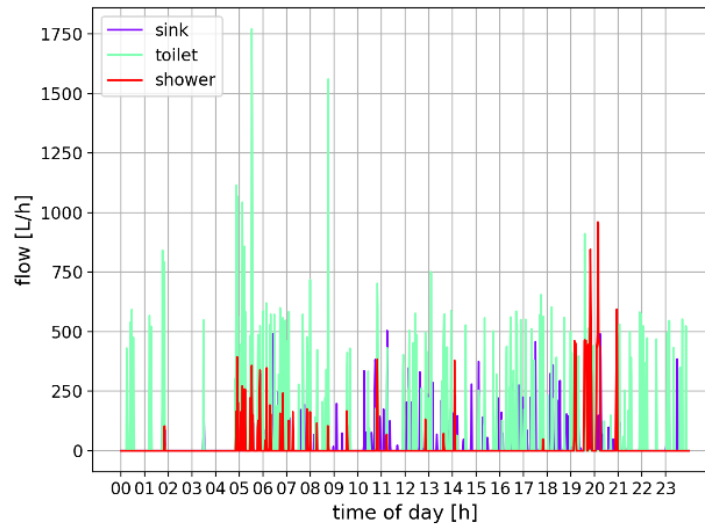


Figura 53 Evenimente de consum zilnic

În Figura 53 este reprezentată o medie pentru intervalele zilnice în ceea ce privește evenimentele de consum, care caracterizează nivelul de consum pentru fiecare moment al zilei.

Se poate observa că evenimentele de duș se înregistrează mai ales dimineața și seara, în timp ce consumul de la toaletă este mai mare dimineața și mai constant în timpul zilei. Consumul de la chiuvetă este mai mare și mai consistent în timpul zilei. Pe timpul nopții, consumul total a fost în general mai mic pentru fiecare punct de consum.

Pentru a evalua tendințele de consum, setul de date a fost preprocesat folosind un filtru de medie mobilă, așa cum se arată în Figura 54.

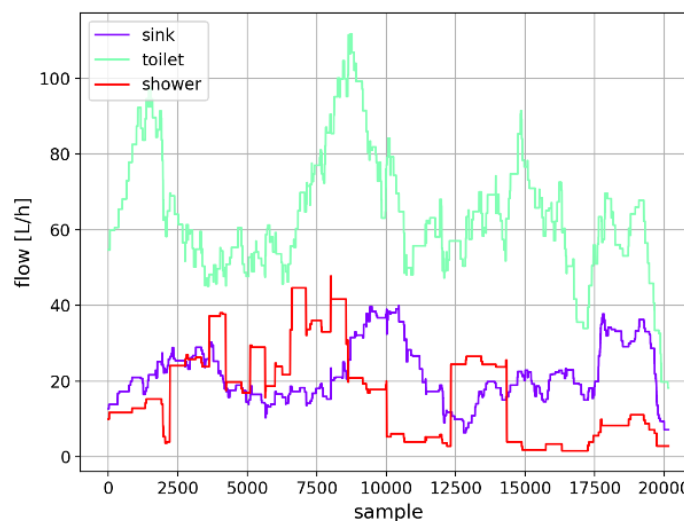


Figura 54 Date de consum filtrate

S-a folosit o matrice de convoluție pentru a implementa filtrul, unde dimensiunea nucleului a fost definită ca o zecime din numărul de mostre. Această valoare a fost selectată pentru a oferi o imagine de ansamblu asupra modelelor de consum, arătând tendințe mai clare și un nivel moderat de variație

pentru a caracteriza fiecare punct de consum. Această metodă dezvăluie în mare parte o comparație între punctele de consum în funcție de volumul mediu.

Se poate observa faptul că este mai puțin consistent consumul de la duș decât pentru celelalte puncte de consum, ceea ce poate contribui la o predicție mai dificilă și la o precizie mai scăzută. Prin urmare, se constată că există multe caracteristici ale evenimentelor individuale de consum care ar trebui să permită diferențierea între diferitele puncte de consum folosind algoritmi de învățare automată. Astfel, evenimentele de consum individuale au fost extrase din seriile cronologice colectate și apoi grupate pe puncte de consum, adică chiuvetă (apă rece/caldă), toaletă și duș.

3.3.2 Clasificarea evenimentelor de consum

Setul de date original a fost utilizat pentru a extrage evenimentele în termeni de durată și volum, ceea ce a generat un set de date de evenimente de consum de 6323 de intrări. Setul de date a fost împărțit în date de antrenament și de testare, folosind o împărțire 80/20.

Algoritmii de clasificare propuși (Decision Tree, Random Forest, Dense Neural Network și Recurrent Neural Network) au fost utilizați pentru a evalua acuratețea clasificării pentru prezicerea punctelor de consum pe baza evenimentelor extrase din date.

Rezultatele acestui experiment sunt incluse în Figura 55 și arată acuratețea fiecărui model pe setul de date de antrenament și testare.

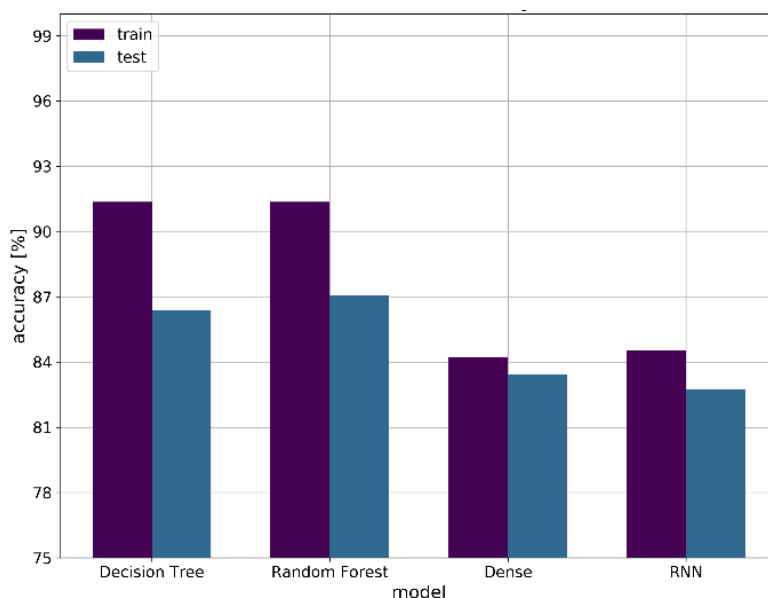


Figura 55 Acuratețea clasificării

Modelul Random Forest oferă o acuratețe puțin mai bună decât modelul Decision Tree, adică 87,1% comparat cu 86,4%, în timp ce modelele de învățare profundă oferă o acuratețe mai mică, adică 83,4% pentru Dense și 82,7% pentru RNN.

Precizia mai scăzută a modelelor Dense și RNN pe setul de date despre evenimentele de consum poate fi explicată prin cantitatea mai mică de date utilizate pentru antrenament, în comparație cu valorile recomandate pentru modelele de învățare profundă. Mai mult, din cauza distribuției inegale

a senzorilor instalați pentru fiecare punct de consum, setul de date a fost dezechilibrat și au existat considerabil mai multe date privind consumul chiuvetei și toaletei, decât în ceea ce privește consumul de apă de la duș.

Figura 56

Figura 56

3.4 Raportarea evenimentelor în infrastructura de apă

Soluția propusă pentru raportarea evenimentelor reprezintă o componentă dedicată în cadrul sistemului propus pentru realizarea scenariilor de operare în infrastructura urbană de apă. Sistemul de raportare integrează modelul de crowdsensing într-o platformă interactivă pe bază de joc serios cu suport blockchain.

3.4.1 Arhitectura jocului serios

Pentru jocurile serioase pe bază de geo-locăție în care raportările sunt distribuite pe hartă, arhitectura jocului este proiectată pentru a obține o distribuție optimă a jucătorilor. Această problemă poate fi definită ca o problemă de optimizare a alocărilor combinată cu un mecanism de stimulare bazat pe recompense. Sarcinile implică raportarea problemelor în contextul infrastructurii de apă urbană și pot fi extinse pentru multe cazuri de utilizare.

Problema optimizării alocărilor poate fi formulată ca o aplicație a problemei rutării vehiculelor - VRP (Vehicle Routing Problem), așa cum se arată în Figura 57, unde jucătorii (în cazul nostru $P_1 - P_3$) sunt modelați ca vehicule și sarcinile (adică $T_1 - T_{10}$) sunt modelați ca clienți, cu scopul de a optimiza costurile (de exemplu, distanța parcursă) în condiții de constrângeri specifice.

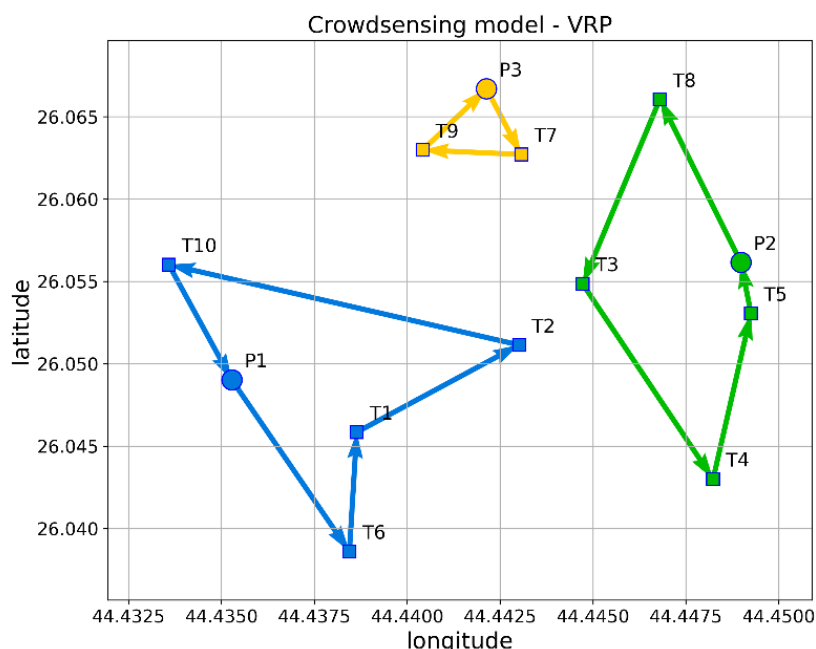


Figura 57 Simularea modelului VRP pentru optimizarea raportărilor

Pentru a crește utilitatea generală a platformei, stimulentele bazate pe recompense sunt contrabalansate de o strategie de reglementare. Reputația reprezintă metoda de validare a participanților și este definită pentru a obține stimulente optime în activitățile de raportare, reducând în același timp riscul de supra-raportare sau de raportare falsă.

În acest sens, problema VRP este formulată având constrângerile de capacitate pentru fiecare jucător în funcție de reputație. La fiecare iterație a algoritmului de optimizare, capacitățile sunt actualizate. În cazul raportării veridice, capacitatea este mărită în funcție de recompensă, ceea ce permite direcționarea către următoarele sarcini.

Pentru a evalua modelul și diferitele tipuri de comportament, am simulat problema optimizării pe mai multe distribuții ale coordonatelor geografice ale jucătorilor. Am avut în vedere coordonate fixe pentru rapoartări și coordonate variabile de pornire pentru jucători, pentru a evalua costul mediu al sistemului (distanța totală parcursă) și, prin urmare, pentru a evalua utilitatea platformei pentru încurajarea participării veridice.

Pentru transpunerea formulării problemei în constrângeri VRP am luat în considerare următoarele: (1) reputația jucătorului este reprezentată de capacitatea vehiculului; (2) efortul jucătorului este reprezentat de nivelul de combustibil al vehiculului; (3) prioritatea sarcinii este reprezentată de cererea clientului.

3.4.2 Evaluarea scalabilității

Având în vedere tehnologiile actuale, dezvoltarea mobilă hibridă a devenit o alternativă viabilă atât pentru prototiparea rapidă, cât și pentru aplicațiile complexe, consumatoare de resurse, cu costuri de dezvoltare mai mici datorită instrumentelor cross-platform și omniprezenței tehnologiilor web în

Întreaga stivă de dezvoltare. Un studiu recent a scos la iveală o comparație bazată pe cercetare între dezvoltarea aplicațiilor native și hibride, în care abordarea hibridă s-a dovedit a fi aproape egală sau chiar superioară în următoarele criterii: Performanță, Interfața cu Utilizatorul, Design și Funcționalitate (Enihe, 2020). Concluzii similare au fost găsite în studii recente similare care au comparat cadrele de dezvoltare a aplicațiilor mobile hibride cu abordarea nativă (Denko, 2021).

În contextul infrastructurii tehnice, baza de date MySQL, utilizată pentru stocarea datelor legate de raportări, este de tip relațional, în timp ce datele colectate de fiecare jucător sunt stocate pentru fiecare raportare, pentru a fi disponibile altor jucători. Prin urmare, de fiecare dată, un număr considerabil de tabele este utilizat pentru a stoca și extrage raportările și caracteristicile asociate. Având în vedere complexitatea sistemului integrat, scalabilitatea reprezintă un aspect important pentru o funcționare optimă.

Prin urmare, au fost prevăzute componente adiționale pentru a gestiona volumul de lucru, așa cum se arată în Figura 58, unde este prezentată arhitectura sistemului împreună cu extensiile propuse pentru scalabilitate în producție.

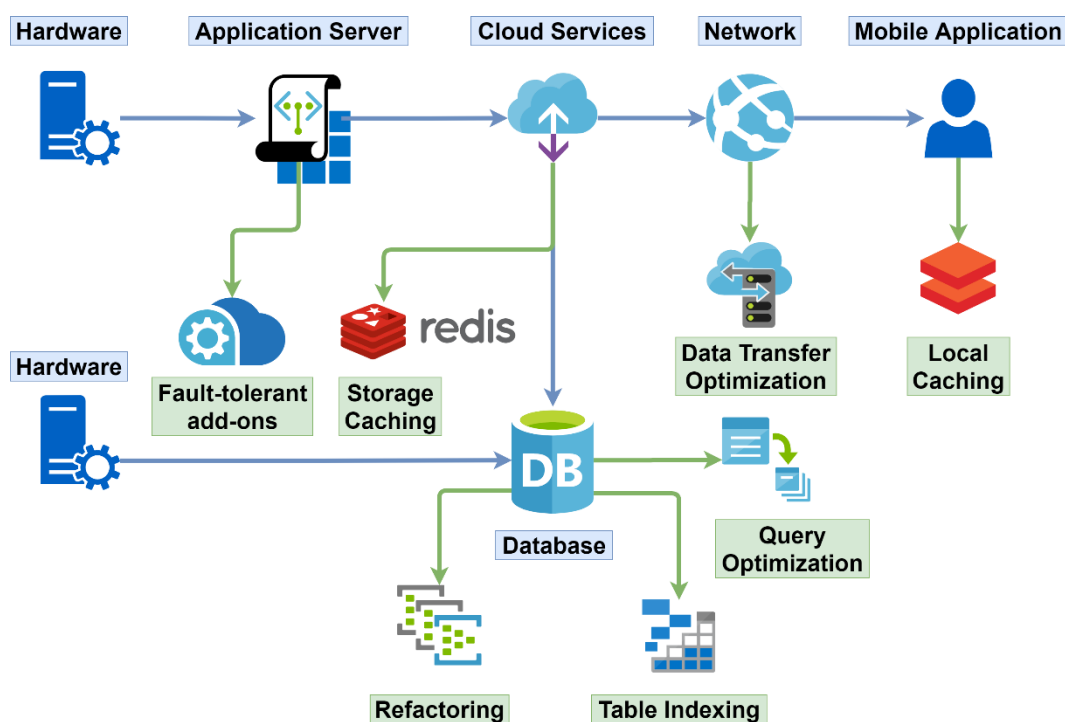


Figura 58 Arhitectura sistemului și extensii de scalabilitate

La nivelul de serviciu (Application Server), au fost adăugate extensii pentru a îmbunătăți toleranța la erori, în timp ce subsistemul REDIS a fost adăugat pentru a crește debitul de procesare a datelor din baza de date. Baza de date a fost concepută pentru a gestiona complexitatea prin indexarea atributelor cheie și optimizarea interogărilor, în timp ce transferul de date a fost optimizat prin utilizarea memorării în cache locale, pentru a reduce cantitatea de date care este transferată între aplicația web și server.

