

REZUMATUL ACTIVITĂȚILOR DESFĂȘURATE ÎN CADRUL PROIECTULUI BIGDATA4GRID ÎN CADRUL ETAPELOR 2020-2022

Obiectivul principal al proiectului a fost propunerea și implementarea unui prototip informatic modular și scalabil, compus din două module funcționale prin care să se eficientizeze managementul operării rețelelor electrice astfel încât să permită aplicarea modelelor de tip Demand Side Management (DSM). Cele două module funcționale sunt:

- **Nivelul M1 – conține modele demonstrative pentru DSM:** i) modele pentru creșterea gradului de conștientizare și implicare a consumatorilor pentru un consum durabil; ii) modele pentru monitorizarea și controlul dispozitivelor electrice; iii) modele de optimizare a consumului la nivel individual; iv) simulator virtual pentru consumatori și prosumatori.
- **Nivelul M2 – conține modele demonstrative pentru funcționarea eficientă a rețelei:** i) modele pentru determinarea profilurilor de consum; ii) modelele de predicție a consumului de energie electrică și a consumului propriu tehnologic; iii) decontarea la 15 minute; iv) modele de tarifare avansate; v) indicatori de performanță (Key Performance Indicators - KPI).

Modelele și funcționalitățile prototipului informatic au fost realizate prin parcurgerea următoarelor **etape și activități**:

Etapa 1 (2020) - Propunerea modelelor de organizare a datelor colectate de la prosumatori și arhitectura prototipului informatic

Act 1.1 - Propunerea modelelor de organizare a datelor colectate de la prosumatori și arhitectura prototipului informatic (WP2 - Big Data Framework - proposal)

Pentru propunerea modelelor de organizare a datelor am analizat principalele surse de date și am identificat o serie de algoritmi și metode necesare pentru prelucrarea acestora astfel încât să putem implementa modelele demonstrative pentru Demand Side Management (DSM) și pentru managementul consumului la nivel agregat (agregator și DSO). Aceștia sunt prezentați în raportul științific al etapei 2020. Pentru managementul datelor am propus o arhitectură pe trei niveluri de procesare și agregare a datelor:

- I. Nivelul locului de consum (*Edge layer*) - corespunde fiecărui consumator de energie electrică și conține un dispozitiv inteligent, cum ar fi Raspberry Pi, care gestionează dispozitivele locale, monitorizează funcționarea acestora dar și a contoarelor inteligente sau controlerele locale/invertoarele în cazul prosumerilor. Se pot implementa o serie de algoritmi locali pentru optimizarea consumului pentru ziua următoare prin care consumatorul programează dispozitivele controlabile pe baza tarifelor orare primite de la agregator; monitorizarea și controlul dispozitivelor care citește consumul local la intervale de 1-15 minute și îl stochează într-o bază de date locală (Edge DB) și apoi le transmite agregate către Fog DB.
- II. Nivelul stației de transformare / nod de consum (*Fog layer*) - corespunde unui nod de consum care conectează mai mulți consumatori dintr-o zonă locală și colectează datele contoarelor inteligente și al dispozitivelor într-o bază de date locală (Fog DB). Pe fiecare nod de consum se pot implementa o serie de algoritmi pentru prognozarea consumului pentru ziua următoare; controlul dispozitivelor flexibile în timp real (Direct Load Control – DLC) care reduce consumul total în caz de deficit prin întreruperea pentru un interval de 15-30 minute a unor dispozitive controlabile;

optimizarea consumului la nivelul comunității în funcție de disponibilitatea surselor locale de generare și a restricțiilor de rețea.

- III. Nivelul central (*Data center sau Cloud layer*) - conține principalul centru de date (Central DB) și aplicațiile necesare agregatorului de energie electrică unde sunt stabilite tarifele orare și restricții legate de consum pentru fiecare nod (load point). La acest nivel se pot implementa o serie de algoritmi pentru stabilirea strategiilor DSM și a schemelor de tarifare, estimarea consumului la nivel agregat, determinarea unor profiluri de consum.

Principalul avantaj al acestei arhitecturi este că permite stocarea și prelucrarea datelor cât mai aproape de sursă, reducând latența în comunicații și evitând probleme de securitate cu privire la datele sensibile ale consumatorilor. Dispozitivele inteligente sunt controlate local de echipamente hardware de tip Raspberry Pi, iar consumul și caracteristicile acestora precum și comportamentul de consum al consumatorului nu traversează nivelul stației.

