



Universitatea
Transilvania
din Brașov

Școala Doctorală Interdisciplinară

Facultatea de Inginerie Electrică și Știința Calculatoarelor

Departamentul de Electronică și Calculatoare

Paul Vlad FERNOAGĂ

Contribuții la Implementarea Inteligenței Artificiale în
Rețele de Instrumentație

Contributions to the Implementing of Artificial Intelligence in
Instrumentation Networks

REZUMAT / ABSTRACT

Conducător științific

Prof. univ. dr. ing. Florin SANDU

D-lui (D-nei)

COMPONENȚA

Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Brașov

Nr. 10618 din 10 iulie 2020

PREȘEDINTE: Prof.dr.ing. Mihai IVANOVICI - Universitatea "Transilvania" din Brașov
CONDUCĂTOR ȘTIINȚIFIC: Prof.dr.ing. Florin SANDU - Universitatea "Transilvania" din Brașov
REFERENȚI: Prof.dr.ing. Adriana SÎRBU - Universitatea Tehnică „Gh. Asachi” din Iași
Prof.dr.ing. Tudor PALADE - Universitatea Tehnică din Cluj-Napoca
Prof.dr.ing. Mihai ROMANCA - Universitatea "Transilvania" din Brașov

Data, ora și locul susținerii publice a tezei de doctorat:

Data 19 Septembrie 2020, ora 11, online

Eventualele aprecieri sau observații asupra conținutului lucrării vă rugăm să le transmiteți în timp util, pe adresa vlad.fernoaga@unitbv.ro

Totodată vă invităm să luați parte la ședința publică de susținere a tezei de doctorat.

Vă mulțumim.

Cuprins

	Pg. teză	Pg. rezumat
Introducere	2	1
Obiectivele cercetării doctorale		
Organizarea tezei de doctorat		
1. Inteligența artificială în mediul distribuit – stadiul actual	10	9
1.1. Taxonomia inteligenței artificiale.....	10	9
1.1.1. Clasificări ale I.A. bazate pe lățime (multilateralitate)	10	9
1.1.2. Etapele de inovare în dezvoltarea I.A.....	11	9
1.1.3. Clasificarea I.A. pe modalități de învățare: I.A. simbolică și învățarea automată	12	10
1.1.4. Clasificarea învățării automate după tipurile de aplicații.....	13	10
1.1.5. Clasificarea învățării automate după modelele de învățare.....	13	10
1.1.6. Clasificarea învățării automate după paradigma algoritmilor.....	14	11
1.1.7. Clasificarea I.A. după adâncime (stratificare).....	18	11
1.1.8. Clasificarea rețelelor neurale artificiale pe baza algoritmilor.....	20	12
1.2. Fluxuri informaționale în mediul distribuit	22	12
1.2.1. De la Centre de Calcul la Centre de Date	22	12
1.2.2. Cloud - ca infrastructură de procesare și stocare a datelor.....	23	12
1.2.3. Modalități de stocare a datelor.....	25	13
1.3. Structuri de date centralizate și locale.....	28	14
1.3.1. Interfețe de comunicare cu date structurate	28	14
1.3.2. Interfețe de comunicare cu date nestructurate	30	14
1.4. Pregătirea datelor în contextul masivelor de date – Big Data Analytics.....	31	15
1.4.1. Map Reduce	31	15
1.4.2. Apache Hadoop	32	15
1.4.3. Masivele de date - bază pentru Inteligența artificială.....	33	15
1.4.4. Servicii pentru managementul masivelor de date.....	34	15
1.5. Sumarul capitolului	34	16
2 Integrarea rețelelor de instrumentație în Cloud	36	17
2.1. Achiziția de date în Cloud	37	17
2.1.1. Soluții pentru managementul datelor eterogene.....	39	18
2.1.2. Modalități de lucru cu date eterogene	41	18
2.1.3. Integrare cu soluții de procesare în masă	43	19
2.2. Edge computing – Cloud computing	48	20
2.2.1. Stocarea temporară aproape de proces	48	20
2.2.2. Procesare la marginea rețelei	50	20
2.3. Implementarea unui Cloud de instrumentație	53	21
2.4. Sumarul capitolului	55	21

3	Optimizarea relației client server în rețele de instrumentație	56	22
3.1.	Noi paradigme de optimizare a relației client server	56	22
3.1.1.	Orientarea pe servicii a arhitecturii client server	57	22
3.1.2.	Moduri de apelare a serviciilor	62	23
3.1.3.	Replicarea resurselor aproape de utilizator.....	64	23
3.1.4.	Infrastructuri cu disponibilitate neîntreruptă	66	23
3.2.	Introducerea I.A. în rețelele inteligente	68	24
3.3.	Inteligența artificială în decizia asupra tranzițiilor optimale	73	25
3.4.	Partajarea inteligenței artificiale între Cloud și Edge - Model as a Service	78	26
3.5.	Optimizarea continuă a rutinelor de servire a întreruperilor	82	27
3.5.1.	Implementarea mecanismului de optimizare	84	27
3.5.2.	Implementarea prototipului	85	28
3.6.	Sumarul capitolului	88	28
4	Rețele inteligente de instrumentație	90	29
4.1.	IVI - Modelul simplificat de control prin stare	90	29
4.2.	Senzori virtuali cu rețele neurale	92	30
4.2.1.	Senzor virtual cu inteligență artificială pentru un stand de testare a motoarelor	92	30
4.2.2.	Antrenarea rețelei neurale	96	30
4.2.3.	Rularea modelului antrenat	99	31
4.2.4.	Alternative de implementare a rețelei neurale	102	33
4.3.	Detectia spectrală potențată de inteligență artificială centralizată.....	115	33
4.3.1.	Definirea modelului de antrenare	116	34
4.3.2.	Arhitectura generică de predicție pentru seturi de frecvență	119	34
4.4.	Sumarul capitolului	121	35
5	Interfațarea inteligență dintre operator și rețeaua de instrumentație.....	123	36
5.1.	Interfețe de comunicare cu agenți conversaționali în rețele de instrumentație	124	36
5.2.	Definirea agenților conversaționali	128	37
5.2.1.	Modalități de exprimare a experienței de utilizare	128	37
5.2.2.	Moduri de interacțiune cu sistemele de business – serviciile din "back-end"	129	38
5.2.3.	Conectarea cu rețelele de instrumentație	130	39
5.2.4.	Inteligența artificială și agenții conversaționali	132	39
5.2.5.	Propunere arhitecturală de integrare cu rețele instrumentale	133	39
5.3.	Demonstrator pentru integrarea agenților conversaționali în administrarea rețelelor de instrumentație.....	135	40
5.3.1.	Integrarea agenților conversaționali Telegram cu servicii Java	136	40
5.3.2.	Integrarea agenților conversaționali nativi în Cloud	138	41
5.3.	Sumarul capitolului	141	41

6	Contorizarea inteligenta în Cloud	142	42
6.1.	Contorul Cloud	142	42
6.1.1.	Sub-sistemul hardware al Contorului – nivel Edge.....	143	42
6.1.2.	Sub-sistemul software la nivelul Edge al contorului	146	44
6.1.3.	Tele-transmisia Edge – Cloud – Edge pentru contorizare.....	147	45
6.1.4.	Procesarea digitală a semnalelor la nivel Cloud	148	45
6.1.5.	Structura integrată a proiectului de Instrumentație Virtuală	152	45
6.2.	Cloud AI pentru citirea contoarelor clasice de utilități	162	47
6.3.	Sumarul capitolului	169	50
7.	Concluzii și contribuții originale	170	51
	Dezvoltări viitoare	183	65
	Validarea și diseminarea rezultatelor științifice în publicații și proiecte de cercetare	185	66
	A) Lucrări proprii indexate ISI	185	66
	B) Lucrări proprii indexate SCOPUS	187	67
	C) Lucrări proprii indexate în alte BDI (EBSCO)	187	67
	D) Proiecte de cercetare	187	67
8	Referințe.....	171	68
8.1	Bibliografie	188	69
8.2	Web-grafie	196	76
	Rezumat (lb. română/lb. engleză)	A-22	77

Table of contents

	Pg. thesis	Pg. resume
Introduction.....	2	1
Objectives of the PhD research		
Organization of the PhD thesis		
1. Artificial intelligence in the distributed environment – state of the art	10	9
1.1. The taxonomy of artificial intelligence (AI).....	10	9
1.1.1. Classifications of AI based on width (multilateralism)	10	9
1.1.2. The stages of innovation in AI development.....	11	9
1.1.3. Classification of AI learning modalities: Symbolic AI and machine learning.....	12	10
1.1.4. Classification of machine learning by types of applications.....	13	10
1.1.5. Classification of machine learning by learning models.....	13	10
1.1.6. Classification of machine learning by algorithm paradigm.....	14	11
1.1.7. Classification I.A. by depth (stratification).....	18	11
1.1.8. Classification of artificial neural networks by algorithms.....	20	12
1.2. Information flows in the distributed environment	22	12
1.2.1. From Computing Center to the Date Center.....	22	12
1.2.2. Cloud - as a data processing and storage infrastructure.....	23	12
1.2.3. Ways of storing data.....	25	13
1.3. Centralized and local data structures.....	28	14
1.3.1. Structured data communication interfaces.....	28	14
1.3.2. Communication interfaces with unstructured data.....	30	14
1.4. Preparing data in the context of data masses - Big Data Analytics.....	31	15
1.4.1. Map Reduce	31	15
1.4.2. Apache Hadoop	32	15
1.4.3. Big Data - the basis for Artificial Intelligence	33	15
1.4.4. Services for big data management	34	15
1.5. Summary	34	16
2 Integration of instrumentation networks in the Cloud	36	17
2.1. Data acquisition in the Cloud.....	37	17
2.1.1. Solutions for the management of heterogeneous data.....	39	18
2.1.2. Ways of working with heterogeneous data	41	18
2.1.3. Integration with mass processing solutions.....	43	19
2.2. Edge computing – Cloud computing	48	20
2.2.1. Caching close to process.....	48	20
2.2.2. Edge processing	50	20
2.3. Implementing an Instrumentation Cloud	53	21
2.4. Summary	55	21

3	Optimizing the client-server relationship in instrumentation networks.....	56	22
3.1.	New paradigms for optimizing the client-server relationship	56	22
3.1.1.	Service-oriented client-server architecture	57	22
3.1.2.	Ways to call services	62	23
3.1.3.	Replication of resources close to the user.....	64	23
3.1.4.	Infrastructure with uninterrupted availability	66	23
3.2.	The introduction of AI in Smart Grids	68	24
3.3.	Artificial intelligence in the decision on optimal transitions	73	25
3.4.	Artificial intelligence sharing between Cloud and Edge - Model as a Service	78	26
3.5.	Continuous optimization of interrupt service routines	82	27
3.5.1.	Implementation of the optimization mechanism.....	84	27
3.5.2.	Implementation of the prototype.....	85	28
3.6.	Summary	88	28
4	Intelligent instrumentation networks	90	29
4.1.	IVI - Simplified state control model	90	29
4.2.	Virtual sensors with neural networks	92	30
4.2.1.	Artificial Intelligence Virtual Sensor for an Engine Test Stand	92	30
4.2.2.	Neural network training	96	30
4.2.3.	Running the trained model	99	31
4.2.4.	Alternatives for neural network implementation.....	102	33
4.3.	Enhanced spectral detection by centralized artificial intelligence.....	115	33
4.3.1.	Defining the training model.....	116	34
4.3.2.	Generic prediction architecture for frequency sets	119	34
4.4.	Summary	121	35
5	Intelligent interfacing between the operator and the instrumentation network.....	123	36
5.1.	Communication interfaces with conversational agents in instrumentation networks	124	36
5.2.	Definition of conversational agents	128	37
5.2.1.	Ways of expressing the user experience	128	37
5.2.2.	Ways of interaction with business systems - back-end services	129	38
5.2.3.	Connecting to instrumentation networks	130	39
5.2.4.	Artificial intelligence and conversational agents	132	39
5.2.5.	Architectural proposal for integration with instrumental networks	133	39
	Demonstrator for the integration of conversational agents in the administration of instrumentation	135	40
5.3.	networks.....		
5.3.1.	Integrating "Telegram" Conversational Agents with Java Services	136	40
5.3.2.	Integrating Native Conversational Agents into the Cloud.....	138	41
5.3.	Summary	141	41

6	Smart metering in the Cloud	142	42
6.1.	Cloud meter	142	42
6.1.1.	Hardware sub-system of the counter – Edge level.....	143	42
6.1.2.	Software sub-system of the counter – Edge level	146	44
6.1.3.	Edge – Cloud – Edge tele-transmission	147	45
6.1.4.	Digital signal pocessing in the Cloud	148	45
6.1.5.	Integrated structure of the virtual instrumentation project	152	45
6.2.	Cloud AI for reading classic utility meters	162	47
6.3.	Summary	169	50
7.	Conclusions and original contribution	170	51
	Future developments	183	65
	Validation and dissemination of scientific results in publications and research projects	185	66
	A) Own ISI indexed articles.....	185	66
	B) Other SCOPUS indexed articles.....	187	67
	C) Own articles indexed in EBSCO journals.....	187	67
	D) Research projects.....	187	67
8	References.....	171	68
8.1	Bibilography	188	69
8.2	Web-graphy	196	76
	Resume (lb. română/lb. engleză)	A-22	77

Introducere

Inteligența Artificială se află în centrul preocupărilor legate de *“Transformarea Digitală”* a societății moderne. La ora actuală multe voci proclamă *“a 4-a revoluție industrială”* (după primele trei marcate de mecanizare, producția pe bandă rulantă și automatizare-electronizare). *Actualitatea* acestei teze de doctorat constă în apartenența ei la efortul global numit *“Industry 4.0”* care are în centru extinderea comunicațiilor om-la-om și om-mașină către *comunicațiile nemijlocite mașină-mașină* (M2M – *“Machine-to-Machine”*).

Tematica tezei de doctorat constă în implementarea Inteligenței Artificiale (I.A.) în rețelele moderne de instrumentație (*“smart metering”*) guvernate de servicii Cloud. [1][2][3]

Oportunitatea prezentei lucrări de doctorat e susținută de numărul mare de inițiative strategice ale organismelor naționale și internaționale cu răspunderi în domeniul *Tehnologiilor Informatice (IT) și de Comunicații (ICT)* – dintre acestea, merită menționat proiectul *“Going Digital”* al OECD (Organizația pentru Cooperare Economică și Dezvoltare) care susține transformarea digitală a societății într-un spectru larg de obiective – de la creștere și până la bunăstare. În urma conferinței *“AI: Intelligent Machines, Smart Policies”* din octombrie 2017, OECD a publicat o amplă lucrare, *“Inteligența Artificială în Societate”* [4] care poate fi o bună referință pentru *motivația și actualitatea tezei de doctorat*.

“Inteligența” reprezintă un concept amplu iar terminologia asociată are multiple implicații și nu poate fi supusă niciunor restricții de utilizare, atâta vreme cât interesele comerciale sunt atât de diverse și de răspândite [W-1].

În centrul acestei teze de doctorat se află două domenii importante care, aparent, au aceeași traducere în românește – *“Rețele Inteligente”* – respectiv, în limba engleză, *“Intelligent Networks”* (IN) și *“Smart Grids”* (SG).

Privitor la traducerea *SG*, autorii unei monografii editate de Societatea Inginerilor Energeticieni din România, [5], prezintă o documentată motivare asupra deciziei lor de a păstra denumirea *SG* și în textul românesc (așa cum va fi cazul și în această lucrare), întrucât *“smart”* e mai puțin decât *“intelligent”* (a se vedea definiția IN & SG de mai jos) iar *“grid”* nu înseamnă *“network”* ci, mai degrabă, acel sistem de linii electrice (de transport și distribuție a energiei) și, prin extensie, acel *“caroiaj”* de conducte (având geometria străzilor unui oraș obișnuit) dacă accepțiunea *SG* se extinde din electrotehnică la *“utilități”* (apă potabilă, agent termic, gaz etc).

“Rețelele Inteligente” (de telecomunicații, calculatoare și instrumentație – *integrate IT*) sunt construite astfel ca înșiși clienții (persoane fizice sau juridice) – și nu numai cei care operează, care dețin sau au produs sistemele – să poată fi în măsură să conceapă, să dezvolte, să instaleze, să configureze, să pună în funcțiune, să administreze și să opereze *serviciile* [6][W-2].

În această lucrare de doctorat, un *apel* în cadrul *IV*, e considerat generic ca un "apel la resurse" umane sau instrumentale – căutare/găsire, conectare/deconectare etc (cu toate metricile și înregistrările aferente).

Cât despre definițiile *SG*, reputatul EPRI (Electric Power Research Institute – S.U.A.) consideră că „un SG este definit *de încorporarea ICT* în fiecare aspect al generării, transportului, distribuției și consumului de electricitate, cu scopul de a minimiza impactul asupra mediului, asupra costurilor, concomitent cu îmbunătățirea situației de pe piața serviciilor, a siguranței și eficienței”. Practic, peste rețeaua de electricitate („*on top*”) se adaugă instrumentația modernă ("*smart metering*") și deasupra ei telecomunicațiile. [143][144]

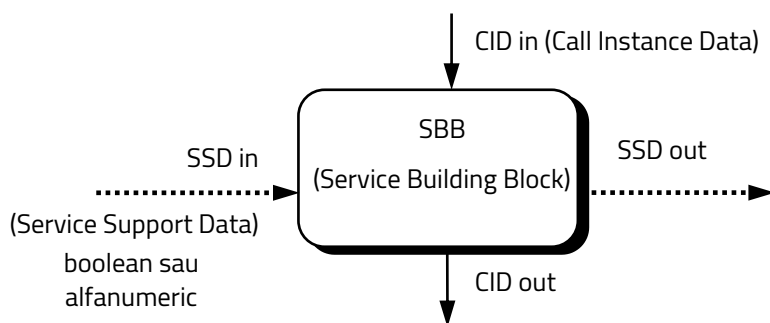
Așadar, un loc central în această teză de doctorat îl au *comunicațiile industriale* cu o semnificativă *orientare pe servicii*, sub aspectul OSS/BSS ("Operational- / Business- Services & Support") – servicii și suport – de operare / de business. Perspectiva *business* – pornind de la cheltuielile investiționale / de operare ("CapEx / OpEx – Capital/Operational Expenditure") extinde conceptele de management ale rețelelor cu aspectele specifice ale *valorii adăugate* (mai ales prin "*servicii cu valoare adăugată*") și ale *monetizării* (inclusiv sub aspectele fiscale și ale capabilităților de auditare).

Fezabilitatea temei este asigurată de sinergia modernă *Calculatoare/Telecomunicații/Instrumentație*, reprezentată la superlativ de "eXtinderea pentru Instrumentație" ("–XI", precum "LXI – LAN eXtension for Instrumentation", sau "PXI – PCI eXtension for Instrumentation" etc) a complexului *ATCA* (Advanced Telecom & Computing Architecture), în așa numita *AXI* ("ATCA eXtension for instrumentation). În acest fel, se adaugă la sub-sistemele (module, plăci – "blades") de *calcul* (procesare, stocare etc) și de *telecomunicații* (comutație și rutare, procesare de pachete etc) sub-sistemele (modulele) de *instrumentație* (senzori-transectoare-"actuatoare", achiziție de semnale, conversii analog-digital / digital-analog, măsurări și acționări avansate). Definitiv pentru ATCA este că toate aceste trei clase de module au, în spatele plăcii de bază (acel "back-plane", "fund de sertar" – "motherboard" cu alimentare comună și baze de timp *comune*) "*textura comună* de inter-comunicații (acea "plasă" – "mesh" de legături "fiecare cu fiecare") ultra-rapidă "*InfiniBand*" inspirată la origine de Apple "FireWire" (legături seriale de mare performanță, cu conectare "din mers" – "hot plugging").

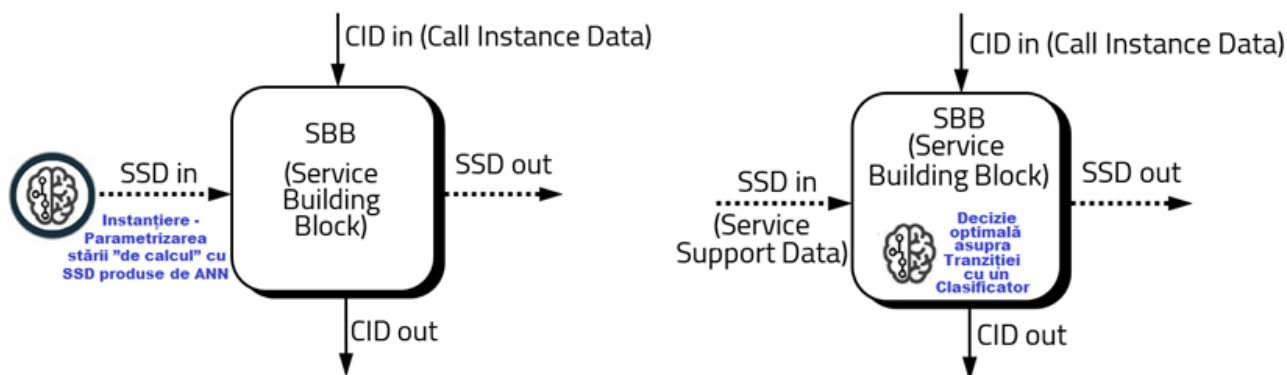
Această infrastructură avansată integrată IT în Cloud – pe baza comunicațiilor IP (Internet Protocols) fără restricții de poziționare sau mobilitate – susține abordarea nouă, propusă de această teză de doctorat: *în rețelele inteligente, orientate pe servicii, controlul prin stare, răspunsul la evenimente ("servirea cererilor de întrerupere") să fie oferit / potențat / mijlocit de către inteligența artificială.*

Obiectivele cercetării doctorale:

1. Extinderea *Rețelelor Inteligente* de telecomunicații, calculatoare și instrumentație – guvernate de capacitatea de a dezvolta și integra serviciile "de sus în jos" (chiar de către clienții individuali sau instituționali) – prin *introducerea inteligenței artificiale*.
2. Implementarea inteligenței artificiale în *automatele cu stări finite* ("FSM – Finite State Machines") ce corespund serviciilor, plecând de la premisa că un serviciu este un automat compus din:
 - "stări" de calcul reprezentate (în diagrama logică, "organigrama" serviciului) ca "blocuri constructive pentru servicii" (SBB, Service Building Blocks – a se vedea figura următoare)
 - "operanți" (mai general *parametri* care *instanțiază* metodele de calcul aferente, transformate astfel în *obiecte*, din clase, în paradigma Programării Obiect-Orientate, POO) care se numesc SSD (Service Support Data)
 - cu "*tranziții*", transmiterea *deciziilor* aferente fiind reprezentată prin *semnalizările* CID (Call Instance Data)



Obiectivele propuse constau în *introducerea inteligenței artificiale atît "pe orizontala datelor" (prin parametrizarea SSD – Service Support Data) cât și "pe verticala decizională" (CID – "Call Instance Data")* prin *intermedierea "calculului evenimentelor"* cel care poate estima cea mai bună tranziție din traiectoria optimă de stări - "strategia" evolutivă a automatului cu stări finite:

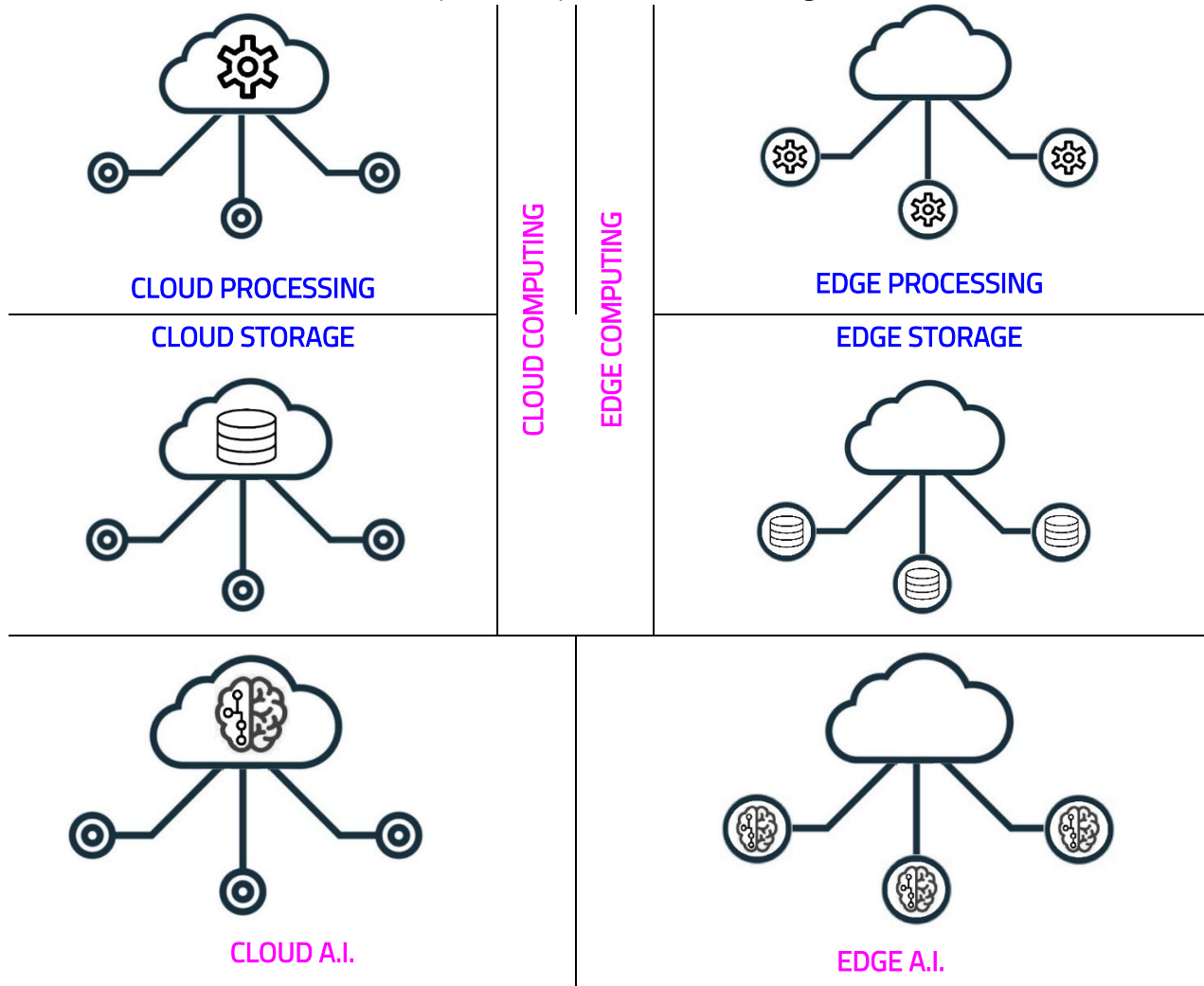


În perspectiva și mai largă a POO (Programarea Orientată pe Obiecte), pentru intermedierea, prin inteligența artificială, a concretizării clasei în obiect, obiectivele specifice sunt:

- calculul / alegerea (de exemplu prin tehnici de clasificare) a *atributelor* în vederea instanțierii *optimale*;

- parametrizarea tranzițiilor - la nivelul atributelor fiecărui "declanșator" ("trigger")
- tratarea diferențiată a declanșatoarelor ("administrative"- "contextuale" sau de tip "cerere" – "notificare" asincronă, conjuncturală etc)
- decizia optimală asupra stării viitoare.

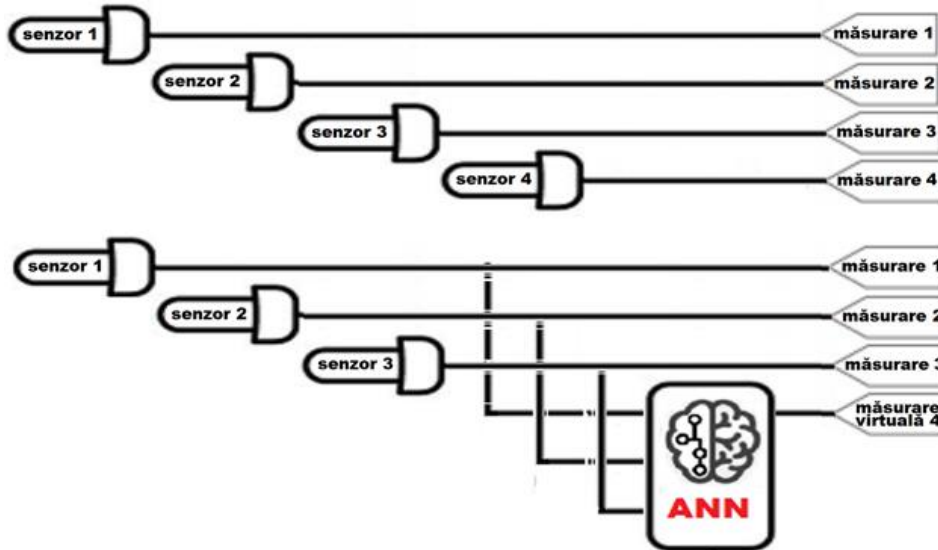
3. Extinderea conceptului "Cloud-computing" / "Edge computing" (de cooperare a calculului centralizat cu cel local, distribuit) prin conceptul de "Cloud AI / Edge AI":



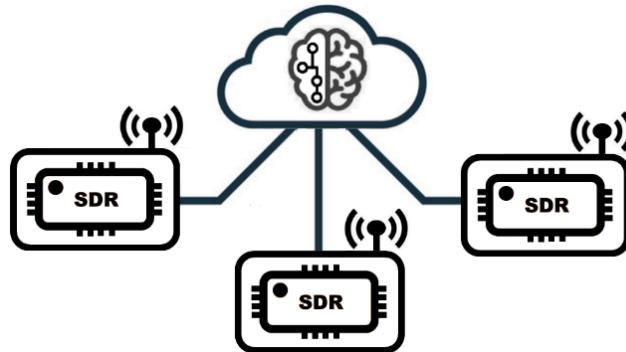
Aceste soluții sunt potențate de recentele progrese în domeniul SBC ("Single Board Computers") și al capacității computaționale a sistemelor "embedded" (cu procesor încorporat) cu aplicații moderne IoT în rețelele de instrumentație. Cu scopul de a optimiza repartizarea calculului Cloud / Edge, prin soluții de comunicații și sincronizare dedicate rețelelor de instrumentație, vor fi tratate distinct funcțiile computaționale de procesare și stocare.

4. Dezvoltarea unui Cloud de instrumentație, cu suport extins pentru informatica instrumentală / instrumentația virtuală și capacități speciale pentru inteligența artificială.