3.5 Blockchain

3.5.1 Setarea pre-rechizitelor pentru experimentarea soluției blockchain

Primul pas pentru experimentarea soluției blockchain, atât independent cât și integrată în cadrul proiectului este reprezentată de satisfacerea dependențelor specifice pentru a putea comunica într-o arhitectura descentralizată. În acest scop, este necesară efectuarea pașilor descriși mai jos.

Instalare Metamask

Instalarea extensiei Metamask pentru browser-ul folosit. Adresa oficială de unde se poate descărca aceasta extensie este: <https://metamask.io/>.

Crearea unui portofel electronic

După instalarea aplicației Metamask, utilizatorul poate crea un portofel electronic așa cum se poate vizualiza în Figurile următoare.

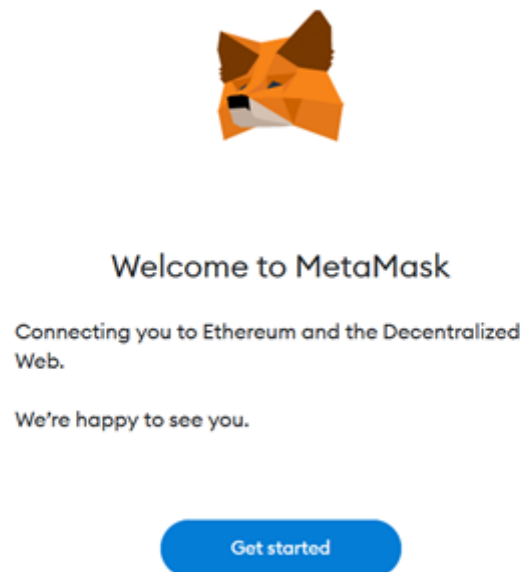
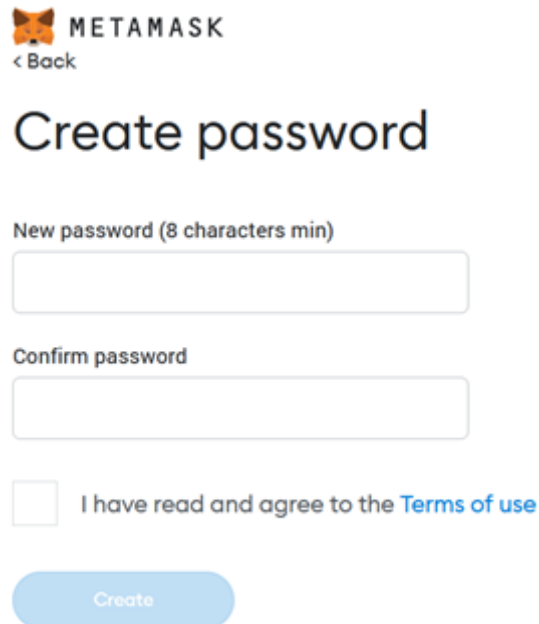


Figura 59 Pagina de start Metamask

- Este necesară crearea unui wallet nou la importul unui wallet existent. Pentru importul unui wallet existent este nevoie de „Cheia secretă de recuperare” a contului. Pentru crearea unui wallet nou se parcurg următorii pași: crearea unei parole si foarte important salvarea „Parolei secrete de recuperare” a contului. De menționat că această parolă este singura metodă prin care se poate avea acces la portofel, iar în cazul pierderii ei, nu se mai poate recupera contul.



The image shows the 'Create password' screen in the Metamask mobile app. At the top, there is the Metamask logo and a '< Back' link. The main heading is 'Create password'. Below it, there is a text prompt 'New password (8 characters min)' followed by a text input field. Underneath that is another text prompt 'Confirm password' followed by a second text input field. Below the input fields is a checkbox and the text 'I have read and agree to the [Terms of use](#)'. At the bottom, there is a blue 'Create' button.

Figura 60 Creare parola Metamask

- După setarea parolei și acceptarea condițiilor, contul de Metamask va apărea ca în Figura 61.

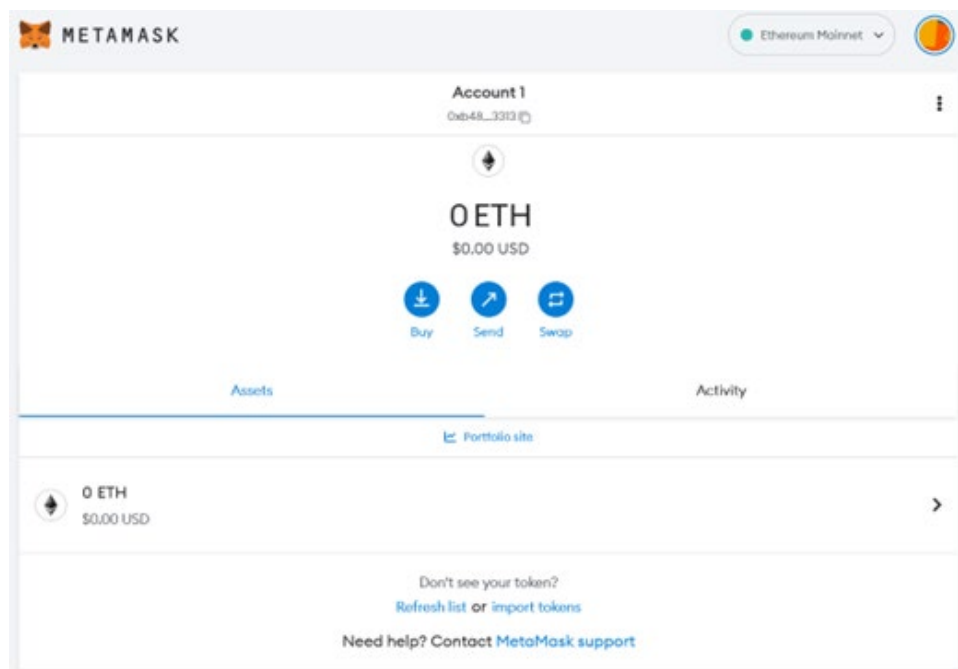


Figura 61 Vizualizare cont Metamask

Obținere de monedă virtuală pe rețeaua de test Ethereum

Acest lucru este necesar pentru a putea efectua tranzacții pe blockchain în cadrul rețelei peste care s-a dezvoltat soluția.

În cadrul sistemului implementat a fost folosită rețeaua de test Ethereum Goerli, o rețea publică accesibilă tuturor utilizatorilor.

Pentru a obține ethereum de test, putem accesa unul dintre faucet-urile oficiale, regăsite și pe site-ul ethereum goerli: <https://goerlifaucet.com/> sau <https://goerli-faucet.mudit.blog/>.

De menționat că pentru efectuarea acestei operațiuni de către rețeaua alchemy, este necesară crearea unui cont. După crearea contului, la 24 de ore, utilizatorul poate cere ethereum de test de la acest faucet prin introducerea adresei portofelului sau electronic și acționarea butonului de trimitere ethereum.

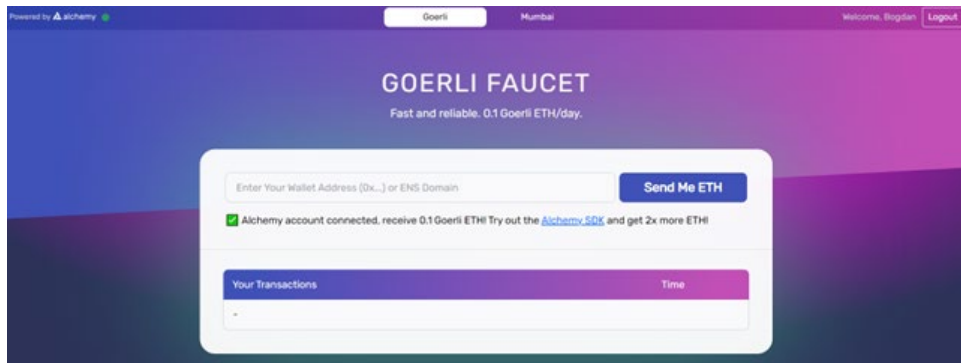


Figura 62 Obținere Ethereum de test prin Alchemy

3.5.2 Experimentarea soluției blockchain

Primul pas în experimentarea soluției blockchain este experimentarea modului de obținere a token-ului WGT (WaterGame Token). Acesta se poate obține prin notificarea în aplicația de crowdsensing a diverselor incidente în infrastructura de apă (Figura 63).

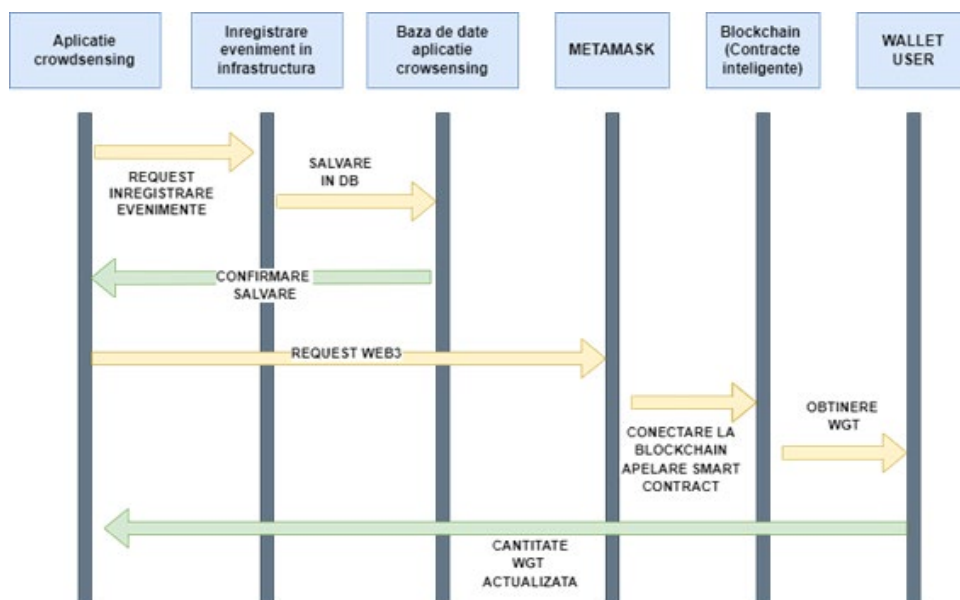


Figura 63 Diagrama de secvență pentru obținerea WGT la înregistrarea unui eveniment în aplicația de crowdsensing

De asemenea, alternativ se mai poate opta pentru a cumpăra moneda prin intermediul ICO-ului (Figura 64).

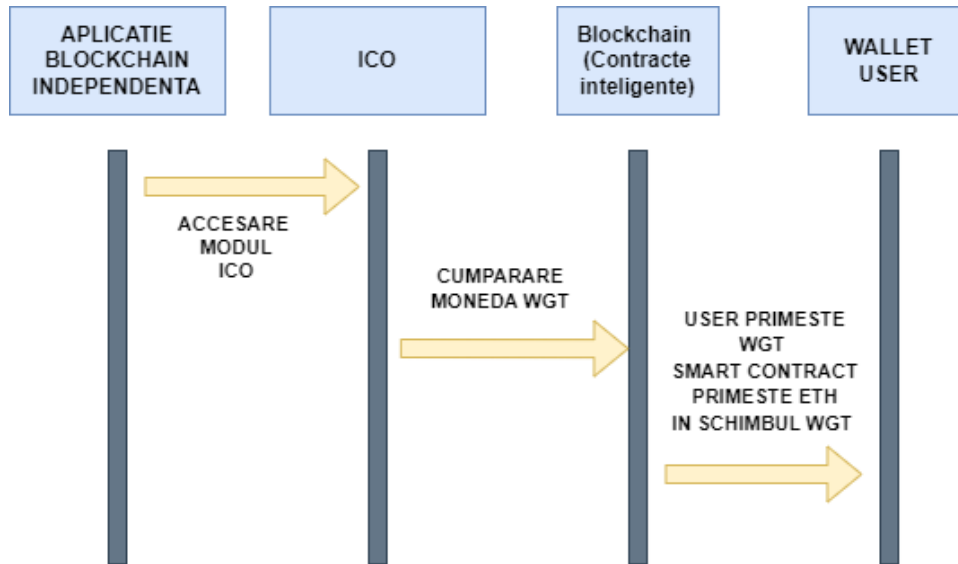


Figura 64 Diagrama de secvență obținere WGT prin ICO

Nu în ultimul rând, aplicația de blockchain permite și obținerea de WGT prin intermediul donației de la un prieten.

Din punct de vedere al folosirii token-ului, principalul mod de utilizare este pentru a cumpăra diverse asset-uri în aplicația de crowdsensing. Acest scenariu este prezentat în Figura 65 prin intermediul unei diagrame de secvență.