Etapa 2 (2021) - Dezvoltarea modelelor pentru organizarea datelor și a modelelor demonstrative pentru DSM. Propunerea modelelor pentru operarea eficientă a rețelelor

Act 2.1 - Dezvoltarea modelelor de organizare a datelor (WP2 - Big Data Framework - development)

Obiectivul principal a fost de a dezvolta un cadru (*framework*) Big Data bazat pe o platformă de comunicare interoperabilă care integrează toate sursele de date pentru a furniza datele necesare cu calitatea definită (în timp real, cu amprentă de timp și cu informații semantice). Datele colectate sunt mai întâi procesate la nivelul locului de consum, stocate în BD Edge și apoi transformate și transmise către nodul Fog și stocate în BD Fog. Platforma de comunicație constă din următoarele elemente: i) scripturi și pachete Python pentru conectarea surselor de date, monitorizarea funcționării acestora și pentru prelucrarea datelor; ii) noduri Node-RED pentru conectarea dispozitivelor inteligente, realizarea comunicației dintre aplicațiile de pe nivelurile fog și edge prin protocoalele MQTT și HTTP, apelul funcțiilor Python și prelucrări locale de mesaje; iii) elemente auxiliare precum noduri pentru realizarea unei interfețe de monitorizare locală destinată consumatorilor. Platforma de comunicație rulează pe nivelul *Edge* pe un dispozitiv de tip Raspberry Pi, însă poate fi cu ușurință implementată și pe un laptop sau PC cu resurse modeste (de generație veche).

Framework-ul Big Data a fost dezvoltat pe cele trei niveluri ale arhitecturii propuse, astfel:

Nivelul Edge – organizează datele colectate din dispozitivele inteligente, invertoare, aplicațiile de monitorizare ale consumatorilor și senzori locali. Aplicațiile dezvoltate la acest nivel impun următoarele cerințe de prelucrare a datelor: i) atomicitate, consistență, independență, durabilitate (ACID) pentru tranzacțiile realizate; ii) securitate și integritate în managementul datelor; iii) resurse de calcul minimale pentru extragerea și prelucrarea datelor. Din aceste considerente modelul de organizare al datelor la acest nivel poate fi un model relațional gestionat de un SGBD relațional care implementează caracteristicile ACID și rulează cu resurse minimale, dar oferă un management optimizat și securizat al datelor (MySQL, MariaDB, PostgreSQL). Majoritatea SGBD-urilor permit și gestiunea datelor în format *json*, astfel schema bazei de date fiind mult mai flexibilă decât variantele tradiționale.

Nivelul Fog – organizează datele agregate primite de la nodurile Edge în baze de date NoSQL dacă volumul acestora este semnificativ (peste 10000 de locuri de consum conectate într-o stație) sau în baze de date relaționale dacă volumul datelor este redus. Pentru generalizare am optat pentru organizarea datelor la acest nivel în baze de date NoSQL utilizând o

reprezentare mixtă de tip document *json* indexat (*key-value with type document*). Nivelurile Fog vor transmite către nivelul Data Center valorile înregistrate de contoarele și dispozitivele inteligente la intervale cuprinse între 15 – 60 minute pentru a fi utilizate în sistemele de tarification și facturare, dar și pentru analize avansate realizate de către agregator.

Nivelul Data Center (Cloud) – organizează datele provenite de la nodurile Fog în baze de date NoSQL sub forma de colecții de documente agregate.

Modul de organizare a datelor, colecțiile de date și fluxul de transformare și reducere a volumului colecțiilor de date a fost prezentat în raportul științific al etapei 2021.

Act 2.2 - Dezvoltarea modelelor pentru DSM (WP3 - Demonstrative Models for DSM)

2.2.1. Măsurile pentru creșterea gradului de conștientizare și implicare a consumatorilor pentru un consum durabil

Proiectarea stimulilor DSM este un pas important, deoarece motivează consumatorii să se comporte în direcția unui consum durabil. Pentru proiectarea setului de stimuli DSM, am analizat rezultatele unor experimente și studii pilot care explorează efectele diferitelor metode de stimulare a consumatorilor în vederea schimbării comportamentului de consum. Setul de stimuli a fost testat și validat folosind chestionare pre- și post-experiment, măsurând impactul și potențialul de a stimula consumatorii.

Setul de stimuli se bazează pe cinci strategii principale de management al consumului (DSM) prin care clădirile de consumatori pot fi eficiente și pot interacționa cu sistemul energetic: măsuri de îmbunătățire a eficienței energetice, reducerea consumului (load shedding), mutarea consumului (load shifting), controlul consumului (modulating, Automated Demand Response - ADR, Direct Load Control - DLC), și generarea (generation) și stocarea energiei. Pentru a implementa și a cuantifica stimulii DSM au fost utilizați algoritmi de optimizare și control al consumului de energie electrică. Acești algoritmi sunt însoțiți de diferite tipuri de contracte (DAL Contract Deferrable Activation Loads Contract, LC Contract Load Curtailment Contract, RI Contract Reduction for Incentive Contract, ToU Contract Time of Use Contract).