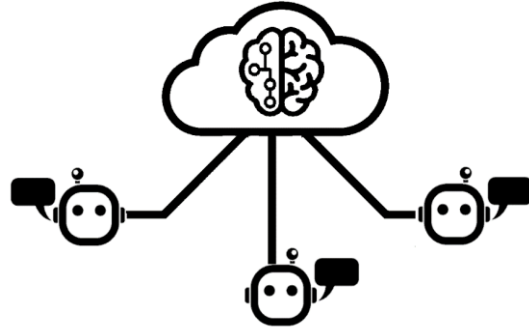
5. Implementarea inteligenței artificiale în relația client-server pentru instrumentația din Cloud.
6. Consolidarea modelului *generic "întrerupere pe pământ / aducerea rutinei de servire a întreruperii din nori"*. Premisa tehnologică o reprezintă performanțele comunicațiilor moderne "ciber-critice" ("CCC – Cyber-Critical Communications") de "time-real" (practic cu constante de timp aflate cu câteva ordine de mărime sub constantele de timp ale proceselor industriale deservite de rețeaua de instrumentație); esențial este *aportul AI în returnarea unei rutine de servire mereu îmbunătățită* (prin mecanismele QoE – Quality of Experience susținute de inteligența artificială).
7. Implementarea conceptului de senzor virtual cu inteligență artificială



8. Extinderea inteligenței artificiale în instrumentația eliberată de restricții spațiale (inclusiv de mobilitate) prin *soluții dedicate "radio cognitiv" SDR* (Radio Define Software)

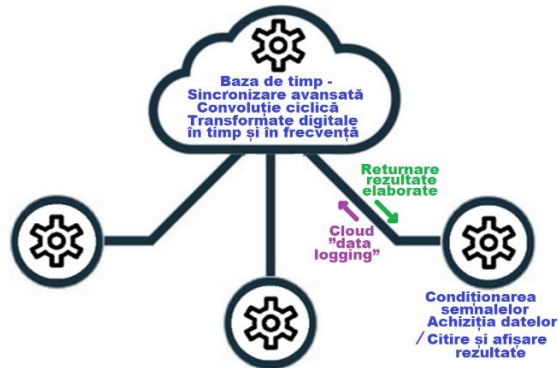


9. Interfațarea inteligență dintre operator și rețeaua de instrumentație. "Umanizarea" răspunsului rețelei prin extinderea interfețelor cu utilizatorul (UI) cu agenți conversaționali (chatbots):

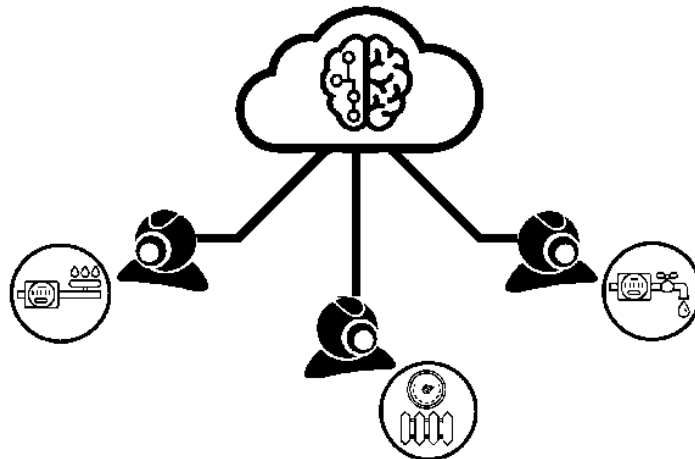


10. Validarea implementărilor inteligenței artificiale în rețele de instrumentație în cadrul unui proiect de cercetare din programul național PN III, pentru “Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare”, pentru “Transfer la operatorul economic”. Obiectivul a fost elaborarea unor soluții cu nivel TRL (“Technology Readiness Level”) ridicat (câte 5 exemplare - “serie zero”) pentru:

- Contorizarea inteligentă a electricității în Cloud – condiționarea și achiziția (nivel Edge computing) precum și teletransmisia curentului și tensiunii către nivelele Cloud storage & Cloud processing (cu implementarea calculelor de puteri reactive printr-o prelucrare exclusiv în domeniul timp, bazată pe transformata Hilbert)



- Citirea optică a contoarelor de utilități din Smart Grid, colectarea și interpretarea informațiilor în “Cloud A.I.”



Organizarea tezei de doctorat

- Teza cuprinde, la început, o listă a abrevierilor folosite în lucrare, o Introducere după care urmează 7 capitole – primele 6 fiind orientate către obiectivele propuse iar ultimul este dedicat evidențierii contribuțiilor originale și validării lor prin lucrări științifice (articole publicate și proiecte de cercetare).
În cele de urmă, secțiunea a 8-a este rezervată referințelor, structurată într-o secțiune de bibliografie cu un număr de 159 titluri și una de web-grafie cu un număr de 18 titluri (considerând și cele 10 articole proprii publicate, totalul e de 187 titluri). La finalul lucrării în secțiunea Anexe se găsește Listing-ul unui program reprezentativ dezvoltat în cadrul cercetării doctorale (Anexa 1) alături de trei articole reprezentative, indexate în ISI (International Scientific Index) WoS (Web of Science) – Anexa 2.
- **Introducerea** (*8 pagini*) are ca scop încadrarea lucrării în contextul direcțiilor de cercetare ale Școlii Doctorale Interdisciplinare și al oportunităților actuale din domeniul tehnologiilor informatice și de comunicații.
- **Capitolul 1, Inteligența artificială în mediul distribuit – stadiul actual** este structurat în patru subcapitole, însumând *26 de pagini* ce conțin *4 figuri* și *2 tabele*, fiind orientat către prezentarea tipologiei I.A. și a aspectelor computaționale în mediul distribuit. În prima secțiune se face o amplă trecere în revistă a modului în care acest domeniu științific este structurat, care sunt criteriile principale după care se clasifică inteligența artificială. Apoi, în secțiunea a doua, sunt analizate tehnicile dedicate de comunicare, procesare și stocare a informației în mediul distribuit. Tot în această secțiune este introdus și conceptul central de *Cloud*. Cea de a treia secțiune se concentrează pe cele două mari categorii de colectare, transmitere și stocare a datelor vehiculate de inteligența artificială: *structurat* și *nestructurat*. Ultima secțiune este responsabilă de trecerea în revistă a tehnicilor și tehnologiilor care fac posibilă procesare volumelor foarte mari de date ce sunt generate în contextului unei digitalizări și conectivități tot mai extinse. Aceste volume mari de date sunt cele care fac oportună existența unor algoritmi cu inteligență artificială robuști.
- **Capitolul 2, Integrarea rețelelor de instrumentație în Cloud, (21 de pagini, 9 figuri, 2 tabele)**, este dedicat *interconectării*, fiind orientat spre infrastructura de comunicații care stă la baza integrării inteligenței artificiale în rețelele de instrumentație. Prima parte studiază modurile în care se poate face achiziția de date de la marginea rețelei (nivel "Edge") către mediul central (nivel "Cloud") folosind interfețe structurate sau nestructurate sau o combinație între cele două. În cea de-a doua parte atenția se mută către distribuția local-central a sarcinilor computaționale și de stocare. Aici am analizat posibilitățile de optimizare a procesării și accesului la date astfel încât dispozitivele de la marginea rețelei să poată fi autonome în anumite contexte. În ultima parte am prezentat un Cloud de instrumentație pe care l-am implementat pe baza tehnologiilor avansate de la National Instruments.
- **Capitolul 3, Optimizarea relației client server în rețele de instrumentație**, structurat în 5 secțiuni care însumează *33 de pagini* cu *21 imagini*, face trecerea la nivelul comunicării ierarhizate a

“cererilor de servire” dintre diferitele entități conectate la Cloud. Scopul acestui capitol este analizarea diferitelor opțiuni de optimizare a relației client - sever cu ajutorul noilor tehnologii existente din Cloud, folosind inteligența artificială ca mediator al comunicării.

- **Capitolul 4, *Rețele inteligente de instrumentație (31 de pagini, 43 de figuri, 10 tabele)***, introduce I.A. în modelele standard (IVI – Interchangeable Virtual Instrumentation) de control prin stare al instrumentației. Dimensiunea puternic aplicativă a capitolului este dată de multiplele implementări ale rețelelor neurale în *senzorii virtuali* precum și de soluțiile *radio cognitiv* potențate de inteligența artificială centralizată. Primul demonstrator folosește I.A. pentru a înlocui senzorii scumpi din industrie cu senzori virtuali implementați cu rețele neurale. Al doilea demonstrator folosește inteligența artificială din Cloud împreună cu tehnologiile SDR (Radio Definit Software) “Spectrum Sensing” pentru a îmbunătăți funcția de Radio *Cognitiv* (în particular strategia de ocupare la emisie a frecvențelor libere- “spectral gaps”).
- **Capitolul 5, *Interfațarea inteligentă dintre operator și rețeaua de instrumentație (18 pagini, 12 figuri și 2 tabele)*** propune o abordare nouă, prin I.A., a modului de comunicare dintre sistem și utilizator. Sunt propuse interfețele OAM (de operare, administrare și mentenanță a rețelei de instrumentație) bazate pe comunicare prin mesaje cu limbaj natural. Aceste tehnologii se regăsesc sub numele de agenți conversaționali - Chatbots. În acest capitol sunt prezentate atât părți conceptuale a agenților conversaționali cât și tehnologiile din spatele acestora. Este cuprinsă metodologia de implementare împreună cu un demonstrator o pune în practică pe aceasta, într-un demers de “umanizare” a răspunsului rețelei cu ajutorul I.A.
- **Capitolul 6, *Contorizarea inteligentă în Cloud, (29 de pagini, 33 de figuri, 1 tabel)***, descrie o altă arie în care inteligența artificială și tehnologiile Cloud pot fi folosite pentru a aduce noi funcționalități pentru utilizator – contorizarea inteligentă. Cercetările au fost validate într-un proiect național PN-III. Este prezentată cercetarea care a dus la achiziția mărimilor electrice (și a altor utilități) – “Cloud Data Logger”, cu procesare “Cloud A.I.” și “citirea contoarelor” prin servicii web. A doua parte a capitolului se orientează către citirea optică a contoarelor clasice cu transmiterea imaginii afișoarelor și procesarea imaginilor în Cloud (OCR folosind inteligența artificială).
- **Capitolul 7 (19 pagini)** este rezervat concluziilor, contribuțiilor, diseminării și dezvoltărilor viitoare. Sunt evidențiate principalele contribuții originale, lucrările publicate (cu indexarea lor și citări) dar și propunerile pentru dezvoltările viitoare.

1. Inteligența artificială în mediul distribuit – stadiul actual

Prezentul capitol face o amplă trecere în revistă a stadiului actual privind *distribuția inteligenței artificiale* (I.A.). După o amplă prezentare a tipologiei I.A. în multiplele ipostaze de ultimă oră, am parcurs fluxurile informaționale în mediul distribuit (în perspectiva diversității instrumentației), noile structuri de date (cu sporirea importanței *stocării* în raport cu *procesarea*) și stadiul actual al masivelor de date (Big Data – Data Mines) și al noilor metode analitice (Data Analytics) de care au nevoie.

1.1. Taxonomia inteligenței artificiale (I.A.)

Inteligența artificială (I.A.) [9] [10] este un domeniu al Tehnologiilor Informatice și de Comunicații (ICT) care se ocupă cu studiul algoritmilor și al modului în care aceștia pot determina un sistem de calcul să răspundă corect la întrebări pentru care nu a fost programat în mod explicit.

1.1.1. Clasificări ale I.A. bazate pe lățime (multilateralitate)

Clasificarea I.A. "după lățime" ("acoperire" – "multilateralitate") va fi urmată și de clasificări "după adâncime" ("profundzime") [14]. Pentru început trebuie remarcat că aplicațiile curente ale I.A. sunt tot mai aproape de omul obișnuit, tot mai folositoare: sortarea robotizată a produselor, detectarea tumorilor, recunoașterea modelelor de trafic sau chiar conducerea mașinii autonome. Acest tip de I.A. dezvoltată în prezent este o inteligență artificială "de tip îngust". Acest lucru înseamnă că mașinile din jurul nostru sunt învățate să execute *o singură sarcină pe care ar putea să o facă un om*, de obicei mult mai bine și mai rapid. Următorul nivel în contextul I.A., este acela de tip **general** [15]. Inteligența artificială de tip general, care mai este numită și "puternică", este aceea care va fi capabilă să facă analogii și schimbări de context asemănătoare cu cele ale creierului uman.

1.1.2. Etapele de inovare în dezvoltarea I.A.

Conform cu DARPA John Launchbury [16] și mai apoi cu Six Kin Development Scott Jones [W-3], dezvoltarea I.A. cunoaște patru etape de inovare. Primul "val" de inovare s-a consumat în perioada 1970-2000, perioadă în care s-au dezvoltat o mulțime de algoritmi și tehnici prin care sistemele au devenit foarte performante în aplicarea logicii deductive și a luării de decizii. Cea de-a doua etapă de inovare care a început în anul 2000 și se estimează ca va dura până în jurul anului 2020, este perioada în care algoritmi au devenit performanți în procesul de învățare și percepție. Cea de-a treia etapă de inovare se estimează ca se va desfășura în jurul anilor 2020-2030, perioadă în care algoritmi vor excela în procesul de învățare, percepție, raționament și vor fi capabili să generalizeze. Cea de-a patra etapă de inovare este estimată să aibă loc după anul 2030, perioadă în care algoritmi și sistemele se vor perfecționa în procesul de a acționa ca un intelect uman.

1.1.3. Clasificarea I.A. pe modalități de învățare: I.A. simbolică și învățarea automată

Din punct de vedere al modalităților de a învăța, I.A. a trecut de la modul de învățare bazat pe reguli, numit și I.A. simbolic, la cel bazat pe cunoaștere, cunoscut sub numele de învățare automată (Machine Learning, "ML") [17]. În prima etapă de inovare, sistemele, care au pătruns pe piață implementau seturi complexe de reguli și instrucțiuni. Astfel, au început să apară sisteme, în general industriale, care mimau un comportament inteligent și autonom, dar care făceau acest lucru într-un mediu izolat și foarte bine conturat. Odată cu avansul tehnologic din domeniul procesărilor de date, dar și a capacităților de stocare a unui volum foarte mare de date, tehnică numită și "big data", performanța algoritmilor de ML a surclasat cu mult algoritmi ce se bazează pe I.A. simbolică. Cea mai eficientă tehnică a învățării automate [19] este "învățarea în adâncime" ("deep learning") care folosește o rețea cu multe straturi de neuroni pentru a putea produce rezultatul corect cu o precizie cât mai mare.

1.1.4. Clasificarea învățării automate după tipurile de aplicații

Clasificarea – este cel mai răspândit tip de aplicație care este implementată cu învățarea automată. Acest tip de algoritm este foarte folosit în recunoașterea automată de caractere, în recunoașterea automată a fețelor ori în domeniul medical sau industrial.

Estimarea continuă – este o altă aplicație răspândită a învățării automate (ML) în care se folosesc algoritmi ce implementează regresia/predicția [21].

Gruparea (Clustering) – este o aplicație a învățării automate (ML) unde se folosesc algoritmi pentru a ordona grupuri pe diferite categorii, în funcție de proprietăți comune.

Achiziționarea de aptitudini – o altă clasă importantă de algoritmi de învățare automată (ML) este aceea care are ca scop însușirea de aptitudini noi pentru a deveni tot mai performanți.

1.1.5. Clasificarea învățării automate după modelele de învățare

Din punct de vedere al metodelor de învățare algoritmi pentru învățarea automată (ML) se împart în trei categorii:

Învățarea supervizată ("învățarea asistată") este procesul prin care algoritmului îi este arătat ce trebuie să învețe pentru fiecare set de date primit [23] [24]. Acest proces se numește etichetarea seturilor de date.

Învățarea nesupervizată este o altă tehnică [25] prin care unii algoritmi ML sunt antrenați fără a li se indica explicit ce trebuie să învețe din respectivele seturi de date, ajungând ca ei "să își dea seama singuri".

Învățarea consolidată ("impusă" – "reinforced learning") este modalitatea în care sistemului îi este dată (impusă) o țintă pe care trebuie să o atingă [26], iar algoritmul va exersa procesul de învățare până la atingerea scopului.

1.1.6. Clasificarea învățării automate după paradigma algoritmilor

Există multe modalități prin care algoritmi se pot împărți după o anumită paradigmă. În cartea "The Master Algorithm" [28], Pedro Domingos propune o abordare prin care algoritmi sunt grupați în cinci categorii. Prima paradigmă după care putem grupa algoritmi este cea a "conectării", unde datele reprezintă blocuri ce au anumite legături între ele. Algoritmi din această categorie sunt cei ce reprezintă rețele neuronale artificiale (ANN – Artificial Neural Networks). A doua paradigmă este cea *evolutivă*, mai exact este vorba despre algoritmi genetici care, prin construcția lor, sunt lăsați să evolueze asemenea unor organisme vii. A treia paradigmă este cea de tip *bayesian* care are la bază știința statisticii și a probabilităților, ce urmărește întărirea sau slăbirea unei ipoteze pe baza evidenței acesteia în datele ce sunt colectate progresiv. Cea de-a patra categorie este aceea a analogiei. În acest caz, se urmărește ca seturile de date ce au lucruri în comun să fie grupate. Cea de-a cincea categorie este cea *simbolică*, ce s-a dezvoltat mai ales în anii '80-'90.

Algoritmi de învățare automată (ML - Machine Learning)

Regresia logistică (Logistic Regression) - este unul dintre cei mai simpli algoritmi de învățare automată, dar și unul dintre cei mai folosiți, pentru că obține performanțe bune în foarte multe scenarii.

Linear Discriminant Analysis (LDA) – este un algoritm asemănător cu regresia logistică, fiind preferat în momentul în care problema de clasificare are mai mult de două clase

K Neighbors Classifier (KNN) -este un algoritm de învățare supervizată care poate să rezolve probleme de clasificare sau de regresie. Algoritmul pleacă de la premisa că elementele similare stau apropiate unele de altele.

Decision Tree Classifier (DTC) – este un alt algoritm de învățare supervizată care folosește o structură de date de tip arbore pentru a clasifica proprietățile setului de date și a obține în final predicțiile

Support Vector Machine (SVM) - face parte din categoria algoritmilor de învățare supervizată folosiți pentru probleme de clasificare.

1.1.7. Clasificarea I.A. după adâncime (stratificare)

Învățarea în adâncime este un domeniu specific al învățării automate (ML - Machine Learning) care încearcă să rezolve probleme la care algoritmi de categoria ML nu sunt performanți. Sarcinile dificile pot fi: împărțirea imaginilor în categorii, prelucrarea sunetelor în vederea recunoașterea vocii, împărțirea în voci feminine și masculine [29] [30].

Pentru asemenea operații, cu date intens nestructurate, au fost dezvoltate rețelele neurale "adânci" – "profunde", care sunt inspirate din modul în care creierul uman funcționează.. O rețea neurală artificială poate fi considerată "adâncă", (Deep Neural Network - DNN) [31] în momentul în care există mai mult de un strat ascuns.

Elemente de bază care compun o rețea neurală sunt : **neuronul, stratul cu neuroni, ponderile, datele de intrare, funcția de activare și algoritmul de învățare, datele de ieșire.**

1.1.8. Clasificarea rețelelor neurale artificiale pe baza algoritmilor

Există mai multe tipuri de rețele neurale unul dintre ele este **rețeaua neurală convoluțională (CNN)**, [35] care este foarte folosită în algoritmi de recunoaștere a imaginilor. **Rețelele neurale recurente (RNN)** [36] sunt o altă clasă de algoritmi ANN care folosește secvențe de date pentru a ajusta neuronii anteriori. Acest tip de algoritmi sunt folosiți cu succes în probleme de recunoaștere de sunet sau de recunoaștere a scrisului de mână. O altă clasă importantă de algoritmi sunt cei pentru **GAN - Generative Adversarial Networks – rețele generate de așa-zisa "adversitate"** care constau, generic, din două rețele neurale care sunt puse să concureze. Aceste două rețele în competiție învață să genereze date noi, foarte apropiate de realitatea celor prezentate în setul de date de antrenare.

1.2. Fluxuri informaționale în mediul distribuit

Acest subcapitol este o radiografie a tehnologiilor și utilităților existente pe piață care permit o nouă abordare în interacțiunea utilizator-sistem sau sistem-sistem, orientată nu pe hardware sau software, ci pe date.

1.2.1. De la Centre de Calcul la Centre de Date

În ultimul deceniu întreaga industrie ICT a investit foarte multe resurse în adoptarea și, apoi, în aducerea la un grad înalt de maturitate a conceptului de Cloud ("calcul în nori"). Cloud-ul este o infrastructură virtualizată care permite oricărui utilizator al platformei să își creeze o întreagă rețea de calculatoare, tradițional numită „Local Area Network”, gestionată printr-o simplă pagină de internet (IP Management).Cloud-ul pune la îndemâna oricui dezvoltator uneltele necesare pentru construirea unui sistem informatic virtualizat complet operațional, ce dispune de cel puțin aceleași resurse ca acelea ale unui sistem tradițional ("on-premise").

1.2.2. Cloud - ca infrastructură de procesare și stocare a datelor

Un avantaj al mediului Cloud în procesarea datelor față de sistemele tradiționale este că infrastructura de rulare este virtualizată și nu depinde de hardware-ul fizic, ceea ce permite dezvoltatorilor să conceapă sisteme de optimizare a răspunsului la o cerere a unui utilizator, aproape în timp real, indiferent de locația utilizatorului.

Ubicuitatea poate fi asigurată fiindcă un operator de servicii poate avea, replicată la nivel virtual, aceeași infrastructura în diverse locații de pe glob, utilizatorul fiind automat conectat la cea mai apropiată dintre acestea și având aceeași experiență de lucru. Un alt avantaj al Cloud-ului este *scalabilitatea*, proprietatea de a se realoca resurse după nevoi, când încărcarea se apropie de limită. Acest lucru ar fi imposibil în cazul unui sistem tradițional de hosting ("găzduire" de date și procese).

Serverless Computing

O altă tehnologie din Cloud care promite optimizarea costurilor de procesare a informației este „serverless computing” care abstractizează și face transparente față de dezvoltator, funcțiile de *server* și infrastructura de deservire [40]. Dezvoltatorul este orientat pe scrierea rutinelor de procesare ca și *funcții*, iar platforma Cloud se ocupă de rularea lor. În plus, platforma monitorizează încărcarea pentru a executa aceste funcții, care poate crește necesarul de resurse până la nivelul potrivit pentru a deservi cererile de procesare într-un timp rezonabil. Teoretic, "serverless computing" este capabil să scaleze la infinit resursele alocate. Pe lângă scalabilitate, tehnologia "serverless computing" oferă și o mobilitate ridicată a proceselor.

1.2.3. Modalități de stocare a datelor

Operarea datelor implică și cunoașterea modalităților de stocare, fiindcă informațiile devin cu adevărat valoroase când sunt memorate și interpretate în raport cu variabila de timp. Modul de stocare a devenit o componentă esențială în optimizarea procesării datelor, sinergia stocare-procesare este evidentă și sub aspectul raportului optim performanță/cost.

Baze de date relaționale

Bazele de date au fost din totdeauna cel mai important mijloc de stocare a datelor. Primele sisteme de baze de date care și-au demonstrat performanța și un nivel de încredere crescut de-a lungul timpului au fost bazele de date relaționale, cunoscute și ca baze de date interogabile – SQL (Structured Query Lanaguage).

Avantajele bazelor de date relaționale:

- Respectă principiile ACID (A- atomicitate, C- consistență, I- izolare, D -durabilitate).
- Posibilitatea de a programa într-un limbaj cu calcul paralel nativ.
- Nivelul de redundanță scăzut – propriu structurării relaționale – asigură în același timp un grad înalt de consistență, direct prin design.
- Sunt pregătite să opereze într-un context distribuit și scalabil.

Bazele de date orientate pe documente (colecții)

Bazele de date relaționale au avut, totuși, mereu un mare dezavantaj – nepotrivirea structurii de date la nivel de aplicație cu structura bazei de date relaționale. A fost mereu nevoie de un strat de traducere între structura tabelară și cea obiect-orientată la nivel de aplicație. Din dorința de a rezolva această problemă și a optimiza procesele de transfer, comunitatea ICT a dezvoltat o nouă tehnologie, bazele de date orientate pe documente, numite simplu și no-SQL care stochează datele în format de obiect. De cele mai multe ori acestea sunt stocate în format JSON (Java Script Object Notation) sau JSON binar. Unele din soluțiile consacrate în acest domeniu sunt MongoDB, Cassandra, Neo4J.

Servicii de stocare hibridă (SQL + noSQL)

După experimentarea sistemelor strict relaționale sau strict orientate pe documente s-a încercat integrarea lor pe mai multe straturi. O nouă alternativă de utilizare a acestor tehnologii a fost pusă la dispoziție și anume, baze de date relaționale, care, la nivel de coloană, permit stocarea de date în format nestrecurat (sub formă de colecție) [45]. Furnizorul de baze de date PostgreSQL în ultimele versiuni a făcut pași importanți în această direcție oferind o interfațare complexă între limbajul SQL și elementele de stocare în format nestructurat [46] [47].

1.3. Structuri de date centralizate și locale

La nivel abstract putem clasifica acest domeniu în două direcții distincte: *Interfețe de comunicații cu date structurate* și *Interfețe de comunicații cu date nestructurate ("schema-less")*.

1.3.1. Interfețe de comunicare cu date structurate

Odată ce Cloud-ul a început să fie adoptat în industria ICT, plus valoarea a rezultat ca urmare a dezvoltării, publicării și refolosirii de servicii. Astfel, standardele de structurare a datelor în contextul comunicării au devenit critice. De multe ori modul în care au fost concepute a fost hotărâtor pentru performanța unui sistem. Pentru interfațarea serviciilor în Cloud, cel mai des întâlnit mod de a comunica între aplicații este folosirea protocolului HTTP prin paradigma REST (REpresentational State Transfer – mesajele HTTP corespund, generic, cu tranzițiile în mașina algoritmică de stare a serviciului respectiv).

Pentru a trimite sau recepționa date prin intermediul REST se folosesc două standarde majore: JSON (Java Script Object Notation) / XML (Extensible Markup Language).

1.3.2. Interfețe de comunicare cu date nestructurate

În ultima perioadă, datorită emergenței tehnologiilor de tip Big Data, au început să apară tot mai multe arhitecturi în care părți din sistem se bazează pe interfețe ce acceptă date nestructurate. Astfel se pot dezvolta sisteme care pot să proceseze... "orice".

Acest gen de interfețe sunt regăsite în zone ale sistemului care oferă o interfațare, în general, de tip "input" pentru rutinele de tip Big Data. Deoarece acest API trebuie să aibă o rată de ingestie foarte mare, iar formatul datelor se poate schimba în timp nu este oportun să se valideze structura la ingestie.

1.4. Pregătirea datelor în contextul masivelor de date – Big Data Analytics

O altă direcție în care optimizare procesării datelor în Cloud este critică, este aceea pentru Big Data. Primul val de adoptare a Big Data a avut loc în companiile mari datorită costurilor mari de operare, dar în prezent tehnologia a început să aibă un cost care o face rentabilă și în companii de nivel mediu.

1.4.1. Map Reduce

Este un model de programare și o implementare asociată pentru procesarea și generarea seturilor mari de date cu un algoritm paralel distribuit pe un cluster Cloud. [49] Un program Map Reduce este alcătuit dintr-o procedură care inițial efectuează filtrarea și sortarea, apoi aplică o metodă de reducere (mult diferită de vechile tehnici de de-fragmentare / re-grupare).

1.4.2. Apache Hadoop

Este un software de tip sursă deschisă (open-source) care facilitează utilizarea unei rețele de mai multe computere pentru a rezolva probleme care implică cantități masive de date și de calcul. Acesta oferă un cadru software pentru stocarea și prelucrarea distribuită a datelor mari folosind modelul de programare Map Reduce [50].

1.4.3. Masivele de date - bază pentru Inteligența artificială

Inteligența artificială este un domeniu al științei calculatoarelor care își are începuturile cu mult înainte ca Big Data să se contureze, diferența fiind de aproximativ 50 de ani. În schimb, până la apariția posibilității de a lucra cu cantități enorme de date, inteligența artificială a avut aplicații foarte restrânse, fiind văzută mai mult ca subiect de cercetare, izolată la nivelul laboratoarelor și nu ca un domeniu care poate produce plusvaloare în economie, fiind prognozat să ajungă la o cotă de piață de 390 miliarde dolari în 2025 (conform grand view reseach [W-5]).

1.4.4. Servicii pentru managementul masivelor de date

Apache Spark și Databricks Spark reprezintă un framework open-source distribuit pentru generarea de aplicații ce se bazează pe cluster-computing. Acesta oferă o interfață pentru programarea clusterelor întregi cu paralelism implicit de date și toleranță la erori. Databricks este o companie fondată de creatorii inițiali ai aplicației Apache Spark și oferă un motor unificat de analiză care urmărește să ajute clienții la procesarea volumelor mari de date și operațiuni de Inteligență Artificială. Produsul oferit de Databricks are ca nucleu software-ul cardul Spark și este prezent în Cloud ca și un SaaS (Software ca și serviciu).

Această companie este fuziunea completă dintre Big Data și Inteligența Artificială, serviciul oferit făcând invizibilă pentru client granița dintre Cloud și AI, clientul beneficiind de o tehnologie integrată, de sine stătătoare.

1.5. Sumarul capitolului

Acest prim capitol introduce stadiul actual al principalelor concepte privitoare la Inteligența Artificială (I.A.) care stau la baza cercetărilor din această teză de doctorat. I.A. este clasificată pe baza etapelor de inovare, a modurilor de învățare și a algoritmilor, fiind analizați mai mulți algoritmi de învățare automată, unii dintre ei fiind aplicați și în capitolele aplicative ale tezei.

Cea de-a doua secțiune analizată este fluxul informațional în mediul distribuit – foarte important pentru I.A. "omniprezentă" – "pervazivă". În această secțiune se analizează modalitățile de procesare și stocare distribuită. Sunt introduse infrastructurile de procesare "Serverless computing" care vor fi .

Cea de-a patra secțiune se referă la modalitatea de structurare și definire a interfețelor de comunicare. Sunt descrise interfețele de comunicare cu date structurate în format JSON / XML, precum și metodele de comunicare cu date nestructurate și scenariile când aceste cazuri de comunicare sunt oportune.

Ultima parte a acestui capitol este rezervată "procesării în masă" - acest scenariu este unul tot mai întâlnit în mediul Cloud deoarece odată cu transpunerea a tot mai multe sisteme și activități în domeniul digital, volumele de date au devenit mari, iar pentru procesarea acestora este necesar să fie utilizate tehnici specifice.

2. Integrarea rețelelor de instrumentație în Cloud

Pentru aducerea inteligenței artificiale la nivelul instrumentației este foarte avantajos ca aceasta din urmă să facă parte dintr-o rețea. Apare astfel necesitatea partajării informației - punerea în comun a unor informații dintr-o zonă izolată a rețelei cu informațiile dintr-o altă zonă, fapt care reprezintă premisele unui sistem capabil să ofere inteligență în răspunsul pentru utilizator.