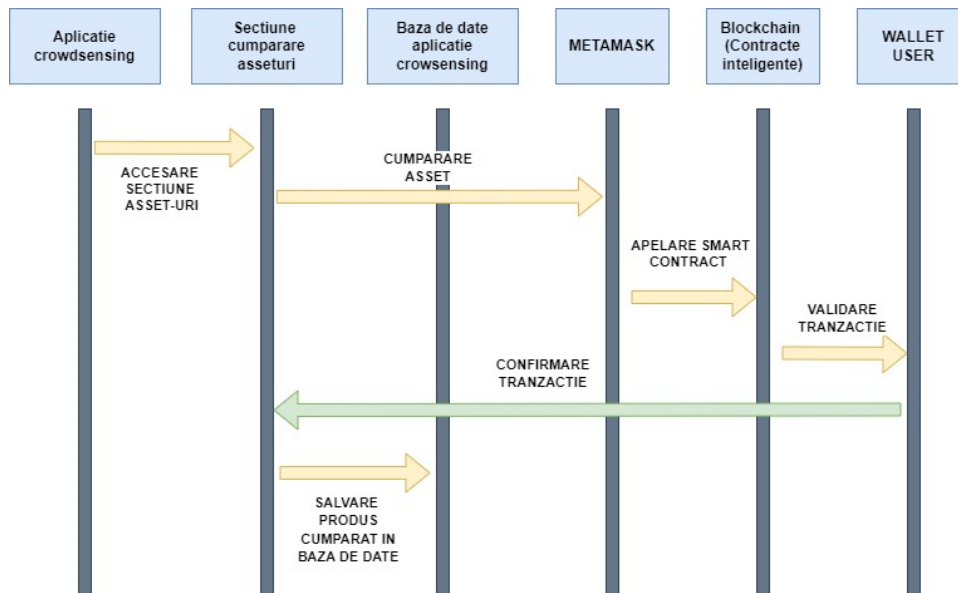


Figura 65 Diagrama de secvență cumpărare asset-uri prin WGT

4 Validarea sistemului și demonstrarea funcționalităților

În această etapă, a fost validată funcționarea sistemului integrat la nivel de laborator. Sunt prezentate instrumentele, metodele, și interacțiunile la nivel de sistem, precum și rezultatele experimentale obținute în contextul scenariilor definite în etapa de proiectare.

4.1 Verificarea funcționării sistemului integrat

În această activitate, au fost realizate integrările propuse, iar sistemul a fost validat în ansamblu, pe baza scenariilor experimentale.

4.1.1 Monitorizare în rețeaua de senzori

Analiza datelor

Pentru a analiza datele primite de la senzori, am utilizat platforma de analiză de date și am antrenat mai multe modele ARIMA pentru predicția consumului de apă. Figura 66 prezintă un exemplu în care am antrenat un model ARIMA pe datele primite de la senzorul 89776. După antrenarea modelului, în partea de jos putem vedea metrica de validare RMSE (în cazul acesta e 2.231 ceea ce e o valoare foarte bună), precum și valorile reale (prezentate pe diagramă folosind culoarea albastră) și valorile prezise de model (marcate cu roșu). Așa cum se vede și în figură, am reușit să creăm un model foarte precis, linia roșie (valori prezise) se suprapune cu linia albastră (valori reale) aproape perfect.

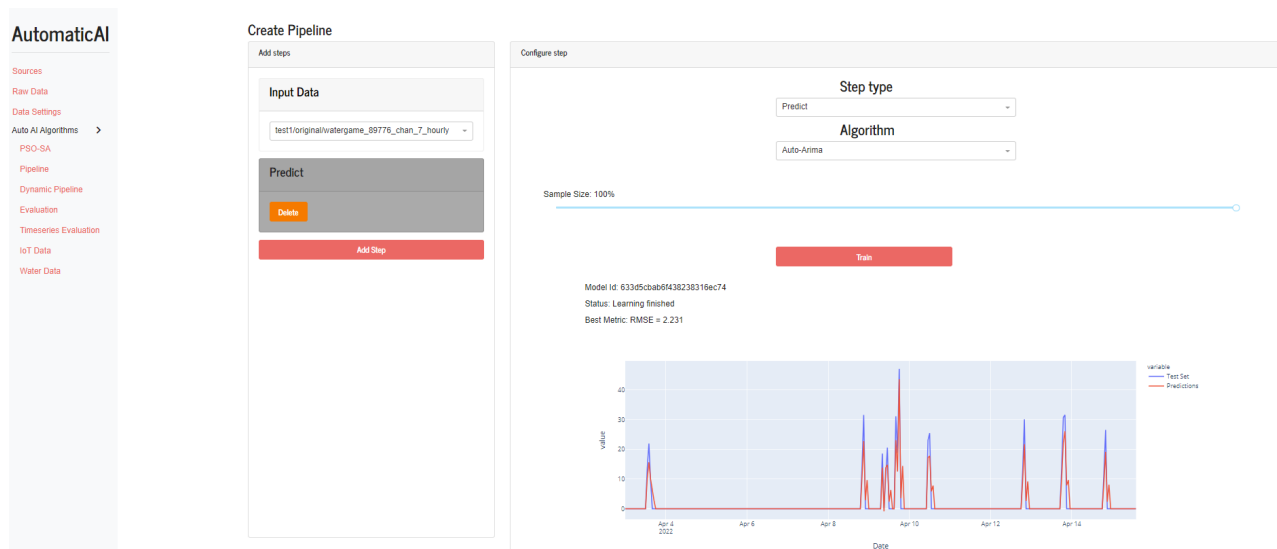


Figura 66 Antrenarea modelului ARIMA pe date citite de la senzori

Pentru fiecare senzor am antrenat modele separate de tip ARIMA, astfel am reușit să maximizăm precizia și să minimizăm impactul unui senzor stricat. Figura 67 prezintă câteva rezultate obținute folosind date reale culese de la senzori.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

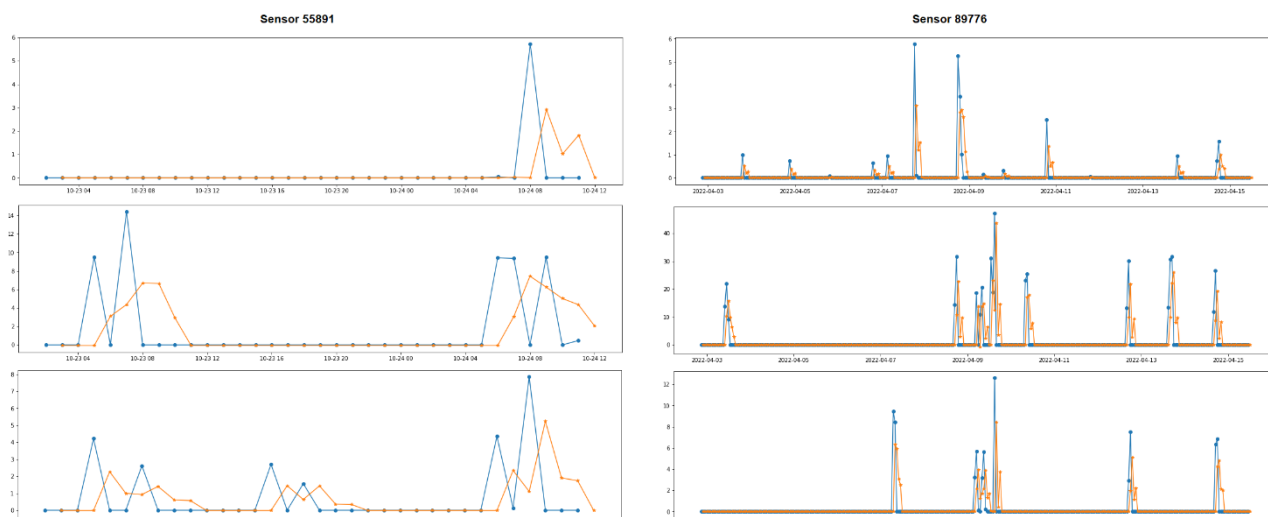


Figura 67 Rezultatele modelelor ARIMA antrenate pe date primite de la senzori

Așa cum se vede în Figura 67, în cazul senzorului 89776 avem un model mult mai precis, ceea ce se datorează faptului că în acest caz am avut mai multe date de antrenare.

Monitorizarea defecțiunilor

Pentru monitorizarea defecțiunilor, Prometheus oferă o interfață grafică pentru vizualizarea metricilor existente și a configurației actuale (Figura 68). Metricile sunt citite de la server la fiecare 5 secunde.

watergame (1/1 up) show less					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8081/metrics	UP	<code>instance="localhost:8081"</code> <code>job="watergame"</code>	2.26s ago	3.365ms	

Figura 68 Interfața Prometheus

Metrica poate fi vizualizată sub formă de tabel, evidențiind-se seriile de timp împreună cu ultima valoare înregistrată, așa cum se poate observa în Figura 69.

heart_beat	Table Graph	Load time: 55ms Resolution: 14s Result series: 2
← Evaluation time →		
<code>heart_beat(app="water_game_server", instance="localhost:8081", job="watergame", sensor_id="19673")</code>		1666030441961
<code>heart_beat(app="water_game_server", instance="localhost:8081", job="watergame", sensor_id="55891")</code>		1666030432987
Remove Panel		

Figura 69 Vizualizare metrice în format tabelar

O altă vizualizare, prezentată în Figura 70, este aceea sub formă de grafic, putând fi observată evoluția în timp a valorii seriilor de timp. Este evidențiată corelarea dintre momentul înregistrării (axa X) și momentul primirii mesajului de către server (axa Y):

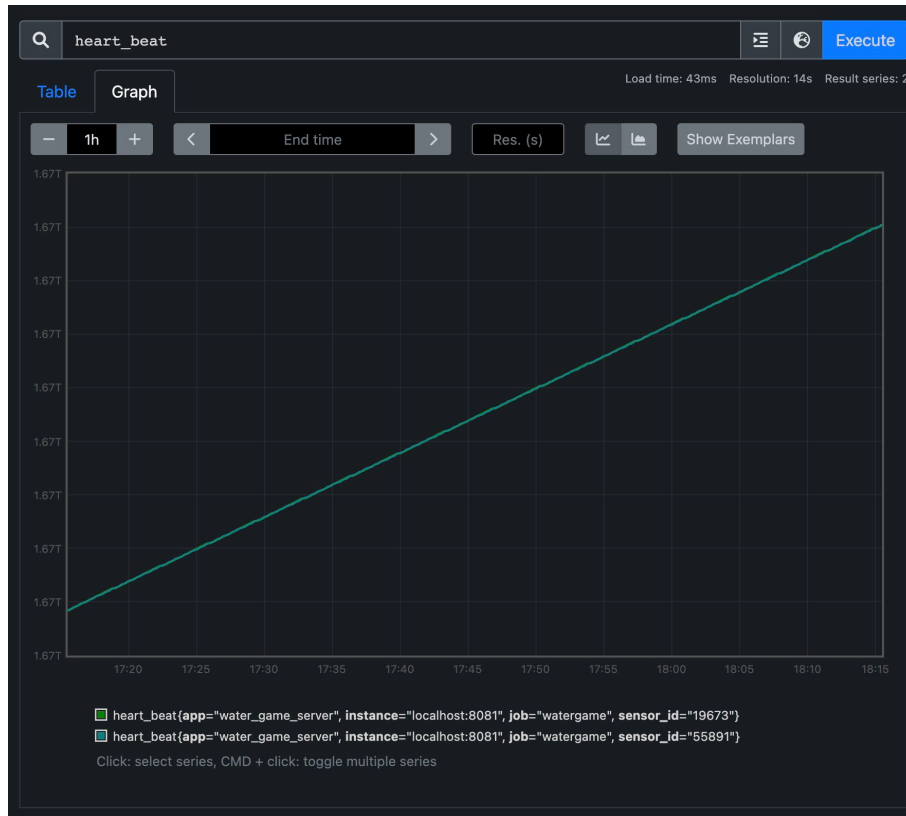


Figura 70 Vizualizare metrice în format grafic

Metrica **heart_beat** a fost agregată sub numele de **hb:time_since_last_seconds** care reprezintă timpul în secunde de la ultimul mesaj heartbeat. Agregarea are loc prin evaluarea la fiecare minut a expresiei: $(\text{time}() * 1000 - \text{last_over_time}(\text{heart_beat}[1y])) / 1000$

În graficul metricii agregate, prezentat în Figura 71, observăm menținerea unui comportament constant în condiții de funcționare normală a sistemului (senzorii trimit date).

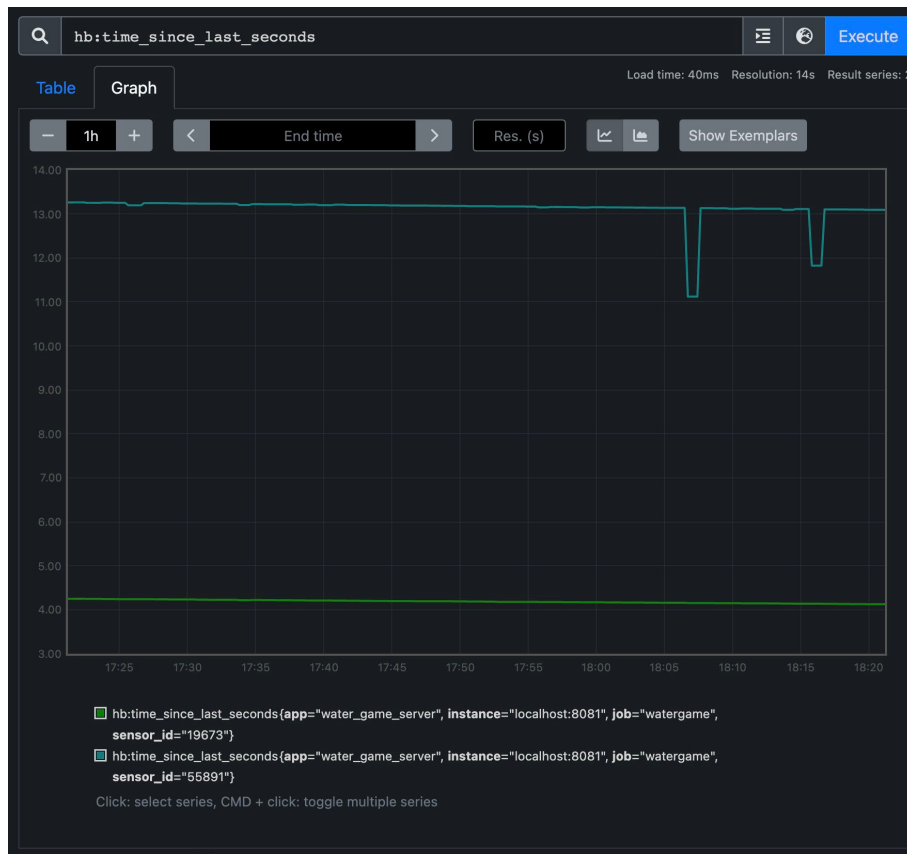


Figura 71 Vizualizare metrice agregate

Se observă o monotonie descrescătoare datorată de diferențele dintre ceasurile senzorilor care publică mesaje regulat și ceasul Prometheus care calculează agregatul periodic.

Alerta este configurată pentru a urmări valoarea metricei agregate (Figura 72).

```

✓ SensorDown (0 active)
name: SensorDown
expr: hb:time_since_last_seconds > 5 * 60
annotations:
  description: Sensor with id {{ $labels.sensor_id }} have sent last heartbeat more than 5 minutes ago
  summary: Sensor {{ $labels.sensor_id }} down
    
```

Figura 72 Configurarea alertei

Declanșarea unei alerte

În situația în care un senzor se oprește din a trimite mesaje de heartbeat, timestamp-ul ultimului mesaj va rămâne constant. Acest comportament se regăsește în captura de ecran din Figura 73.