Analizând chestionarele derulate înainte și după experimentul instalării contorizării inteligente din Irlanda, utilizând analiza factorială de confirmare, putem concluziona că măsurile pentru creșterea gradului de conștientizare și implicare a consumatorilor pentru un consum durabil sunt fezabile și pot fi implementate pe scară largă. Recomandăm realizarea studiilor pilot care să includă chestionare pentru un eșantion reprezentativ de consumatori. De asemenea, recomandăm realizarea unor analize care să confirme ipotezele considerate cu privire la fezabilitatea măsurilor de creștere a gradului de conștientizare.

2.2.2. Monitorizarea și controlul dispozitivelor electrice.

Algoritmul pentru monitorizarea și controlul dispozitivelor electrice (*Load Monitor & Control – LMC*) presupune următoarele două componente:

- Componenta de monitorizare a consumului citește valorile înregistrate de dispozitiv prin intermediul prizelor inteligente (putere, curent, tensiune, stare), salvează aceste citiri în baza de date locală a nodului Edge și trimite datele către BD Fog. În plus, verifică periodic (la 30-60 secunde) lista de dispozitive programate să funcționeze și trimite către componenta de control comanda corespunzătoare (pornire/oprire).
- Componenta de control comunică cu nodul Fog prin protocolul MQTT și primește lista dispozitivelor de controlat de la modelul DLC. În plus, primește de la componenta de monitorizare comenzile de pornire/oprire a dispozitivelor programate să funcționeze.

În cazul dispozitivelor de încălzire/răcire se implementează o subrutină de verificare a temperaturii din locuință pentru a porni/opri aparatele de tipul AC, radiatoare, plasmă termice doar cazul depășirii unui anumit prag de conform setat de utilizator. Prin urmare, un senzor de temperatură este conectat la LMC și asigură monitorizarea temperaturii camerei. În cazul în care dispozitivele electrice de încălzire/răcire sunt oprite de către DLC, ele nu vor fi pornite până când diferența dintre temperatura camerei și temperatura preferată a utilizatorului nu este mai mare de 1-2 grade.

Algoritmul LMC este detaliat în raportul științific al etapei 2021. Versiunea actuală a algoritmului LMC este implementată în Node Red pe Raspberry Pi folosind prize inteligente TP-Link pentru a monitoriza și controla aparatele.

2.2.3. Modele de optimizare a consumului la nivel individual

Modelele de optimizare propuse la nivel individual sunt integrate cu modelele de optimizare și DLC realizate la nivelul furnizorilor/agregatorilor. Modul de integrare a acestor modele este sintetizată în figura următoare.

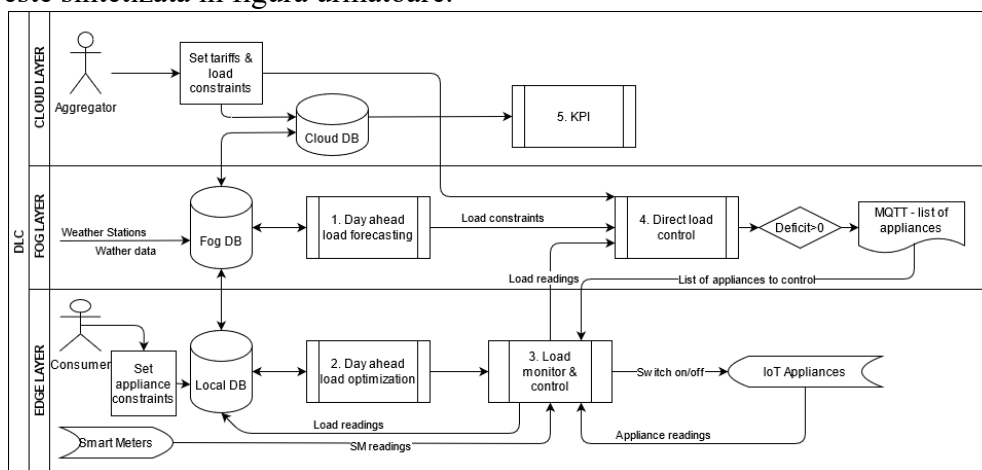


Figura 1 – Integrarea modelelor de optimizare și DLC

Modelul de optimizare la nivelul consumatorului - Day-Ahead Load Optimization (DALO)

Consumatorii de energie electrică își pot optimiza consumul individual pentru a minimiza costurile cu energia electrică. În primul rând, consumatorul trebuie să precizeze pentru ziua următoare următorii parametri pentru fiecare dispozitiv controlabil: numărul de cicluri de funcționare și constrângerile privind orele de pornire/oprire. La acestea se adaugă tarifele orare stabilite de agregator/furnizor, capacitatea și timpul necesar de funcționare pentru fiecare dispozitiv. Algoritmul DALO determină orarul optim de funcționare ale dispozitivelor minimizând astfel costurile zilnice cu energia electrică. Algoritmul DALO este detaliat în raportul științific al etapei 2021.