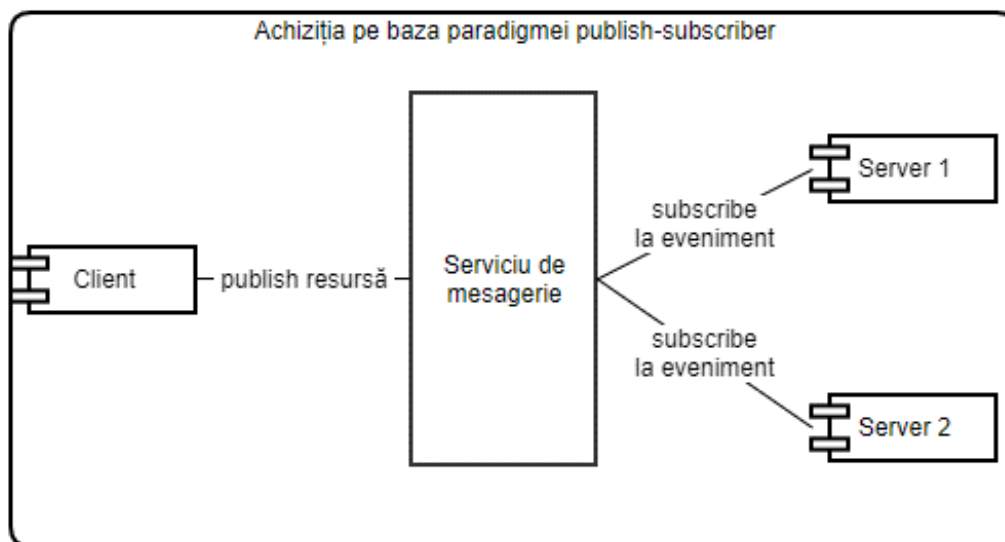
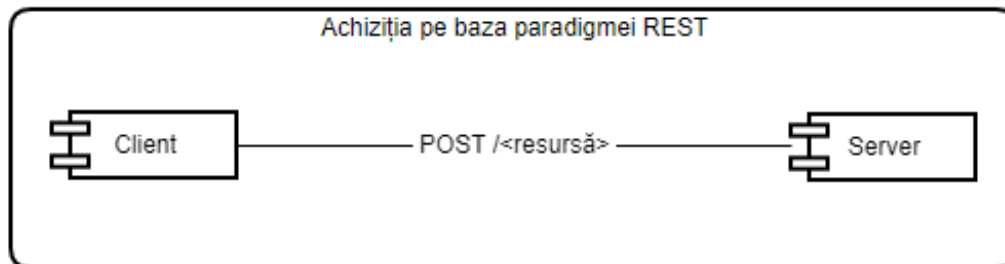
Primul pas este definirea a conceptului de "Cloud pentru instrumentație". În definirea infrastructurii vor fi considerate următoarele puncte:

- aspecte de configurare și comunicare la nivelul rețelei;
- definirea elementelor hardware care fac parte din Cloud-ul de instrumentație;
- definirea elementelor software pentru instrumentație;
- servere și rutine de servire;
- entități de stocare a datelor;
- aspecte de conectare și interconectare;

Un al pas fundamental este rețeaua Cloud pentru instrumentație care este necesar să poată conecta informațiile de la nivelurile inferioare cu unități de procesare la nivel înalt care au capacitatea de a centraliza și de a oferi o vedere de ansamblu asupra sistemului. În același timp trebuie considerate nevoile de descentralizare sau autonomie ale fiecărui element la nivelul său ("Edge" – "la marginea rețelei").

2.1. Achiziția de date în Cloud

Pentru a putea efectua operații "inteligente" la nivelul rețelei este nevoie de date, pentru că acestea oferă contextul necesar. Acesta este motivul pentru care achiziționarea datelor de la nivelul inferior al rețelei în mediul centralizat este un pas esențial care poate avea diferite abordări în funcție de cazul de utilizare.. Modelul bazat pe resurse face apel la protocolul HTTP și paradigma REST (REpresentational State Transfer) unde conectarea între client și server se face pe bază de resurse. Folosirea acestui model de achiziție poate funcționa foarte bine în multe cazuri de utilizare, mai ales în momentul în care datele sunt structurate, iar cel care achiziționează se așteaptă la un anumit format.



Comunicarea REST și publish-subscriber

2.1.1. Soluții pentru managementul datelor eterogene

O contribuție metodologică propusă în acest context: atunci când se proiectează un sistem complex, pentru a ține compromisul tehnologic la un nivel cât mai scăzut, fără să fie afectată funcționalitatea, arhitectura trebuie abordată "feliat" - fiecare "felie" fiind responsabilă de o sarcină specifică. Pentru această funcționalitate izolată se va ține cont de constrângerile existente, iar pe baza lor se va alege tehnologia de stocare adecvată. În acest mod se poate optimiza procesarea datelor în Cloud cu adevărat. Sunt prezentate în cele ce urmează două realizări științifice care au făcut uz de eterogenitatea datelor și modul prin care se gestionează aceste diferențe.

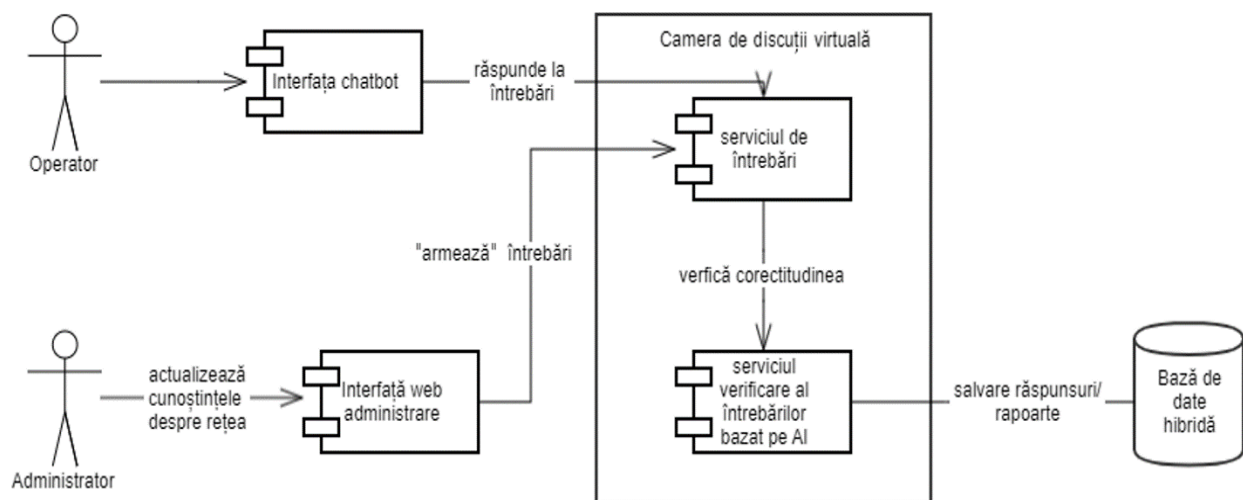
2.1.2. Modalități de lucru cu date eterogene

În cadrul cercetării din articolului [7.A.3] care exploatează interfațarea către utilizator prin terminale de tip *chats*-au folosit în același context date structurate, cât și date nestructurate. Această încărcare ce există la nivelul structurii a trebuit omogenizată până la nivelul utilizatorului și interacțiunii lui cu datele. Astfel utilizatorul va avea senzația că poate comunica cu un sistem predictiv care oferă informații într-un mod nestructurat.

Provocarea acestei cercetări a fost găsirea unei arhitecturi de comunicare între *utilizator* (operator) - *serviciul de management* (aflat și configurat în Cloud – menționat anterior) și *consumatorul de date* (rețeaua de instrumentație).

Scopul acestei arhitecturi este să permită în același timp o structurare a datelor în vederea creării de rapoarte (date structurate), dar totodată să ofere acel grad de *flexibilitate* întâlnită în comunicarea verbală sau scrisă tipică unui sistem conversațional.

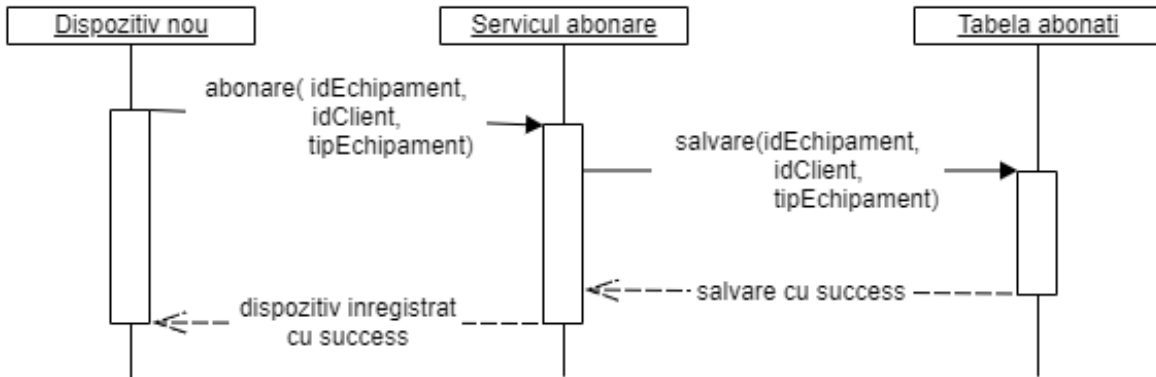
În urma cercetărilor am ajuns să exploatăm beneficiile ambelor tipuri de comunicare prin interfețe și anume am înglobat armonios un set de *interfețe de comunicare cu date nestructurate* cu un alt set de *interfețe de comunicare cu date structurate*



*Arhitectura de bază a sistemului conversațional rețea – operator/administrator
ce omogenizează interfețele cu date structurate și nestructurate*

2.1.3. Integrare cu soluții de procesare în masă

Odată cu aducerea datelor în Cloud și expansiunea rețelelor a avut loc o creștere a volumului de date și a diversității acestora. Astfel apare problema modului în care se pot integra fluxurile de date provenite de la marginea rețelei ținând cont de variabilitatea structurilor de date transmise de dispozitivele de la periferie. Acest gen de provocare apare în sisteme care necesită integrarea cu surse de date eterogene. Am abordat o provocare asemănătoare în cadrul articolului [7.A.4] care a avut ca obiect aducerea de date de la diverse contoare inteligente direct în Cloud. Originalitatea soluției de gestiune a unor dispozitive eterogene (fără a necesita alocarea de etichete Cloud „tokens”) a fost mecanismul unitar de „abonare” pe baza identificatorului unic MAC al dispozitivului.



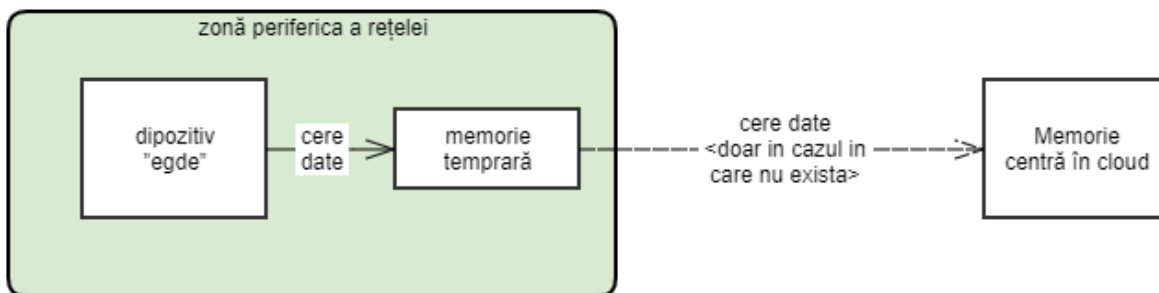
Secvența de abonare a unui serviciu

2.2. Edge computing – Cloud computing

Pentru a potența la maxim capabilitățile unei rețele instrumentale în Cloud este necesar să oferim un grad de flexibilitate dispozitivelor periferice. Pentru anumite operații, instrumentele de la marginea rețelei au nevoie de autonomie în procesarea informațiilor și executarea calculelor / transformărilor direct la marginea rețelei. O arhitectură Cloud trebuie să poată oferi această tranziție a procesului din Cloud spre margine sau în sens invers într-un mod facil și transparent.

1.2.1. Stocarea temporară aproape de proces

Prin stocarea aproape de proces se înțelege capacitatea unui sistem Cloud de a-și optimiza comunicarea în așa fel încât informațiile să nu fie păstrate doar central, ci să înceapă a fi răspândite prin rețea, respectiv spre marginea ei.

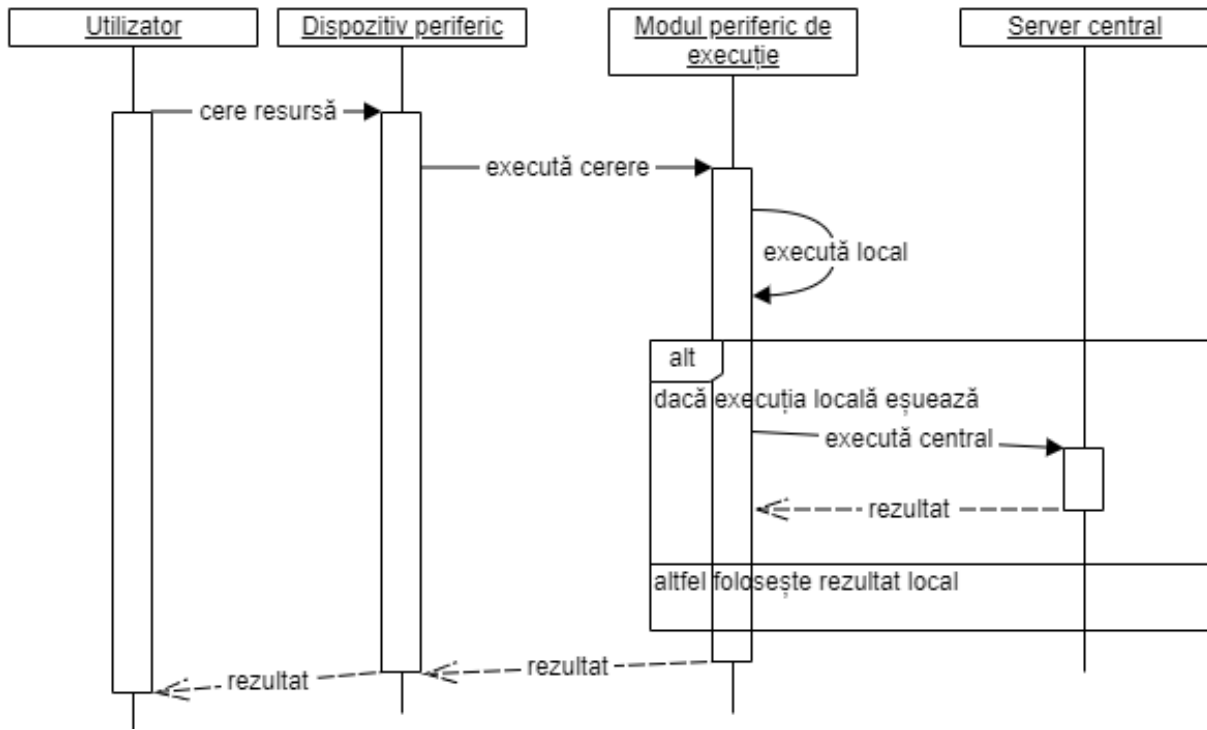


Relația Edge – memorie centrală Cloud

2.2.2. Procesare la marginea rețelei

Prezența datelor în proximitatea proceselor și a dispozitivelor de la marginea rețelei rezolvă doar o parte a problemei, pentru a avea "Edge computing" este nevoie ca dispozitivul de la margine să fie pregătit din punct de vedere hardware și software să execute rutine local.

În momentul de față există produsele din gama Raspberry Pi, Intel Movidius, Google Coral – specializat pe execuția de rutine tipice inteligenței artificiale la marginea rețelei și multe altele.



Diagramă de secvență de procesare Edge

2.3. Implementarea unui Cloud de instrumentație

Am realizat o extensie "Cloud pentru Instrumentație" a sistemului de 6 "server blades" Dell 1850 "Power Edge" de la Departamentul de Electronică și Calculatoare al Universității "Transilvania".

2.4. Sumarul capitolului

Acest capitol furnizează o analiză a componentelor de infrastructură și a tehnicilor necesare pentru dezvoltarea unui mediu Cloud care servește ca și infrastructură de dezvoltare a serviciilor de instrumentație (în paradigma "instrumentația ca serviciu" – o extensie a conceptului de "instrumentație virtuală"). În acest context provocările actuale sunt referitoare la volumele mari de date eterogene și la tehnicile de calcul și procesare "la marginea rețelei" (la "nivelul Edge"). În acest capitol am prezentat premisele introducerii inteligenței artificiale la "nivelul Rețea" (de instrumentație).

Achiziția de date este un aspect important, tratat din diferite perspective: pe de-o parte sunt tratate paradigmele comunicării: sincron prin "strângeri de mână" ("handshakes") HTTP sau asincron folosind

3. Optimizarea relației client server în rețele de instrumentație

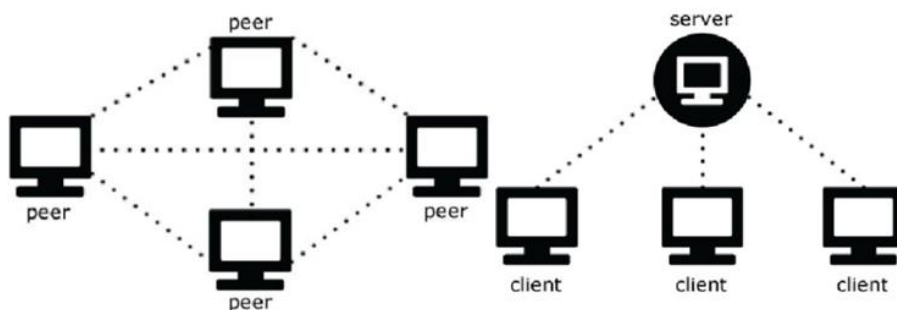
Acest capitol prezintă tehnologiile și arhitecturile existente în momentul de față pentru îmbunătățirea managementului stărilor și proceselor de alocare de resurse de la server către clientul instrumental. Pentru a se realiza această optimizare, în secțiunile ce urmează sunt prezentate diferite aspecte tehnologice ale internetului și rețelelor care optimizează sau îngreunează comunicarea între client și server. Astfel, am abordat aspectele arhitecturale ale relației între client și server, modurile în care cele două părți pot comunica și care sunt implicațiile infrastructurii Cloud în perspectiva optimizării.

3.1. Noi paradigme de optimizare a relație client server

Astfel, de-a lungul timpului, au evoluat două topologii importante de comunicare în sisteme distribuite:

- Topologiile de comunicare de la nod la nod – “peer to peer”;
- Topologiile de comunicare client -server .

Diferența dintre aceste două topologii este contextul pe care îl oferă: comunicarea *peer to peer* [66] oferă un context *descentralizat*, iar comunicarea *client - server* oferă un context centralizat.



Conexiunea într-o rețea peer-to-peer vs conexiunea client-server

3.1.1. Orientarea pe servicii a arhitecturii client server

Arhitecturile de tip client – server [69][70] s-au dovedit, în timp, de mare utilitate pentru construcția diferitelor sisteme informatice. În prezent, marea majoritate a sistemelor cu care interacționăm în mediul digital sunt bazate pe acest tip de arhitectură. Comunicarea între aplicația client și cea server se face pe baza unui contract, care în termeni tehnici este numit API (Application Program Interface). Acesta definește interfața prin care cele două entități au agreeat să comunice. În peste 90% din cazuri, aceste interfețe sunt implementate folosind protocolul HTTP (Hyper Text Transfer Protocol).

3.1.2. Moduri de apelare a serviciilor

O retrospectivă asupra arhitecturilor dedicate sistemelor ce funcționează în internet, indică un volum considerabil de muncă depus în găsirea unei soluții optime pentru managementul stărilor din interiorul aplicațiilor Internet. Câteva dintre paradigmele de comunicare între două noduri din rețea sunt următoarele:

- Proceduri apelate la distanță - RCP (Remote Procedure Calls);
- Comunicare prin paradigma emițător–abonat (publish-subscribe);
- Comunicare prin cozi de mesaje;
- Comunicare prin paradigma SOAP (Simple Object Access Protocol);
- Comunicare prin paradigma REST (Representational State Transfer).

În momentul de față, avem două paradigme importante de comunicare:

- comunicare cu stare – "statefull";
- comunicarea fără stare – "stateless".

3.1.3. Replicarea resurselor aproape de utilizator

Până la răspândirea serviciilor publice ca și Centre de Date care pot oferi acoperire la nivel global, cu o proximitate scăzută față de utilizator, aplicațiile și serviciile web erau în general rulate pe sisteme proprietare ("on premise"). După ce mediul Cloud a devenit *public*, au început să apară multe servicii și funcționalități prin care comunicarea dintre client și server a fost încontinuu optimizată, astfel încât utilizatorul final să aibă o experiență plăcută și constantă indiferent de locația geografică, ceea ce oferă acestei aplicații transparență geografică. Mai concret este analizată în continuare modalitatea prin care serviciile din Cloud se folosesc de replicarea resurselor aproape de utilizator pentru a oferi îmbunătăți QoE pentru utilizator, indiferent de unde accesează resursa.

3.1.4. Infrastructuri cu disponibilitate neîntreruptă

O problemă veche a infrastructurilor tradiționale a fost asigurarea funcționării neîntrerupte a aplicațiilor. Practic, în momentul în care era nevoie să funcționeze mai mult de o aplicație, atunci se întâmpla o defecțiune de ordin fizic, se defecta serverul sau hard-disk-ul sau rețeaua se întrerupea. Aceste probleme de ordin fizic afectează de cele mai multe ori încrederea în sistemul online. Pentru companiile care oferă servicii de găzduire sau pentru centrele de date ale companiilor private acest aspect este o provocare continuă. Un sistem de top, spre care tind toți marii furnizori de servicii Cloud (Amazon, IBM, Azure, Google) are o disponibilitate de 99,999% sau chiar 99,9999%, acest lucru însemnând o indisponibilitate de 5,26 minute, respectiv de 26,30 secunde pe durata unui an.

3.2. Introducerea I.A. în rețelele inteligente

“Rețelele Inteligente” – IN (“Intelligent Networks”) sunt construite pentru “a liberaliza accesul” la ciclul de viață al serviciilor, de către înșiși beneficiarii acestora – în concordanță cu abordarea Agilă din ultimii ani, o implicare în specificarea funcțională, definirea / adaptarea modelelor de servicii, dezvoltarea lor, testarea modulară, testarea de integrare, desfășurarea / instalarea / configurarea și apoi exploatarea lor într-un model business de succes [88] [89].

În această abordare “de sus în jos” (“top down”), nivelele IN sunt:

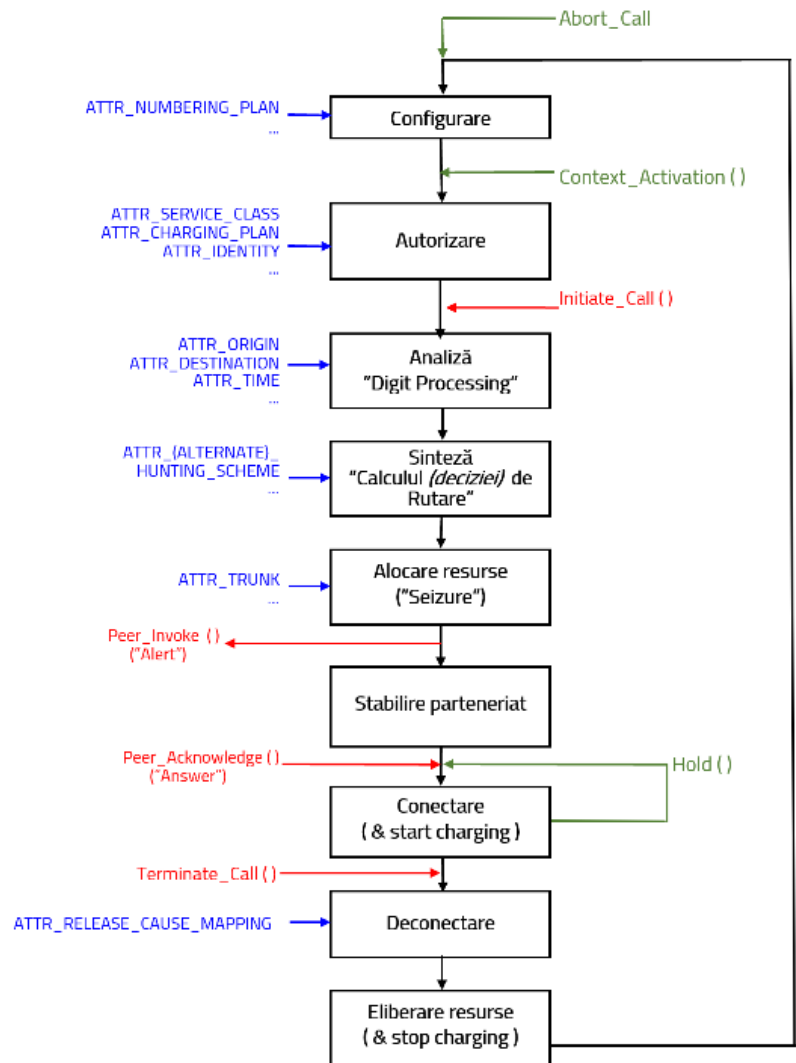
- SCE (Service Creation Environment)
- SMP (Service Management Point)
- SCP (Service Control Point)
- SSP (Service Switching Point)

Pentru a ilustra modul în care am contribuit la introducerea inteligenței artificiale la toate aceste nivele, am luat în considerare, un apel la modul generic, ca *apel la resurse* (umane și/sau instrumentale).

De asemenea, am acordat o deosebită importanță modelelor de *control prin stare* în rețelele inteligente.

Un serviciu e un automat:

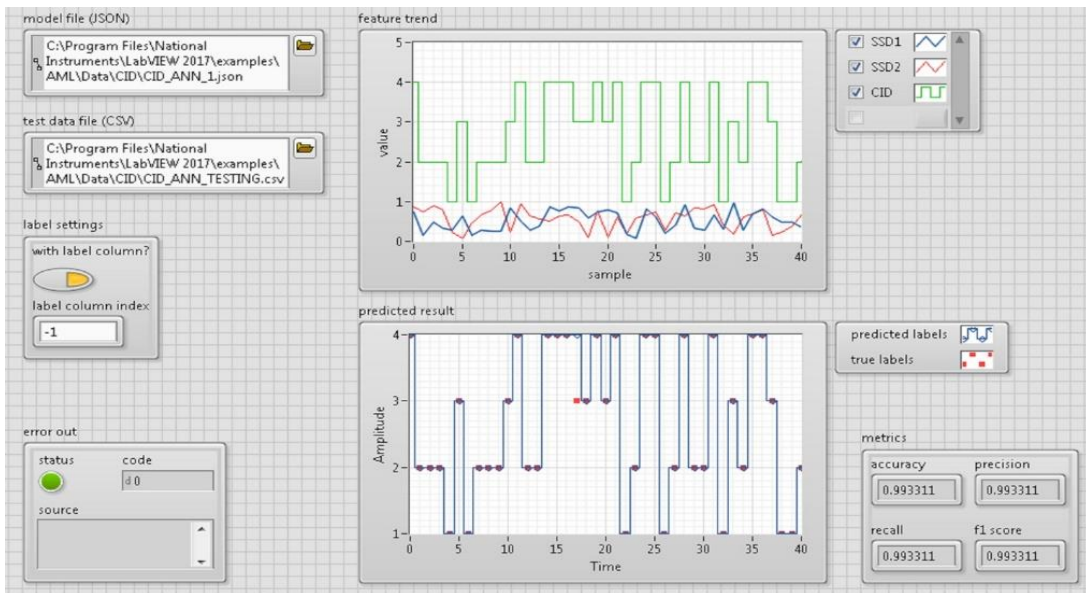
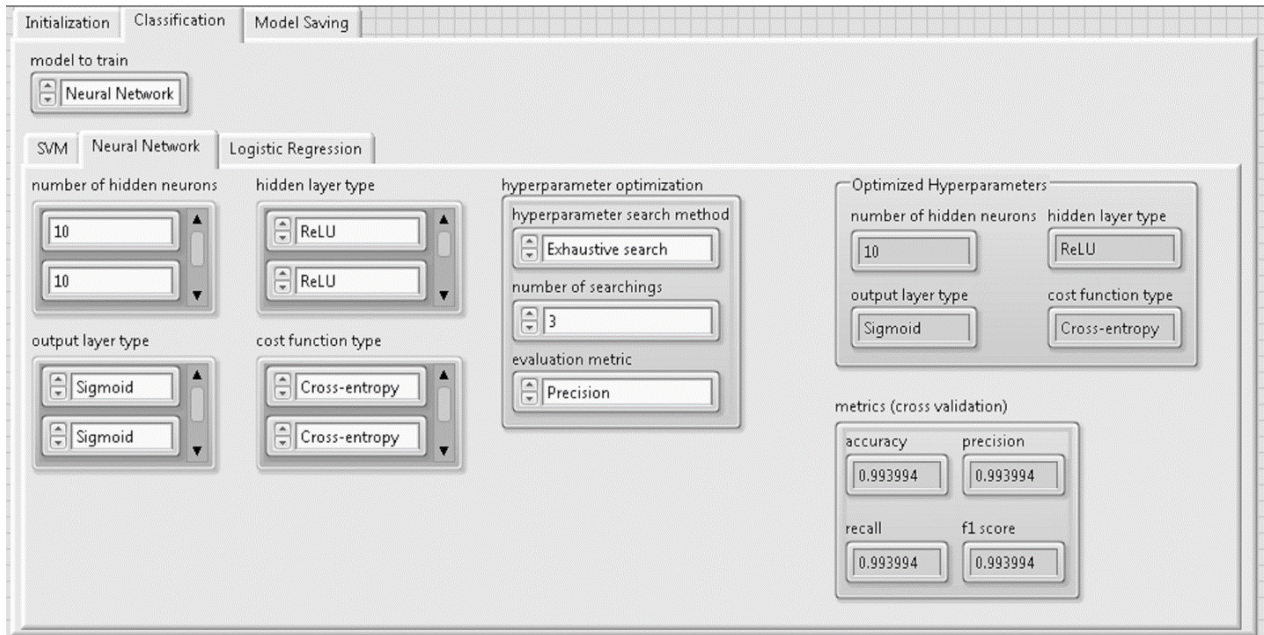
- cu “stări” de calcul reprezentate ca “blocuri constructive pentru servicii” (SBB)
- cu “operanzi” - denumit SSD (Service Support Data)
- cu “tranziții”, transmiterea *deciziilor* aferente fiind reprezentată prin *semnalizările* – numite în Introducere CID (Call Instance Data)



Schema logică a automatului

3.3. Inteligența artificială în decizia asupra tranzițiilor optime

Am construit un instrument virtual LabVIEW de antrenare a unui clasificator cu două intrări SSD1 și SSD2 normalizate (între 0 și 1) și decizie asupra tranzițiilor optime (câte 4 stări, codificate simplu, binar, după cum $SSD1$ și $SSD2 < 0,5$ sau $\geq 0,5$). Am utilizat o facilitate performantă de preselectie a modelului (ANN), de *dimensionarea automată* acestuia și găsirea celor mai potrivite funcții de activare pe anumite criterii (funcții cost ale aproximărilor succesive) – în paradigma *MaaS (Model as a Service)*



3.4. Partajarea inteligenței artificiale între Cloud și Edge - Model as a Service

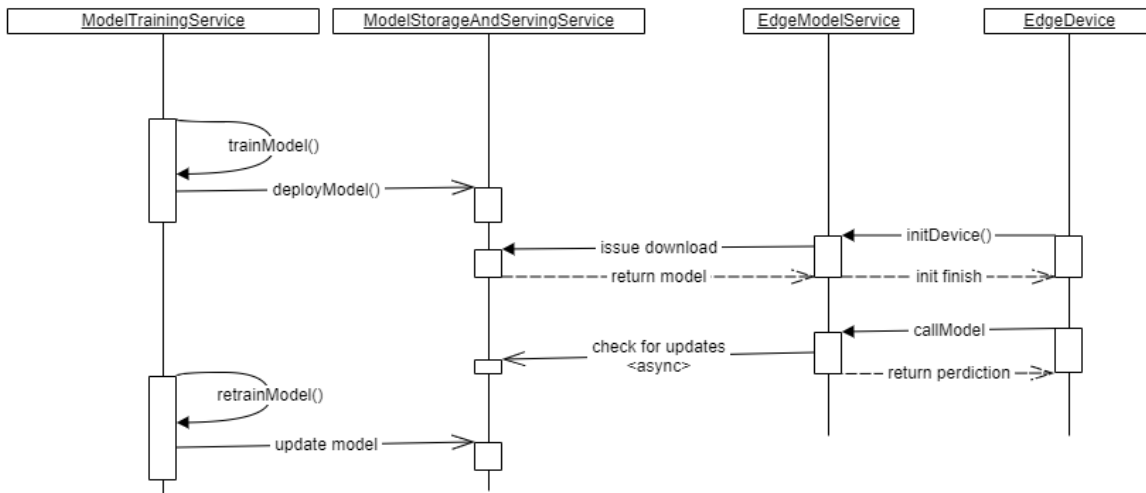
În sistemele distribuite ale următorului deceniu, inteligența artificială va deveni "status quo", se va răspândii din mediile Cloud, centralizate, către marginile rețelei și dispozitivele aflate acolo. Acest lucru se va face prin oferirea modelor AI ca și serviciu către periferia rețelei, iar modelul AI (coeficienții - rezultatul antrenării) este coborât, către "marginea rețelei" (la nivelul Edge) pentru a efectua operațiile și deciziile inteligente aproape de date. Implicit sistemele care vor aborda o asemenea arhitectură vor beneficia de calități precum: Latență redusă în momentul deciziei; Independența dispozitivului periferic față de zona centrală de procesare a sistemului – Cloud.