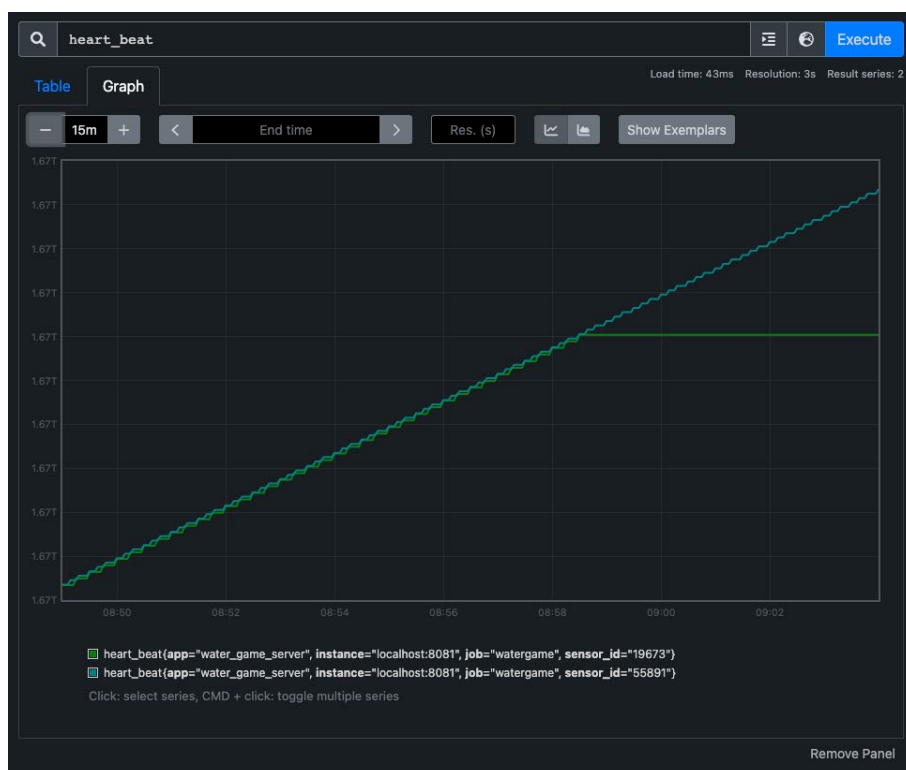


Figura 73 Declanșarea defecțiunii

Se observă că pentru senzorul cu id-ul 19673 ultimul mesaj a fost primit la 09:58:23 GMT+0.

În Figura 74, metrica agregată care reprezintă timpul de la ultimul mesaj va crește.

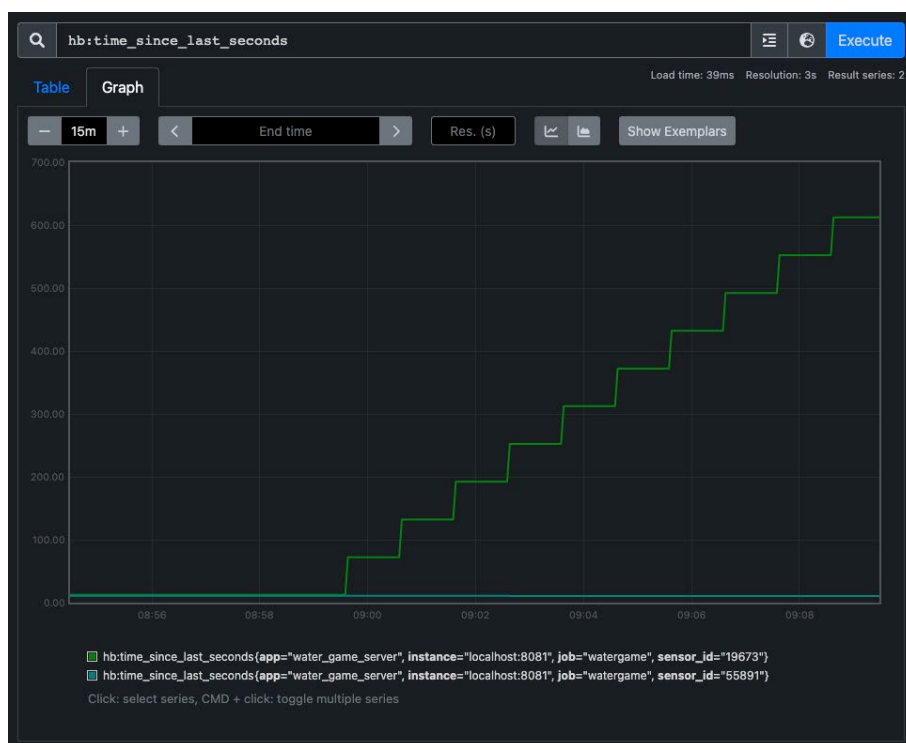


Figura 74 Detecția defecțiunilor prin metrica agregată

La 5 minute de la ultimul mesaj se declanșează automat alerta (Figura 75).



Figura 75 Declanșarea alertei

Alerta este trimisă către serverul aplicației care generează o notificare către furnizor, ce poate fi citită din aplicația mobilă (Figura 76).

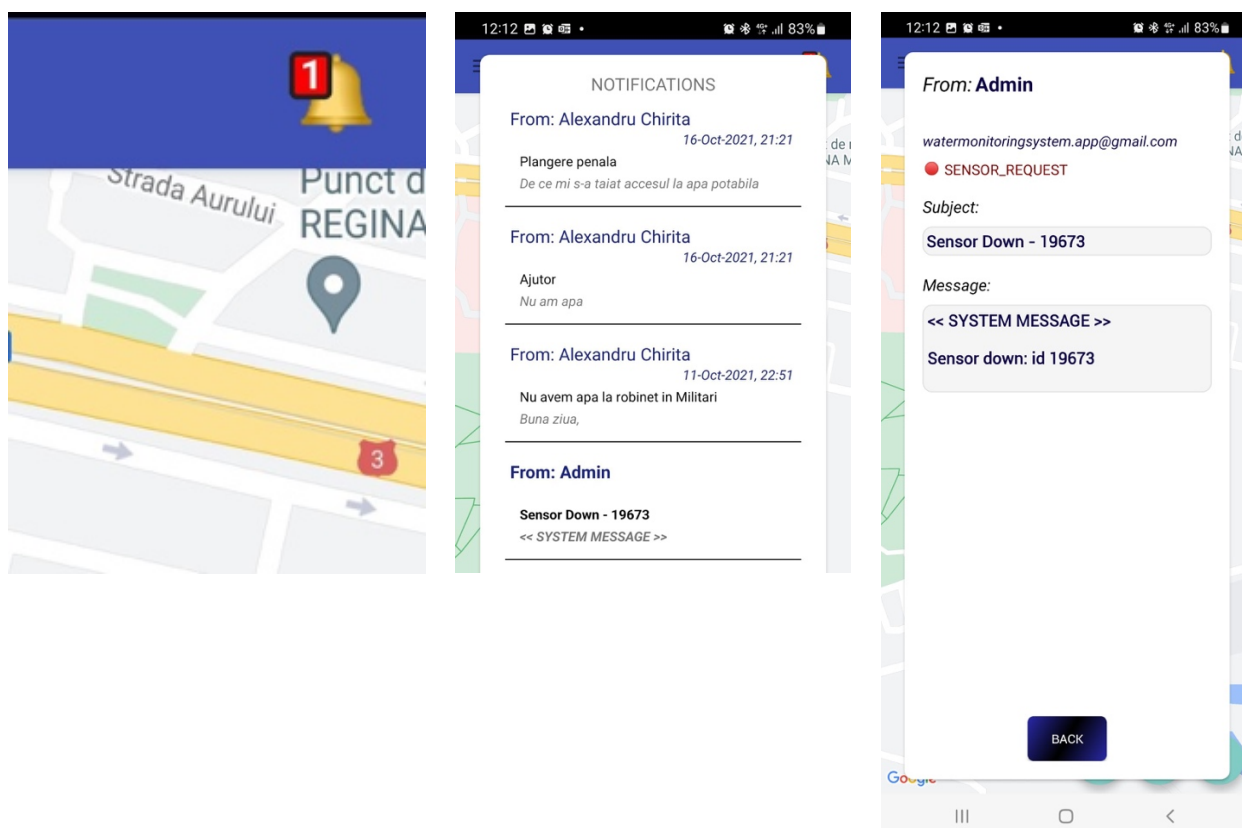


Figura 76 Notificare în aplicația WaterMonitoringSystem

Atunci când senzorul revine cu mesaje la server, se actualizează automat metrica agregată și statusul alertei, așa cum este evidențiat în Figura 77 și Figura 78.

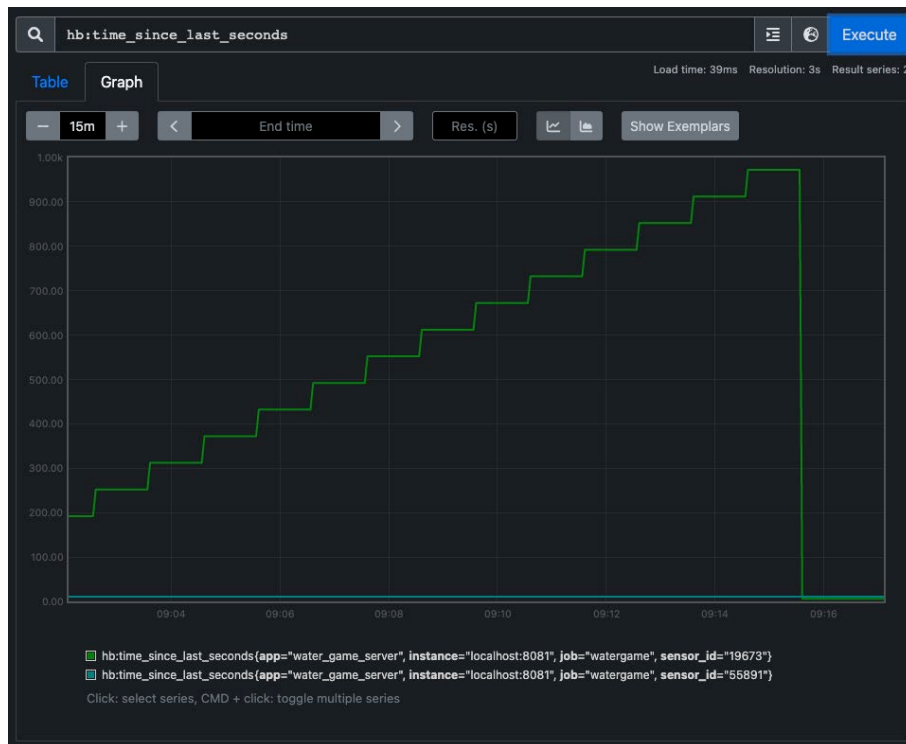


Figura 77 Actualizare metrică agregată

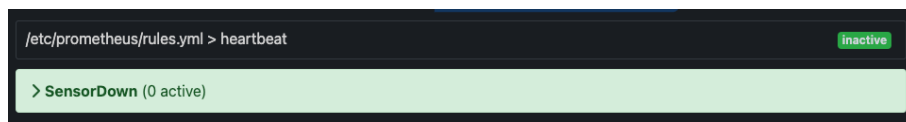


Figura 78 Resetarea alertei

Pentru a limita numărul de notificări, rezolvarea unei alerte nu generează o notificare, deoarece informația legată de funcționalitatea senzorului poate fi extrasă din interfața grafică a Prometheus.

4.1.2 Vizualizare și suport decizional în rețeaua de senzori

Metodele de procesare a datelor în contextul scenariilor prezentate în Capitolul 2.1.2 și 2.1.3 sunt implementate ca script-uri Python și integrate în aplicația de procesare a datelor descrisă în Capitolul 3.2. Rularea scenariilor de grupare a modelelor de consum este evidențiată prin diagrama de activități din Figura 79:

- Gruparea modelelor de consum se realizează prin scripturile `extract_combined.py` (extrage datele din fișiere CSV asociate fiecărui consumator într-o reprezentare matriceală), și `show_extracted.py` (rulează algoritmul de grupare și efectuează reprezentarea vizuală a modelelor de consum ca serii de timp)
- Gruparea evenimentelor de consum are la bază reprezentarea combinată a datelor, obținută prin rularea script-ului `extract_combined.py`. Prin script-ul `extract_events.py`, se parcurg seriile de timp asociate fiecărui consumator și sunt extrase evenimentele de consum reprezentate prin durată și volum. Script-ul `process_events.py` încarcă evenimentele extrase,

rulează algoritmul de grupare în funcție de durată și volum, și efectuează reprezentarea vizuală a rezultatelor grupării.

- Clasificarea evenimentelor de consum presupune două etape: antrenare și testare. În etapa de antrenare sunt rulate script-urile corespunzătoare algoritmilor de învățare profundă / deep learning (train_deep.py) și învățare automată / machine learning (train_dtree.py). Se antrenează astfel modelele și se evaluează acuratețea pe setul de date de validare. Pentru evaluarea modelelor în clasificarea efectivă a evenimentelor de consum, se utilizează script-urile eval_results_overview.py (evaluarea acurateții modelelor selectate) și eval_conf_matrix.py (realizarea matricei de confuzie).

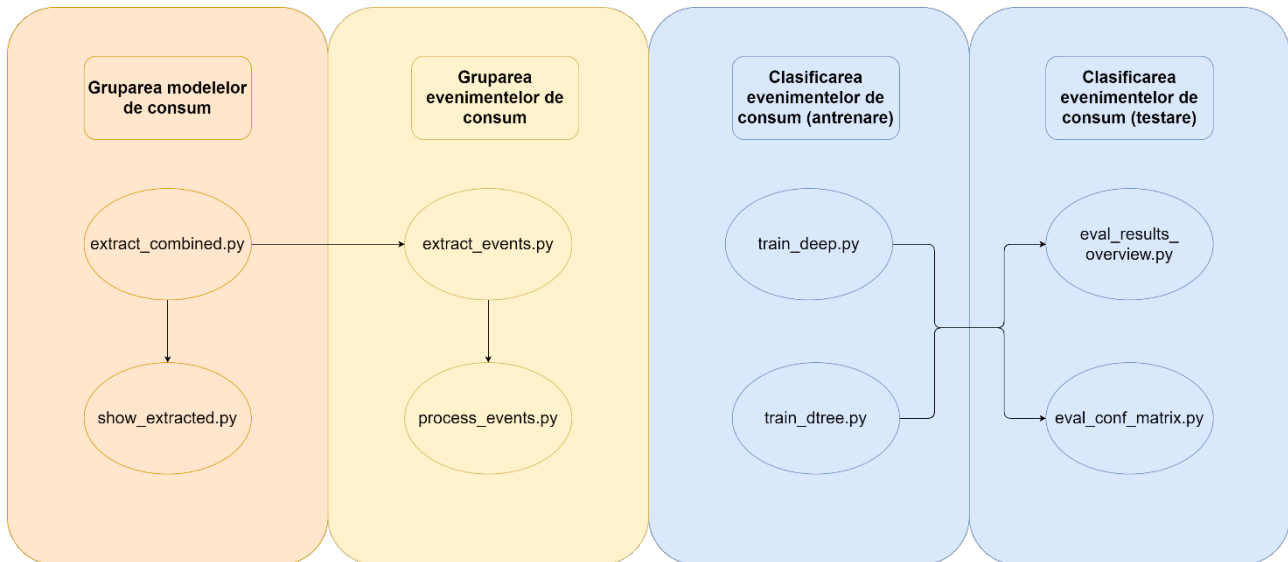


Figura 79 Diagrama de activități pentru rularea script-urilor de procesare

Script-urile de procesare și datele extrase din baza de date a sistemului de colectare sunt accesibile pe GitHub: <https://github.com/alexp25/watergame-other>

4.1.3 Raportarea evenimentelor cu suport blockchain

Aplicația Watergame este disponibilă în format web, adaptată pentru dispozitive mobile. Aceasta prezintă funcționalitățile principale din soluția de raportare a evenimentelor în infrastructura de apă în mediul urban, conform scenariilor identificate în etapa de proiectare.