Modelul de control al flexibilităților - Direct Load Control (DLC)

Algoritmul DLC rulează pe nodurile Fog și verifică la fiecare interval de timp t (15 minute sau 30 minute în funcție de necesitățile agregatorului) dacă consumul total pe nodul de rețea depășește maximumul estimat sau valoarea maximă impusă de operatorul de rețea (restricțiile de rețea) și calculează deficitul. Apoi, determină lista dispozitivelor întreruptibile (I) sau de stocare (B) care funcționează în prezent și care pot fi întrerupte timp de 15 minute, ținând cont de următoarele reguli:

- 1) Numărul maxim de întreruperi per dispozitiv (NoI_j) este 4, adică maxim 1 oră pe zi pentru fiecare dispozitiv;

- 2) Intervalul minim între 2 întreruperi consecutive (Δt_{ij}) este 120 minute;
- 3) Dispozitivele care vor fi întrerupte sunt selectate în ordinea inversă a întreruperilor anterioare.

Aceste reguli sunt implementate în algoritmul DLC detaliat în raportul științific al etapei 2021. Regulile DLC pot fi adaptate, iar numărul maxim de întreruperi sau timpul dintre întreruperi pot fi considerate ca parametri, conform acordurilor dintre agregator și consumatori privind participarea în programele de flexibilități. DLC este implementat în Python și rulează în paralel pe nodurile Fog la fiecare 15 minute pentru a determina lista dispozitivelor care vor fi întrerupte pe fiecare nod de rețea.

2.2.4. Dezvoltarea unui simulator virtual pentru consumatori și prosumatori

Algoritmii propuși pentru nivelul M1 au fost integrați într-un simulator virtual pentru a motiva consumatorii să monitorizeze, să controleze și să optimizeze consumul de energie electrică și să crească gradul de acceptare și implicare într-un consum durabil și sustenabil.

Prin simularea diferitelor scenarii de consum și vizualizarea rezultatelor optimizării, consumatorii pot înțelege și evalua beneficiile optimizării consumului, a înlocuirii electrocasnicelor energofage și avantajele consumului din surse proprii de energie în cazul prosumatorilor.

Simulatorul virtual a fost dezvoltat astfel încât să ruleze pe un dispozitiv de tip Raspberry PI cu cerințe minime de resurse și costuri de achiziție. Tehnologiile utilizate sunt: MySQL pentru managementul datelor colectate din inverter, smart meter, API web și dispozitive inteligente/prize; Python pentru prelucrarea datelor și implementarea algoritmilor; Flask ca micro-framework pentru dezvoltarea aplicației web și Bootstrap 5.0 pentru realizarea interfețelor web. Pe baza simulatorului, în etapa 2022 a fost dezvoltată aplicația *My SASM dashboard* prezentată în secțiunea 3.

[Etapa 3 \(2022\) - Dezvoltarea modelelor pentru operarea eficientă a rețelelor electrice de distribuție. Implementarea prototipului informatic BigData4Grid în mediul de testare](#)

[Act 3.1 - Dezvoltarea modelelor pentru operarea eficientă a rețelelor electrice de distribuție \(WP4 - Demonstrative Models for Efficient Grid Operation\)](#)

3.1.1. Determinarea profilurilor de consum

Pentru determinarea profilurilor de consum în vederea aplicării pe scară largă a programelor de DR, am luat în considerare percepțiile și așteptările consumatorilor utilizând analiza factorială de confirmare (CFA) de ordinul întâi și ierarhică, cum ar fi CFA de ordinul doi și bifactorială. Simulările au fost realizate pe date din chestionarul preliminar (pre-trial) oferit de Comisia Irlandeză pentru Reglementarea Energiei. Cel mai bun model CFA bifactorial este construit oferind rezultate excelente de performanță: Comparative Fit Index (CFI) = 0,95 în medie, Standardize Root Mean Square Residual (SRMR) = 0,03, Root Mean Square Error of Aproximation (RMEAS) = 0,02. Modelul CFA bifactorial oferă cele mai interesante rezultate și confirmă ipoteza că cei cinci factori latenți au un impact semnificativ asupra gradului de conștientizare a consumatorilor. Analizele sunt detaliate în raportul etapei 2022.

3.1.2. Modelul de predicție a consumului de energie electrică și a consumului propriu tehnologic

Am dezvoltat modelul *Day-Ahead Load Forecasting (DALF)* prin care se estimează consumul agregat la nivelul fiecărui nod de rețea pentru următoarele 24 de ore utilizând modele de învățare automată (ML) precum rețele neuronale artificiale (ANN), rețele neuronale de tip deep learning (DNN) sau modele de tip ansamblu. DALF utilizează mai multe modele de ML

pentru antrenament și validare încrucișată: *i*) modelul Multi Layer Perceptron (MLP) care implementează ANN; *ii*) Gradient Boosting Regressor (GBR); *iii*) eXtreme Gradient Boosting (XGB); *iv*) Random Forest Regressor (RF). Modelul selectează automat algoritmul cu cea mai bună performanță la pasul de validare încrucișată, evaluat prin coeficientul de determinare R^2 , eroarea medie absolută procentuală (MAPE) și eroarea medie pătratică (RMSE). Modelul selectat furnizează valorile estimate ale consumului agregat ($FC_t^{(l)}$) pe fiecare nod de rețea l (corespondent unui nod *Fog*) și apoi le încarcă agregat pe nodul *Cloud* corespunzător distribuitorului/agregatorului de energie electrică.