Noutatea pe care o propunem în aceasta teză este dinamizarea acestui model de lucru, prin propunerea unei specificații de comunicare și execuție, în care coeficienții unui model AI antrenat sunt coborâți în mod constant la periferia rețelei, stocați și folosiți direct de către dispozitivele periferice.

Calitățile sistemelor ce folosesc MaaS:

- Putere mare de calcul în generarea modelelor.
- Capabilitatea de centralizare a unor volume mari de date.
- Ajustarea și rafinarea modelului cu o perspectivă de ansamblu asupra datelor – Big Data.
- Decuplarea dispozitivului de la marginea rețelei de centrală pentru anumite evenimente.
- Capabilitatea dispozitivelor periferice de a lua independent decizii inteligente.

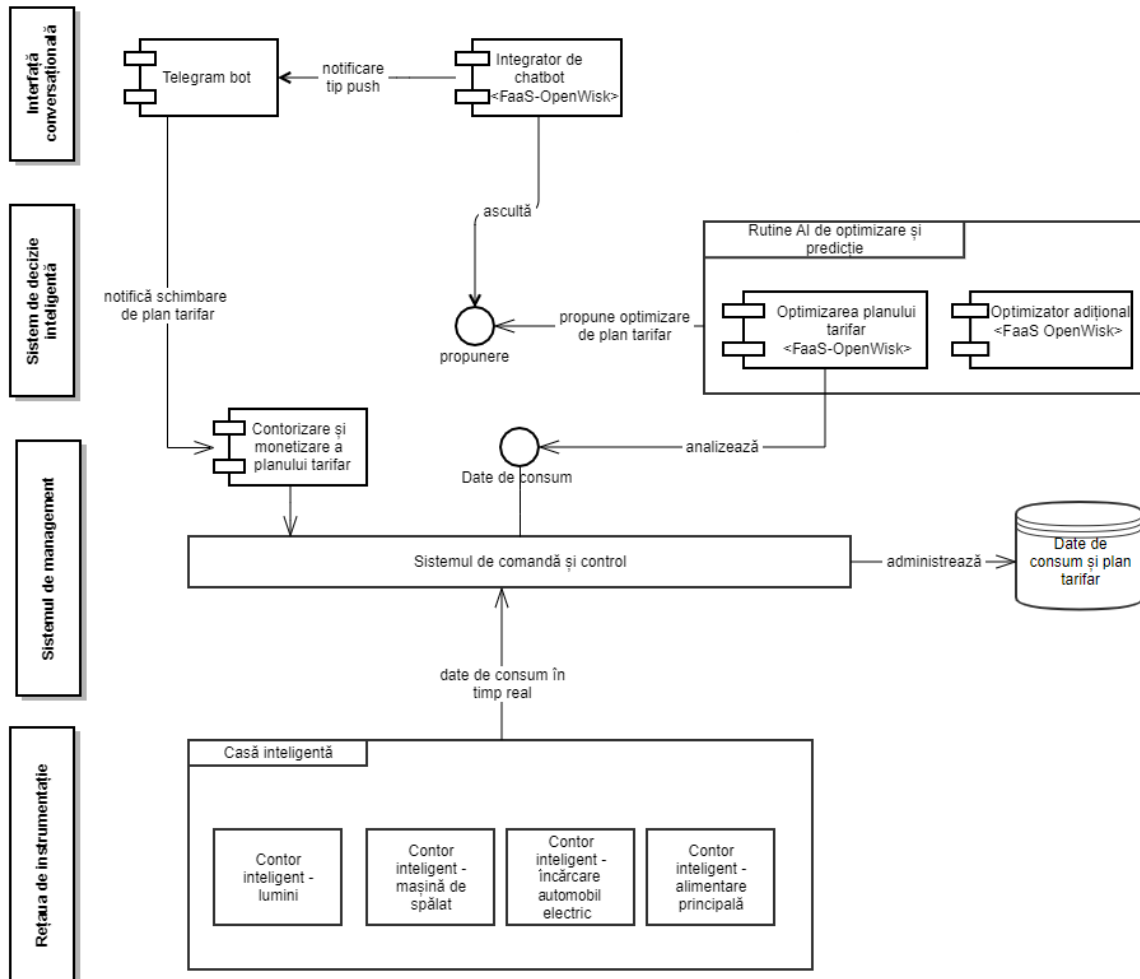
Modul de funcționare al unui asemenea model este prezentat în diagrama ce urmează [91].



3.5. Optimizarea continuă a rutinelor de servire a întreprinderilor

Un alt scenariu în care inteligența artificială este exploatată la nivelul infrastructurii Cloud este *oferirea unui răspuns mereu optimizat* clientului în situația în care acesta face o cerere de servire.

În acest concept serviciul se folosește de experiența din trecut pentru a decide asupra stării viitoare. În acest fel propunem aducerea deciziei bazate pe rutine inteligente la nivel de Service Building Block (SSB). Pentru a demonstra modul în care un asemenea concept poate să opereze am exemplificat niște scenarii unde acest concept este aplicabil, ca apoi să propun un demonstrator ce se folosește de infrastructura Cloud IBM Bluemix și tehnologia OpenWisk.



Arhitectura de sistem

3.5.1. Implementarea mecanismului de optimizare

Am introdus la nivelul rutinelor de "servire a întreruperilor" conceptul de FaaS ("Function as a Service") implementat cu IBM/Apache "OpenWhisk".

- FaaS aparține categoriei mai largi "serverless computing" (din perechea client-server, acesta din urmă e în Cloud), în particular, pentru un eveniment, o "întrerupere" (a unui proces generic de funcționare la nivel sistem), rutina de "servire a întreruperii" este adusă din Cloud, mecanism extrem de important pentru implementarea inteligenței artificiale în rețelele de instrumentație.

3.5.2. Implementare prototipului

Am apelat la unele din cele mai noi tehnologii Cloud de procesare și stocare a datelor care, împreună cu integrarea tehnicilor de inteligență artificială, permit o interacțiune versatilă cu utilizatorii, un set de servicii noi care oferă posibilitatea de reacție în timp real și, implicit, reduceri de costuri. Din punct de vedere al dezvoltatorului de servicii, furnizarea unei arhitecturi native direct în Cloud oferă un grad ridicat de flexibilitate când devin necesare noi creșteri de performanță.

Implementarea FaaS, care rulează într-un mediu "server-agnostic" (serverless), s-a realizat în Cloud IBM Bluemix cu ajutorul tehnologiei tip sursă deschisă OpenWhisk ce se află sub umbrela Apache.

Am implementat *acumulatorul de date* ca serviciu *node.js* care rulează într-o mașină virtuală aflată în Cloud-ul IBM Bluemix. Acest serviciu colectează informația de la senzorii care furnizează datele de consum. Acestea sunt apoi adaptate la modelul intern de date și salvate în baza de date *noSQL* Cloudant.

3.6. Sumarul capitolului

Acest capitol a abordat relația dintre client și server în contextul Cloud, cu scopul optimizării acestora. S-a plecat de la analiza diferitelor modele de comunicare între client și server (client-server și peer-to-peer), ca apoi să se prezinte topologia client-server care este mai utilizată în contextul diferitelor demonstratoare din această lucrare. Pentru acest tip de arhitectură este mai întâi analizat modul detaliat de funcționare pentru a crea mai departe premisele necesare optimizării comunicațiilor bazate pe conceptul "arhitecturilor elastice". La nivel de protocol de comunicare sunt prezentate diferențiat modurile de comunicare sincron și asincron împreună cu aspectele lor specifice în perspectiva îmbunătățirii sau îngreunării traficului. Tot în acest context sunt analizate situațiile în care este indicat să fie utilizat unul sau celălalt tip de comunicare, totul în contextul maximizării experienței de utilizare – QoE.

Am propus un concept de "coborâre" a modelului antrenat din Cloud ("din nori pe pământ") și rularea predicțiilor la nivelul de jos (Edge). Aceasta, combinată cu un sistem de memorie temporară, va readuce un grad sporit de independență la nivelul dispozitivelor periferice. Soluțiile prezentate concretizează Inteligența Artificială ca și optimizator de tranziții – am ilustrat, printr-un clasificator cu rețea neurală decizia asupra CID (Call Instance Data), decizia asupra stării următoare recomandate.

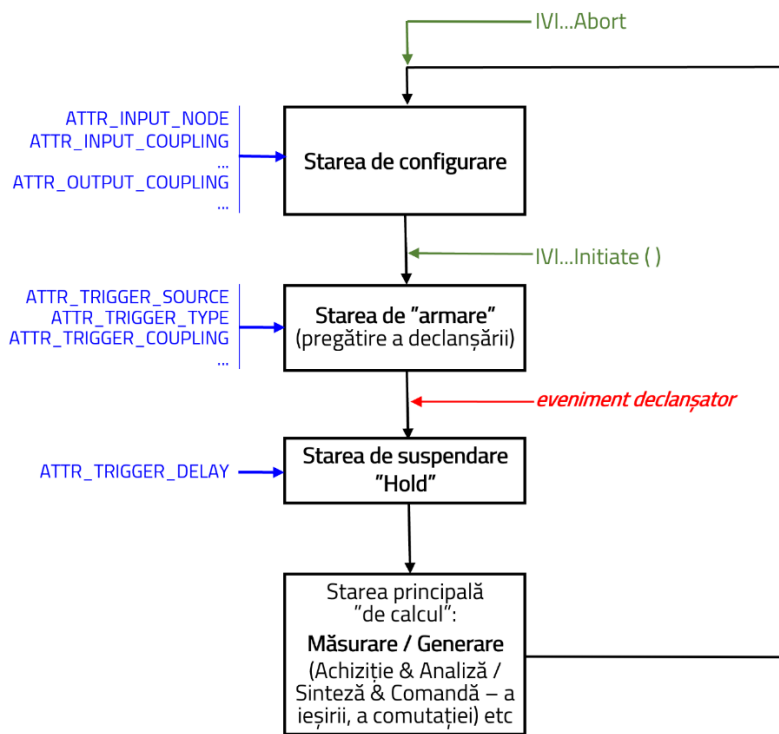
În aceeași ordine de idei acest capitol expune și un studiu de caz prin care, cu ajutorul tehnicilor Cloud și infrastructurii din IBM Bluemix, am ilustrat modul prin care inteligența artificială poate fi introdusă ca strat mediator între client și rutina de servire, pentru a oferi un *răspuns continuu optimizat* pentru client. Așadar, pe lângă utilizarea I.A. în alegerea optimală – *optimizarea deciziei asupra rutinei de servire* (menționată mai sus) *am optimizat și rutina însăși* ("conținutul" – programul rutinei de servire).

4. Rețele inteligente de instrumentație

Capitolul ce urmează tratează modul de aplicare a inteligenței artificiale la nivelul rețelelor inteligente pentru a îmbogăți funcționalitatea la nivelul instrumentației. La nivel conceptual, primul paragraf prezintă standardul de controlabilitate a instrumentației (IVI – Interchangeable Virtual Instrumentation) ca mașină algoritmică de stare ce particularizează modelul de control pe bază de evenimente prezentat în capitolul 3 (cu cele două modalități de adăugare I.A. – la nivel CID și la nivel SSD. Paragrafele următoare consideră două abordări, gradual: prima utilizează Cloud pentru modelare, iar calculul predicției se face local, direct de către logica programată a instrumentului care are stocat în memoria proprie modelul antrenat, cea de a doua fiind centralizată, în care modelele cu inteligența artificială sunt antrenate în Cloud iar rezultatul predicției e dat tot din Cloud.

4.1. IVI - Modelul simplificat de control prin stare

Acest model poate fi aplicat cu succes în optimizarea continuă a rutinelor ce servesc un răspuns continuu optimizat. Conform principiului 20/80 al lui Pareto [94] (formulat de mai bine de un secol – principiu conform căruia aproximativ 20% din cauze determină, în general, aproximativ 80% din efecte) am prezentat într-o perspectivă unificată modelul *comportamental* IVI pentru "cele 20 procente din instrumentație care se folosesc 80% din timp".



4.2. Senzori virtuali cu rețele neurale

În contextul acestei cercetări atenția a fost concentrată asupra modului în care se poate utiliza inteligența artificială la nivel Edge – direct aplicată rețelelor de instrumentație. În următoarele secțiuni sunt descrise contextul problemei și analiza mai multor soluții bazate pe rețele neurale care ajută la rezolvarea problemei descrise.

4.2.1. Sensor virtual cu inteligență artificială pentru un stand de testare a motoarelor

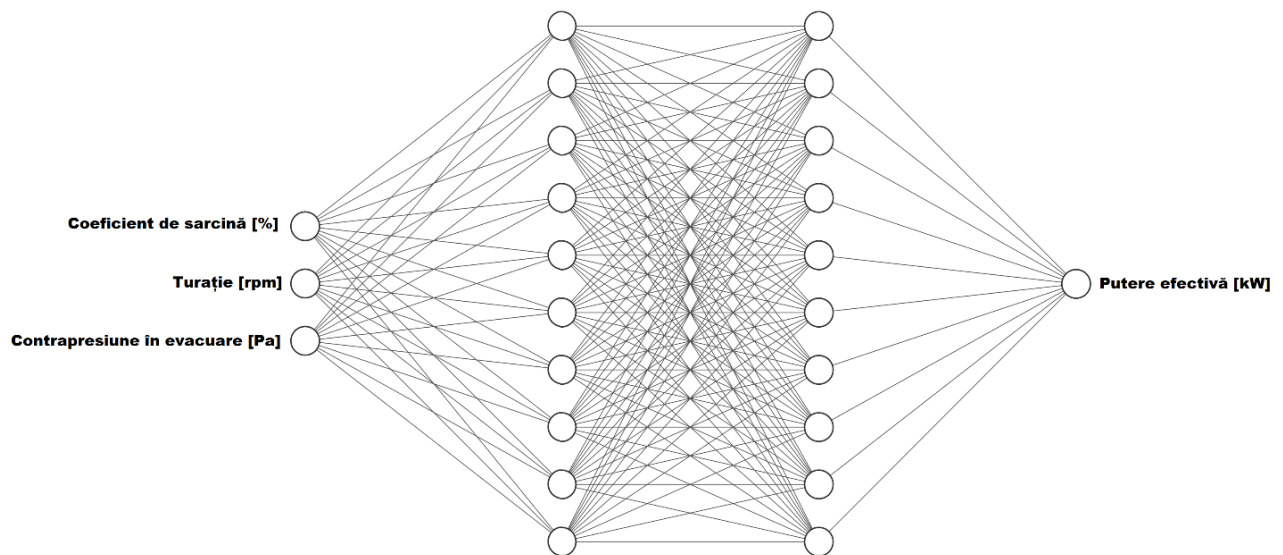
Studiul de caz se referă la instrumentația specifică motoarelor cu ardere internă (MAI) cu aprindere prin compresie (MAC) - Diesel, a căror funcționare impune un proces rapid de schimbare a gazelor și de evacuare a acestora din cilindru.

În acest studiu, procesul este evaluat prin căderea de presiune a gazelor arse pe traseul de evacuare (Δp).

Această mărime însumează rezistențele gazo-dinamice pentru fiecare componentă din instalația de evacuare.

4.2.2. Antrenarea rețelei neurale

Arhitectura aleasă [3 x 10 x 10 x 1] este prezentată mai jos și corespunde studiului de caz detaliat anterior – un stand de testare a motoarelor Diesel:



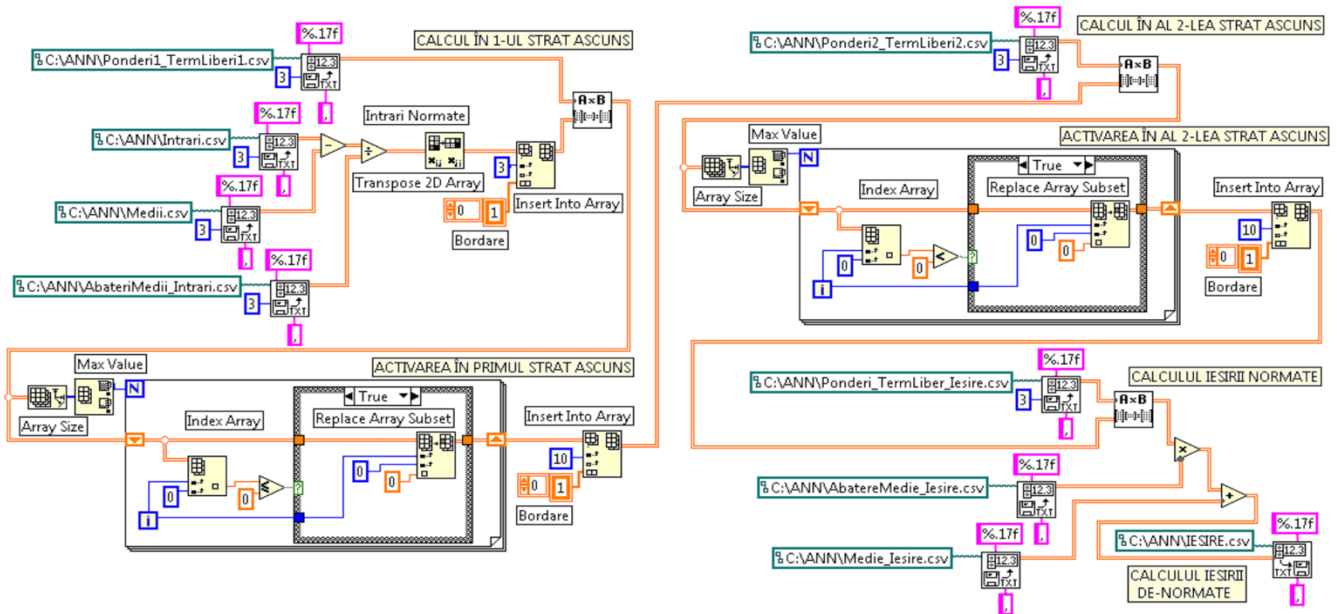
4.2.3. Rularea modelului antrenat

În contextul rulării au fost abordate două posibilități, prima este cea centralizată cu rularea modelului ca și program LabView, iar a doua opțiune este rularea lui la nivelul edge pe un echipament arduino.

Rularea centralizată în LabView:

Programul LabVIEW de mai jos are ca intrare fișierele .CSV (Comma Separated Values) ce corespund intrării propriu-zise (3 valori *ne-normate*) și fișierelor text sus-menționate (data_mean.txt, data_std_test și, respectiv, hidden1_weight.txt *bordat cu* hidden1_bias.txt, hidden2_weight.txt *bordat cu* hidden2_bias.txt, out_weight.txt *bordat cu* out_bias.txt) și furnizează ieșirea *de-normată* tot într-un fișier CSV. Un avantaj al acestei implementări e *scalabilitatea la diverse dimensiuni ale celor două straturi ascunse (pot avea și numere diferite de neuroni)*.

Se observă că iterațiile de activare se desfășoară de un număr de ori calculat direct măsurând dimensiunea seturilor de coeficienți introduse în calcul.



Programul LabView de rulare a modelului

Rularea ca și Edge A.I.

Întrucât *statistica intrărilor se modifică* cu constante de timp având câteva ordine de mărime peste constantele de timp ale proceselor industriale monitorizate de rețeaua de instrumentație, putem programa în mod extrem de compact *Edge A.I.* cu resurse computaționale minimale: coeficienții ANN precum și media și abaterea medie pătratică (deviația standard) se pot înscrie direct în codul de dimensiuni minime care este încorporat ("embedded") în micro-sistemul cu logică programată.

Am dezvoltat un cod *foarte compact* (în vederea implementării pe *hardware minimal*) al *senzorului virtual*, cod exemplificat în cele ce urmează:

```

package ro.unitbv;
import java.util.Random;
import org.sintef.jarduino.InvalidPinTypeException;
import org.sintef.jarduino.JArduino;
import org.sintef.jarduino.JArduino;
public class EdgeAI extends JArduino {
    private WeightReader wReader = new LocalWeightReader();
    // Read the weights for hidden layer1 already containing the additional bias
    // parameters
    double[][] layer1Weights = wReader.getWeightsHiddenLayer1();
    // Read the weights for hidden layer2 already containing the additional bias
    // parameters
    double[][] layer2Weights = wReader.getWeightsHiddenLayer2();
    // Read the weights for output already containing the additional bias parameters
    double[][] outputLayerWeights = wReader.getWeightsOutputLayer();
    Statistic[] statistics = wReader.getParameterStatistic();
    @Override
    protected void loop() throws InvalidPinTypeException {
        // Run prediction using randomized inputs between range [0, 4095]
        double[][] inputParams = getRandomizedInputParams();
        // Print normalized inputs
        System.out.println("Normalized");
        System.out.println(inputParams[0][0]);
        System.out.println(inputParams[1][0]);
        System.out.println(inputParams[2][0]);
        // Print denormalized inputs
        System.out.println("Denormalized");
        System.out.println((inputParams[0][0] * statistics[0].std) + statistics[0].mean);
        System.out.println((inputParams[1][0] * statistics[1].std) + statistics[1].mean);
        System.out.println((inputParams[2][0] * statistics[2].std) + statistics[2].mean);
        double prediction = runPrediction(inputParams);
        // Print normalized prediction
        System.out.println("Normalized output");
        System.out.println(prediction);
        // Print denormalized prediction
        System.out.println("Denormalized output");
        System.out.println(prediction * statistics[3].std + statistics[3].mean);
    }

    double runPrediction(double[][] inputParams) {
        // Run activations on first layer
        double[][] outputLayer1 = multiply(inputParams, layer1Weights);
        applyRelu(outputLayer1);
        outputLayer1 = extendRowBy1(outputLayer1);
        outputLayer1[outputLayer1.length-1][0]=1;
        // Run Activations on second layer
        double[][] outputLayer2 = multiply(outputLayer1, layer2Weights);
        applyRelu(outputLayer2);
        outputLayer2 = extendRowBy1(outputLayer2);
        outputLayer2[outputLayer2.length-1][0]=1;
        double predictionResult = multiply(outputLayer2, outputLayerWeights)[0][0];
        return predictionResult;
    }

    private void applyRelu(double activations[][]){
        for (int row = 0; row < activations.length; row++)
            for (int col = 0; col < activations[row].length; col++)
                activations[row][col] = activations[row][col] < 0 ? 0 : activations[row][col];
    }

    private double[][] multiply(double[][] input, double[][] currentLayerWeights) {
        double[][] result = new double[currentLayerWeights.length][input[0].length];
        for (int row = 0; row < result.length; row++)
            for (int col = 0; col < result[row].length; col++)
                result[row][col] = multiplyMatricesCell(input, currentLayerWeights, row, col);
        return result;
    }

    private double[][] extendRowBy1(double[][] mat){
        double[][] result = new double[mat.length+1][mat[0].length];
        for(int i =0; i<mat.length;i++)
            for(int j=0; j<mat[i].length;j++)
                result[i][j]= mat[i][j];
        return result;
    }

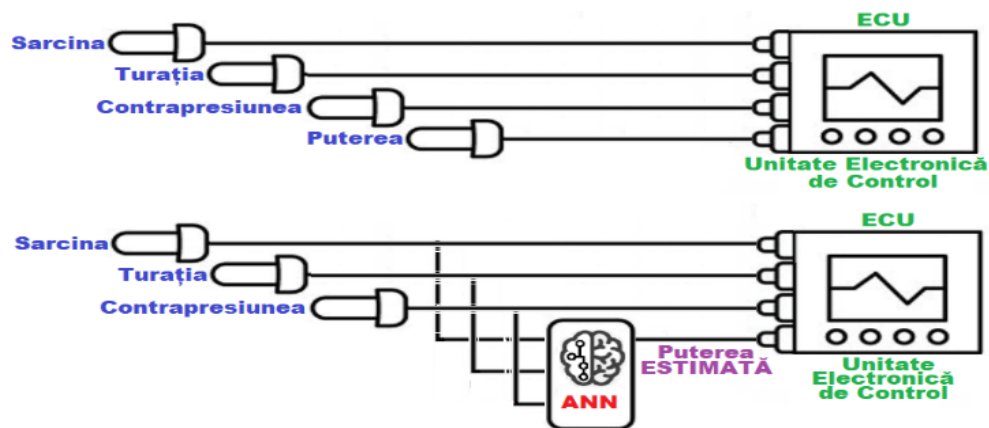
    private double multiplyMatricesCell(double[][] mat1, double[][] mat2, int row, int col) {
        double cell = 0;
        for (int i = 0; i < mat1.length; i++)
            cell += mat1[i][col] * mat2[row][i];
        return cell;
    }

    private double[][] getRandomizedInputParams() {
        double[][] inputParam = new double[4][1];
        inputParam[0][0] = (double)(new Random().nextInt(4095) - 2048) / 2048; // PE
        inputParam[1][0] = (double)(new Random().nextInt(4095) - 2048) / 2048; // RPM
        inputParam[2][0] = (double)(new Random().nextInt(4095) - 2048) / 2048; // delta_p
        inputParam[3][0] = 1; // Bias
        return inputParam;
    }

    @Override
    protected void setup() throws InvalidPinTypeException {}
}

```

Imaginea de mai jos prezintă modul în care codul de predicție dezvoltat mai sus poate fi integrat la nivelul ECU ca și senzor virtual.

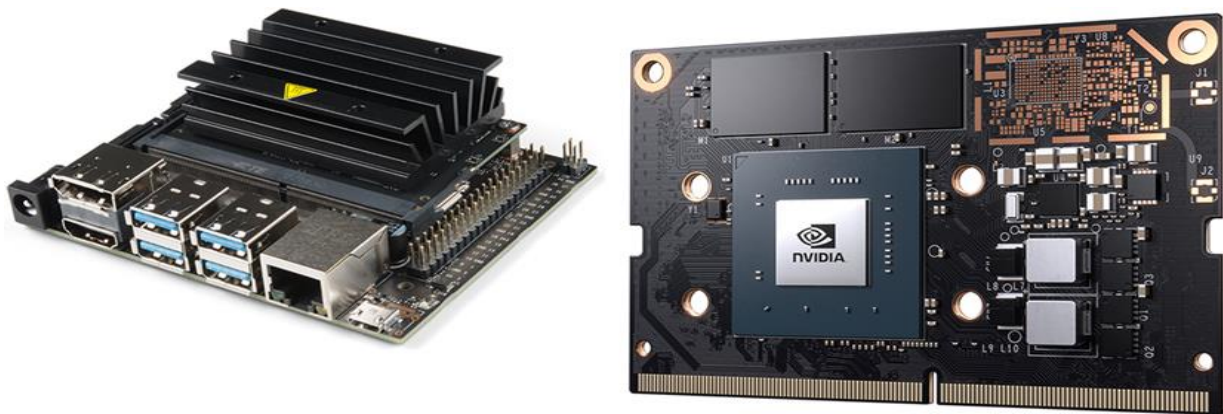


4.2.4. Alternative de implementare a rețelei neurale

În contextul acestui studiu de caz au fost analizate mai multe alternative de implementare.

Printre acestea amintim:

- *Implementarea Weka* (Waikato Environment for Knowledge Analysis) este un software oferit gratuit de către Universitatea din Waikato, Noua Zeelandă.
- *Implementarea MATLAB* (MathWorks™) cu avantajele: posibilității de a rula în Cloud și de a genera modele (similare GAN) care sunt puse în competiție în faza de *training*
- *Implementarea Keras* – se referă la folosirea rețelelor neurale cu ajutorul programului cadru Keras ce oferă o interfață simplificată peste backend-ul de Tensorflow.
- *Implementarea metodelor XGBoost* - metodele eXtended Gradient Boosting (XGBoost) [W-12] sunt orientate pe optimizarea iterativă a clasificatoarelor cu I.A., prin *combinarea "forțată"* ("impusă" – "boosted") – fără invalidare – a unor clasificatori aparent mai slabi.
- *Micro-sistem Edge cu accelerare hardware pentru Inteligența Artificială* - are la bază cea mai spectaculoasă soluție accesibilă comercial cu hardware dedicat pentru accelerarea calculelor de inteligență artificială, kit-ul de dezvoltare NVIDIA Jetson Nano (V3).



4.3. Detecția spectrală potențată de inteligență artificială centralizată

SDR (Software-Defined Radio) – *Radio Definite Software* – reprezintă o combinație modernă de sub-sisteme de bandă largă ("capete de RF – radio frecvență, cu antene active, amplificatoare și mixere – "down converters") cu procesoare tot mai puternice (de uz general sau dedicate prelucrării de pachete sau prelucrării de semnale digitale). Această putere de calcul și programabilitatea avansată face ca SDR să devină o platformă ideală pentru implementarea inteligenței artificiale în sporirea eficienței "modem" (modulării-demodulării) și "co-dec" (codificării-decodificării). "I.A. profundă" aplicată la SDR a dus la conceptul "DeepSig".

În această direcție de radio cognitiv și spectrum sensing [100], problema ce se impune a fi rezolvată cu ajutorul inteligenței artificiale - învățarea automată ("machine learning") - este aceea de a prezice cu o acuratețe ridicată dacă și în ce moment din viitor o anumită frecvență va fi ocupată sau liberă pentru a fi folosită. Pentru a cerea asemenea sisteme de predicție este oportună folosirea inteligenței artificiale.