Aplicația dispune de un mecanism de autentificare pe bază de email și parolă, prezentat în Figura 80. Datele de autentificare sunt înregistrate în format criptat în baza de date MySQL, folosind algoritmul BCrypt.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

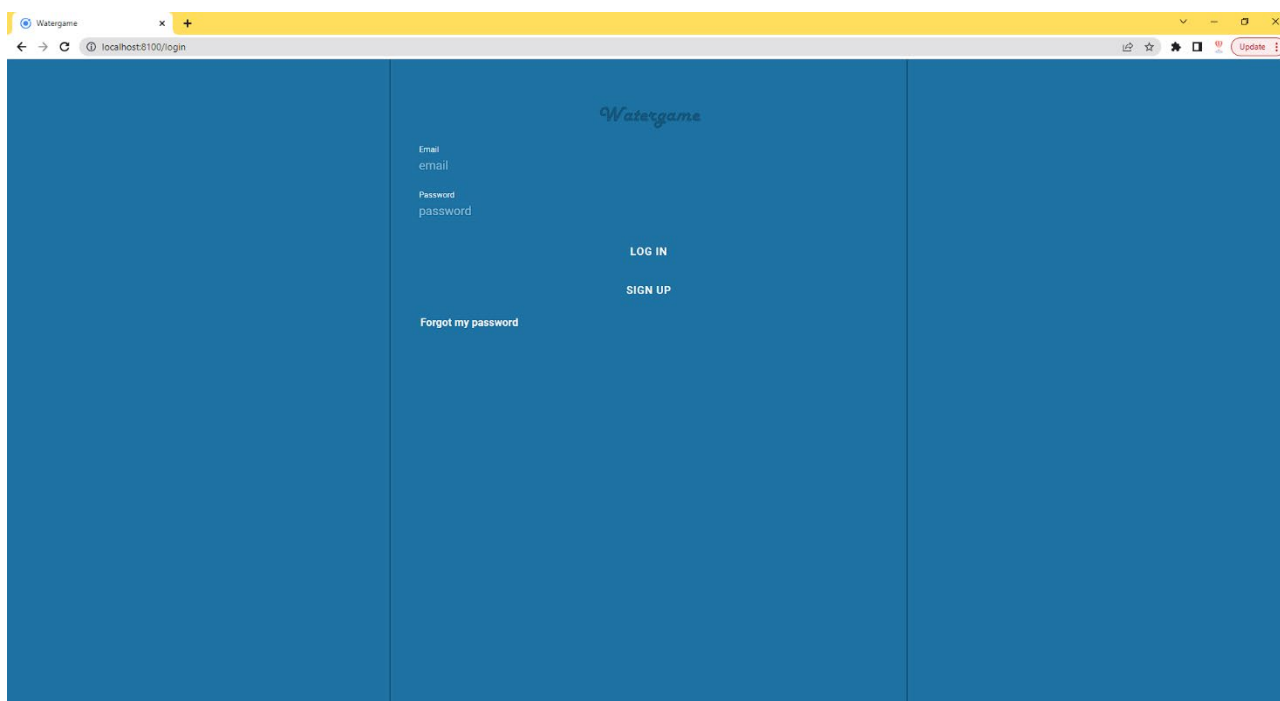


Figura 80 Autentificarea în aplicația Watergame

După autentificarea în aplicație, utilizatorul este direcționat către pagina principală, de unde poate accesa meniul principal, așa cum este prezentat în Figura 81. Meniul prezintă cele 3 secțiuni principale ale aplicației: profilul utilizatorului, zona comercială și harta interactivă.

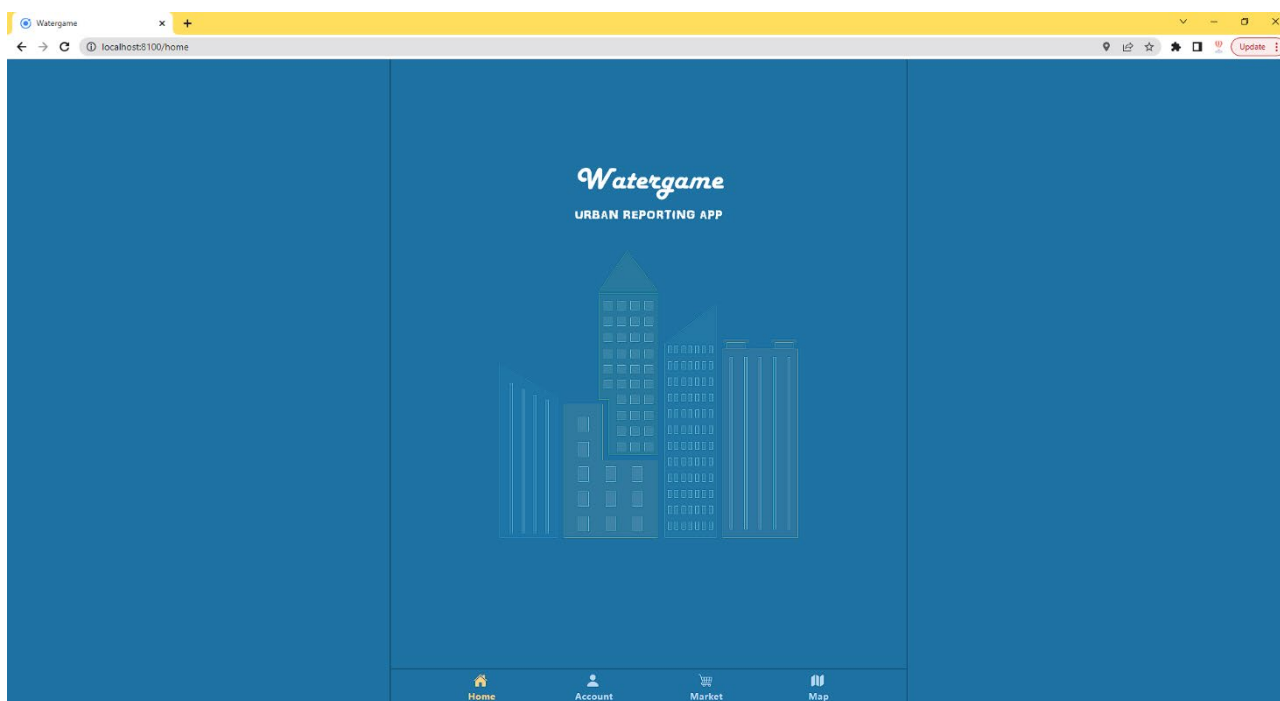


Figura 81 Pagina principală a aplicației Watergame

Profilul utilizatorului este disponibil în pagina de profil, așa cum este prezentat în Figura 82, și constă într-o reprezentare vizuală a progresului acestuia în cadrul jocului serios, conform rolurilor definite în etapa de proiectare: Finder (raportarea evenimentelor), Fixer (soluționarea raportărilor), Validator (validarea raportărilor). Pe baza acestui profil se calculează reputația utilizatorilor ce este definită pentru a calcula ponderea acțiunilor sale în procesul de recompensare interactivă.

Pagina de profil mai permite realizarea operațiunilor cu privire la contul utilizatorului în aplicație, precum modificarea numelui, a parolei, a pozei de profil, delogarea și ștergerea contului.

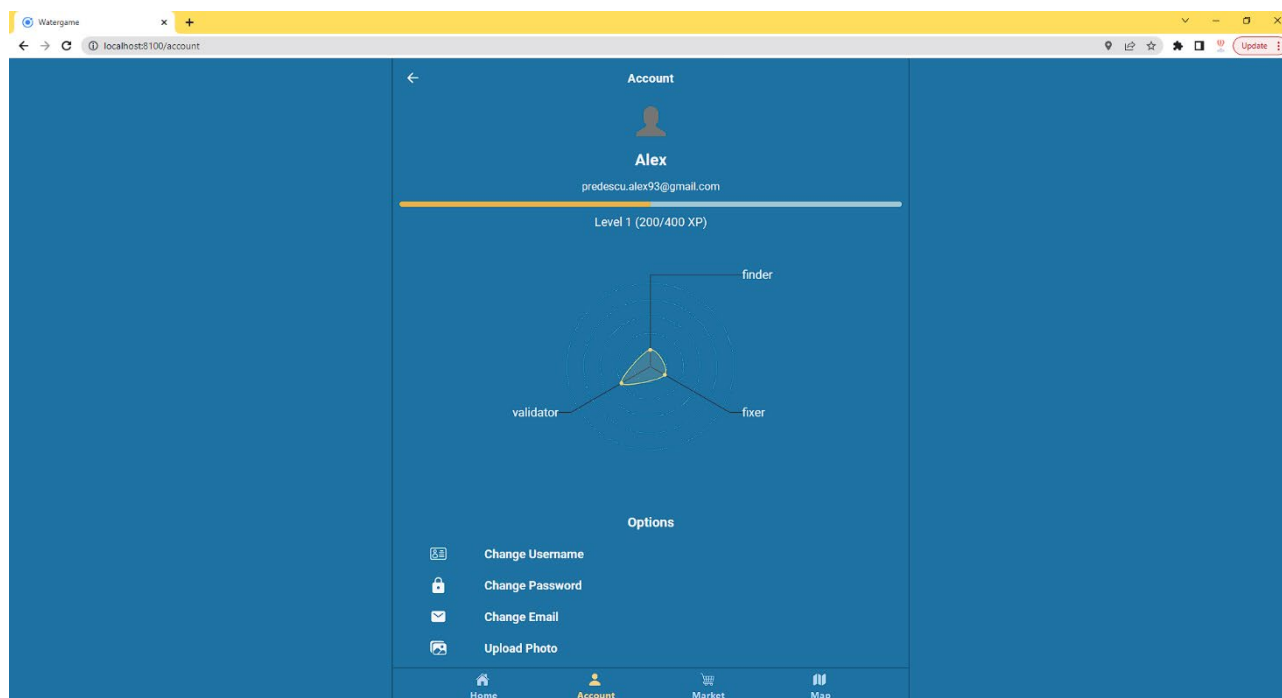


Figura 82 Profilul utilizatorului în aplicația Watergame

Zona comercială din aplicație este reprezentată prin catalogul de produse ce pot fi tranzacționate folosind moneda virtuală WGT (Watergame Token). Modelul de valorificare a recompenselor este construit pe baza sistemului Blockchain, ce constituie suportul pentru realizarea operațiunilor de validare și preschimbare a monedelor virtuale în beneficii reale, aplicabile în contextul furnizării de servicii în rețeaua de distribuție a apei. În Figura 83 este evidențiată modalitatea de conectare la Blockchain prin extensia MetaMask, ce permite realizarea de operațiuni cu moneda virtuală WGT.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

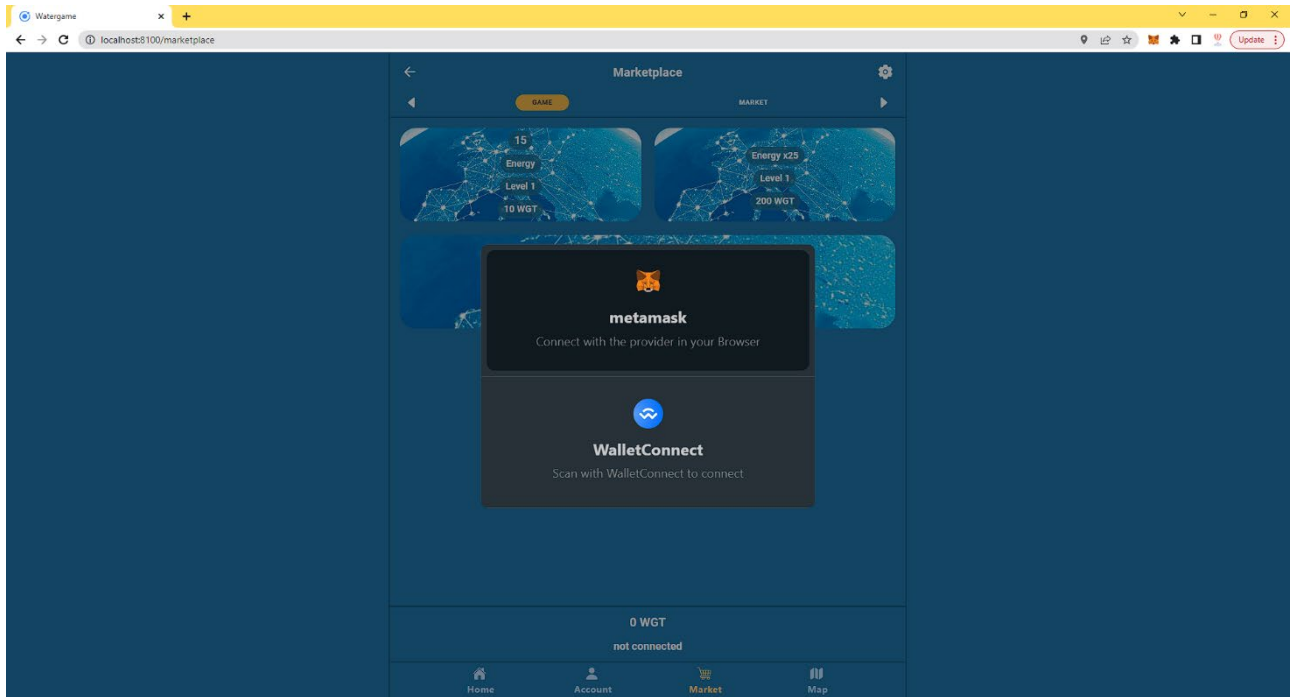


Figura 83 Magazinul virtual cu extensia de blockchain

În Figura 84, este prezentat modul de realizare a conexiunii la contul utilizatorului prin MetaMask. În acest context, se autorizează efectuarea tranzacțiilor prin aplicația Watergame.