Variabilele de intrare ($X_t^{(l)}$) ale algoritmului DALF sunt formate din variabile care modelează consumul istoric agregat ($CX_t^{(l)}$) la care se adaugă și valorile istorice ale parametrilor meteorologici ($WX_t^{(l)}$) extrași cu ajutorul unor funcții API de pe site-urile de profil (openweather și accuweather). Valorile de consum sunt extrase din BD Fog pe ultimele 30 de zile la nivelul fiecărui nod de consum și reprezintă consumul mediu, total, max, min și abaterea standard pentru fiecare interval t (15 sau 30 minute). De asemenea, tipul zilei - *DayCode* (zi lucrătoare/zi de weekend) este considerată ca și variabilă de intrare.

$$FC_t^{(l)} := DALF(X_t^{(l)}) := DALF(WX_t^{(l)}, CX_t^{(l)}, t, DayCode) \quad (1)$$

Algoritmul DALF este detaliat în raportul științific al etapei 2022 și poate fi utilizat și pentru estimarea consumului propriu tehnologic (CPT), modificând variabila de intrare $X_t^{(l)} = CPT_t^{(l)}$.

3.1.3. Decontarea la 15 minute

Întregul flux propus al procesului de tranzacționare locală care include decontarea la 15 minute constă din mai multe etape: I) Ofertele și cererile sunt generate pe baza previziunilor de generare și consum; II) Prima etapă: licitație pentru ziua următoare. Surplusul de energie fotovoltaică este comercializat prin contracte smart la nivel comunitar. Două mecanisme de stabilire a prețurilor la licitație (Pay-as-Bid și Uniform Price) și două modele de decontare (versiunea clasică și versiunea îmbunătățită) sunt încorporate în algoritm; III) Rezultatele sunt evaluate pentru a identifica dacă mai există ofertă și cerere ca urmare a diferențelor de preț. Astfel, unele oferte și cereri rămân nesatisfăcute; IV) Etapa a doua: licitația cu 6 ore înainte se desfășoară cu oferte și cereri actualizate; V) Sunt calculați și comparați indicatorii de tranzacționare pentru a măsura performanța pieței. În funcționarea în timp real, pot apărea abateri de la program din cauza previziunilor imperfecte sau a întreruperilor neplanificate. Aceste abateri ar putea fi temperate și gestionate printr-un proces de licitație cvasi-continuu care permite consumatorilor și producătorilor să își ajusteze strategiile aproape în timp real în funcționarea sistemului. Prin urmare, licitația se repetă la fiecare interval de timp, având în vedere abaterile inevitabile. Ofertele de incrementare și decrementare ar putea rezolva astfel de abateri, indicând prețurile la care producătorii (consumatorii) sunt dispuși să-și mărească/reducă producția (consumul). De exemplu, întreruperea unui generator va duce la un deficit care ar putea fi compensat de un alt generator ținând cont de ofertele de incrementare. Prin urmare, sunt necesare mecanisme de decontare pentru a gestiona aceste abateri. Într-un sistem de decontare unică, licitarea pentru ziua următoare este utilizată doar pentru a programa funcționarea generatoarelor. În schimb, într-un sistem cu două sau mai multe decontări, licitația pentru ziua următoare este luată în considerare atât pentru programare, cât și pentru decontare. Abaterile sau diferențele dintre operarea în timp real și programul pentru ziua următoare sunt plătite ex-post. De obicei, în urma decontării clasice, producătorul sau vânzătorul care nu a