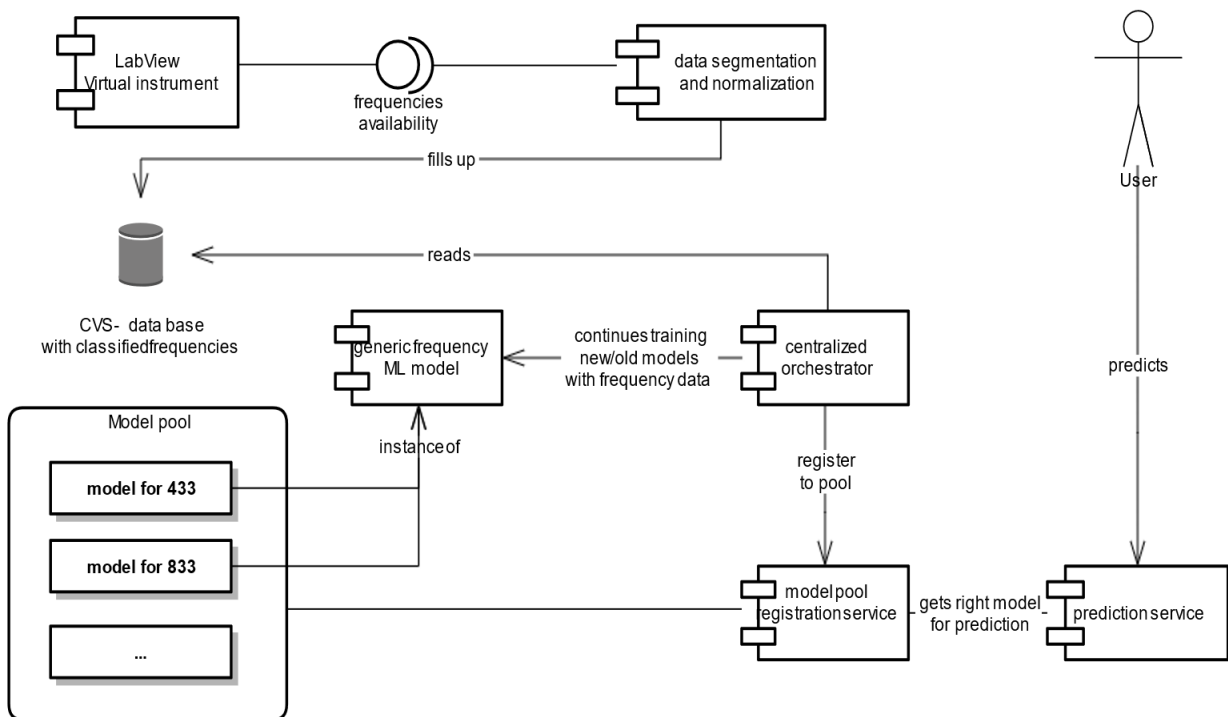
4.3.1. Definirea modelului de antrenare

Metodologia pe care am folosit-o în definirea acestui sistem a constat în următoarele etape:

- Normalizarea frecvențelor
- Analizarea și preprocesarea setului de date
- Normalizarea Datelor
- Antrenarea modelului
- Validarea modelului

4.3.2. Arhitectura generică de predicție pentru seturi de frecvență

În următoare secțiune este detaliat conceptul arhitectural al acestui sistem de predicție a disponibilității în rețea per (*interval de*) *frecvență* și *interval de timp*. În diagrama următoare este descris în mod vizual arhitectura de sistem a acestui concept.



4.4. Sumarul capitolului

Capitolul 4 se concentrează asupra implementării inteligenței artificiale în rețelele de instrumentație – o accepțiune consolidată fiind “rețele inteligente de instrumentație”. În această perspectivă unificată, pe parcursul capitolului sunt abordate apoi aplicații practice din domenii diverse ale instrumentației – de la standurile de testare a motoarelor termice și până la radio definite software. Ceea ce au în comun toate aceste abordări este tocmai modalitatea de implementare a inteligenței artificiale ca strat intermediat între instrument și utilizator.

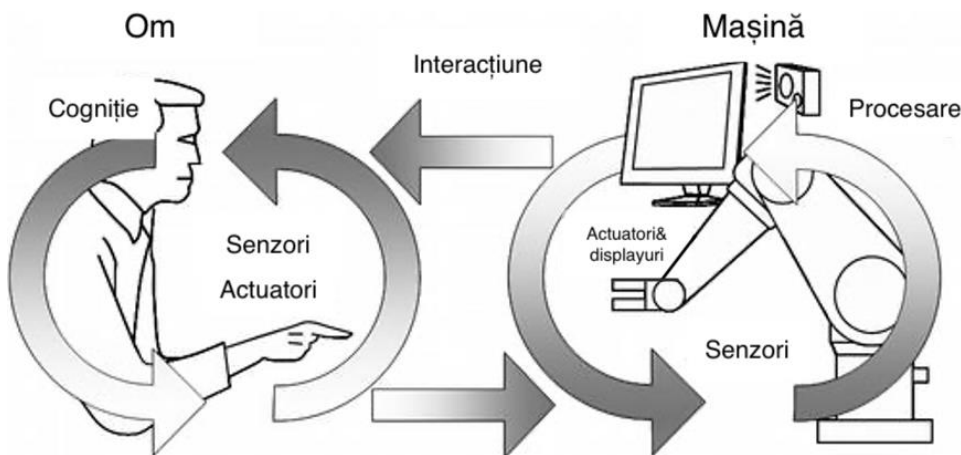
5. Interfațarea inteligentă dintre operator și rețeaua de instrumentație

Societatea a căutat întotdeauna un mod cât mai prietenos, facil și rapid de a interacționa cu "unelte" de care are nevoie. În trecut interacțiunea operatorilor cu instrumentația era predominant analogică, cuprinzând monitorizarea diferitelor afișaje, cadrane, indicatori etc. Odată cu integrarea rețelelor și a web-ului interacțiunea a devenit digitală, mai întâi prin aplicații desktop, iar apoi prin aplicații web și mobile. Am publicat cercetările asupra serviciilor web pentru instrumentație cu titlul "*Tele-measurement with Virtual Instrumentation using Web-Services*" [7.A.5] în volumul indexat ISI al unei conferințe. Următorul pas în îmbunătățirea experienței de utilizator este introducerea inteligenței artificiale ca mediator al comunicării între instrument și operator. Pentru "*umanizarea*" operării – administrării rețelelor de instrumentație cu ajutorul Inteligenței Artificiale, în acest capitol este introdus conceptul de *chatbot* ca mod de prezentare/comunicare a informațiilor către utilizator.

5.1. Interfețe de comunicare cu agenți conversaționali în rețele de instrumentație

Aproape tot ceea ce se întâmplă mai important este urmarea unui proces de comunicare. În lumea actuală se disting mai multe tipuri de comunicare, după tipul de entități participante:

- Comunicarea clasică de la om la om ("H2H") care de obicei se produce verbal sau în scris.
- Comunicarea de la Om la Mașină ("H2M") [121] și invers, de la Mașină la Om ("M2H") este realizată datorită tehnologiilor moderne.



Ilustrație a procesului de comunicare Om-Mașină, sursa [122]

În continuare, sunt enumerate principalele atribute ale Human to Machine (H2M) în contextul Chatbots și diferența față de paradigma de comunicare H2M tradițională.

Criteriu	Contextul chatbot	Sisteme tradiționale
Comanda unei acțiuni	Interacțiunea se face prin intermediul limbajului natural, nu este nevoie de sintaxa specială.	Utilizatorul trebuie să cunoască funcția asociată fiecărui buton sau care este comanda exactă pentru a executa o anumită funcție.
Modul de feedback	Interacțiunea se bazează pe dialog. Sistemul din spate trebuie să distingă dacă o interogare este neclară și să ofere utilizatorului posibilitatea de a clarifica interogarea.	Sistemele tradiționale sunt bazate pe comandă. Utilizatorul introduce o interogare și programul trebuie să o execute, dacă comanda este greșită, atunci programul returnează o eroare.
Contextul de interacțiune	Întreaga interacțiune se desfășoară în ecranul de chat – intrările, dar și ieșirile.	La fel se poate întâmpla și cu CLI tradiționale. Pentru GUI există multe schimbări de context pentru efectuarea unei operații.
Suportul multimedia	Sistemele de mesagerie nu sunt pur și simplu CLI, deoarece se poate comunica cu utilizatorul și conținut multimedia, precum pagini web, emoji-uri, imagini, videoclipuri și voce.	Command Line Interface nu integrează suport multimedia. De obicei este în afara scopului lor.
Orientarea spre scop	Sistemele de mesagerie sunt direct orientate pe scop. De exemplu, dacă doriți să găsiți informații, trebuie doar să interogați/întrebați sistemul și veți primi rapid răspunsul.	Este mai dificil ca GUI să fie orientate spre scop. De obicei pentru a obține informații (de exemplu, de pe o pagină Wiki despre automobile) trebuie să faceți click de cel puțin 3 ori pentru a obține informația dorită.

5.2. Definirea agenților conversaționali

Calitățile pe care un chatbot trebuie să le aibă [128]:

- Capabilitatea de a răspunde unor volume mari de cereri venite de la consumatori diverși.
- Să aibă o personalitate.
- Să răspundă precis și corect la cereri.
- Sistemul de agenți conversaționali este indicat să ofere o legătură cu un interlocutor uman în momentul în care reușește să rezolve o problemă.

5.2.1. Modalități de exprimare a experienței de utilizare

Paradigma de comunicare se modifică în cazul interfețelor cu utilizatorul prin intermediul mediilor conversaționale. Pentru interfețele expuse ca pagină web, există o flexibilitate mare prin care se poate interacționa cu utilizatorul pentru că există spațiu și o colecție mare de elemente grafice dispuse pe ecran. În scenariile web experiența de utilizare și *calitatea ei* (QoE – Quality of Experience) se calculează în raport cu cât de repede și ușor poate un utilizator să găsească informația de care are nevoie.

Calitățile unui modul conversațional din punct de vedere al experienței cu utilizatorul sunt:

1. Capacitatea de a reține experiențele din trecut cu utilizatorul.
2. Capacitatea algoritmului NLP de extragere a informațiilor cheie
3. Capacitatea agentul conversațional de a converge rapid către rezultat

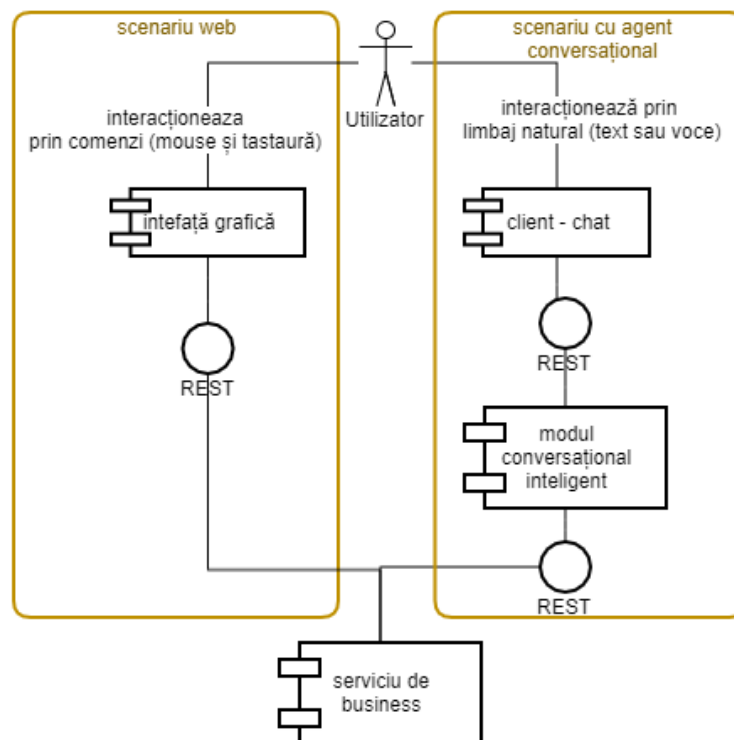
5.2.2. Moduri de interacțiune cu sistemele de business – serviciile din "back-end"

Pentru că interacțiunea cu utilizatorul se întâmplă printr-un limbaj natural, iar utilizatorul nu își exprimă direct dorința față de ceea ce vrea să facă este nevoie de un modul de mapare a interacțiunilor.

Responsabilitățile acestui modul sunt:

- *Identificarea și menținerea contextului* în care utilizatorul se află (
- *Înțelegerea limbajul natural și maparea lui la acțiuni* (
- *Transformarea acțiunilor în apeluri către serviciile de business.*

Acest modul este esențial și este cel care creează inteligență și conferă plusvaloare unui sistem conversațional. Pentru a face trecerea de la limbajul natural la acțiuni și apeluri către serviciile de business este necesară integrarea tehnicilor NLP explicate în secțiunea anterioară.



Diferențierea scenariului web de cel conversațional în integrarea cu serviciile de business

5.2.3. Conectarea cu rețelele de instrumentație

Rețelele de instrumentație au devenit inteligente, trimit și primesc date și comenzi în timp real. Există deja capacitatea de a reacționa la evenimente în mod inteligent, la nivel de rețea instrumentală.

În același timp este nevoie ca senzorii din rețea să comunice într-un mod cât mai util și plăcut cu utilizatorul. Pe piață se observă o creștere a interesului utilizatorilor către interfețe cu o inteligență tot mai mare. Așteptările utilizatorului sunt ca fiecare tehnologie cu care interacționează să conțină interfețe inteligente și, în anumite cazuri, conversaționale. Pentru rețelele de instrumentație și aplicațiile practice asociate este o oportunitate acest trend pentru că există o șansă în crearea de plusvaloare pentru utilizatorul final, dar și pentru furnizorii de servicii. Cu ajutorul interfețelor conversaționale utilizatorii vor avea parte de experiențe personalizate în relația cu infrastructurile inteligente (casa inteligentă, mașina sau chiar infrastructura publică

5.2.4. Inteligența artificială și agenții conversaționali

Instrumentele de inteligență artificială joacă un rol esențial în crearea agenților conversaționali [133]. Principalul rol al inteligenței artificiale este să asigure o conversație naturală a utilizatorului cu mașina și să ofere un nivel ridicat în exprimarea mesajului. Pentru a se putea ajunge la acest nivel este nevoie de implementarea tehnicilor NPL – procesarea naturală de limbaj. Așadar fiecare agent conversațional trebuie să conțină un modul care înțelege limbajul natural și extrage cuvintele cheie pe care le mapează la acțiuni și operații din stratul de business.

5.2.5. Propunere arhitecturală de integrare cu rețele instrumentale

Pentru integrarea [134] a unui agent conversațional cu rețele de instrumentație este nevoie de următoarele componente:

1. Aplicația de chat (Telegram, Slack, Facebook Messenger, etc.)
2. Componenta de integrare cu furnizorul de chat
3. Componenta agentului conversațional inteligent
4. Serviciile de business
5. Rețeaua de instrumentație

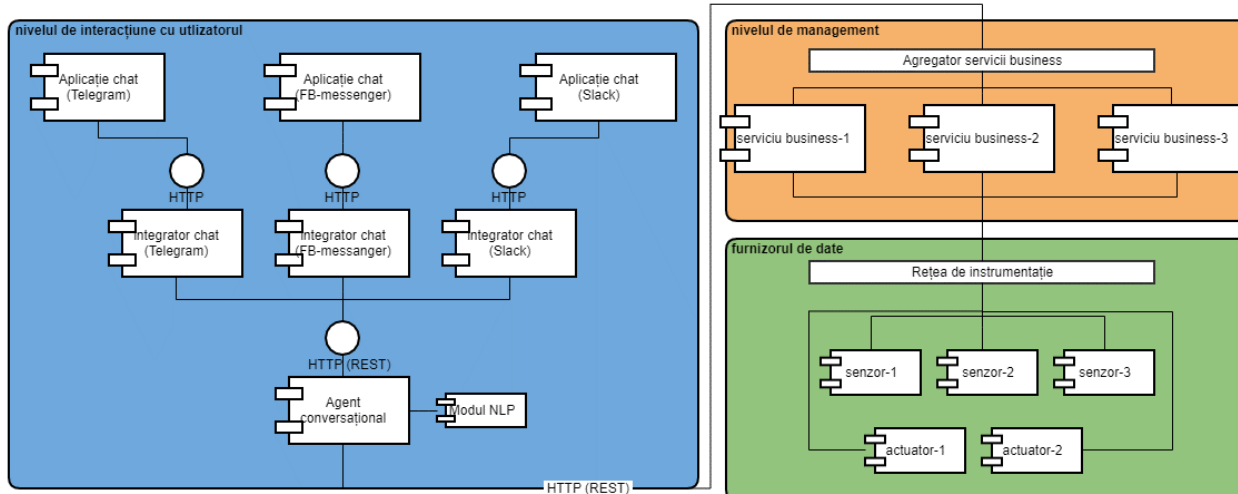


Diagrama arhitecturală a unei rețele de instrumentație ce inter-operează cu utilizatorul prin interfețe conversaționale

5.3. Demonstrator pentru integrarea agenților conversaționali în administrarea rețelelor de instrumentație

Prezintă două demonstratoare ale modului în care se poate face interfațarea cu un agent conversațional definit pentru interacțiunea cu rețeaua de instrumentație. Primul scenariu prezintă modul de integrare al unui agent conversațional cu serviciile de business definite ca servicii web scrise în programul cadru Spring – Java și pretabile atât pentru mediul Cloud, cât și pentru scenarii on-premise sau de Cloud privat. Cel de al doilea scenariu se referă la un demonstrator definit în JavaScript și orientat pentru integrarea ca servicii native Cloud [138] – servicii FaaS (Function as a Service) în Cloud-ul IBM Bluemix. Cel de-al doilea prototip este continuarea studiului de caz din capitolul 3.5 "Optimizarea continuă a rutinelor de servire" [139]. În cazul ambelor demonstratoare s-a ales integrarea cu platforma de chat/chatbot Telegram.

5.3.1. Integrarea agenților conversaționali Telegram cu servicii Java

Pentru a conecta clientul chat Telegram la serviciile de business este necesară implementarea unui serviciu de tip integrator care comunică cu API-ul Telegram pentru a asculta mesajele transmise și pentru a putea răspunde utilizatorului. Pentru acest demonstrator acest serviciu este implementat cu ajutorul limbajului de programare Java, folosind biblioteca cadru Spring 1.5.9.

5.2.2 Integrarea agențiilor conversaționale nativi în Cloud

Această secțiune extinde prototipul din studiul de caz de la capitolul "3.5 Optimizarea continuă a rutinelor de servire" și prezintă o modalitate de integrare a procesării Cloud – Edge cu interfețe de comunicare conversaționale care potențează răspunsul continuu optimizat al rutinelor de servire printr-o interacțiune directă, promptă și de tip "push-notification". Această oferă o rată mai mare de angrenare a clientului față de soluțiile clasice. Nivelul Chatbot ajută la facilitarea asupra deciziei complexe într-un mod automat / semi automat. Complexitatea actului decizional și "responsabilitățile" pot fi partajate.

5.3. Sumarul capitolului

Acest capitol s-a focalizat pe modul de expunere a datelor și acțiunilor dinspre rețeaua de instrumentație către utilizator și invers. Este prezentată o analiză a tehnologiilor care fac posibilă interfațarea cu clientul în limbaj natural, am prezentat aspecte precum procesarea limbajului natural (NLP) și tehnologiile care se pot folosi pentru implementarea unor asemenea funcționalități. Este propus un model arhitectural care face posibilă comunicarea cu utilizatorul prin chatbots. De asemenea am realizat o paralelă între comunicarea clasică prin pagini web și cea textuală analizând modul în care sistemul implementează translatarea cerințelor și acțiunilor făcute de utilizatori în activități business.

Secțiunea a treia a capitolului se concentrează pe implementarea unor demonstratoare care arată fezabilitatea tehnică și funcțională a acestui concept. Primul demonstrator este implementat în Java și este un serviciu de sine stătător ce poate rula atât în Cloud, cât și în centre de date private. Cel de-al doilea demonstrator este orientat spre o implementare nativă Cloud, în limbajul JavaScript urmărind conceptul de *serverless computing*. Acest demonstrator este continuarea studiului de caz început în capitolul 3.5 sub aspectul optimizării continue a rutinelor de servire.

6. Contorizarea inteligenta în Cloud

Acest capitol grupează majoritatea cercetărilor doctorale aplicative în domeniul Smart Grid.

6.1. Contorul Cloud

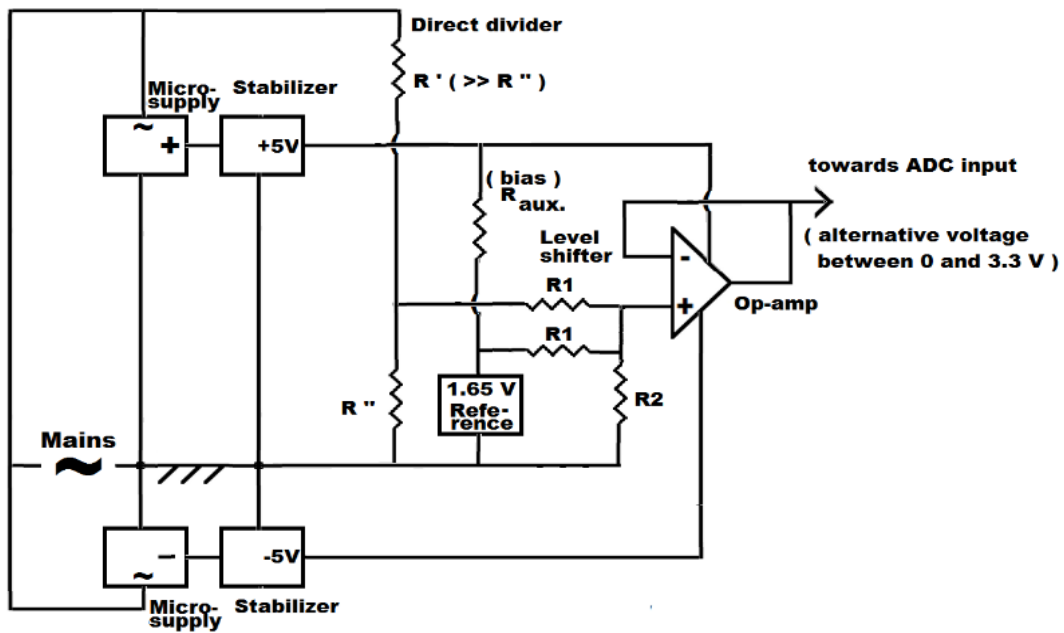
Validarea contribuțiilor originale propuse în acest capitol s-a făcut în cadrul Programului Național de cercetare PN III – Programul 2 “Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare”, Domeniul 3 “Energie, mediu și schimbări climatice”, Subdomeniul 3.3. “Orașul inteligent”. Proiectul este de tipul “Transfer la operatorul economic”, are codul PN-III-P2-2.1-PTE-2016-0064, denumirea “Măsurarea consolidată și transmiterea parametrilor energetici spre punctele de colectare” și acronimul “CON-INTEL”.

Scopul CON-INTEL a fost extinderea conceptului Smart Grid prin realizarea unui contor integrat de energie electrică, gaz metan și apă pentru măsurarea consolidată și transmiterea în timp real a parametrilor energetici spre punctele de monitorizare și control. Pentru a contribui la această abordare unitară, am propus *Contorizarea în Cloud* care, în contextul acestei lucrări de doctorat ilustrează repartiția modernă Cloud computing / Edge computing a proceselor din rețeaua de instrumentație. *Impactul* CON-INTEL în domeniul implicării diversificate în Smart Grid a “prosumatorilor” (în dublu rol de producători-consumatori) a promovat un *nou model de business* bazat pe capabilități de evaluare asistată (de I.A.) în timp real sau cumulativ/statistic a costurilor. *Caracterul de noutate* al contribuțiilor implementate a constat mai ales în repartizarea Cloud/Edge a proceselor specifice de condiționare a semnalelor, achiziția adaptivă și optimizarea proporției de calcule în domeniul frecvență și în domeniul timp – o serie de soluții avansate de instrumentație și telematice concretizate în sisteme DAQ / DSP / Informatică Instrumentală aduse la un nivel TRL (Technology Readiness Level) ridicat, cu câte 5 prototipuri funcționale, testate pe standurile de la Institutul de Cercetare-Dezvoltare al Universității “Transilvania”.

6.1.1. Sub-sistemul hardware al Contorului – nivel *Edge*

Privitor la *tensiune*, un prim aspect inovator este măsurarea directă, evitând transformatoarele (care au nelinearități inerente cauzate de imperfecțiunile de constructive și de material). Deoarece circuitul integrat a traductorului de curent asigură o scalare automată a ieșirii în funcție între 0 și tensiunea de alimentare (3.3V DC), a fost implementată, după divizarea rezistivă (soluție îndrăznească, de *punere a masei de semnal mic în comun cu masa de putere*) o condiționare similară și pentru semnalul de tensiune, cu o referință de 1,65V dedicată și un amplificator operațional.

Soluția propusă și implementată constă în divizarea rezistivă directă a tensiunii de rețea (cu valori aproximative $R' = 110\text{ k}$ și $R'' = 0,5\text{ k}$) așa cum se arată în figura următoare.



*Sub-sistemul analogic al Contorului Cloud:
mini-surse stabilizate $\pm 5\text{V}$ DC, divizarea și condiționarea tensiunii alternative*



Sub-sistemul hardware al Contorului Cloud

6.1.2. Sub-sistemul software la nivelul Edge al contorului

Sub-sistemul software pentru ESP32 asigură achiziția și tele-transmisia (prin intermediul router-ului Wi-Fi din Smart- Grid-ul domotic) a semnalelor de tensiune și curent. Am programat citirea rapidă, simplă (de pe cele două intrări ADC, pinii 0 și 3, fără nicio multiplexare), a unei perechi de eșantioane { u , i }, unul după altul. Așa cum se observă din explicațiile ("comentariile") pe care le-am inclus chiar în extrasul de cod următor, programarea ESP32 s-a făcut pentru o soluție ingenioasă de returnare Cloud →Edge a valorii Energiei active (în kWh), una din cele mai importante calculate în contorul Cloud CON-INTEL, în însăși replica HTTP ("answer") de încheiere a conexiunii TCP la nivelul serverului de instrumentație (confirmarea 200 OK)

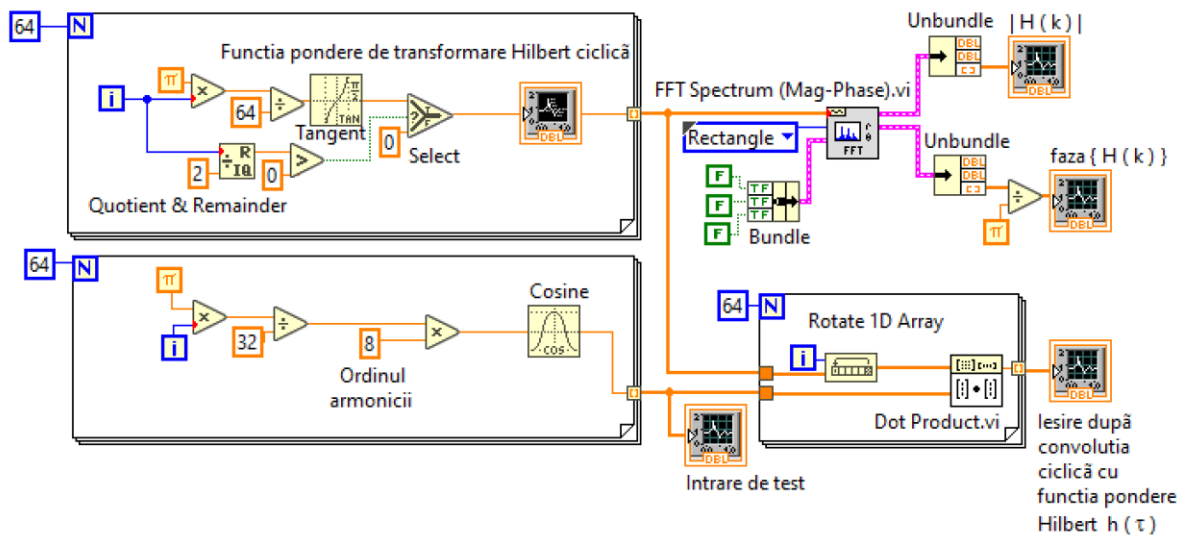
```
1 #include <LiquidCrystal_I2C.h>
2 /*
3 | AnalogReadSerial
4 | Citirea a două intrări analogice, pe pinii 0 și 3; rezultatul e "tipărit
5 | la serial monitor. Utilitarul serial plotter (Tools > Serial Plotter men
6 | permite afișarea grafică pe calculatorul de dezvoltare
7 | */
8 #include <Wire.h>
9 #include <WiFi.h>
10 #include <WiFiMulti.h>
11 #include <LiquidCrystal_I2C.h>
12 WiFiMulti WiFiMulti;
13 LiquidCrystal_I2C lcd(0x3F, 16, 2);
14 int Voltage[149];
15 int Current[149];
16 byte mac[6];
17 char b[2];
18 String MacStr;
19 void setup() {
20 | Serial.begin(115200);
21 | delay(10);
22 | // initializarea LCD
23 | lcd.init();
24 | // aprinderea luminii de fundal LCD
25 | lcd.backlight();
26 | lcd.setCursor(0, 0); // deplasare cursor la coloana 0, linia 0
27 | lcd.print(" "); // tipărire mesaj pe LCD
28 | lcd.setCursor(0, 0);
29 | // readucere cursor la coloana 0, linia 0
30 | lcd.print("ConIntel init"); // tipărire mesaj pe LCD
31 | // conectare la rețeaua WiFi
32 | WiFiMulti.addAP("conintel", "letnino");
33 | Serial.println();
34 | Serial.println();
35 | Serial.print("The MAC is:");
36 | Serial.println(WiFi.macAddress());
37 | Serial.print("Wait for WiFi... ");
38 | while (WiFiMulti.run() != WL_CONNECTED) {
39 | | Serial.print(".");
40 | | delay(500);
41 | }
42 | Serial.println("");
43 | Serial.println("WiFi connected");
44 | Serial.println("IP address: ");
45 | Serial.println(WiFi.localIP());
46 | MacStr = "MAC=";
47 | WiFi.macAddress(mac);
48 | MacStr = MacStr + WiFi.macAddress();
49 | delay(500);
50 | }
51 // buclare:
52 void loop() {
53 | const uint16_t port = 8090;
54 | const char * host = "vlab.unitbv.ro"; // IP sau DNS
55 | String payload;
56 | String answer = "";
57 | Serial.print("connecting to ");
58 | Serial.println(host);
59 | // clasa WiFiClient este utilizată pt crearea conexiunilor TCP
60 |
61 | WiFiClient client;
62 | lcd.setCursor(0, 0); // readucere cursor la coloana 0, linia 0
63 | lcd.print(" "); // tipărire mesaj pe LCD
64 | lcd.setCursor(0, 0); // readucere cursor la coloana 0, linia 0
65 | lcd.print("ConIntel running"); // tipărire mesaj pe LCD
66 | Serial.println(0);
67 | // Citirea a două intrări analogice, pe pinii 0 și 3:
68 | delay(100);
69 | for (int i = 0; i < 149; i++) {
70 | | Voltage[i] = analogRead(A0);
71 | | Current[i] = analogRead(A3);
72 | | delayMicroseconds(140);
73 | }
74 | if (!client.connect(host, port)) {
75 | | Serial.println("connection failed");
76 | | Serial.println("wait 5 sec...");
77 | | delay(5000);
78 | | return;
79 | }
80 | Serial.println("connection successful");
81 | // urmează trimiterea cererii la server:
82 | // client.print("Send this data to server");
83 | // "tipărirea" valorilor achiziționate:
84 | payload = MacStr; // adăugarea etichetelor
85 | payload = payload + ", Voltage,";
86 | for (int i = 0; i < 149; i++) {
87 | | payload = payload + String(Voltage[i], HEX) + ",";
88 | | //
89 | | reprezentare hexa pentru reducerea numărului de cifre și
90 | | încadrarea în unitatea de transfer TCP
91 | | //
92 | | delay(1);
93 | }
94 | Serial.print(payload);
95 | payload = payload + " Current,";
96 | for (int i = 0; i < 149; i++) {
97 | | payload = payload + String(Current[i], HEX) + ",";
98 | | delay(1);
99 | }
100 | payload = payload + "EOS";
101 | Serial.println("");
102 | Serial.print("Length of sent data:");
103 | Serial.println(client.print(payload));
104 | Serial.println(payload);
105 | // e preluată replica de la server
106 | // String line = client.readStringUntil('\r');
107 | // client.println(line);
108 | answer = client.readStringUntil('kWh.');
```

6.1.3. Tele-transmisia Edge – Cloud – Edge pentru contorizare

Instrumentul Virtual principal "Citire & parsing pentru unitățile de transfer TCP și returnarea selectivă a valorilor contorizate" (a cărei diagramă e prezentată mai jos) folosește cu ingeniozitate avantajul (sus-menționat) al posibilității de a returna și o replică (aici Energia Activă, cumulată anterior în Variabila Partajată aferentă contorului respectiv) la sfârșitul fiecărei conexiuni TCP, în vederea "coborârii" la nivelul Edge Computing, pentru afișare pe contorul respectiv. Procesarea "cu discriminare MAC" (rutine de servire individualizate pe principiul "abonării la serviciile Cloud" prezentate în capitolul 2) se face pe baza structurii Case.