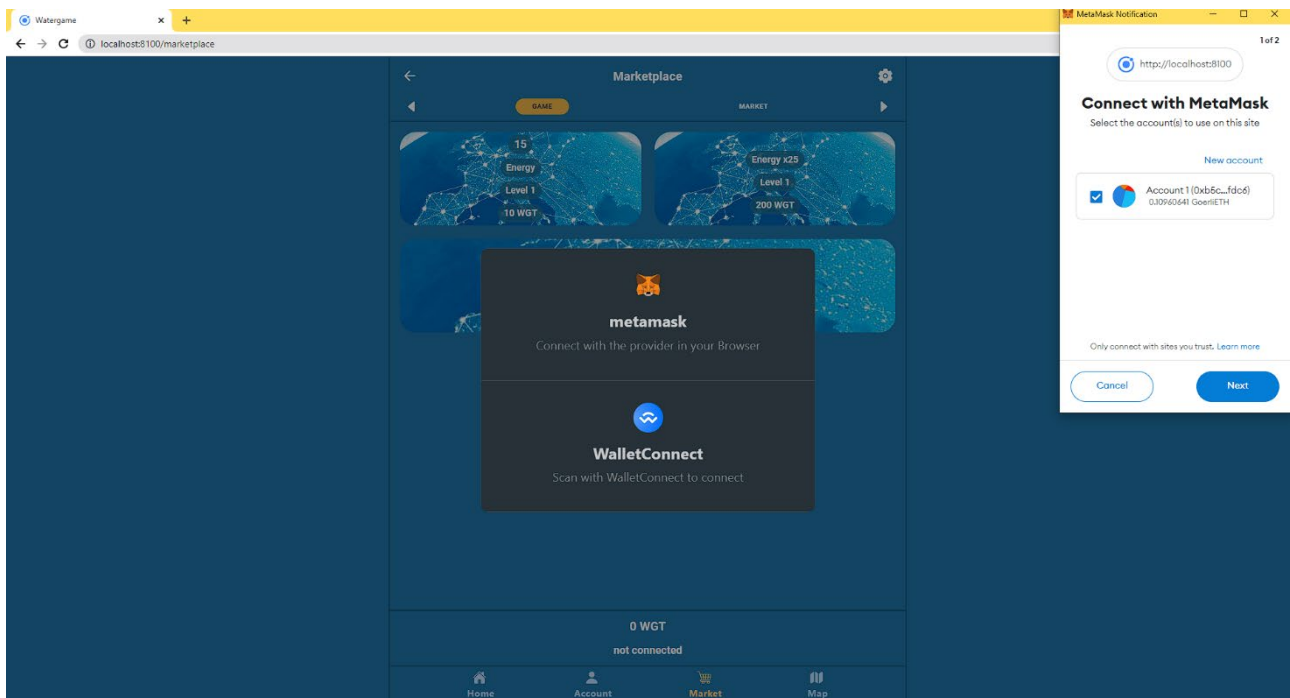


Figura 84 Autentificarea pe rețeaua blockchain folosind extensia MetaMask

În urma autentificării în rețeaua Blockchain, sunt preluate informațiile despre contul utilizatorului (identificatorul contului, balanța) și afișate în aplicație. Catalogul de produse devine disponibil, așa

cum se observă în Figura 85, și listează produsele disponibile pentru a fi cumpărate folosind moneda virtuală WGT. În exemplu, sunt listate două potențiale reduceri oferite de furnizorul de servicii.

Moneda virtuală poate fi obținută în urma activităților desfășurate în cadrul aplicației, și este asimilată recompenselor care pot fi acordate conform scenariilor propuse pentru rolurile Finder, Fixer și Validator.

Achiziționarea unui produs din catalog presupune transferul unei cantități de monedă virtuală din contul utilizatorului în contul furnizorului, în urma căruia produsul va fi înregistrat pe contul utilizatorului, în aplicația Watergame.



Figura 85 Vizualizarea produselor disponibile prin intermediul monedei virtuale

Pagina de raportare interactivă are la bază o hartă Google Maps, în care se pot vizualiza raportările de incidente din vecinătatea utilizatorului, reprezentate sub formă de puncte (markere), așa cum se pot observa în Figura 86.

Localizarea utilizatorului pe hartă este preluată de la serviciul de localizare disponibil în browser (e.g., localizare prin GPS), și este reprezentată prin marker-ul albastru, pentru utilizatorul de tip Finder (galben sau roșu pentru utilizatorii de tip Fixer sau Validator).

Utilizatorii pot adăuga raportări noi, sau pot vizualiza raportările încărcate de ceilalți utilizatori în mod interactiv. Localizarea acestora se poate ajusta prin selectarea și deplasarea punctelor pe hartă.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

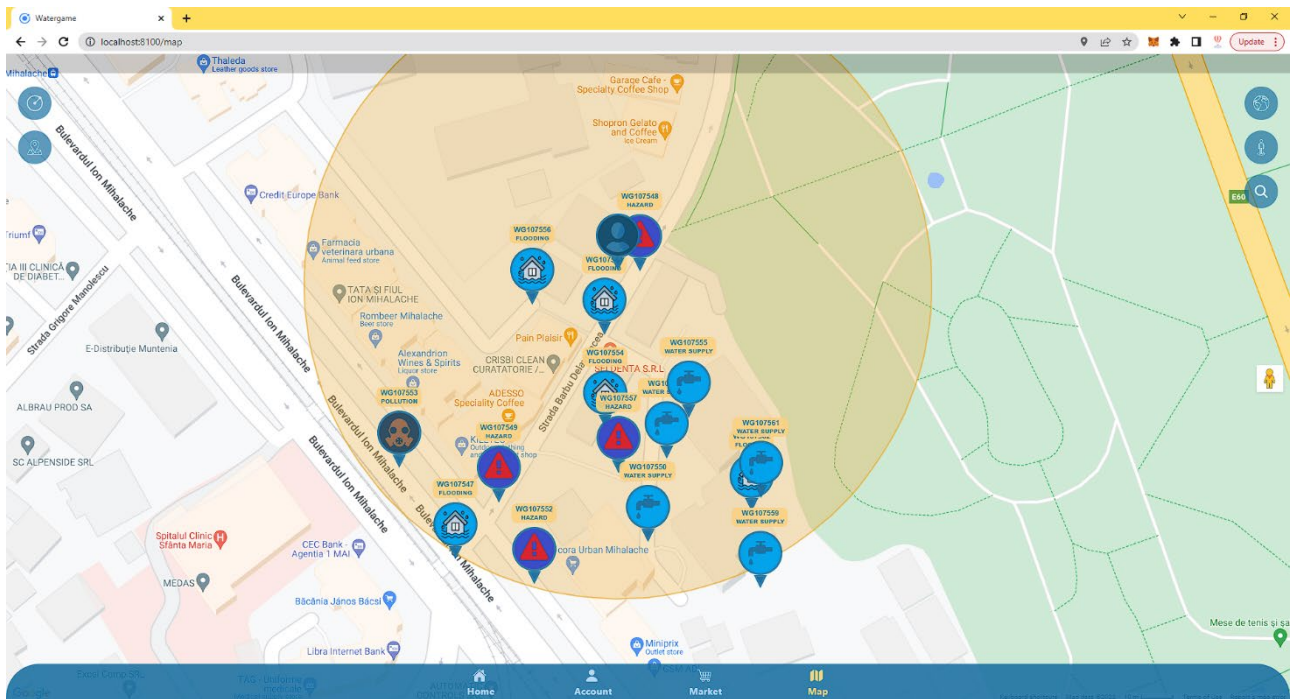


Figura 86 Vizualizarea raportărilor pe harta interactivă

În cadrul unei raportări, se poate selecta categoria din care face parte, așa cum este prezentat în Figura 86, pentru o încadrare și reprezentare vizuală corespunzătoare contextului general. Apoi, se poate introduce o descriere și încărca o poză, așa cum se poate observa în Figura 88.

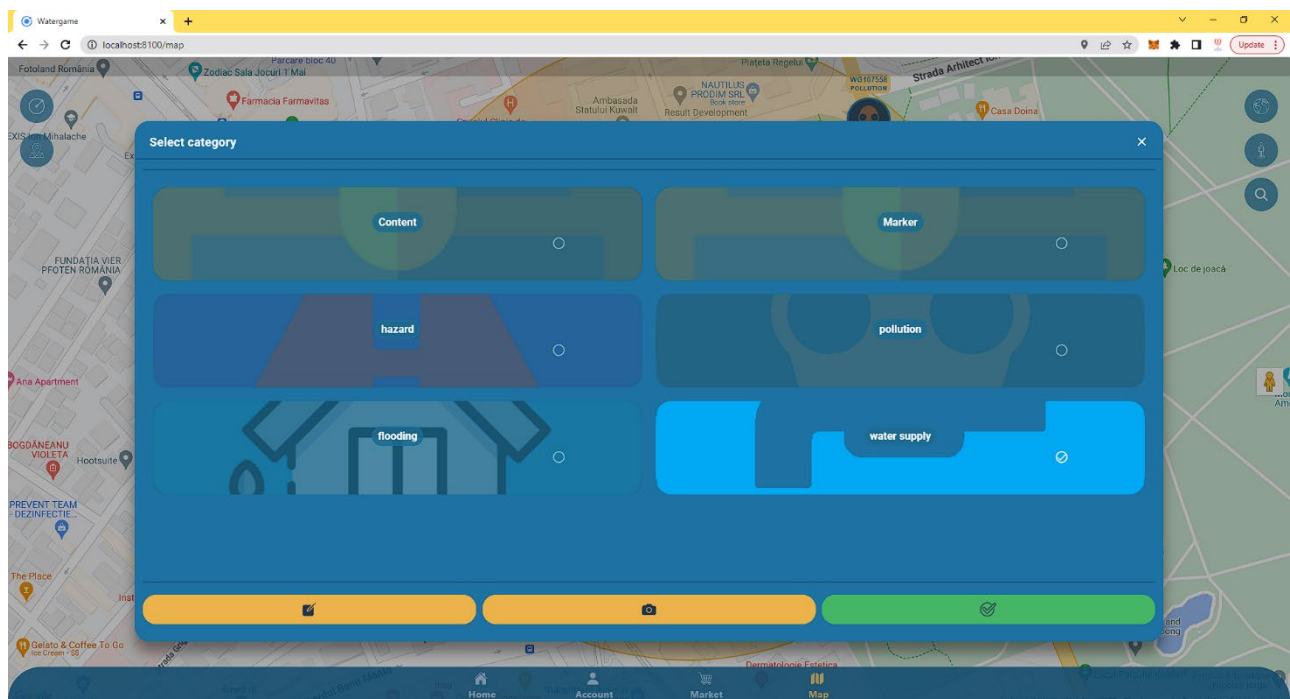


Figura 87 Adăugarea raportărilor pe hartă. Selectarea categoriei.

Fiecare raportare este înregistrată în baza de date cu un identificator unic. Informațiile despre raportări pot fi vizualizate public, iar aplicația permite validarea, soluționarea, actualizarea, sau ștergerea unei raportări.

Atunci când utilizatorii interacționează (validează sau soluționează raportările), informațiile aferente sunt înregistrate în baza de date. Un utilizator poate înregistra validarea unei raportări o singură dată, în timp ce pot exista mai multe validări din partea utilizatorilor diferiți.

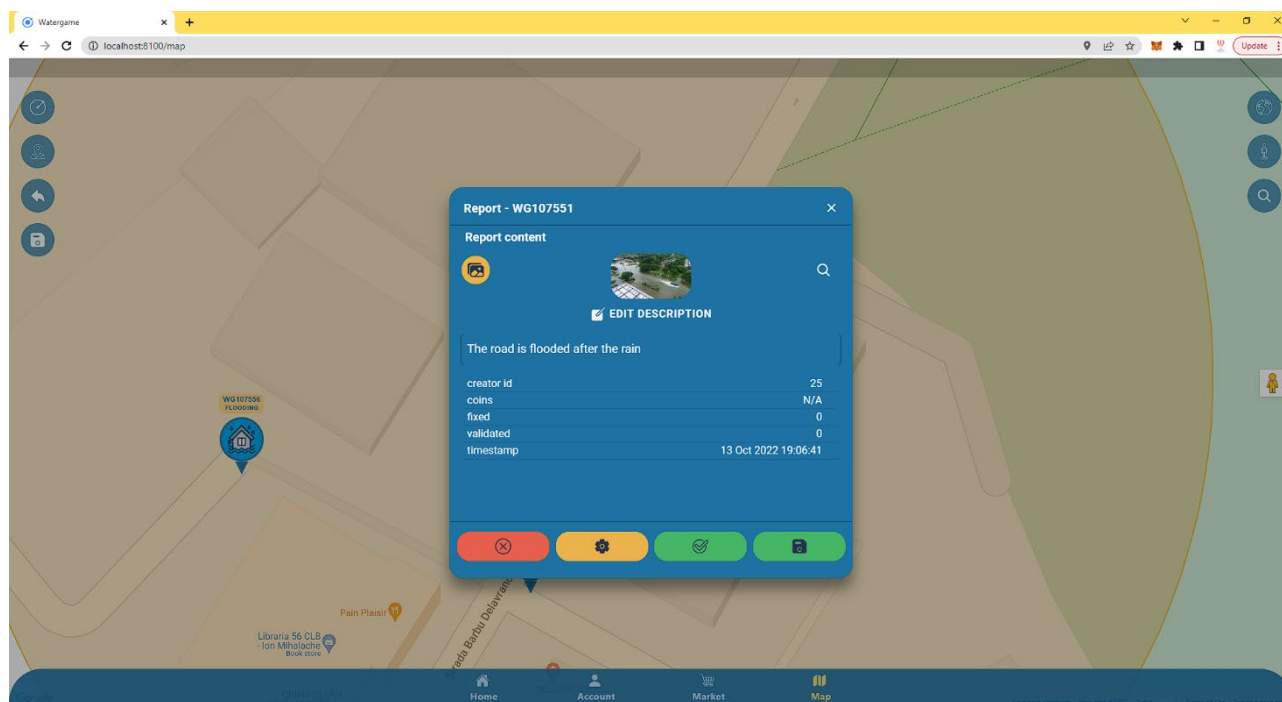


Figura 88 Adăugarea raportărilor pe hartă. Editarea conținutului.

4.2 Evaluarea rezultatelor experimentale obținute din funcționarea sistemului

În această activitate, au fost evaluate rezultatele experimentale obținute în urma evaluării sistemului în ansamblu, pe baza scenariilor experimentale.

4.2.1 Monitorizarea consumului de apă

Scenariul 1

Validarea scenariului UTCN: Monitorizarea consumului de apă în puncte multiple în cadrul unei locuințe unifamiliale în care se colectează date neprocesate direct de la punctele de monitorizare.

În cadrul locuinței s-au determinat punctele de măsurare ale consumului de apă rece la nivelul băilor, a bucătăriei și a centralei termice producătoare de apă caldă menajeră. S-au determinat un total de 8 puncte de măsurare pentru consumatorii de apă rece: 3 puncte la chiuvete, 2 puncte la cadă/duș, 2 puncte la rezervoarele de WC și un punct la admisia de apă rece a centralei termice (Figura 89).