livrat plătește diferența dintre prețul de vânzare cu amănuntul pentru cumpărători (care ar putea fi un plan tarifar) și prețul ofertei înmulțit cu cantitatea nelivrată. De asemenea, un consumator sau un cumpărător care nu a consumat va fi penalizat cu diferența dintre prețul licitat și prețul producătorului pentru vânzători (care ar putea fi un feed-in-tariff). Un sistem cu trei decontări ia în considerare, de asemenea, decontarea cu oră înainte pentru abaterile de la programul pentru ziua următoare și apoi o decontare în timp real pentru abaterile de la programul cu oră înainte. Astfel, un sistem de decontare multiplă atenuează jocul/speculația, deoarece ofertele pentru ziua următoare sunt obligatorii din punct de vedere financiar și, prin urmare, sunt respectate. Atât producătorii, cât și consumatorii ar trebui să respecte programul deoarece dezechilibrele sunt penalizate și scad încasările, cresc plățile. Algoritmul clasic de decontare este descris mai sus. Aceasta nu recompensează participantul care are o abatere care echilibrează efectiv tranzacția. Prin urmare, nu există nicio recompensă indiferent de semnul individual al abaterii dacă valoarea reală este mai mare decât valoarea licitației. În acest caz, încasarea pentru vânzători se calculează prin înmulțirea valorii licitate cu prețul licitației, adunând surplusul înmulțit cu prețul furnizorului pentru vânzători. În mod similar, plata se calculează prin înmulțirea valorii licitate cu prețul licitației, adunând deficitul înmulțit cu prețul de vânzare cu amănuntul pentru cumpărători. Evident, dacă nu există abatere, rezultatele se obțin prin înmulțirea valorii licitate, egală cu valoarea livrată/consumată, cu prețul de compensare. Algoritmul de decontare propus este detaliat în raportul științific al etapei 2022.

3.1.4. Modele de tarificare avansate

Pentru evaluarea diverselor sisteme de tarificare aplicabile la nivelul consumatorilor am dezvoltat un algoritm fiabil de optimizare la nivel agregat bazat pe teoria jocurilor pentru a obține cea mai bună recompensă pentru o comunitate formată din 11 case cu numeroase electrocasnice, implementând un tarif dinamic de tip Time of Use (ToU). Algoritmul de optimizare a costurilor se adresează tuturor tipurilor de electrocasnice programabile, neîntreruptibile și întreruptibile. Tariful dinamic ToU calculează iterativ tarifele off-peak, shoulder și peak pe baza consumului total orar al comunității, atribuind tarife mai mari la nivel de consum ridicat, descurajând astfel consumul la vârf de sarcină. Tarifele calculate sunt introduse pentru algoritmul de optimizare care oferă o soluție de minimizare a costurilor prin deplasarea consumului electrocasnicilor programabile către ore cu tarife mai mici. Deoarece tarifele ToU sunt calculate pe baza matricei totale de consum orar, algoritmul îmbunătățește minimizarea costurilor prin optimizare repetitivă, oferind, ca urmare, reduceri de costuri de până la 30%. Simulările fără constrângeri, văzute ca având cel mai înalt grad de flexibilitate, conduc la noi vârfuri care se traduc prin necesitatea de a impune constrângeri. Simulările au fost efectuate cu preferințele anumitor clienți (~20% ore de funcționare restricționate) sau cu constrângeri impuse de furnizor (în cazul în care preferințele consumatorilor nu sunt disponibile) care limitează gradul de flexibilitate al electrocasnicilor programabile. Indicele de aplatizare poate ajunge până la 0,83. Tarifele ToU fixe au fost implementate la începutul strategiilor DSM. Cel mult au avut rezultate bune în primele luni de implementare, apoi nu au reușit să îmbunătățească DR, deoarece alura curbei tarifului și a curbei de sarcină diferă semnificativ ca urmare a schimbării comportamentului consumatorilor. Prin urmare, un design flexibil al tarifului ToU este definit ca o funcție de cost pătratică în funcție de consumul orar:

$$t^h = f(A^h) = a \times (A^h)^2 + b \times A^h + c \quad (2)$$

Notațiile se găsesc aici: <https://doi.org/10.1007/s10957-021-01876-1>

Astfel, am calculat tarifele în funcție de consumul orar al comunității la orele de vârf, de zi și în afara orelor de vârf. Vectorul tarifar ToU are trei componente corespunzătoare orelor

de vârf, de zi și celorlalte ore. Tariful de vârf se calculează ca media tarifelor t^h pentru orele de vârf (între 17 și 21), extrasă din curba de sarcină. Tariful în afara orelor de vârf se calculează ca media tarifelor t^h pentru orele de vârf (între 1 și 4 și 24). Tariful de tip *shoulder* se calculează ca medie a tarifelor t^h pentru orele rămase.

$$\begin{aligned} ToU &= \{t^{h_{rate}} | t^{h_{rate}} > 0, t^{h_{rate}} \in \{t^{h_{peak_{rate}}}, t^{h_{offpeak_{rate}}}, t^{h_{shoulder_{rate}}}\}, \forall h = \overline{1,24} \quad (3) \\ t^{h_{offpeak_{rate}}} &= \text{mean}(t^h), \forall h = \overline{1,4} \text{ and } h = 24 \\ t^{h_{shoulder_{rate}}} &= \text{mean}(t^h), \forall h = \overline{5,16} \text{ and } h = \overline{22,23} \\ t^{h_{peak_{rate}}} &= \text{mean}(t^h), \forall h = \overline{17,21} \end{aligned}$$

Integrarea pe scară largă a sistemelor de contorizare inteligentă împreună cu electrocasnicele inteligente și sistemele de management al consumului permit implementarea diferitelor scheme tarifare avansate, cum ar fi: tarife bazate pe momentul de consum (ToU), prețul de vârf critic, tariful în timp real etc. Aceste noi scheme ajută furnizorii să interacționeze cu consumatorii și să ofere stimulente pentru aplatizarea curbei de sarcină. Astfel, consumatorii de energie electrică își programează funcționarea electrocasnicilor pentru a minimiza costul energiei electrice.