6.1.4. Procesarea digitală a semnalelor la nivelul Cloud al contorului

Am implementat o soluție originală de calcul al puterilor reactive, promovată cadrul prestigioasei școli de *calitatea energiei electrice* de la Universitatea "Transilvania" din Brașov [140]. Pentru a ilustra aceste proprietăți, am realizat în LabVIEW un *demonstrator Hilbert* prezentat mai jos:



6.1.5. Structura integrată a proiectului de Instrumentație Virtuală

Instrumentele virtuale (VI) principale sunt: "Managerul pentru Contoare", "Citire și parsing pentru TCP MTU și returnarea selectivă a valorilor care au fost contorizate", "Contor1" (DSP), precum și a setului de variabile partajate (Active_Energy_1 ,..., MTU ,..., Time_Stamp_5) și serviciul web ("Furnizor1") asociat. Instrumentul Virtual principal dedicat serviciului web asociat (aici "Furnizor2") de tip REST (Representational State Transfer).

Acest serviciu web publică valorile (însoțite de marcajul temporal – "Time Stamp") nu numai pentru Energia Activă, E ci și pentru Energia Reactive Pozitivă, VARh(+) și Negativă VARh(-), conform standardului IEC/EN 62053.

Se încadrează valorile numerice ale energiei în valori de tip string cu 8 caractere (inclusiv punctul zecimal care e înlocuit cu virgula) fiind concatenate cu etichete sugestive. Se observă că discriminatorul pentru Structura Case este acum chiar NLC.

Testarea contoarelor Cloud într-un scenariu electrocasnic

Am înseriat, cu o mașină de spălat automată (programată pentru un ciclu de spălare de 2 ore și 5 minute) cele 5 contoare:



Pentru început, se observă afișajul "ConIntel init" (inițializarea electrică) la punerea contoarelor sub tensiune.



Urmează inițializarea conexiunii cu router-ul WiFi al casei – Se afișează "ConIntel running"



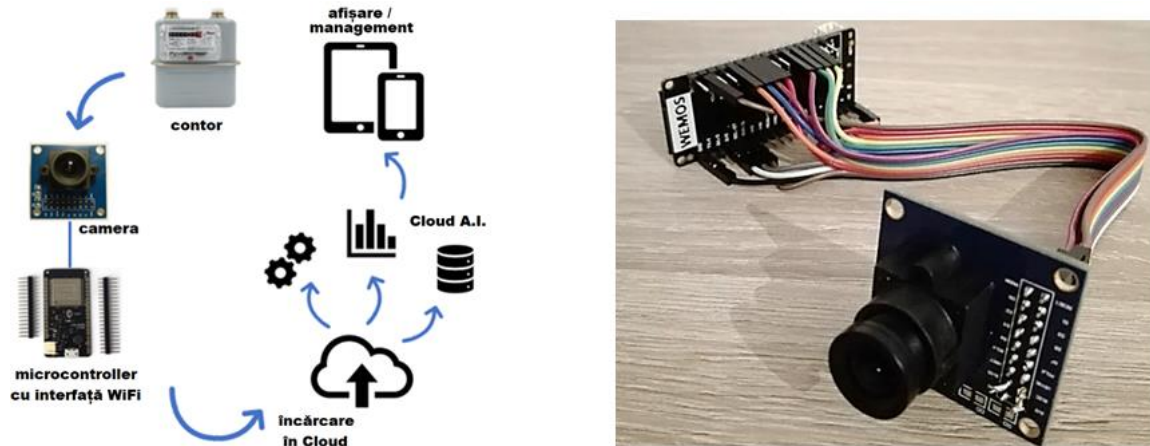
Iată modul de vizualizare pe terminalul mobil, prin servicii web, a datelor publicate ca *variabile partajate* de la contorul Cloud.



6.2. Cloud AI pentru citirea contoarelor clasice de utilități

Am implementat OCR – recunoașterea optică a caracterelor – în telemetria pentru contoarele tradiționale de utilități (apă, gaze naturale, căldură, energie electrică etc.) ca o soluție de extindere cu costuri reduse a capacităților acestor contoare și, implicit, a timpului lor de viață. Pe de altă parte, „separarea galvanică” asigurată de achiziția optică a valorilor contoarelor afișate oferă încrederea unei citiri sigure – pentru contoarele de gaze naturale și pentru alte categorii de instrumentație în medii

casnice sau industriale cu risc. Soluția implementată e eficientă din punct de vedere al costurilor și reprezintă un demonstrator Smart Grid pentru Cloud AI combinat cu posibilitatea de a transmite fluxuri de la dispozitive simple (cum ar fi o cameră „ArduCam” pentru soluțiile încorporate în Arduino bazate pe ESP32).



Arhitectura sistemului OCR pentru contorizare de la distanță împreună cu camera de achiziție de imagini embedded, cu compresie locală și încărcare în Cloud prin WiFi

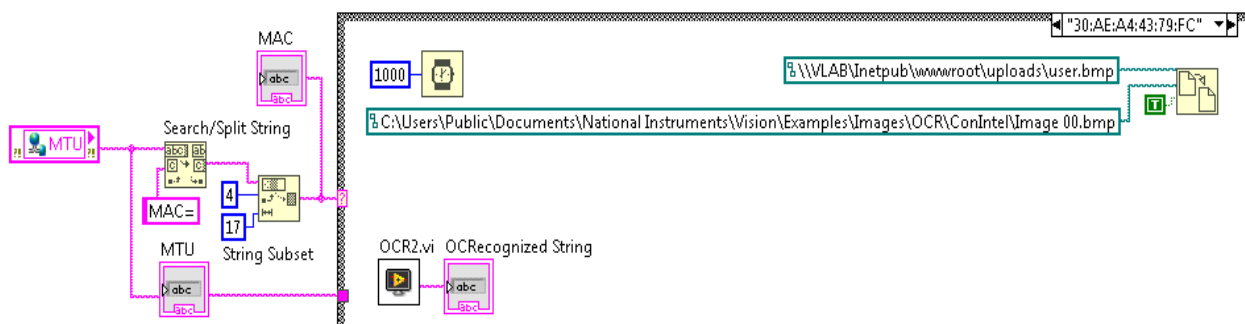
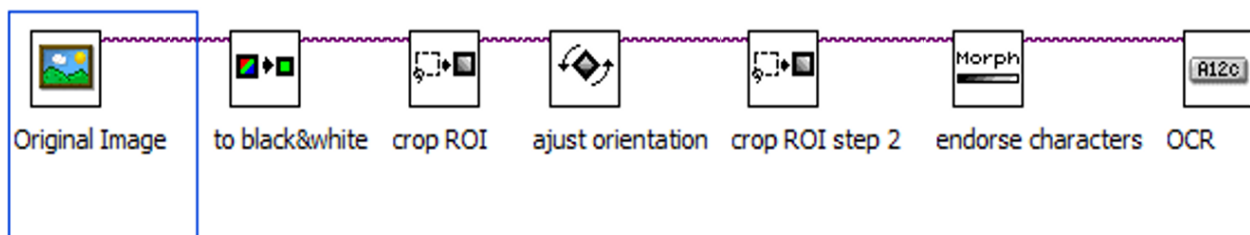
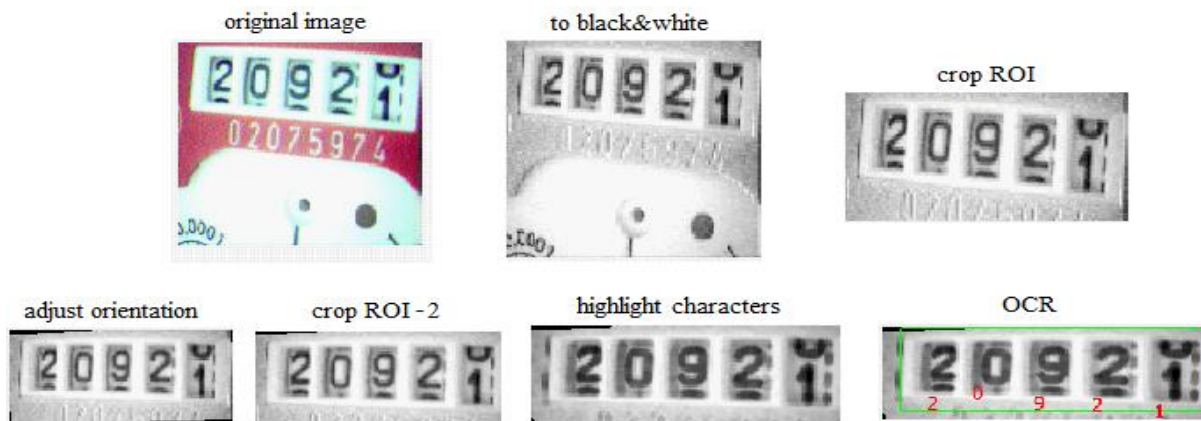


Diagrama instrumentului virtual principal, "serving" VI, cu parsing de extragere a MAC și rutina de servire a cererii OCR

Pentru a obține un mecanism stabil pentru recunoașterea într-o manieră sigură a indicilor, am investigat, selectat și implementat mai mulți pași de procesare a imaginii pentru a re-poziționa și pentru a evidenția caracterele:





Pașii de pre-procesare configurați cu NI-Vision și exemplificarea de imagini pre-procesate înainte de OCR



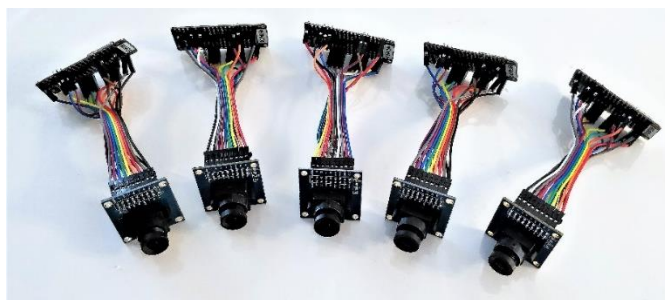
Citire la distanță a unui contor de apă



Citire la distanță a unui contor de gaz

Dezvoltarea modului de citire optică a indecșilor contoarelor pentru diferite utilități oferă o variantă pentru dezvoltarea Contorului integrat de energie electrică, gaz metan și apă pentru măsurare consolidată și transmisie în timp real către punctele de monitorizare și control.

Cele 5 sub-sisteme de citire cu recunoașterea optică a caracterelor:



6.3. Sumarul capitolului

Acest capitol prezintă soluțiile pe care le-am dezvoltat în cadrul Programului Național de cercetare PN III – Programul 2 “Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare”, Domeniul 3 “Energie, mediu și schimbări climatice”. Aceste realizări au avut ca scop implementarea unor concepte viabile pentru măsurarea mărimilor electrice de la distanță. Pentru a reuși acest lucru s-a abordat o comunicare digitalizată unde consumul de energie este calculat la nivel central pe baza eșantioanelor de curent și tensiune achiziționate prin intermediul unor dispozitive “Edge” care le transmit în serii (grupate în TCP MTU – Message Transfer Units) către “înregistratorul” Cloud “Data Logger”. Serviciile din Cloud sunt responsabile să furnizeze valorile de consum către clienți prin intermediul serviciilor web, returnate în mod ingenios (ca parametri ai comenzilor HTTP de închidere a conexiunilor TCP) înapoi către dispozitive Edge (de la “marginea rețelei”) pentru a fi afișate și pe ecranul acestora. Realizările specifice contorizării electrice folosind tehnici Cloud / Edge au fost publicate în lucrarea [7.A.4] în volumul indexat ISI al unei conferințe.

Parte a doua al acestui capitol se referă o altă realizare științifică în cadrul măsurării mărimilor de consum de la contoare clasice de utilități. În acest context a fost folosită – la nivelul Cloud AI – tehnica procesării de imagini (achiziționate și transmise de la nivelul Edge) precum și metodologia de recunoaștere optică a caracterelor pentru a citi în mod constant consumul de la contoarele clasice („legacy”). În cadrul acestei cercetări a fost folosită pentru validare citirea de pe contoare de apă (apometre) de la care s-au preluat imagini de training și validare. În acest scop s-a folosit National Instruments, LabVIEW pentru integrare și modului NI Vision pentru procesul de recunoaștere de caractere. Aceste realizări au fost publicate în cadrul lucrării „OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids,” [7.A.1] în volumul indexat ISI al unei conferințe.

7. Concluzii și contribuții originale

Acest capitol prezintă principalele concluzii și rezultate ale cercetării doctorale și marchează (©) contribuțiile originale în direcția integrării inteligenței artificiale în rețelele de instrumentație. În partea finală sunt prezentate posibile dezvoltări viitoare și o listă a principalelor realizări științifice (articole proprii, indexate în baze de date internaționale și un contract de cercetare PNCDI) care validează cercetările doctorale.

Teza de doctorat este centrată pe *Rețelele Inteligente* având ca scop să le consolideze / să le potențeze / să le extindă "inteligența" prin I.A. (Inteligența Artificială). Cea care se pretează la introducerea Inteligenței Artificiale este însăși organizarea pentru *control prin stare* care se află la baza standardelor de telecomunicații – în particular comunicații industriale, în rețelele de instrumentație care fac obiectul cercetărilor.

Structura de *automate cu stări finite / mașini algoritmice de stare* este esența arhitecturii *orientate pe servicii* care este mai presus (e "transparentă", e "agnostică") în raport cu "segmentarea" implementării concrete, hardware sau software (decisă ulterior pe criterii de disponibilitate a resurselor).

- Modul în care se face "desfășurarea" ("*deployment*") implementării unor funcționalități în hardware și (a "celor care rămân") în software va reprezenta, de fapt, prima fază a *deciziei optimale* asupra pre-istanțierii. Așa cum am arătat în capitolul 3, conceptele de livrare continuă ("*continuous delivery – development & deployment*") se pot extinde, în perspectiva acestei lucrări, prin Inteligența Artificială, în conceptul de *optimizare continuă*.
- Implementarea I.A. este hotărâtoare și pentru decizia repartizării sarcinii computaționale la nivel central sau local – Cloud, respectiv Edge ("la marginea rețelei"). În perspectiva acestei lucrări, I.A. ajunge la nivel Edge chiar și în sub-sistemul software *orientat pe obiecte*, prin *asistarea deciziei și a parametrizării (istanțierii)*.

© Am arătat că însuși nivelul SCE (Service Creation Environment) ca strat IN (Intelligent Network) – primul în abordarea "de sus în jos" ("*top-down*") a acestor rețele – prin *blocurile constructive ale serviciilor* (SBB – Service Building Blocks), este nivelul unde *I.A. poate fi introdusă atât "pe orizontala datelor" (prin parametrizarea SSD – Service Support Data) cât și "pe verticala decizională" (CID – "Call Instance Data") prin intermedierea "calculului evenimentelor"* cel care poate estima cea mai bună tranziție din traiectoria optimală de stări - "strategia" evolutivă a automatului cu stări finite.

- Am ilustrat *introducerea I.A. în decizia "asistată" asupra stării viitoare* printr-un *clasificator cu rețea neurală* implementat pe baza sub-sistemului National Instruments LabVIEW AML (Analysis and Machine Learning), pentru alegerea uneia din cele patru strategii pentru *"prosumatori"* ("producători, nu numai consumatori" de energie) bazate pe noile mecanisme DSF (Demand-Side Flexibility) din Smart Grids Domotice – setul de soluții moderne, electro-energetice și de utilități în casa inteligentă pentru care am elaborat contribuțiile prezentate în capitolul 6, implementate în proiectul PN-III "Con-Intel" la care a participat și Universitatea "Transilvania".
- Pentru parametrizarea SIB (aducerea la intrarea SSD a unor valori calculate prin I.A.) se poate utiliza rețeaua neurală (cu două straturi ascunse având câte 10 neuroni) pe care am elaborat-o în secțiunea 4.2 "Senzori virtuali cu rețele neurale" din capitolul 4 (folosind PyTorch – varianta pentru Python a bibliotecilor Torch cu metode dedicate Inteligenței Artificiale).

Această abordare contribuie la *sinergia TIC* (tehnologiilor informaționale și de comunicații – ICT, "Information & Communication Technologies") prin *introducerea I.A. în trecerea de la clasă la obiect*, (trecere care e esența POO – Programarea Obiect-Orientată), atât în faza de *instanțiere-parametrizare* cât și în *"controlul prin stare"* al *"dezvoltării continue"* (cea mai nouă tendință în software). Modelele pe care le-am ilustrat re-orientează limbajele de modelare universale (UML – Universal Modelling Languages) prin formalizarea controlului prin stare (SC – State Control), pe care am concretizat-o în specificații SC-XML într-o serie de studii de caz care extrapolează dezvoltarea "test-driven" (știut fiind că majoritatea caracteristicilor funcționale sunt specificate euristic, prin însuși modul lor de testare).

[aceste contribuții au fost prezentate în § 6.1, § 4.2 și validate în proiectul 7.D-1]

© Am realizat o extensie "Cloud pentru Instrumentație" a sistemului de 6 "server blades" Dell 1850 "Power Edge" (cu câte 2 procesoare Intel Quad-Core Xeon, hypervisor Vmware, IaaS OpenStack – administrate prin servicii DevStack) de la Departamentul de Electronică și Calculatoare al Universității "Transilvania", prin instalarea *unui Sistem de Informatică Instrumentală* centrat pe soluțiile software de la National Instruments (NI): LabVIEW 2019 Development System, LabVIEW NGX (Next Generation eXtensions – orientat pe mediul distribuit, apelul la proceduri de la distanță, RPC – Remote Procedure Calls, precum și ubicuitate a legăturilor din orice instrument virtual care devin "transparente la distanță" fiind intermediare TCP – Transfer Control Protocol); Server NI VISA (Virtual Instrumentation Software Architecture) NI DIAdem (sistem BigData / Data Analytics), NI AML (Analysis and Machine Learning – subsistemul de bază AI); NI MAX (Measurement & Automation Explorer + sistem expert IVI Wizard);

NI SDR Device Bundle (subsistemul software pentru Software Defined Radio – destinat soluțiilor de Radio Cognitiv centrate pe USRP – Universal Software Radio Peripherals de la NI – ETTUS Research); NI Vision / NI IMAQ (pentru achiziția și procesarea de imagini în contextul vederii artificiale, în care se încadrează și soluțiile OCR – Optical Character Recognition). *[această contribuție a fost prezentată în § 2.3 și validată în cadrul articolului 7.A-1]*

© Am extins conceptul Cloud Computing / Edge Computing (la nivelul inferior fiind, în ultimii ani, multe sisteme “embedded” – cu calculator încorporat, fie la nivel microcontroller / SoC – System on Chip, fie la nivel SBC – “Single Board Computer”) cu conceptul *Cloud A.I. / Edge A.I.* – potențat, în ultimii ani, de o serie de Acceleratoare Hardware ale calculelor *locale* tip I.A., dintre care am experimentat Intel Movidius și NVIDIA Jetson). În cadrul cercetărilor prezentate în capitolul 3 – “Optimizarea relației client server in rețele de instrumentație” și cu studiul de caz aplicat din capitolul 4.3 referitor la implementare metodelor XGSBoost folosind NVIDIA Jetson la nivel de Edge. *[această contribuție a fost prezentată în § 4.2.4].*

Această separare Cloud-Edge are avantajul de a permite o serie de “mixaje” – soluții combinate:

- Edge ML (Machine Learning) într-un scenariu orientat pe transmiterea unui minimum de date personale (protejate de GDPR – “General Data Protection Regulation”) despre utilizator către Cloud, unde ajung strict doar cele “de lucru”. Modelul AI Local poate fi particularizat, în condițiile în care Modelul Cloud AI poate rămâne cât mai generic (minim parametrizat) – o altă nuanță a implicării AI în trecerea de la Clasă la Obiect.
 - de exemplu, asistentul virtual Amazon “Alexa” nu returnează în Cloud ci păstrează local ceea ce învață de la deținător;
 - un alt exemplu: asistentul virtual Apple “Siri” face doar local asocieri ale datelor din email și calendar cu contactele.
- Există, așadar, motive de reținere asupra unor scenarii cu “învățare în Cloud” și rămâne o provocare hibridizarea training local / descărcarea de modele pre-antrenate. Doar ce e *generic* este expus ca serviciu (MaaS).
- Crearea unor date de training quasi-anonimizate poate duce la răspunsuri “prea generice” - în contextul agenților conversaționali pe care i-am dezvoltat în capitolul 5, un chatbot poate părea astfel plictisitor de repetitiv.

© Am integrat o serie de tehnici moderne de Cloud Computing (“Serverless / Functionless – Computing) într-un mecanism de *optimizare în Cloud a rutinelor de servire* pe principiul *evenimentul (“întreruperea”) care are loc “pe pământ” declanșează o “rutină de servire a întreruperii” care este adusă “din nori” (din Cloud) mereu optimizată (cu intermedierea I.A. la nivel de parametrizare și/sau de decizie) prin mecanisme QoE (Quality of Experience).*

- Am dezvoltat o soluție IBM BlueMix / OpenWhisk menită să extindă conceptul dezvoltării (și implementării) continue (ilustrat de tehnicile recente DevOps) cu conceptul optimizării continue a răspunsului din Cloud la întreruperile de la nivelul IaaS (Infrastructure as a Service) al rețelelor de instrumentație – Demonstratorul pe care l-am realizat este orientat pe Smart Grids Domestic (în contextul soluțiilor moderne pe care le-am implementat în proiectul PN-III "Con-Intel").
 - Am ilustrat modul de desfășurare ("coborâre din Cloud") – "deployment" al unei specificații de rețea în forma SC-XML sus menționată printr-o implementare care ilustrează conceptul MaaS ("Model as a Service") – modelul tip I.A. (de exemplu o rețea neurală) fiind elaborat în Cloud prin proceduri intensive computațional (de exemplu GAN – Generative Adversarial Networks).
- © Cercetările prezentate în capitolul 4, "Rețele Inteligente de Instrumentație" au urmărit o *sinergie Rețea (și Cloud) – Instrumentație – Rețele Inteligente – Comunicații – Inteligență Artificială* (cu o deschidere către *Smart Grid* care s-a concretizat, mai ales, în capitolele următoare). În ultimii ani s-au răspândit rețelele *pervasive* (prezente peste tot, de la cele mai mici dispozitive interconectate prin comunicații de proximitate NFC – "Near Field Communications", și până la Internetul Obiectelor, IoT – "Internet of Things") organizate ca WSN (Wireless Sensors & Actuators Networks) cu sisteme de operare încorporate ("embedded OS" – Operating Systems) minimale, Tiny OS" – la toate nivelele. [*aceste contribuții a fost prezentate în § 3.5*]
- © Am introdus I.A. în conceptul "cutiei albe" ("white box model") – modelul comportamental ("behavioral model") al controlabilității standard a instrumentației – IVI ("Interchangeable Virtual Instrument"). *Decizia asistată* a fost introdusă în înseși comenzile automatelor cu stări finite ce reprezintă *drivere-le* instrumentației iar mecanismul sus-menționat al instanțierii a fost preluat în calculul *atributelor* (valorile date parametrilor care sunt trimiși "rutinelor de servire a întreruperilor" odată cu declanșarea – cu "întreruperea" propriu-zisă).
- Abordarea "întreruperilor" – "declanșatoarelor" ("triggers") aferente modelelor IVI pentru instrumentație a avut la bază o trecere în revistă a *modurilor de "armare"* (pre-instanțiere a declanșării) și a *tipurilor de trigger-e*: a fost preluat modelul DP ("Detection Points") din Rețelele Inteligente de Comunicații care distinge ("detectează", "decelează") *cererile de întrerupere* propriu-zise (ce declanșează principalele "rutine de servire" ale automatului IVI cu stări finite) de *notificări* – evenimente ce declanșează doar procese secundare (de exemplu contorizări diverse tip "charging").

Controlul inteligent poate fi local / centralizat / distribuit – inclusiv cu forme de replicare locală a controlului central, o formă de "caching" ce permite trecerea de la gestiunea orientată pe conexiune la gestiunea orientată pe sesiune (de management al rețelei de instrumentație). Așa cum am arătat, oportunitatea introducerii I.A. în controlul distribuit ("în Cloud") o constituie *comunicațiile cibercritice* ("cyber-critical communications") de mare performanță, potențate și de sisteme de securizare și prioritizare dar bazate, în principal, pe constante de timp aflate cu câteva ordine de mărime sub cele ale instrumentației.

În această perspectivă, am considerat oportune cercetările prezentate în capitolul 2, "Integrarea rețelelor de instrumentație în Cloud" [aceste contribuții au prezentate în § 4.1, § 3.3 și au fost validate în cadrul articolului 7.A.4 și al proiectului 7.D-1]

- © Am implementat un *demonstrator Edge computing / Cloud A.I.* care ilustrează (pe principiile REST – Representational State Transfer), la nivel client, preluarea de către mesajele HTTP a funcțiilor de semnalizare / declanșare în modelul de control prin stare (implementare pe micro-sisteme ESP32 cu conexiune locală WiFi).
- Așa cum am detaliat în capitolul 6, micro-sistemul dezvoltat și prototipat în cadrul proiectului PN-III "Con-Intel", transmite prin video-camera proprie imaginea indicatorului unui contor de utilități (gaz, apă etc) care e "citit" (prin OCR – Optical Character Recognition) de către Cloud A.I. (unde este implementată și funcția DB – Data Base).

[această contribuție a fost prezentată în § 6.2 și validată în cadrul articolului 7.A.1 și în cadrul proiectului 7.D-1]

- © Am propus o *soluție de gestiune a unor dispozitive eterogene (fără a necesita alocarea de etichete Cloud "tokens") a cărei originalitate constă în mecanismul unitar de "abonare,, a acestora pe baza identificatorului unic MAC la un sistem de servire a cererilor de întrerupere.*
- Datagramelor TCP MTU (Message Transfer Units) recepționate, sunt "*parametrizate,,* cu acest MAC, astfel încât "*parsarea*" acestuia – extragerea din MTU și "*introducerea în selectorul unei structuri LabVIEW Case lansează direct rutina de servire respectivă.*" "Abonarea" este, în acest caz (din punct de vedere al structurilor de date *relaționale*), o intermediere *1:N* (cardinalitate "one-to-many") – o singură rutină de servire pentru toate cele *N* adrese MAC ale dispozitivelor *de același fel, "tratate" în același fel.*

[această contribuție a fost prezentată în § 2.1 și validată în cadrul articolului 7.A.4 și a în cadrul proiectului 7.D-1]

- © Am implementat un *sistem de senzori virtuali bazați pe I.A.* concretizat într-o soluție Edge A.I. pentru un stand de testare a motoarelor Diesel. Rețeaua neurală implementată cu bibliotecii PyTorch are 3 intrări (*turația, sarcina și căderea de presiune* pe evacuare) în primul strat ascuns cu 10 neuroni, urmat de al doilea strat ascuns (tot cu 10 neuroni) și terminat cu 1 ieșire (estimarea *puterii*), având ReLU („Rectified Linear Unit”) ca funcție de activare pentru straturile ascunse; pentru neuronul de ieșire nu există nicio funcție de activare; optimizatorul de antrenare a rețelei neurale e de tip SGD ("Stochastic Gradient Descent") iar datele de antrenare sunt pre-normate folosind media și deviația standard (o abordare simplă, tolerantă la eventuale intrări afectate de "zgomote" – erori sporadice de măsurare).
- Am adus, în acest fel, inteligența artificială la extremitatea sub-sistemului Edge A.I., chiar *la nivelul sensor-traductor inteligent* (nivel intrinsec pentru instrumentație).

- Este prezentată o implementare LabVIEW a acestei ANN, cu un potențial deosebit pentru extinderea "sigma delta" (în același context computațional) a procesării "diferenței (semnul și valoarea absolută a acesteia) dintre "realitate" (ieșirea unei senzori fizic) și "așteptări" (ieșirea senzorului virtual) cu toleranță sporită la toate cele trei nivelele (procesare, fizic, virtual) dacă se menține performanța dar scad costurile respectiv performanță sporită pe ansamblu (raportată la costul sau performanța subsistemelor).

- Sunt prezentate și discutate alternative de implementare:

- Weka (regresie arborescentă, pe sub-domenii)

O concluzie interesantă e dată de faptul că, deși în exploatarea motorului, sarcina nu e în trepte, datele de la standul de testare (cu frâna electrodinamică programată în trepte) au produs "o contaminare a modelului".

Structurarea arborescentă pe un număr de ramuri egal cu numărul de "trepte" este similară unui alt efect, „efectul de memorare” (a unui număr prea mic de seturi de antrenare, care poate fi chiar de ordinul coeficienților în modele DNN – Deep Neural Networks iar acestea, prin absurd, pot aloca câte un neuron individual care se activează doar pe un set individual de intrare și furnizează doar respectiva valoare de antrenare.

- Tensorflow cu abstractizare Keras (WNN – Wide(r) Neural Network – cu mai mulți neuroni pe strat dar fără mai multe straturi – cum au DNN, Deep Neural Network) cu același număr de intrări și ieșiri dar cu câte 64 de neuroni pe cele două straturi ascunse, unde funcțiile de activare sunt ReLU -pe primul, și Sigmoidă -pe al doilea. În același context, am mai testat, cu rezultate foarte bune, și o configurație alternativă având un singur strat ascuns cu 120 neuroni, funcție de activare tanh, funcție de optimizare SGD.
- Matlab (MathWorks TM) – cu avantaje deosebite precum rularea în Cloud (am folosit "MATLAB online"), posibilitatea de a genera modele care sunt puse în competiție în faza de training, cu opțiunea de a rula modelul cel mai performant; în studiul de caz al senzorului virtual pentru standul de testare a motoarelor, acest model a fost Regresorul Liniar Gaussian cu optimizare Bayesiană.
- ANNHUB – o implementare performantă a algoritmului Levenberg-Marquardt, cu funcționarea ANN tip "senzor virtual" intermediată de un API (Application Programming Interface) care permite calculele de predicție doar acelor clienți care au Licența ANNHUB.