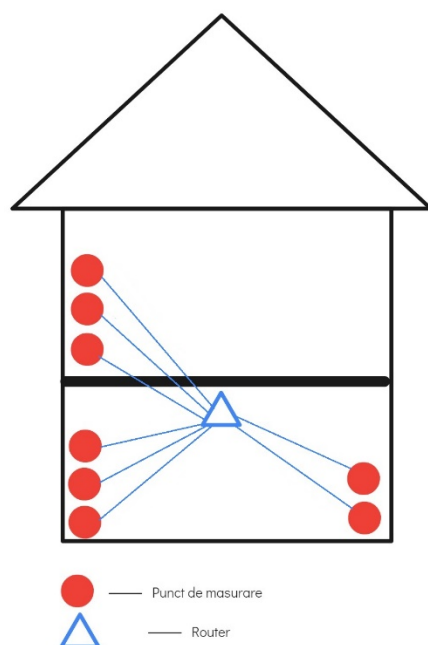


Figura 89 Colectare directă a datelor măsurate

Pentru fiecare dintre punctele de măsurare s-a montat câte un debitmetru pentru colectarea datelor de consum, debitmetru conectat la câte un nod NodeMCU. Alimentarea nodului s-a realizat prin intermediul adaptoarelor AC/DC conectate în apropierea punctelor de consum. La nivelul întregii locuințe fiecare nod a transmis datele neprocesate de debit direct la serverul de colectare a datelor aflat în Cloud (la nivelul infrastructurii UTCN). În Tabel 2 sunt prezentate datele măsurate și transmise neprelucrate și anume valoarea frecvenței furnizate de către debitmetrul YF-S201.

Tabel 2 Măsurători transmise neprelucrate

Nr. senzor	Data	Ora	Frecvența
1	30/06/2022	12:10	13.5
2	30/06/2022	12:37	15
3	30/06/2022	14:07	14.5
4	30/06/2022	15:48	15.5
5	30/06/2022	16:04	16
6	30/06/2022	17:04	17.5
7	30/06/2022	10:09	18
8	30/06/2022	18:23	17.5

Colectarea datelor s-a realizat în mod cursiv, fără întreruperi la punctele de colectare care nu erau expuse eventualelor intervenții din exterior (punctele cum sunt chiuvetele, unde prezența unui mobilier de chiuvetă ajută la protejarea debitmetrului și a nodului) și cu întreruperi la nivelul punctelor de colectare cum este cazul vaselor de wc sau chiar al dușurilor, unde debitmetrul și nodul trebuie protejate împotriva intervențiilor din exterior sau a umidității. Pentru acest lucru, nodul și debitmetrul

trebuie integrate într-un dispozitiv cu o carcasă rezistentă la umezeală și la acțiuni mecanice din exterior. În cazul testat, acțiunile mecanice externe sunt realizate în special de locatarii minori din cadrul locuinței unifamiliale.

Validarea relevanței datelor de consum pentru punctele în care s-au realizat măsurătorile, în mod cumulativ, se efectuează prin comparație cu datele de consum furnizate de către apometrul montat de către compania de apă la nivelul conductei principale de alimentare cu apă a locuinței unifamiliale. Datele de consum obținute pe scurte perioade de timp - 1, 2, 3 zile - sunt valide în raport cu datele de consum furnizate de apometrul general.

Scenariul 2

Validarea scenariului UTCN: Monitorizarea consumului de apă în puncte multiple în cadrul unei locuințe unifamiliale în care se colectează date neprocesate direct de la punctele de monitorizare, date care ulterior sunt agregate și preprocesate la nivelul unui dispozitiv local.

În cadrul aceleiași locuințe s-a adăugat un punct de agregare a datelor colectate de noduri (Figura 90). La nivelul unui dispozitiv Raspberry Pi datele neprocesate de consum, provenite de la toate cele 8 puncte de măsurare au fost colectate prin intermediul aplicației de tip server MQTT (Message Queuing Telemetry Transport) denumită Mosquitto. Mosquitto este o aplicație open-source de tip server MQTT care acționează ca un broker de mesaje în cadrul unei rețele wireless locale nesigure, unde anumite mesaje se mai pot pierde pe parcursul transmisiei.

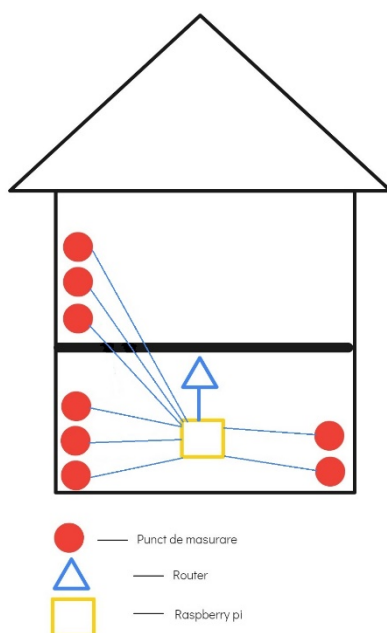


Figura 90 Dispozitiv local de prelucrare a datelor

Datele de consum calculate la nivelul dispozitivului Raspberry Pi sunt transmise ulterior în Cloud (la nivelul infrastructurii UTCN). Un exemplu este prezentat în Tabel 3.

Tabel 3 Date prelucrate după măsurare

Nr. senzor	Data	Ora	Litri/oră
1	15/07/2022	8:29	120
2	15/07/2022	8:32	102
3	15/07/2022	8:40	142
4	15/07/2022	8:36	126
5	15/07/2022	8:44	104
6	15/07/2022	8:49	108
7	15/07/2022	9:15	96
8	15/07/2022	9:33	84

Din nou, datele de consum obținute pentru teste efectuate pe perioade scurte de timp - 1, 2, 3 zile - sunt valide în raport cu datele de consum furnizate de către apometrul companiei furnizoare de apă.

4.2.2 Monitorizare și suport decizional

Scenariul A UPB: Pentru a monitoriza consumul de apă în vederea obținerii unui input consistent pentru suportul decizional, din setul de date original au fost extrase evenimentele individuale de consum. Evenimentele sunt caracterizate prin durata și volum total. Astfel, am analizat evenimentele extrase pentru fiecare tipar de colectare și am realizat gruparea în funcție de caracteristicile identificate, adică durata calculată în minute ($\times 60$ s) și volumul calculat în litri (L). În general, s-au găsit multe evenimente cu o durată egală sau mai mică de un minut (adică timpul de eșantionare al sistemului), și un volum total mai mic de 1 L, care au fost eliminate, pentru a filtra caracteristicile nedorite, de exemplu, scurgeri minore.

Clusterelor sunt reprezentate în Figura 91 și Figura 92 pentru evenimentele colectate de la senzorii de la chiuveta (apă caldă și apă rece), în timp ce evenimentele de consum de apă de toaletă și duș sunt grupate în Figura 93 și Figura 94. Clusterelor sunt prezentate în culori diferite, în timp ce centroizii lor sunt reprezentate de punctele galbene. Un număr optim de 3 grupuri a fost găsit utilizând metoda cotului și analiza siluetei clusterelor, cu excepția datelor de toaletă, care au necesitat 4 grupuri pentru o separare optimă. Cu toate acestea, am folosit 3 clusterelor pentru fiecare punct de colectare pentru a putea compara rezultatele.

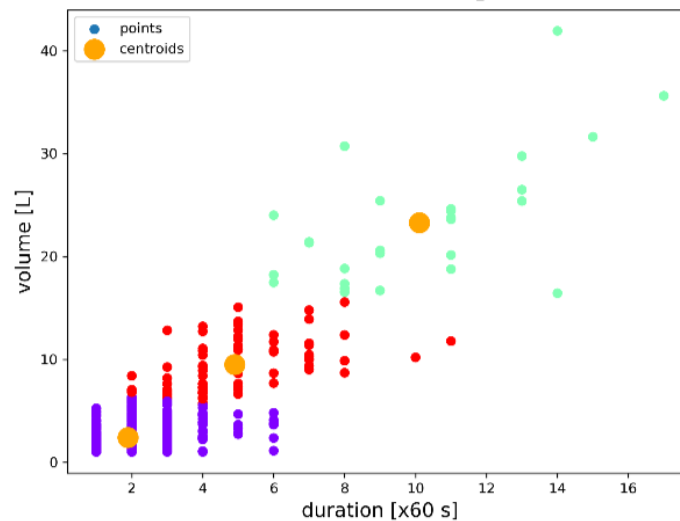


Figura 91 Gruparea evenimentelor (chiuveta apă caldă)

Astfel, se arată că există o variabilitate mai mare în consumul de apă rece de la robinet în comparație cu apa caldă, ceea ce se poate explica prin preferința față de apa caldă pentru spălarea mâinilor.

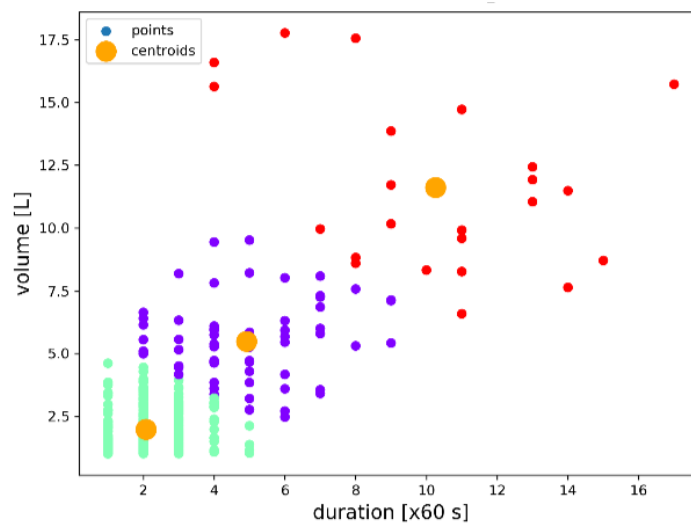


Figura 92 Gruparea evenimentelor (chiuveta apă rece)

Un rezultat foarte diferit se obține din gruparea evenimentelor de consum de apă de la toaletă, așa cum se arată în Figura 93, în sensul că durata este clar subliniată între diferite puncte de măsurare și volumul este în general aliniat cu durata. Acest comportament observat din date poate fi explicat prin volumul fix al rezervoarelor de apă de toaletă, care necesită o anumită perioadă de timp pentru a fi reumplute.

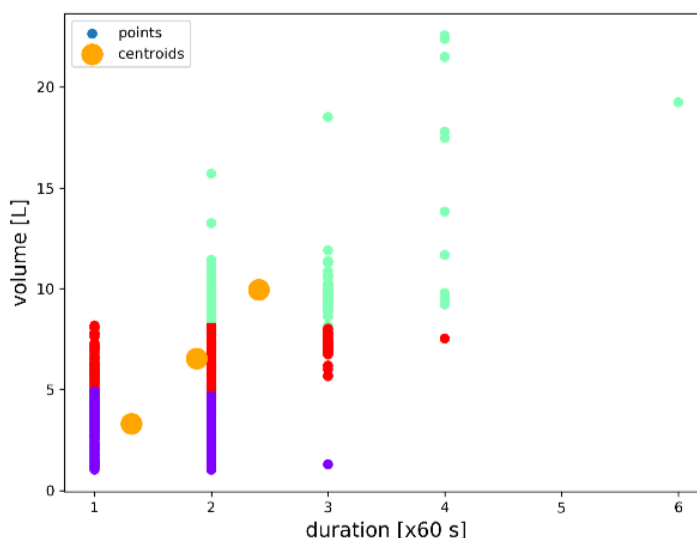


Figura 93 Gruparea evenimentelor (toaletă)

Au fost mai puține evenimente de duș în comparație cu celelalte activități înregistrate, așa cum se arată în Figura 94. Gruparea bazată pe evenimente relevă 3 tipuri de evenimente, crescând liniar atât ca durată, cât și ca volum, ceea ce explică un nivel mai mare de variabilitate în ceea ce privește apa de la duș.

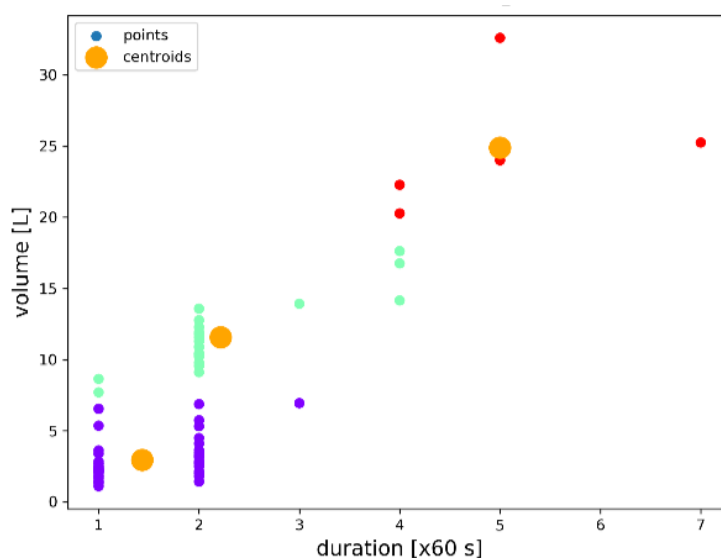


Figura 94 Gruparea evenimentelor (duș)

O monitorizare îndelungată asupra fiecărei grupări de evenimente sugerează că există, în general, un nivel mai mare de variabilitate pentru al treilea cluster, în comparație cu primele două grupări, având o durată și un volum mai mare. Prin urmare, am evaluat distribuția globală a clusterelor în termeni de dispersie. Rezultatele sunt reprezentate în Figura 95 pentru fiecare dintre aparatele măsurate și setul de date combinat. Într-adevăr, o dispersie mai mare este relevată pentru al treilea cluster în fiecare scenariu. Acest lucru sugerează că există mai multe anomalii asociate cu un consum mai mare de apă atât în ceea ce privește volumul, cât și durata, iar comportamentele de

consum mai scăzute sunt mai răspândite. În plus, graficul relevă o comparație între diferite puncte de colectare și/sau evenimente privind variabilitatea în ceea ce privește consumul. După cum era de așteptat, se constată că cea mai mică variabilitate o prezintă consumul de toaletă. Datele despre duș și apa rece arată o variabilitate similară, în timp ce apa caldă este cea mai variabilă în ceea ce privește comportamentul consumatorilor.

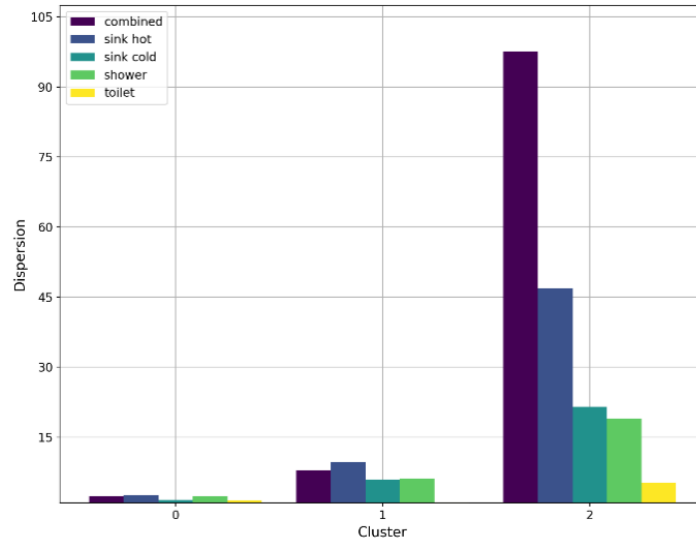


Figura 95 Distribuția clusterelor (folosind dispersia)

Rezultatele subliniază modul prin care comportamentul consumatorilor ar trebui încurajat spre reducerea consumului de apă caldă pentru a se alinia la rezultate mai durabile, deoarece există un nivel ridicat de variabilitate între diferiți consumatori.