Am propus un model de joc de semnalizare pentru optimizare (*Signaling Game Optimization - SGO*) care se realizează la nodurile Fog și Edge și implică mai mulți pași de optimizare. Un tarif dinamic avansat este calculat iterativ în funcție de nivelul de consum. Procesul de optimizare a jocului se oprește atunci când nu mai există nicio îmbunătățire suplimentară a recompensei jucătorilor.

Trei tarife cu structuri diferite sunt descrise și implementate cu algoritmul de optimizare a jocului de semnalizare (SGO) propus pentru a înțelege implicațiile acestora. O funcție *real time tariff* - RTT este definită ca o funcție de cost pătratică:

$$RTT^h = a \times (C^h)^2 + b \times C^h + c \quad (4)$$

Având în vedere profilul de consum al comunității, putem seta tariful ca un tarif dinamic, stabilind tarifele orare în funcție de consumul mediu orar al comunității la orele de vârf, de zi și în afara orelor de vârf.

$$ToU^h = f(C^h) = \begin{cases} t^{h_{peak}}, & \text{if } h = \overline{17,21} \\ t^{h_{offpeak}}, & \text{if } h = \overline{0,4} \\ t^{h_{day}}, & \text{if } h = \overline{5,16} \text{ or } h = \overline{22,23} \end{cases} \quad (5)$$

În plus, am propus o combinație hibridă de RTT și structură tarifară ToU (denumită în continuare *hRTT*). Pentru a calcula tariful, sunt prevăzuți diferiți coeficienți a, b și c pentru orele de vârf, de zi și în afara orelor de vârf.

$$\begin{aligned} hTRR^h &= (t^{h_{peak}}, t^{h_{offpeak}}, t^{h_{day}}) = f(C^h) \\ t^{h_{peak}} &= 0.036 \times (C^h)^2 + 0.1413 \times C^h + 0.4 \\ t^{h_{offpeak}} &= 0.013 \times (C^h)^2 + 0.0024 \times C^h + 0.15 \\ t^{h_{day}} &= 0.019 \times (C^h)^2 + 0.0242 \times C^h + 0.35 \end{aligned} \quad (6)$$

Rezultatele simulărilor sunt prezentate în raportul științific al etapei 2022.

3.1.5. Indicatori de performanță (KPI)

Pentru evaluarea modelelor de tarifare avansate am calculat o serie de indicatori de performanță precum ar fi: economii (Savings), câștig (Gain), indicele de aplatizare (FI) și Peak Average to Ratio (PAR). Economii consumatorilor reprezintă diferența dintre costul inițial și costul cu algoritmul SGO, de asemenea, economiile pot fi calculate ca diferență între costul de optimizare clasic (obținut după prima iterare) și costul SGO. Câștigul consumatorilor reprezintă procentul de reducere a costurilor, FI este raportul dintre consumul mediu și

consumul la vârf, în timp ce PAR este raportul dintre consumul la vârf la pătrat și consumul mediu la pătrat.

$$\begin{aligned}
 FI &= \frac{\text{mean}(C^h)}{\max(C^h)} \\
 PAR &= \frac{(\text{mean}(C^h))^2}{(\max(C^h))^2} \\
 Gain &= 100 - \frac{\sum_{i=1}^n \text{Cost}_i \times 100}{\sum_{i=1}^n \text{Cost}_i^0} \\
 Savings &= \sum_{i=1}^n \text{Cost}_i^0 - \sum_{i=1}^n \text{Cost}_i
 \end{aligned} \tag{7}$$

Acești indicatori au fost calculați pentru un set de date reprezentând 11 case moderne cu peste 300 de electrocasnice care formează o mică comunitate rezidențială și care sunt luate în considerare în simularea bazată pe teoria jocurilor care urmărește optimizarea consumului de energie electrică. Rezultatele sunt prezentate în raportul științific al etapei 2022.

Act 3.2 - Implementarea prototipului informatic BigData4Grid în mediul de testare

3.2.1. Dezvoltarea interfețelor cu utilizatorii

Modelele dezvoltate în cadrul proiectului au fost implementate în cadrul *prototipului informatic BigData4Grid* structurat pe două niveluri funcționale: *nivelul M1 destinat consumatorilor/prosumatorilor și nivelul M2 destinat agregatorilor/distribuitorilor de energie electrică*. Prototipul a fost dezvoltat pe arhitectura *Edge-Fog-Cloud* utilizând limbaje de programare, sisteme de gestiune a bazelor de date, protocoale de comunicație și medii de dezvoltare *open-source*: Python, MySQL, MongoDB, MQTT, HTTP, Flask și Bootstrap.