[aceste contribuții au fost prezentate în § 4.2]

© Am implementat două micro-sisteme Edge cu scopul de a atinge *maximul de performanță locală* și, respectiv, *maximul de compactitate* (miniaturizare hardware și "cod minimal") – folosind tot conceptul de senzor virtual:

- Micro-sistemul Edge *cu accelerare hardware* pentru Inteligența Artificială are la bază cea mai spectaculoasă soluție accesibilă comercial la ora actuală, NVIDIA Jetson Nano (V3), gestionat de un procesor ARM A57 cu 4 nuclee (ce rulează un micro-server "flask") și având în centru SOM ("System on Module") NVIDIA "Maxwell", o *unitate specială de procesare paralelă* cu 128 "CUDA (Compute Unified Device Architecture) cores".
- O implementare *foarte compactă* Edge AI "in a Nutshell" ("într-o coajă de nucă"), bazată pe aceleași module Espressif "ESP" pe care le-am utilizat în multe din soluțiile prezentate în teză. Placa superminiatură din seria TTGO produsă de LILYGO™ are dimensiuni de ordinul centimetrilor, incluzând și un mini-display OLED pe care sunt afișate cele trei intrări (sarcina, turația și contrapresiunea) și ieșirea (puterea estimată). Coeficienții ANN au fost puși direct în codul Arduino – această "compactare software" presupune că statistica celor 3 intrări nu se modifică în timp – doar la intervale lungi mai poate fi nevoie de o "re-calibrare statistică", o „re-softare” cu actualizarea coeficienților ANN re-calculați în Cloud A.I.

Micro-sistemul Edge A.I. cu NVIDIA Jetson Nano (V3) poate rula și calcule de antrenare, în cele mai avansate calcule DNN, pe când, la extrema super-miniatură, deși există și cameră digitală încorporată nu se pot rula nici măcar citiri QR-code, darămite calcule de procesare a imaginilor achiziționate. Prin această lucrare de doctorat am încercat să dezvolt soluții la limita tehnologică ("cutting edge") pe care producătorii se precipită s-o anunțe – de exemplu prin marcaje "AI Cloud inside" pe module din familia ESP.

Ubicuitatea în rețele de instrumentație are la bază tot mai mult integrarea în mediul distribuit printr-o multitudine de comunicații radio, de la cele pe distanțe mari la cele de proximitate (NFC – Near Field Communications). Un rol important în digitalizarea acestei integrări radio o joacă, în ultimii ani SDR (Software Defined Radio). *Radio Definite Software* constau, principial, dintr-o parte analogică redusă numai la circuitele de antenă, urmate / precedate de convertoare analog-digital / digital-analog extrem de performante care fac legătura cu un sub-sistem DSP (Digital Signal Processing) cu grad ridicat de programabilitate și, deci, cu potențial de a încorpora Inteligența Artificială. Miniaturizarea CPS (Cyber Physical Systems) a extins conceptul "Smart Dust" ("pulberea" dispozitivelor miniatură) la [159] cu o mare deschidere către soluții de integrare IoT – instrumentație – Cloud.

[aceste contribuții au fost prezentate în § 4.2, § 4.3 și validate în cadrul articolelor 7.A-5, 7.A-3, 7.C-2]

© Am implementat o soluție *Radio Cognitiv cu inteligență artificială centralizată*. Cu scopul de a *consolida, prin I.A., funcția "radio cognitiv"* ("cogniție" a situației spectrale pe o bandă largă – chiar "panoramică"), cu capabilitate "spectrum sensing" (de senzor al ocupării/disponibilității pe zone spectrale) am dezvoltat o soluție SDR cu *"I.A. pe verticală" unor canale radio* (a unor intervale spectrale) cu calcule centralizate în Cloud – *"paralelism I.A."* care permite *o abordare multi-bandă* a detecției și, apoi, a deciziei "pe ansamblu" de alocare a resurselor.

- Această soluție poate fi premisa unui serviciu generic care să estimeze disponibilitatea pentru *o colecție* mai largă de canale (un ansamblu multi-bandă).
- Implementarea I.A. în SDR permite ca modelul propus să devină *adaptabil în timp și spațiu*, astfel încât să învețe continuu folosind valorile curente (ale ocupării canalelor) pe care rețeaua radio le stochează în SDR Cloud, acesta din urmă fiind capabil să publice ulterior o „prognoză spectrală” pentru fiecare locație (zonă de acoperire, celulă radio etc).
 - Adaptabilitatea în spațiu se referă la capacitatea sistemului de a descoperi noi emițătoare în teren și de a antrena automat un model de acoperire și "comportament spectral" aferent unei zone (referite prin GIS – Geographical Information Systems) – o abordare unificată a ocupării atât pe hărțile geografice cât și pe hărțile spectrale.

După faza de învățare asistată un astfel de model bazat pe I.A. se poate înregistra automat (prin procedurile de *abonare* prezentate în teză) la un serviciu centralizat și poate începe să furnizeze predicții pentru zona spectrală (și geografică) respectivă. În acest fel se poate obține un sistem care are "transparentă" temporală și spațială – *"agnostic"* în raport cu timpul, cu poziția și chiar cu mobilitatea (în aceeași ordine de idei a "ubicuității I.A." menționată în primele capitole). Practic, USRP pot să se acomodeze la orice locație nouă, începând imediat să colecteze date și fiind capabile să emită rapid previziuni pentru disponibilitatea spectrală imediată și viitoare (modele diurne, modele săptămânale etc).

Am extins detecția spectrală multicanal la observarea ocupării spectrale pe termen lung. Prin aceste cercetări am reușit să aduc I.A. în SDR prin dezvoltarea unui model de învățare automată per canal (pe „verticală”), cu mare potențial pentru abordarea *cognitivă a spectrului radio multi-bandă*. Modelul a fost testat într-un mediu integrat National Instruments (NI) – ETTUS / NI USRP (Universal Software Radio Peripheral) orientat pe servicii, controlat "de sus în jos" ("top-down") de LabVIEW – în particular de sub-setul LabVIEW "Communications System Design Suite". În abordarea unificată a acestui „Cloud de instrumentație” pe care l-am dezvoltat (și care e descris în capitolul 2), sunt gestionate în cadrul „infrastructurii ca serviciu” (IaaS).

Demonstratorul implementat s-a bazat pe un scenariu de utilizare PMR (Private Mobile Radio) cu 8 canale. Decizia de alocare a unui canal sau a altuia se poate lua "în cunoștință de cauză" folosind modelele de predicție (asupra disponibilității viitoare) bazate pe învățarea automată (AI – ML) a comportamentelor radio (a "obiceiurilor spectrale" de ocupare a benzilor de frecvență) ale stațiilor coexistente în zona respectivă.

Cercetările doctorale au investigat și potențialul de "introducere a Inteligenței Artificiale (I.A.) în OAM – *operarea, administrarea și mentenanța rețelelor de instrumentație*. În perspectiva TMN (Telecom Management Networks) se pretează eficientizării prin I.A. toate cele 5 funcții de control pentru "FCAPS" (Fault, Configuration, Accounting, Performance & Security): managementul erorilor (detecție, înregistrare, raportare, izolare), managementul conturilor (colectarea, salvarea și livrarea informațiilor de contorizare și de plată), managementul performanțelor (colectarea, salvarea și livrarea datelor statistice de funcționare, în vederea optimizării rețelei - prin planificare / re-alocare de resurse), managementul configurațiilor (instalarea instrumentației, setarea de parametri, configurarea capacităților utilizate) și, tot mai important în ultimii ani, managementul securității (administrarea funcțiilor de autorizare și de acces multiplu, protecția la intruziune).

În ultimii ani și progresele tehnologice în I.A. au fost rapid absorbite și integrate în procesele de "*support-client*" (principalul domeniu de aplicare), în educația online etc, permițând dezvoltarea de soluții eficiente și inovatoare capabile să acționeze ca *resurse conversaționale om-mașină* care să poată fi adaptate pentru necesitățile operatorului uman al rețelelor de instrumentație (de exemplu Smart Grid, la nivelul ocupat tradițional de dispeceratele SCADA – Supervisory Control & Data Acquisition). Am integrat tehnologia chatbots în experiența OAM ca punte între tehnologia tot mai avansată a rețelelor de instrumentație și necesitatea de supraveghere a lor (ce poate fi privită, într-o perspectivă mai largă, și ca măsură de *securitate*). Tehnologia chatbots poate fi folosită și în alte domenii datorită flexibilității și capacității sale de *adaptare la ritmul specific individual*, fără a provoca stres sau epuizare pentru operatorul uman. Chatbots pot funcționa și cu *interactivitate distribuită* (pentru mai mulți operatori ai rețelei), păstrând un grad ridicat de personalizare a experienței de lucru în regim conversațional. Am combinat tehnologii software moderne, cum ar fi dezvoltarea bazată pe Cloud, micro-servicii, tehnologii Java și chatbots pentru a crea o buclă închisă semi-automatizată care să permită monitorizare dar și *decizie asistată* (asupra regimului de funcționare a rețelei de instrumentație) în contextul mai larg al *strategiilor*, al *politicilor de acces și de control* (prin aceasta, contribuțiile aduse pot fi extinse și în domenii precum controlul traficului sau proceduri de aprobare ierarhică în timp real).

[aceste contribuții au fost prezentate în § 4.3 și validate în cadrul articolelor 7.A-1, 7.A-6, 7.A-7]

- © Interfațarea inteligentă operator - rețea de instrumentație reprezintă o contribuție la „umanizarea” interfeței H2M (om-mașină) prin implementarea de „agenți conversaționali (”chat-bots”).
- Am implementat sinteza în Cloud a răspunsului dat de chat-bots, pe principiile ”function-less computing”, pe același principiu al *optimizării continue* (de către I.A. centralizată) a răspunsului din Cloud la evenimente de la marginea (Edge) a rețelei (răspunsul ”din nori” la cererile ”de pe pământ”) – premisele acestei soluții sunt performanțele recente ale comunicațiilor ”ciber-critice” (”cyber-critical communications”) cu prioritarizare a *transmiterii deciziilor*.
 - Baza de cunoștințe (”knowledge base”) din Cloud și mecanismele ”Policy Control” pot susține aportul Inteligenței Artificiale peste tot unde operatorul uman trebuie: 1) să analizeze datele ca ”simptome”, 2) să le interpreteze ca ”diagnostice” – începând cu evaluarea gradului de severitate (prioritarizarea erorilor, de simplele avertizări la cele critice), 3) să ia *decizii* asupra ”alarmelor” și, mai ales, asupra unor măsuri corective (începând cu izolarea sub-sistemelor defectuoase și continuând cu re-planificarea alocării de resurse), 4) să schimbe strategiile pe termen lung, politicile de exploatare a rețelei de instrumentație, în sensul unei (re-)planificări ”durabile”.
 - Soluția implementată evită plasarea operatorului uman într-o poziție secundară față de complexitatea tehnologică a rețelei – aspectele tehnice ne-esențiale sunt ”mascate” prin afișarea de statistici relevante și acordarea accesului la pârghiile de intervenție care țin de un anumit profil al drepturilor de acces – așadar un ”feedback personalizat”.
 - Într-o abordare arhitecturală modernă, am separat straturile de implementare, pentru o soluție cu scalabilitate și interoperabilitate mărite, cu grad sporit de disponibilitate. Din punct de vedere tehnologic, am oferit o soluție multi-platformă (”cross platform”) care este durabilă - fiind accesibilă de la toate tipurile de sisteme de operare și pentru toate aplicațiile client.

Data fiind preocuparea (din mai multe secțiuni ale tezei) asupra QoE (Quality of Experience), modul în care chatbots determină și înregistrează *experiența de utilizare* reprezintă o *extindere a mecanismelor de ”charging”* care, cu aportul Inteligenței Artificiale rezidente în agenții conversaționali poate contribui la mecanisme avansate de *taxare a calității serviciilor / calității experienței* (o nouă extindere a mecanismelor de taxare la durată și volum aferente, care extind comutația de circuite și comutația de pachete cu *comutația de servicii*).

- Așa cum am menționat anterior, tocmai mecanismele de înregistrare a experienței de utilizare (prezentate în capitolul 5) pot închide acea buclă de *optimizare continuă* a modului în care e returnată ”din nori,, o rutină *mereu îmbunătățită* de servire a cererilor de întrerupere provenite ”de pe pământ”.

În paradigma UX/UI ("User eXperience / User Interfacing"), soluția de interfațare a operatorului uman cu rețeaua de instrumentație asistată de agenți conversaționali cu inteligență artificială are un mare potențial de aplicare și în alte domenii (civile și chiar militare) cu forme mixte (mașină-om-mașină), care îmbină cerințele de timp real cu cerințele de asumare a răspunderilor în anumite faze ale *deciziei semi-automatizate*. Deschiderea către "controlul politicilor" de *gestiune asistată* – "policy control" (prin I.A.) – este foarte mare în "managementul complexității" / "managementul configurațiilor", în toate sistemele "umbrelă" (rețele de management al rețelelor) TMN sus-menționate în contextul recomandării TM3000 a ITU (International Telecom Union).

Suita de soluții originale pe care le-am dezvoltat în cadrul proiectului PN-III "Con-Intel" pot fi grupate pentru

- "Contorul web de electricitate" și pentru
- "Citirea în Cloud a indicatoarelor de la contoarele obișnuite de utilități".
- Din prima categorie, merită menționate următoarele realizări: care pun în valoare progrese tehnologice recente, atât în sistemele încorporate cât și în comunicațiile industriale. Aceste noi dezvoltări aduc o nouă provocare în reechilibrarea calculului Edge și a Cloud Computing, în special în Smart Grids.

Am conceput o serie de soluții pentru "Contoare în Cloud" cu *măsurarea* locală și cu *teletransmisia* de eșantioanele de tensiune și curent pentru *calculul la nivel central* al puterii active și reactive într-un server Cloud de contorizare inteligentă cu capacități de instrumentație virtuală.

[aceste contribuții au fost prezentate în § 5.2, § 5.3 și validate în cadrul articolelor 7.A-4, 7.A-2, 7.A-3]

© Sistemul de achiziție directă - fără transformatoare de tensiune sau curent - și de teletransmisie a formelor de undă din rețeaua electrică":

- Sub-sistemul hardware al Contorului – nivel Edge – înglobează soluție îndrăznească, de *punere a masei de semnal mic în comun cu masa de putere*, cu divizarea rezistivă directă a tensiunii de rețea, circuit de condiționare a tensiunii prelevate (scalare și deplasare de nivel pentru încadrarea formei de undă în intervalul 0-3,3V), senzor de curent *de bandă largă* (mult superior oricărui transformator de curent), bazat pe efectul Hall, cu circuit integrat propriu de auto-scalare, micro-modulul (LoLin32 bazat pe ESP32) de achiziție a semnalelor de tensiune și curent, conversie A-D pe 12 biți și tele-transmisie către Cloud.

[această contribuție a fost prezentată în § 6.1.2 și validată în cadrul proiectului 7.D-1]

- © "Înregistratorul de date în Cloud" – "Cloud Data Logger": "digitizor" / înregistrator" multi-canal și de mare capacitate al datelor în Cloud, o soluție integrată cu cele de *agregare "Big Data"* (inclusiv pentru colecții de date nestructurate) pe care le-am propus.
- La nivel Edge:
 - Se realizează achiziția DAQ (Data Acquisition) + Condiționarea + SH (Sample & Hold) Eșantionarea & Reținerea (în vederea cuantizării și conversiei analog-digital) + ADC
 - Se realizează împachetarea – în MTU (Message Transfer Units), cu adăugarea unui preambul MAC (ID-ul expeditorului, principalul discriminator în vederea servirii cererilor de întrerupere), și a altor etichete ("*data tagging*") – consistente cu toate modelele de „*markup*” (marcare – utilă pentru procedurile de *parsing* la nivel Cloud)
 - Așa cum am arătat mai sus, la nivel Edge este implementată RESTful semnalizarea evenimentelor de control prin stare prin cereri ("de întrerupere") HTTP, foarte folosită fiind metoda GET care cuprinde și *attribute* (în sensul *SSD – Service Support Data*). Astfel, o soluție ingenioasă de returnare Cloud → Edge a valorii Energiei active (în kWh), una din cele mai importante calculate în contorul Cloud CON-INTEL, a fost includerea ei în însăși replica HTTP ("answer") de încheiere a conexiunii TCP la nivelul serverului de instrumentație (confirmarea 200 OK) la transmisia fiecărui MTU.
 - La nivel Cloud:
 - După înregistrarea centralizată a datelor achiziționate (Cloud Data Logging) urmează analiza (Cloud Data Analytics) bazată pe interoperabilitatea NI LabVIEW și NI DIAdem. Pentru transmiterea datelor între procesare și nivelul de analiză, o bază de date PostgreSQL a fost utilizată pentru intermediere. La modul general, DIAdem este capabil să extragă informații brute despre măsurători din baza de date și corelează această informație cu meta informațiile atașate, cum ar fi marcajele temporale ("time stamps") și adresa MAC a dispozitivului.
 - Limitările în lungimea unităților de transmisie TCP (MTU) au impus soluții de reprezentare sexagesimală a eșantioanelor și, mai ales, un calcul quasi-PLL al numărului exact de *eșantioane / perioadă* în perspectiva unor formule speciale de calcul în domeniul temporal al puterii reactive care implică o transformare Hilbert prealabilă a tensiunii.
 - Agregarea "colecțiilor de date" de la ADC multiple și precise beneficiază și de optimizarea lungimii secvențelor:
 - cu adaptarea quasi-*PLL* a perioadei semnalelor măsurate, în vederea trecerii la convoluțiile ciclice și a creșterii preciziei DFT – Discrete Fourier Transforms sau chiar
 - cu adaptarea directă PLL a frecvențelor de eșantionare (ca fiind, de obicei, frecvența fundamentală a semnalului achiziționat înmulțită cu o putere de 2).

Astfel de soluții (exemplificate în capitolul 6) ar permite stocarea "Big Data" a unor colecții de fluxuri brute, cu potențial Cloud Data Analytics (contorizarea în Cloud fiind doar una din multiplele posibilități).

Pentru a evita redundanța inutilă și, implicit, risipa spațiului de stocare, trebuie luate în considerare constantele de timp ale proceselor monitorizate – în soluția de Contor Cloud implementată, la "nivelul Cloud" (așa cum se va vedea mai jos) rularea repetitivă a Servirii VI are un timp de așteptare pre-setabil în diagrama fiecărei rutine de tratare a întreruperilor.

[aceste contribuții au fost prezentate în § 2.2, § 6.1 și au fost validate în cadrul proiectului, 7.D-1]

© DSP - Procesarea digitală a semnalelor la nivelul Cloud al contorului – implementează în LabVIEW (pe serverul de instrumentație) o soluție originală de *calcul în domeniul timp al puterilor reactive*, promovată în cadrul prestigioasei școli de calitate a energiei electrice de la Universitatea "Transilvania" din Brașov.

- pentru calculele de *convoluție ciclică* de mare precizie, am implementat o procedură PLL numeric de determinare a perioadei semnalelor de tensiune și curent;
- asupra tensiunii am aplicat transformarea Hilbert discretă și am putut aplica formula temporală a puterii active $P(\{i\}, \{u\})$ și pentru calculul puterii reactive, $Q = P(\{i\}, H\{u\})$
 - avantajele computaționale ale acestei soluții DSP sunt calculele doar cu numere reale și doar asupra unei singure serii de eșantioane (de obicei asupra tensiunii $\{u\}$, mai puțin deformată în raport cu sinusoida ideală; dezavantajul formulei Q tradiționale fiind necesitatea transformării Fourier prealabile $\{I\}$ și $\{U\}$, calculele făcându-se cu numere complexe, și pentru ambele serii de eșantioane, $\{i\}$ și $\{u\}$).

Un alt avantaj al soluției de contorizare a fost marcarea temporală ("time stamping") în Cloud. Așadar, managementul timpului – "baza de timp" pentru rețele de instrumentație poate fi o capacitate centralizată, cu mari beneficii de precizie și de sincronizare pentru clienți multipli.

Am realizat seria de 5 prototipuri pentru care am prezentat, în capitolul 6, procedurile de calibrare, testarea într-un scenariu electrocasnic și o spectaculoasă testarea a contoarelor Cloud pentru puteri reactive negative pe un super-capacitor industrial performant, produs de SCHNEIDER ELECTRIC.

"Publicarea" rezultatelor tele-măsurării în Cloud s-a făcut cu serviciile web National Instruments, foarte avantajos fiind sistemul de variabile partajate (Shared Variables), direct din instrumentele virtuale LabVIEW, prin PSP (Publish & Subscribe Protocol). Compatibilitatea HTML5 (cu "adaptarea conținutului la terminal") ușurează "citirea contorului" și de pe "thin clients" – "clienți subțiri" (ca resurse) de pe tablete sau smartphones (un exemplu sugestiv fiind NI Data Dashboard, un client mini-LabVIEW pentru Android).

[aceste contribuții au fost prezentate în § 6.1 și validate în cadrul articolului 7.A-2, 7.A-5]

© Am implementat o soluție de *"citire în Cloud"* a contoarelor tradiționale de *"utilități"* (apă, gaze naturale, energie electrică etc.), pe baza OCR ("Optical Character Recognition") – *recunoașterea optică a caracterelor*.

Această tele-măsurare bazată pe Cloud A.I. poate fi adoptată ca o extensie cu costuri modice pentru capabilitățile acestor contoare și pentru timpul lor de viață.

Totodată, „separarea galvanică” oferită de achiziția optică a valorilor contorului afișate oferă încrederea unei citiri sigure - pentru contoarele de gaze naturale și instrumentație din medii similare.

- La nivel *Edge computing*, soluția implementată se bazează pe
 - simpla fotografiere a afișoarelor de contoare cu dispozitive ieftine precum „ArduCam” (cameră pentru soluții încorporate Arduino) în montură compactă, pe micro-sistemul ESP32
 - „calcul Edge” – local, pentru formatarea .BMP a imaginilor
 - tele-transmisia prin WiFi via router-ul ("hot-spot") al clădirii inteligente.
- La nivel *Cloud A.I.*, în serverul de instrumentație bazat pe subsistemele software National Instruments - NI Vision Development Module și LabVIEW, am implementat procesările de imagini preliminare (conversia în tonuri de gri – "gray-scale", segmentarea – decuparea regiunii de interes, rotirea – "tilt", re-segmentarea, accentuarea conturilor), apoi OCR și, în fine, agregarea valorilor măsurate în baze de date.

Soluția originală a fost implementată la un nivel TRL (Technology Readiness Level) superior – serie de 5 prototipuri – în cadrul proiectului de cercetare PN III "CON-INTEL" (pachetul de lucru pentru contorizarea inteligentă în Cloud).

Soluția OCR a contribuit la extinderea Smart Grid (SG) electric ca "SG de utilități" integrată, cu gestiune unitară (prin sistemul unificat de "abonare" IoT cu MAC "tokens") a instrumentației în mediu distribuit. Specifică aceste contribuții este echilibrarea procesării ("computational load balancing") locale și centralizate (implicând și o fază de învățare asistată pentru OCR).

În aceeași abordare centrată pe sistemul "Cloud de Instrumentație" pe care l-am dezvoltat (descriș în capitolul 2) am ales să folosesc tot pachetul de licențe NI Academic Site - pentru achiziție de imagini, procesare și recunoaștere a caracterelor – avantajul fiind integrarea mai ușoară cu NI LabVIEW (care are interfețe consolidate cu modulul NI Vision Assistant), cu celelalte instrumente virtuale realizate și cu managementul lor unificat. Un alt beneficiu pe termen lung – în sensul durabilității / sustenabilității soluțiilor dezvoltate – este posibilitatea de a modifica orice detalii de implementare atât timp cât nu se schimbă interfețele, fără riscul de a bloca sistemul "Cloud de instrumentație".

[aceste contribuții au fost prezentate în § 6.2 și validate în cadrul articolului 7.A-6]

Dezvoltări viitoare

În urma cercetărilor făcute în contextul acestei teze de doctorat am identificat mai multe direcții viitoare de cercetare:

- Extinderea modelelor bazate pe inteligență artificială care permit medierea comunicațiilor între client și server pentru oferirea unor răspunsuri, mereu optimizate care se sunt bazate pe înțelegerea nevoilor actuale ale clientului (cu implicarea *ontologiilor* din *semantic web*).
- Integrarea conceptelor de Model as a Service, definite în capitolul 3.4 în dispozitivele care au nevoie de autonomie și sunt expuse pierderilor (întreruperilor) de semnal. Practic acest concept se poate folosi în contextul vehiculelor autonome care sunt predispuse pentru anumite momente să piardă conectivitatea "car-to-car" și/sau "car-to-infrastructură" (pe autostrăzile moderne). Tocmai conceptul MaaS le poate ajuta să opereze în condiții normale pe timpul în care nu există conectivitate.
- Eforturile în direcția optimizării codului pentru I.A. în vederea rulării sale la nivel Edge sunt promițătoare – de exemplu sunt în lucru noile biblioteci Arduino TinyML ("Machine Learning") care permit compilarea modelelor de învățare automată cu noul sub-set TensorFlow™ Lite și încărcarea lor pe micro-module folosind Arduino IDE ("Integrated Development Environment").
- Modelele de ocupare spectrală (propușe în capitolul 4), cu potențialul unei abordări unificate a hărților geografice și a hărților spectrale pot beneficia de LISP (Location from Identity Separation Protocols) în dezvoltarea unor LBS (Location Based Services) care să integreze mecanismele de rutare dependente de destinație / origine / timp cu mecanisme radio-relev, în perspectiva comunicațiilor peer-to-peer din 5G, în anii care urmează.
 - *Managementul identității* poate beneficia de *mecanismul unificat de "abonare" pe baza adreselor MAC* propus în această teză de doctorat, mecanism care permite o filtrare robustă a fluxurilor prevenind "inundarea" ("data flooding") sistemelor de colectare a datelor.
 - *Managementul locațiilor* poate fi, de asemenea, o soluție sustenabilă și pentru Smart Grids (SG) energetice extinse ca "SG de utilități" întrucât furnizorii din acest domeniu sunt orientați pe "locurile de consum" – (N)LC.
 - De exemplu, măsurile de compensare pot fi orientate nu pe ID ci pe NLC și eventuale "grupuri de interes" locale ("ad-hoc") se pot aborda în sens colaborativ – gestionând prin inteligența artificială resurse "puse în comun" în proximitate. Iar dacă comunicare rețelei de energice cu clientul este mediată cu ajutorul modelelor AI, conform cu primul punct din această enumerare, se poate ajunge la niște modele disruptive asupra modului în care consumatorul se abonează la energie. Se poate face trecerea de la plata pe cantitatea de energie consumată la un model de plată pe serviciu de energie (e.g. încărcarea mașinii, cicluri de spălare etc.)

- Sustenabilitatea soluției OAM intermediată de agenți conversaționali constă și în potențialul de a crește, pe seama inteligenței artificiale, proporția de chat pe baza NLP (Natural Language Processing) în raport cu simpla manipulare a indicilor. Datele stocate prin interacțiunea cu agenții conversaționali permit proceduri ample Data Analytics cu evidențierea "indicatorilor cheie" de performanță (KPI – Key-Point Indicators) metrice care pot contribui la auditarea rețelelor de instrumentație (exemplul mai important fiind Smart Grids energetice – așa cum am arătat în capitolele 5 și 6).
- Pentru a îmbunătăți precizia de sincronizare, un client NTP (Network Time Protocol) în combinație cu un RTC (Real Time Clock) poate fi adăugat și la nivel Edge, astfel încât marcajele temporale ("time stamping") să se poată face și la client.
 - Verificarea stării clientului RTC și NTP este necesară pentru a decide dacă timpul furnizat va fi utilizabil sau nu (o valoare implicită de 00:00 0000-00-00 este trimisă pentru a marca un eventual ceas nevalid).
- Datele "achiziționate în Cloud" măsurate în format „brut” ar putea fi transmise *împreună cu meta-date privind metodele proprii de procesare*. De asemenea, aceste „meta-date” pot cuprinde reprezentarea și/sau semnificația ("rostul" măsurărilor – "metadata semantice"). Aceste meta-date ar putea fi modificate într-un singur loc, de exemplu pe server, fiind actualizate automat la toți clienții. Un sistem de "cache" evoluat asociat acestui "Cloud Data-Logger" ar putea înregistra (în plus față de valorile intrinseci tele-măsurate) inclusiv antetul GET (foarte susceptibil de a se mai repeta).
- Resursele instrumentale astfel "publicate" pot fi cuprinse în scheme colaborative de agregare fiind "căutabile" și susceptibile de a fi "descoperite", "de sus în jos" – din Cloud –putând fi grupate („federate”) prin intermediul serviciilor web.

Validarea și diseminarea rezultatelor științifice în publicații și proiecte de cercetare

A) Lucrări proprii indexate ISI

[1] **V. P. Fernoaga**, G.A. Stelea, A. Balan and F. Sandu, "OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), Iasi, 2018, pp. 398-403. doi: 10.1109/SIITME.2018.8599200. *[lucrare indexată și în SCOPUS]*

[2] GA Stelea, **V Fernoaga**, C Gavrila, D Robu, "Web-Service Based Thin Client for Tele-Measurement", The International Scientific Conference eLearning and Software for Education, Volume 2, pp. 128-134, Carol I" National Defence University; DOI: 10.12753/2066-026X-18-088, 2018

[3] **V Fernoagă**, G.A. Stelea, C. Gavrilă, F. Sandu, "Intelligent Education Assistant Powered by Chatbots", The International Scientific Conference eLearning and Software for Education, Volume 2, pp. 376-383, Carol I" National Defence University, DOI: 10.12753/2066-026X-18-122, 2018. *1 citare în revistă BDI IEEE Explore.*

[4] **V. Fernoaga**, G.A. Stelea, D. Robu, F. Sandu, "Communication Solutions for Power Measurement in the Cloud", 2018 International Conference on Communications (COMM), Bucharest, 2018, pp. 397-402. doi: 10.1109/ICComm.2018.8484758. 1 citare în revistă BDI - IEEE Explore.

[5] D.N.Robu, **P.V.Fernoagă**, G.A.Stelea, F.Sandu, "Tele-measurement with virtual instrumentation using web-services", October 2017 DOI: 10.1109/SIITME.2017.8259932 Conference: 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME) *2 citări în reviste BDI - IEEE Explore și 1 citare într-o revistă BDI – EBSCO*

[6] **VP Fernoaga**, R. Curpen, C. Nutiu, F.Sandu - Multichannel Software Defined Radio with Spectral Decision via Centralized Artificial Intelligence – proceedings of the ISI – IEEE "18th RoEduNet Conference: Networking in Education and Research", Galați, 2019 *[lucrare indexată și în SCOPUS]*

[7] R. Curpen, **VP Fernoaga**, D. Robu, F.Sandu – „Open-LTE Call Emulator in Software Defined Radio” – proceedings of the ISI – IEEE "18th RoEduNet Conference: Networking in Education and Research", Galați, 2019 *[lucrare indexată și în SCOPUS]*

B) Alte lucrări proprii indexate SCOPUS

[1] G.A. Stelea, V. Popescu, M. Murrioni, T. Balan and **V. Fernoaga**, „Enhanced Resource Management for Web Based Thin Clients Using Cross-Platform Progressive Offline Capabilities”, 14th EAI International Conference on Cognitive Radio Oriented Wireless Networks - CROWNCOM 2019 | Poznan, Poland [conferință de nivel 1 în clasificarea Julkaisu] pp. 361-372, Springer Cham *Indice Hirsch Scopus h=2*

C) Lucrări proprii indexate în jurnale EBSCO

[1] G.A. Stelea, **V. Fernoaga**, C. Gavrilă, V. Popescu, M. Murrioni, "Mobile Accessible Rich Internet Web Application enhanced with AMP publishing technology", Review of the "Henri Coandă" Air Force Academy No.1 (39)/2019, DOI: 10.19062/1842-9238.2019.17.1.9;

[2] Stelea, G.A., Gavrilă, C., **Fernoaga, V.P.**: Real-Time Data Analytics with Semantic Web Metadata and Web Services - BULLETIN OF THE TRANSILVANIA UNIVERSITY OF BRASOV - VOL. 10 (59) No.2 - 2017 SERIES I - ENGINEERING SCIENCES, ISSN 2065-2119 (Print), ISSN 2065-2127 (CD-ROM);

D) Proiecte de cercetare

[1] Proiectul PN-III-P2-2.1-PTE-2016-0064 "CON-INTEL – Măsurarea consolidată și transmiterea parametrilor energetici spre punctele de colectare" - Finanțat de UEFISCDI - Program PNCDI III – Programul 2 "Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare", Domeniul 3 "Energie, mediu și schimbări climatice", Subdomeniul 3.3. "Orașul inteligent". Proiectul de tipul "Transfer la operatorul economic".