În continuare, datele de la chiuvetă au fost combinate (apă caldă și apă rece), creând un alt scenariu pentru a oferi observații relevante cu privire la modelele de consum de apă.

Scenariul B UPB: În urma monitorizării consumului de apă, se dorește determinarea sursei unui eveniment de consum de apă, aceasta fiind caracterizată prin durata (măsurată în minute) și volumul de apă consumat (măsurat în litri). În acest scenariu, am folosit patru algoritmi de clasificare, doi algoritmi cu o abordare de învățare automată: Decision Tree și Random Forest, și doi algoritmi deep learning: adică Dense Neural Network și Recurrent Neural Network.

Pentru evaluarea calității rezultatelor, a fost efectuată o analiză bazată pe matricea de confuzie, așa cum este reprezentată în Figura 96, care arată o bună separare între punctele de consum prevăzute. Modelul Random Forest, care a produs cea mai mare acuratețe dintre modelele de clasificare propuse, a fost capabil să prezică punctul de consum corect pentru datele de toaletă și duș, în timp ce datele de la chiuvetă au fost uneori confundate cu datele de la toaletă.

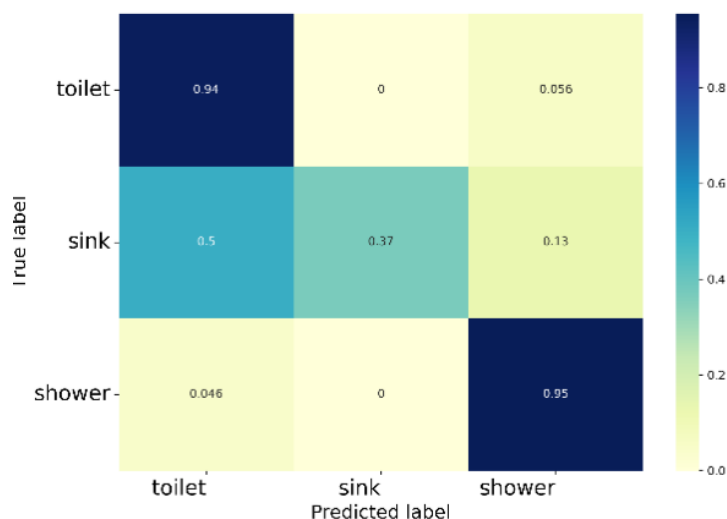


Figura 96 Matricea de confuzie

4.2.3 Raportarea evenimentelor cu suport blockchain

Rezultatul principal al dezvoltării aplicației Watergame pentru raportarea evenimentelor este obținerea unui model de laborator care integrează toate funcționalitățile propuse, și demonstrează capacitatea de integrare a scenariilor de gamificare cu tehnologia blockchain. În urma experimentării sistemului, evidența tranzacțiilor efectuate cu moneda virtuală WGT este accesibilă în rețeaua publică.

Astfel, fiecare acțiune efectuată de utilizatori prin intermediul criptomonedei, atât de către administratorul monedei, cât și de utilizatorii aplicației de crowdsensing sau prin intermediul altor metode publice de interacțiune cu blockchain va fi salvată într-un mod descentralizat în rețeaua publică de test Goerli sub forma unor tranzacții.

Fiecare tranzacție poate reprezenta una din următoarele operațiuni: creare contracte inteligente, apelare metode din contracte inteligente, cumpărare/vânzare token WGT, transfer token WGT și nu numai.

Toate aceste informații se vor putea accesa la adresa: <https://goerli.etherscan.io/>. Un exemplu de astfel de tranzacție este cea prin care a fost publicată în rețeaua de test moneda virtuală: <https://goerli.etherscan.io/address/0xf3696ccc3cced64c20a97b555cd524522c9d0943>

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

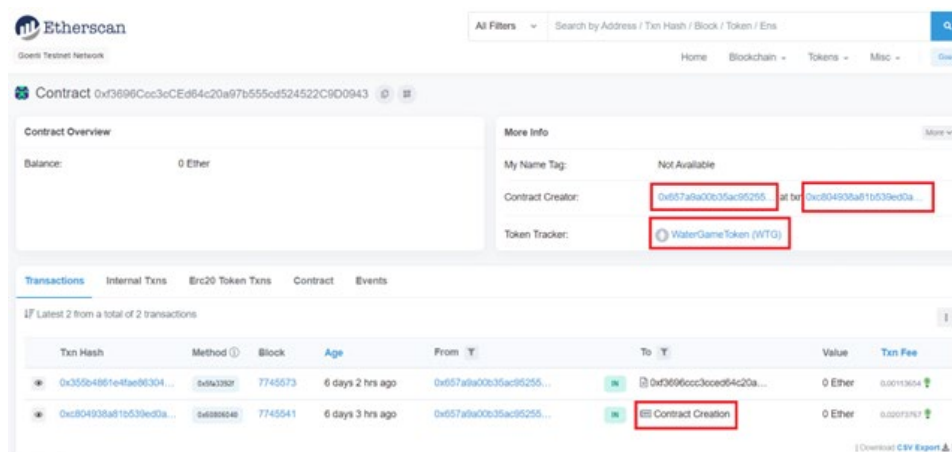


Figura 97 Pagina token-ului WGT

În Figura 97 se poate observa pagina publică a contractului monedei virtuale WGT. În cadrul acestei pagini se observă adresa contractului, adresa care a publicat moneda în rețea, adresa tranzacției corespunzătoare publicării sau numele monedei (WGT) .

Un alt exemplu de vizualizare a unei operații pe blockchain este pagina unei tranzacții (Figura 98). Aici se pot regăsi informații precum: hash-ul tranzacției, statusul, blocul în care a fost salvată, data și ora, adresa persoanei care a inițiat tranzacția și cu ce contract a interacționat sau taxele aferente tranzacției.

WATERGAME – Etapa 3. Integrarea, verificarea și validarea sistemului

[illegible]

Figura 98 Vizualizarea unei tranzacții în etherscan

5 Concluzii

În această etapă a proiectului s-a realizat integrarea componentelor realizate în etapele anterioare de către cei doi parteneri (UPB și UTCN). S-au realizat experimente plecând de la cazurile de utilizare propuse în faza a doua a proiectului, având ca scop validarea funcționării individuale a componentelor implementate, după care s-au realizat experimente pentru validarea funcționării sistemului integrat.

Conform analizei realizate, sistemul funcționează conform specificațiilor, dar în situația în care acesta este pus în producție, mai sunt necesare testări în condiții de stres ce pot apărea ca urmare a multitudinii de senzori și consumatori din sistem și / sau evenimente ce pot fi raportate de către cetățeni. De asemenea, trebuie investigat costul soluției generat de salvarea informațiilor în blockchain și disponibilitatea autorităților legale de a suporta acest cost, precum și flexibilitatea acestora în oferirea unei serii de avantaje celor care raportează sau verifică incidentele descoperite la infrastructura urbană.

În monitorizarea consumului de apă s-a constatat că toaleta necesită cel mai mare volum de apă în comparație cu celelalte puncte de colectare, iar evenimentele de consum legate de toaletă au fost mai consistente în ceea ce privește volumul și durata activității. Acest lucru se explică prin faptul că rezervoarele de toaletă au în principiu același volum și, prin urmare, necesită același timp pentru a fi reumplute. O soluție de reducere a consumului de apă în acest caz ar fi înlocuirea rezervorului cu unul mai mic.

Evenimentele cu cel mai mic consum de apă sunt reprezentate de duș care are loc doar o dată sau de două ori pe zi. Pot exista variații în funcție de sezon sau de numărul de membri ai familiei care locuiesc într-o gospodărie, ceea ce poate explica o variabilitate mai mare atunci când sunt analizate mai multe gospodării.

Mai departe, s-a demonstrat că, în ceea ce privește variabilitatea consumului de apă, chiuveta este în top. Fie că ne spălăm pe mâini, pe dinți, gătim sau spălăm vase, chiuveta este al doilea consumator ca mărime, când luăm în considerare atât consumul de apă caldă, cât și de apă rece. Acest lucru se poate explica și prin faptul că apa de la chiuvetă este lăsată să curgă chiar și atunci când ne săpunim mâinile, ne spălăm pe dinți, spălăm vasele, ceea ce permite diverse metode de optimizare, de la creșterea gradului de conștientizare cu privire la consumul inefficient de apă, la instalarea robinetelor de apă pe bază de senzori.

Rezultatele obținute prin metodele de clasificare oferă un pas necesar spre predicția diferitelor activități de consum de apă în funcție de evenimentele extrase din datele colectate de la senzori. În acest sens, un sistem de sprijin decizional poate include metodele de prelucrare a datelor propuse pentru eficientizarea consumului de apă care conduc spre sustenabilitatea rețelei de distribuție a apei, având totodată un impact favorabil asupra mediului.

Descoperirea tiparelor de consum care reprezintă un consum inadecvat de apă este un aspect important în proiectarea unui sistem de sprijinire a deciziilor, în care se pot lua decizii cu privire la educarea consumatorilor, precum și la infrastructura de apă. Astfel, monitorizarea consumului de apă și eficiența acestuia aduce multiple beneficii în ceea ce privește durabilitatea și funcționarea eficientă a infrastructurii urbane de apă.

Asigurarea durabilității rețelei de apă constă în buna funcționare a tuturor punctelor de colectare dintr-o gospodărie. Pentru eficientizarea procesului de luare a deciziilor privind conservarea apei, este important să fim conștienți de faptul că obiceiurile consumatorilor sunt diferite. Cu toate acestea, comportamentul oamenilor poate fi îmbunătățit prin creșterea gradului de conștientizare cu privire la modul de gestionare a resurselor de apă.

Monitorizarea consumului de apă reprezintă un factor esențial în procesul decizional legat de eficiența apei. Extragerea activităților consumatorilor ajută la observarea unei imagini de ansamblu asupra obiceiurilor fiecărui consumator.

Rezultatele sunt promițătoare și ajută atât consumatorul să înțeleagă care dintre activitățile zilnice generează un consum mare de apă cât și furnizorul să înțeleagă mai bine cererea de apă, oferind în același timp perspective pentru funcționarea durabilă a apei urbane.

Cercetarea realizată în cadrul proiectului a fost diseminată prin intermediul

Bibliografie

- Bell C. (2016). MySQL for the Internet of Things.
- Chatzigeorgakidis, G. (2018). Household Water Consumption (Average), HELIX, 2018. <https://data.hellenicdataservice.gr/dataset/236a0883-91b6-4f57-af9a-8d54ae7b154e> (accessed Apr. 20, 2021).
- Chen S., Tang X., Wang H., Zhao H. & Guo M. (2016). Towards Scalable and Reliable In-Memory Storage System: A Case Study with Redis.
- Denko, B., Pečnik, Š., & Fister Jr., I. (2021). A Comprehensive Comparison of Hybrid Mobile Application Development Frameworks. International Journal of Security and Privacy in Pervasive Computing (IJSPPC), 13(1), 78-90. <http://doi.org/10.4018/IJSPPC.2021010105>
- developer.nvidia.com. (2022). Jetson Nano Developer Kit. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- Enihe, Raphael & Joshua, Jimmy. (2020). HYBRID MOBILE APPLICATION DEVELOPMENT: A BETTER ALTERNATIVE TO NATIVE. 10.11216/gsj.2020.05.39825.
- espressif.com. (2022). ESP8266. <https://www.espressif.com/en/products/socs/esp8266>
- fraunhoferiosb.github.io. (2022). The SensorThings API Data Model. <https://fraunhoferiosb.github.io/FROST-Server/sensorthingsapi/requestingData/STA-Data-Model.html>
- Harris C. (2020). Array programming with NumPy. Nature, vol. 585, no. 7825, pp. 357–362.
- Hochreiter S. & Schmidhuber J. (1997). Long Short-Term Memory. Neural Comput., vol. 9, no. 8, pp. 1735–1780.
- Mishra B. & Kertesz A. (2020). The Use of MQTT in M2M and IoT Systems: A Survey. IEEE Access, vol. 8, pp. 201071–201086.
- Nair A. (2022). Beginner's Guide To K-Means Clustering. Analytics India Magazine. <https://analyticsindiamag.com/beginners-guide-to-k-means-clustering/> (accessed Jan. 31, 2022).
- Nakano, M., Takahashi, A. & Takahashi S. (2021). Generalized exponential moving average (EMA) model with particle filtering and anomaly detection. Expert Systems with Applications. Volume 73. 187-200.
- NestJS (2022). Documentation | NestJS – A progressive Node.js framework (accesat 20 Oct 2022)
- Nodemcu.readthedocs.io. (2022). NodeMCU. <https://nodemcu.readthedocs.io/en/latest/>
- optimusdigital.ro. (2022). YF-S201 Debitmeter. <https://www.optimusdigital.ro/ro/senzori-senzori-hall/3078-debitmetru-yf-s201-1-30-l-min.html>
- Parihar Y. (2019). A review of use of Nodemcu ESP8266 in IoT products. Internet of Things and Nodemcu, vol. 6, p. 1085.

- Pedregosa F. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830.
- Poyrazoglu G. (2021). Determination of Price Zones during Transition from Uniform to Zonal Electricity Market: A Case Study for Turkey. *Energies*, vol. 14.
- Raiyn J. & Toledo T. (2014). Real-Time Road Traffic Anomaly Detection. *Journal of Transportation Technologies*. 256-266.
- Rousseeuw P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, vol. 20, pp. 53–65.
- S. Chen, X. Tang, H. Wang, H. Zhao and M. Guo (2016). Towards Scalable and Reliable In-Memory Storage System: A Case Study with Redis. *IEEE Trustcom/BigDataSE/ISPA*, pp. 1660-1667, doi: 10.1109/TrustCom.2016.0255.
- Syakur M., Khotimah B., Rochman E. & Satoto B. (2018). Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster. *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 336, p. 12017.
- Tim I. (2015). Anomaly detection on social media using ARIMA models. Uppsala University. Department of Information Technology.
- Toni T., Resa S. P., Rezzy E. C., Solichatus Z., Youngjo L. & Rung C. C. (2021). Employing long short-term memory and Facebook prophet model in air temperature forecasting. *Communications in Statistics - Simulation and Computation*.
- Yu Y., Si X., Hu C. & Zhang J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.*
- Zhagparov Z., Buribayev S., Joldasbayev A. & Zhassuzak M. (2021). Building a System for Predicting the Yield of Grain Crops Based On Machine Learning Using the XGBRegressor Algorithm. *IEEE International Conference on Smart Information Systems and Technologies (SIST)*. 1-5.
- Zhou Y. & Li J. (2019). Research of Network Traffic Anomaly Detection Model Based on Multilevel Autoregression. *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 380-384.