Pentru *nivelul M1* a fost dezvoltat un kit cu următoarele componente:

- dispozitiv Raspberry Pi 3 Model B, CPU 1.2GHz Broadcom BCM2837, 1 GB RAM. Pe acesta au fost instalate pachetele Python necesare implementării algoritmilor, o instanță MySQL (MariaDB) pentru gestiunea datelor și Node-RED pentru implementarea fluxurilor de comunicație bazate pe MQTT și HTTP;
- senzor Sense HAT pentru măsurarea temperaturii și umidității din interior. Pentru o precizie mai mare au fost experimentați și senzorii Air Things de monitorizare a temperaturii, umidității, calității aerului și emisiilor de radon;
- prize inteligente de tip TP-LINK HS100, HS110, KP115 pentru monitorizarea și controlul dispozitivelor electrice.

Comunicația dintre dispozitivul Raspberry Pi, prizele inteligente, senzorii de temperatură și umiditate, precum și invertoare/smart metere se realizează prin protocoalele de comunicație MQTT și HTTP.

Modelele pentru DSM au fost implementate sub forma unei aplicații web destinate consumatorilor/prosumatorilor și care integrează modelele de monitorizare și control (algoritmul LMC), optimizare a consumului (algoritmul DALO) și estimare a generării din sistemele fotovoltaice în cazul prosumatorilor (Forecast PV). Aplicația *My SASM dashboard* extinde simulatorul virtual dezvoltat în etapa 2021 și permite utilizatorilor un control efectiv al dispozitivelor, precum și planificarea optimă a funcționării dispozitivelor electrice în funcție de planul tarifar și de disponibilitatea generării din surse proprii în cazul prosumatorilor.

Pentru *nivelul M2*, modelele propuse în etapa curentă au fost integrate în cadrul prototipului sub forma unei aplicații (*M2 Dashboard*) destinate furnizorilor, agregatorilor sau

DSO prin care aceștia pot monitoriza consumul agregat pe fiecare nod/total și pot estima consumul pentru ziua următoare la intervale de 15 minute.

3.2.2. Implementarea prototipului în mediul de testare

Prototipul a fost implementat într-un mediu de testare configurat împreună cu partenerul ICPE. S-a urmărit implementarea pe arhitectura *Edge-Fog-Cloud* cu următoarele componente:

- *Nodurile Edge* – constau din kit-urile destinate consumatorilor/prosumatorilor (dispozitivele Raspberry Pi, prizele inteligente și senzori). Acestea sunt conectate la invertoare și contoare inteligente pentru colectarea datelor de consum și generare din surse proprii;
- *Nodurile Fog* – au fost configurate patru noduri *Fog* cu diferite configurații hardware: Fog 1 și 2 – procesor i7, 2.60 GHz, 32 GB RAM, Windows 10; Fog 3 – procesor i9, 2.4 GHz, 32 GB RAM, Windows 10; Fog 4 – procesor i7, 2.4 GHz, 32 GB RAM, Ubuntu. Toate nodurile rulează MongoDB 5.0 pe containere Docker folosind imaginea MongoDB Community Docker. Pentru a evalua performanța nodurilor *Fog* am utilizat același set de date pe fiecare nod cu 5.567 de consumatori.
- *Nodul Cloud* – a fost configurat un singur nod de tip Cloud care rulează pe un server PowerEdge Rack R440 cu procesor Intel Xeon Silver 4208 2,1 GHz și 32 GB RAM. Pentru a compara valorile cu nodurile *Fog*, MongoDB 5.0 este setat pe un container Docker.

3.2.3. Testarea și validarea prototipului

Pentru testare, am utilizat atât seturi de date colectate din surse ICPE (prosumatori, consumatori, noduri), surse web pentru date meteo (accuweather și openweather), cât și seturi de date disponibile precum *SmartMeter Energy Consumption Data in London Households* (<https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>) în care sunt datele de consum provenite de la peste 5500 consumatori cu contoare inteligente. Datele au fost prelucrate și integrate în colecțiile MongoDB. Kit-urile pentru nodurile *Edge* au fost testate pentru 5 prosumatori din România cu puteri instalate între 3-9 kW având invertoare Huawei și Growatt.

În evaluarea performanțelor prototipului s-au luat în considerare atât metrici specifice modelelor de predicție și optimizare cât metrici prin care se evaluează performanțele tehnice ale prototipului: gradul de încărcare și utilizare al resurselor de calcul pe fiecare nod (CPU, hard disk, memorie internă), timpul de răspuns, latența în comunicație și procesare, spațiu de stocare necesare colecțiilor de date, numărul de task-uri realizate cu succes/eșuate.