8. Referințe

8.1. Bibliografie

- [1] Y. Yan, Y. Qian, H. Sharif and D. Tipper - "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges" - IEEE Communications Surveys & Tutorials, Vol.15, Issue 1, Feb.2012, pp 5-20, ISSN 1553-877X
- [2] K.Budka, J.Deshpande, M.Thottan - "Communication Networks for Smart Grids: Making Smart Grid Real (Computer Communications and Networks)", Springer 2014 , ISBN 978-1-4471-6301-5
- [3] S. Erlinghagen, B. Lichtensteiger and J. Markard, Smart meter communication standards in Europe – a comparison, In: Renewable and Sustainable Energy Reviews, 43 (2015), pp. 1249-1262.
- [4] Proceedings of the International Conference "AI: Intelligent Machines, Smart Policies" – OECD – octombrie 2017 – <http://oe.cd/ai2017>
- [5] F.Vatră, P.Postolache, C.A.Vatră, A.Poida – "Smart Grids. Introducere pentru profesioniști", ISBN:978-973-87456-8-1, 640
- [6] D.Robu și C.Costache (editori): Rețele Inteligente de Telecomunicații - Editura Universității Transilvania din Brașov, ISBN:978-606-19-0820-2, An Apariție: 2016, Nr Pagini: 559
- [7] Khorram M., Faria P., Abrishambaf O., Vale Z. (2020) Economic Impact of an Optimization-Based SCADA Model for an Office Building. In: Madureira A., Abraham A., Gandhi N., Varela M. (eds) Hybrid Intelligent Systems. HIS 2018. Advances in Intelligent Systems and Computing, vol 923. Springer, Cham
- [8] Q. Mao, F. Hu and Q. Hao, "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2595-2621, Fourthquarter 2018, doi: 10.1109/COMST.2018.2846401.
- [9] Nilsson, N. J. (2014). *Principles of artificial intelligence*. Morgan Kaufmann.
- [10] Russell, S., & Norvig, P. (2002). *Artificial intelligence: a modern approach*.
- [11] Müller V.C., Bostrom N. (2016) Future Progress in Artificial Intelligence: A Survey of Expert Opinion. In: Müller V. (eds) *Fundamental Issues of Artificial Intelligence*. Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science), vol 376. Springer, Cham
- [12] McCorduck, P., & Cfe, C. (2004). *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. CRC Press.
- [13] Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the Turing Test* (pp. 23-65). Springer, Dordrecht.
- [14] Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Pearson education.
- [15] Grace, K., Salvatier, J., Dafoe, A., Zhang, B., & Evans, O. (2017). When will AI exceed human performance. *Evidence from AI Experts*. Disponibil em: *Acesso em, 24(08)*.
- [16] Launchbury, J. (2017). A DARPA perspective on artificial intelligence. *DARPA slides*.
- [17] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- [18] Flasiński, M. (2016). Symbolic Artificial Intelligence. In *Introduction to Artificial Intelligence* (pp. 15-22). Springer, Cham.
- [19] Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning. *Neural and Statistical Classification, 13(1994)*, 1-298.

- [20] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160, 3-24.
- [21] Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6), 352-359.
- [22] Soni, N., & Ganatra, A. (2012). Categorization of several clustering algorithms from different perspective: a review. *International Journal of*.
- [23] Zhang, M. L., & Zhou, Z. H. (2013). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8), 1819-1837.
- [24] Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. *National Science Review*, 5(1), 44-53.
- [25] Bengio, Y., Courville, A. C., & Vincent, P. (2012). Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 1, 2012.
- [26] Modha, D. S. (2014). *U.S. Patent No. 8,874,498*. Washington, DC: U.S. Patent and Trademark Office.
- [27] Hassabis, D. (2016). AlphaGo: using machine learning to master the ancient game of Go. *Google Blog*, 27.
- [28] Domingos, Pedro. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.
- [29] Goyal, S., & Benjamin, P. (2014). Object recognition using deep neural networks: A survey. *arXiv preprint arXiv:1412.3684*.
- [30] Simpson, A. J., Roma, G., & Plumbley, M. D. (2015, August). Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network. In *International Conference on Latent Variable Analysis and Signal Separation* (pp. 429-436). Springer, Cham.
- [31] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.
- [32] Sibi, P., Jones, S. A., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, 47(3), 1264-1268.
- [33] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press, page 195
- [34] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press, page 290
- [35] Li, Y. D., Hao, Z. B., & Lei, H. (2016). Survey of convolutional neural network. *Journal of Computer Applications*, 36(9), 2508-2515.
- [36] Pearlmutter, B. A. (1995). Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural networks*, 1212-1228.
- [37] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
- [38] Isaacson, W. (2014). *The Innovators: How a Group of Hackers, Geniuses, and Geeks Created the Digital Revolution*.
- [39] Acquisition, monitoring and control smart system for efficient energy consumption, autori Sorin-Aurel Moraru, Milian Badea, Delia Elisabeta Ungureanu, Dominic Mircea Kristaly și Adrian Duță, The 6th Annual Winter Global Business Conference 2018, Tignes (France), January 29th- February 2nd, 2018
- [40] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing* (pp. 1-20). Springer, Singapore.

- [41] Kuntsevich, A., Nasirifard, P., & Jacobsen, H. A. (2018, December). A Distributed Analysis and Benchmarking Framework for Apache OpenWhisk Serverless Platform. In *Proceedings of the 19th International Middleware Conference (Posters)* (pp. 3-4).
- [42] Codd, E. F. (1989). Relational database: a practical foundation for productivity. In *Readings in Artificial Intelligence and Databases* (pp. 60-68). Morgan Kaufmann.
- [43] Bhargava, G., & Gadia, S. K. (1993). Relational database systems with zero information loss. *IEEE Transactions on Knowledge and Data engineering*, 5(1), 76-87.
- [44] Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on NoSQL database. In *2011 6th international conference on pervasive computing and applications* (pp. 363-366). IEEE.
- [45] S. Hecht și R. Jablonski, „NoSQL evaluation: A use case oriented survey,„ în International Conference on Cloud and Service Computing, Hong Kong, 2011.
- [46] Goyal, S., Srivastava, P. P., & Kumar, A. (2015, October). An overview of hybrid databases. In *2015 International Conference on Green Computing and Internet of Things (ICGCloT)* (pp. 285-288). IEEE.
- [47] Vyawahare, H. R., Karde, P. P., & Thakare, V. M. (2018, August). A hybrid database approach using graph and relational database. In *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)* (pp. 1-4). IEEE.
- [48] Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML data interchange formats: a case study. *Caine*, 9, 157-162.
- [49] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)* (pp. 1-10). IEEE.
- [50] Bhandarkar, M. (2010, April). MapReduce programming with apache Hadoop. In *2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)* (pp. 1-1). IEEE.
- [51] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113.
- [52] Osanaiye, O., Chen, S., Yan, Z., Lu, R., Choo, K. K. R., & Dlodlo, M. (2017).
- [53] From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access*, 5, 8284-8300.
- [54] Yi, S., Li, C., & Li, Q. (2015, June). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37-42).
- [55] More, P. (2015). Review of implementing fog computing. *International Journal of Research in Engineering and Technology*, 4(06), 335-338.
- [56] Dabhi, A. A., PTJR, P., & Chaudhary, K. (2017). Fog computing: A review and conceptual architecture, issues, applications and its challenges. *IJARIE-ISSN (O)-2395-4396*, 3.
- [57] Li, M., Ye, F., Kim, M., Chen, H., & Lei, H. (2011, May). A scalable and elastic publish/subscribe service. In *2011 IEEE International Parallel & Distributed Processing Symposium* (pp. 1254-1265). IEEE.
- [58] Lampesberger, H. (2016). Technologies for Web and cloud service interaction: a survey. *Service Oriented Computing and Applications*, 10(2), 71-110.
- [59] Bjeladinovic, S. (2018). A fresh approach for hybrid SQL/NoSQL database design based on data structuredness. *Enterprise Information Systems*, 12(8-9), 1202-1220.
- [60] James, B. E., & Asagba, P. O. (2017). Hybrid database system for big data storage and management. *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, 7(3/4), 15-27.
- [61] Furche, T., Gottlob, G., Grasso, G., Guo, X., Orsi, G., Schallhart, C., & Wang, C. (2014). DIADEM: thousands of websites to a single database. *Proceedings of the VLDB Endowment*, 7(14), 1845-1856.

- [62] Furche, T., Gottlob, G., Grasso, G., Gunes, O., Guo, X., Kravchenko, A., ... & Wang, C. (2012, April). DIADEM: domain-centric, intelligent, automated data extraction methodology. In *Proceedings of the 21st International Conference on World Wide Web* (pp. 267-270).
- [63] Diadem and Labview sursa: <https://www.ni.com/ro-ro/innovations/case-studies/19/northrop-grumman-uses-ni-labview-and-diadem-for-rapid-telemetry-data-management.html>
- [64] Cache & cloud Han, H., Lee, Y. C., Shin, W., Jung, H., Yeom, H. Y., & Zomaya, A. Y. (2011). Cashing in on the Cache in the Cloud. *IEEE Transactions on Parallel and Distributed Systems*, 23(8), 1387-1399.
- [65] Slevin, J. (2007). Internet. *The Blackwell encyclopedia of sociology*.
- [66] Ripeanu, M. (2001, August). Peer-to-peer architecture case study: Gnutella network. In *Proceedings first international conference on peer-to-peer computing* (pp. 99-100). IEEE.
- [67] Callaghan, M. J., Harkin, J., McColgan, E., McGinnity, T. M., & Maguire, L. P. (2007). Client-server architecture for collaborative remote experimentation. *Journal of Network and Computer Applications*, 30(4), 1295-1308.
- [68] Glombitza, N., Pfisterer, D., & Fischer, S. (2010, May). Using state machines for a model driven development of web service-based sensor network applications. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications* (pp. 2-7).
- [69] H. Oluwatosin, „Client-server model,” *Journal of Computer Engineering*, Vol.16, nr.1, p-ISSN: 2278-8727, pp. 67-71, 2014.
- [70] J.Kim, R. Baratto și J. Nieh, „A thin-client architecture for mobile wireless web,” în *Proceedings of the 15th international conference on World Wide Web (WWW '06)*, New York, NY, USA.
- [71] KHAN, Dost & Mohamudally și Nawaz, „From Mainframe to Cloud Computing: A Study of Programming Paradigms with the Evolution of Client-Server Architecture,” *Journal on Computing*, vol. 3, pp. 21-27, 2011.
- [72] Schwartz, B., Ward, A., Monterosso, J., Lyubomirsky, S., White, K., & Lehman, D. R. (2002). Maximizing versus satisficing: Happiness is a matter of choice. *Journal of personality and social psychology*, 83(5), 1178.
- [73] Evans, E. (2004). *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional.
- [74] Moharana, S. S., Ramesh, R. D., & Powar, D. (2013). Analysis of load balancers in cloud computing. *International Journal of Computer Science and Engineering*, 2(2), 101-108.
- [75] Dawoud, W., Takouna, I., & Meinel, C. (2011, December). Elastic virtual machine for fine-grained cloud resource provisioning. In *International Conference on Computing and Communication Systems* (pp. 11-25). Springer, Berlin, Heidelberg.
- [76] Rescorla, E. (2000). Http over tls.
- [77] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext transfer protocol-HTTP/1.1.
- [78] Fette, I., & Melnikov, A. (2011). The websocket protocol.
- [79] Guinard, D., Ion, I., & Mayer, S. (2011, December). In search of an internet of things service architecture: REST or WS-*? A developers' perspective. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services* (pp. 326-337). Springer, Berlin, Heidelberg.
- [80] Fielding, R. T. (2000). REST: architectural styles and the design of network-based software architectures. *Doctoral dissertation, University of California*.

- [81] G. Orsini, D. Bade și W. Lamersdorf, „CloudAware: Empowering context-aware self-adaptation for mobile applications, Transactions on Emerging Telecommunications Technologies,” în 29(4):e3210, 2017.
- [82] Saroiu, S., Gummadi, K. P., Dunn, R. J., Gribble, S. D., & Levy, H. M. (2002). An analysis of internet content delivery systems. *ACM SIGOPS Operating Systems Review*, 36(SI), 315-327.
- [83] Gilbert, S., & Lynch, N. (2012). Perspectives on the CAP Theorem. *Computer*, 45(2), 30-36.
- [84] „The twelve-factor app,” [Interactiv]. Available: <https://12factor.net/>.
- [85] Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc.
- [86] Rodríguez, G., Soria, Á., & Campo, M. (2016). Artificial intelligence in service-oriented software design. *Engineering Applications of Artificial Intelligence*, 53, 86-104.
- [87] Surs imagine arhitectură cu disponibilitate întreruptă <https://www.3pillarglobal.com/insights/build-highly-available-systems-odfellamazon-web->
- [88] R. Noldus - CAMEL: intelligent networks for the GSM, GPRS and UMTS network. John Wiley & Sons, 2006.
- [89] Zuidweg, J. (2015). *Creating value-added services and applications for converged communications networks*. Artech House.
- [90] Kuthe, D. Developing Cloud Computing Architecture for Modeling (Model as a Service) using Data Assimilation Techniques (2019). *International Journal of Computer Applications*, 975, 8887.
- [91] Glombitza, N., Pfisterer, D., & Fischer, S. (2010, May). Using state machines for a model driven development of web service-based sensor network applications. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications* (pp. 2-7).
- [92] Sigüenza, Á., Murillo, J. L. B., Bernat, J., & Gómez, L. A. H. (2010, November). Using SCXML for Semantic Sensor Networks. In *SSN*.
- [93] Barnett, J. (2017). Introduction to SCXML. In *Multimodal Interaction with W3C Standards* (pp. 81-107). Springer, Cham.
- [94] Sanders, R. (1987). The Pareto principle: its use and abuse. *Journal of Services Marketing*.
- [95] Bode, F. (2003). IVI comes of age: An overview of IVI specifications with current status. *IEEE aerospace and electronic systems magazine*, 18(8), 31-34.
- [96] Dillinger, M., Madani, K., & Alonistioti, N. (2005). *Software defined radio: Architectures, systems and functions*. John Wiley & Sons.
- [97] Kaabouch, N. (Ed.). (2014). *Handbook of Research on Software-Defined and Cognitive Radio Technologies for Dynamic Spectrum Management*. IGI global.
- [98] Akyildiz, I. F., Lee, W. Y., Vuran, M. C., & Mohanty, S. - A survey on spectrum management in cognitive radio networks. *IEEE Communications Magazine* - Vol. 46, Issue 4, April 2008, pp 40 – 48
- [99] Popescu, O.; Abraham, S.; and El-Tawab, S., "A Mobile Platform Using Software Defined Radios For Wireless Communication Systems Experimentation" (2017). Engineering Technology Faculty Publications-70, Old Dominion University-VA, USA
- [100] Subhedar, M., & Birajdar, G. (2011). Spectrum sensing techniques in cognitive radio networks: A survey. *International Journal of Next-Generation Networks*, 3(2), 37-51.
- [101] Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc."
- [102] Coelho, L. P., & Richert, W. (2015). *Building machine learning systems with Python*. Packt Publishing Ltd.

- [103] Pedregosa, Fabian, et al. (2011). Scikit-learn: Machine learning in Python, *Journal of machine learning research* (pp. 2825-2830)
- [104] Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á. L., Heredia, I., ... & Hluchý, L. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, 52(1), 77-124.
- [105] Chris Albon. (2018) *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning*, 1st Edition, "O'Reilly Media, Inc."
- [106] D.Srivastava, L. Bhambhu, Data classification using support vector machines, (2010), *Journal of Theoretical and Applied Information Technol-ogy* 12(1):1-7
- [107] Wyglinski, A. M., Orofino, D. P., Ettus, M. N., & Rondeau, T. W. (2016). Revolutionizing software defined radio: case studies in hardware, soft-ware, and education. *IEEE Communications Magazine*, 54(1), 68-75.
- [108] Mitola, J. - Cognitive radio architecture. In: *Cooperation in wireless networks: Principles and applications*. Springer, Dordrecht, 2006. p. 243-311.
- [109] Zheng, Y., Rajasegarar, S., & Leckie, C. (2015, April). Parking availability prediction for sensor-enabled car parks in smart cities. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*(pp. 1-6). IEEE.
- [110] Sursă imagine Citroën C4 Picasso TM 2013
- [111] Castillo, F., Witrant, E., Dugard, L., & Talon, V. (2013). Exhaust manifold pressure estimation diesel equipped with a VGT turbocharger (No. 2013-01-1752). *SAE Technical Paper*.
- [112] Wang, Y. Y., & Haskara, I. (2010, June). Exhaust pressure estimation and its application to variable geometry turbine and wastegate diagnostics. In *Proceedings of the 2010 American Control Conference* (pp. 658-663). IEEE.
- [113] Mayer, A. (2004). Number-based emission limits, VERT-DPF-verification procedure and experience with 8000 retrofits. VERT, Switzerland.
- [114] Ian H. Witten, Eibe Frank și Mark A. Hall - „Data Mining: Practical Machine Learning Tools and Techniques”
- [115] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [116] Mittal, S. (2019). A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform. *Journal of Systems Architecture*.
- [117] Han, Q., & Cho, D. (2016, August). Characterizing the technological evolution of smartphones: insights from performance benchmarks. In *Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World* (pp. 1-8).
- [118] Coccia, M. (2019). The theory of technological parasitism for the measurement of the evolution of technology and technological forecasting. *Technological Forecasting and Social Change*, 141, 289-304.
- [119] T. Bălan , F. Sandu, D.N. Robu - Integrarea Sistemelor de Calcul și Telecomunicații, Editura Universității Transilvania din Brașov 2015 ISBN:978-606-19-0609-3
- [120] Shum, H. Y., He, X. D., & Li, D. (2018). From Eliza to Xiaolce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1), 10-26.
- [121] Song D., Oh E. Y. and Rice M., Interacting with a conversational agent system for educational purposes in online courses, 10th International Conference on Human System Interactions, HSI 2017, Ulsan, South Korea, Pag 78-82

- [122] Sursă imagine comunicare om masină <http://www.doyouknow.in/Articles/Technology/Introduction-To-Human-Machine-Interface-HMI-Human-Machine-Interface-Design-Human-Machine-Interface-Software.aspx> [vizitat la data de 26 Mai 2020]
- [123] Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *interactions*, 24(4), 38-42.
- [124] Suchman, L. A. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge university press.
- [125] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug), 2493-2537.
- [126] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60).
- [127] Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, 37(1), 51-89.
- [128] Shawar, B. A., & Atwell, E. (2007, January). Chatbots: are they really useful?. In *Ldv forum* (Vol. 22, No. 1, pp. 29-49).
- [129] Njenga S.T., Oboko R.O., Omwenga E. I. and Muuro E. M., 2017 Regulating group cognitive conflicts using intelligent agents in collaborative M-learning, 2017 IEEE AFRICON, Cape Town, South Africa, Pag 38-43
- [130] Porter, T., & Miller, R. (2016, May). Investigating the three-click rule: a pilot study. In *Proceedings of the Eleventh Midwest Association for Information Systems Conference* (pp. 1-7).
- [131] Zeldman, Jeffery (30 May 2001). *Taking Your Talent to the Web: Making the Transition from Graphic Design to Web Design*. New Riders. pp. 448. ISBN 978-0-7357-1073-3.
- [132] Bowerman, G. F. (2012). *U.S. Patent No. 8,250,192*. Washington, DC: U.S. Patent and Trademark Office.
- [133] Shakhovska, N., Basystiuk, O., & Shakhovska, K. (2019). Development of the speech-to-text chatbot interface based on Google API. In *CEUR Workshop Proceedings* (Vol. 2386, pp. 212-221).
- [134] [Satu M. S., Parvez M. H. and Al-Mamun S., Review of integrated applications with AIML based chatbots, International Conference on Computer and Information Engineering, ICCIE 2015, Rajshahi, Bangladesh Pag 87-90
- [135] Io, H. N., & Lee, C. B. (2017, December). Chatbots and conversational agents: A bibliometric analysis. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 215-219). IEEE.
- [136] Muslih, M., Supardi, D., Multipli, E., Nyaman, Y. M., & Rismawan, A. (2018, September). Developing Smart Workspace Based IOT with Artificial Intelligence Using Telegram Chatbot. In *2018 International Conference on Computing, Engineering, and Design (ICCED)* (pp. 230-234). IEEE.
- [137] Johannsen, F., Leist, S., Konadl, D., & Basche, M. (2018). Comparison of Commercial Chatbot solutions for Supporting Customer Interaction.
- [138] Yan, M., Castro, P., Cheng, P., & Ishakian, V. (2016, December). Building a chatbot with serverless computing. In *Proceedings of the 1st International Workshop on Mashups of Things and APIs* (pp. 1-4).
- [139] D'silva G.M., Thakare S., More S. and Kuriakose J., 2017, Real world smart Chatbot for customer care using a software as a service (SaaS) architecture, International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2017, Palladam, India, Pag 658-664
- [140] Sandu, F., Szekely, I., Szabo, W., & Scutaru, G. (2007). Virtual instrumentation for time-domain active and reactive power measurement and compensation in industrial networks. In *Proc. of 15th IMEKO TC 4 Symposium on Novelties in Electrical Measurements and Instrumentation-Iasi*.

- [141] Sandu, F., & Szabo, W. (1998, May). Switched Capacitor Hilbert Transformer For Periodical Signals. In Proceedings of the 6th International Conference on Optimization of Electrical and Electronic Equipments (Vol. 3, pp. 679-682). IEEE.
- [142] A. Cataliotti, D. Di Cara, P. A. Di Franco, A. E. Emanuel and S. Nuccio, "Frequency response of Measurement Current Transformers," 2008 IEEE Instrumentation and Measurement Technology Conference, Victoria, BC, 2008, pp. 1254-1258
- [143] D.Alahakoon, X.Yu - "Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey" - IEEE Transactions on Industrial Informatics, Vol.12, Issue 1, March 2015, pp 425-436, ISSN 1551-3203
- [144] J.Momoh "Smart Grid: Fundamentals of Design and Analysis" - J.Wiley & sons, 2012 - ISBN 978-0-470-88939-8
- [145] T. K. Hazra, D. P. Singh and N. Daga, "Optical character recognition using KNN on custom image dataset," 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Bangkok,
- [146] M. Y. Akpınar, E. Emekligil and S. Arslan, "Extracting table data from images using optical character recognition text," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018,
- [147] J. Xie, "Optical Character Recognition Based on Least Square Support Vector Machine," 2009 Third International Symposium on Intelligent Information Technology Application, Shanghai, 2009, pp. 626-629. doi: 10.1109/IITA.2009.327
- [148] Barve, Sameeksha. "Optical character recognition using artificial neural network." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 1.4 (2012): pp-131.
- [149] S. Afroge, B. Ahmed and F. Mahmud, "Optical character recognition using back propagation neural network," 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi,
- [150] Mithe, Ravina, Supriya Indalkar, and Nilam Divekar. "Optical character recognition." International journal of recent technology and engineering (IJRTE) 2.1 (2013): 72-75.
- [151] P. A. Khaustov, V. G. Spitsyn and E. I. Maksimova, "Algorithm for optical handwritten characters recognition based on structural components extraction," 2016 11th International Forum on Strategic Technology (IFOST), Novosibirsk, 2016, pp. 355-358. doi: 10.1109/IFOST.2016.7884126
- [152] Chandarana, Jagruti, and Mayank Kapadia. "Optical character recognition." International Journal of Emerging Technology and Advanced Engineering 4.5 (2014): 219-223.
- [153] Singh, Sukhpreet. "Optical character recognition techniques: a survey." Journal of emerging Trends in Computing and information Sciences 4.6 (2013): 545-550.
- [154] T Balan, P Birla, C Marcu, I Onceru – "IoT web-shared variables–publish, collect and analysis in the Cloud" - Review of the Air Force Academy, Brasov – RO, No.3 (35)/2017 - ISSN 2069-4733
- [155] Jiqiang Song, Zuo Li, M. R. Lyu and Shijie Cai, "Recognition of merged characters based on forepart prediction, necessity-sufficiency matching, and character-adaptive masking," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 35, no. 1, pp. 2-11, Feb. 2005. doi: 10.1109/TSMCB.2004.837588
- [156] A. Coates et al., "Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning," 2011 International Conference on Document Analysis and Recognition, Beijing, 2011, pp. 440-445. doi: 10.1109/ICDAR.2011.95
- [157] Patel, Chirag, Atul Patel, and Dharmendra Patel. "Optical character recognition by open source OCR tool tesseract: A case study." International Journal of Computer Applications 55.10 (2012).

- [158] Kurzweil, Raymond C., et al. "Tilt adjustment for optical character recognition in portable reading machine." U.S. Patent No. 8,320,708. 27 Nov.2012.
- [159] D. Robu, F. Sandu, T. Balan, M. Serban- Radio Dust – Decentralized Spectrum Sensing for Cognitive Broadcasting – Proceedings of the 6th Győr Symposium and 3rd Hungarian-Polish and 1st Hungarian-Romanian Joint Conference on Computational Intelligence (2014)

8.2. Web-grafie

- [W-1] Rețele inteligente <https://www.telekom.com/en/media/details/intelligent-networks-353142> [accesat Mai 2020]
- [W-2] Rețele inteligente cu inteligență artificială <https://www.ericsson.com/en/blog/2019/11/ai-enabling-the-intelligent-network> [accesat Mai 2020]
- [W-3] Six Kin Development profil: <https://www.sixkin.com/about-us> [accesat Mai 2020]
- [W-4] Distribuția în algoritmi KNN, sursă imagine: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> [accesat Mai 2020]
- [W-5] Cota de piața al AI <https://www.grandviewresearch.com/press-release/global-artificial-intelligence-ai-market> [accesat Mai 2020]
- [W-6] DIAdem software bazat pe configurare special conceput pentru a ajuta inginerii și oamenii de știință: https://www.digitalengineering247.com/pics/0712/05957_DIAdem_Flyer_FG.pdf
- [W-7] Apache "Metron" Model as a Service: <https://metron.apache.org/current-book/metron-analytics/metron-maas-service> [accesat Mai 2020]
- [W-8] Sursă imagine motor termic www.fae.es/en/products/exhaust-gas-pressure-sensor [accesat Mai 2020]
- [W-9] Donaldson comp. (2018), Engine Horsepower & Exhaust Flow Guide; <https://www.donaldson.com/content/dam/donaldson/engine-hydraulics-bulk/catalogs/Exhaust/North-America/F110028-ENG/Exhaust-Product-Guide.pdf> [accesat Mai 2020]
- [W-10] PyTorch <https://pytorch.org/docs/stable/optim.html#> [accesat Mai 2020]
- [W-11] ANNHUB pagină oficială – ANS Center <https://www.anscenter.com/> [accesat Mai 2020]
- [W-12] Documentație XGBoost <https://xgboost.readthedocs.io/en/latest/> [accesat Mai 2020]
- [W-13] Sursă imagine NLP <https://www.quora.com/What-are-good-examples-of-shallow-natural-language-processing-SNLP> [accesat Mai 2020]
- [W-14] PHP 5 File Upload – w3schools. "The world's largest web developer site" – https://www.w3schools.com/php/php_file_upload.asp – [accesat in Mai 2020]
- [W-15] Google – The Cloud Vision API – <https://cloud.google.com/vision> [accesat in Mai 2020]
- [W-16] MQTT (Message Queuing Telemetry Transport) – the standard ISO/IEC PRF 20922)ISO/IEC PRF 20922 – <http://mqtt.org> – [accesat in Mai 2020]
- [W-17] IA îngustă vs IA generală sursă imagine: www.sharper.ai/taxonomy-ai/ – [accesat in Mai 2020]
- [W-18] „Cache-invalidation,” [accesat in Mai 2020]. <https://yihui.name/en/2018/06/cache-invalidation/>.

Rezumat

Scopul tezei de doctorat "*Contribuții la Implementarea Inteligenței Artificiale în Rețele de Instrumentație*" este de a consolida, potența și extinde rețelele de instrumentație prin introducerea Inteligenței Artificiale (I.A.). Cea care se pretează la I.A. este însăși organizarea pentru *control prin stare* care se află la baza standardelor de telecomunicații – în particular comunicații industriale, în rețelele de instrumentație care fac obiectul cercetărilor. Am arătat că însuși nivelul SCE (Service Creation Environment) ca strat IN (Intelligent Networks) – primul în abordarea "de sus în jos" a acestor rețele – prin blocurile constructive ale serviciilor SBB (Service Building Blocks), este nivelul unde I.A. poate fi introdusă. Implementarea I.A. se poate face atât "pe orizontala datelor" prin instanțierea parametrilor de intrare (produși de rețele neurale) cât și pe "verticala comenzilor" prin intermedierea (cu clasificatori) a "calculului evenimentelor". Astfel se poate estima cea mai bună tranziție din traiectoria optimală de stări – "strategia" evolutivă a automatului cu stări finite. Am realizat o extensie "Cloud pentru Instrumentație" a sistemului de 6 "server blades" Dell 1850 "Power Edge" de la Departamentul de Electronică și Calculatoare al Universității "Transilvania", prin instalarea *unui Sistem de Informatică Instrumentală* centrat pe soluțiile software de la National Instruments. Am extins conceptul Cloud Computing / Edge Computing cu conceptul *Cloud A.I. / Edge A.I.* Acesta a fost ilustrat prin implementarea a două demonstratoare, unul care pune accentul pe I.A. centralizată într-un scenariu de *radio cognitiv* și un alt demonstrator ce analizează mai multe alternative de implementarea a I.A. la nivel Edge pentru *senzori virtuali*. Am integrat, în cadrul unui studiu de caz dezvoltat cu IBM Bluemix și Apache OpenWhisk o serie de tehnici moderne de Cloud Computing ("Serverless / Functionless – Computing") într-un mecanism de optimizare în Cloud a rutinelor de servire pe principiul: evenimentul care are loc la marginea rețelei declanșează o "rutină de servire a întreruperii" care este adusă din Cloud, mereu optimizată. Acest lucru se face prin intermedierea I.A. la nivel de parametrizare și/sau de decizie prin mecanisme QoE (Quality of Experience). Am propus o soluție de interfațare între operator și rețeaua de instrumentație care, prin intermediul *agenților conversaționali*, "umanizează" răspunsul rețelei și contribuie la *decizia asistată*. Am implementat două soluții de "contorizare în Cloud" electrică / non-electrică: Prima soluție este nativă Cloud și se folosește de transformarea directă a valorilor de tensiune și curent achiziționate. A doua soluție este orientată pe contoarele de "utilități" (apă, gaze naturale, căldură etc.): pe baza tehnologiilor OCR ("Optical Character Recognition") – recunoașterea optică a caracterelor – citește non-invaziv consumul de la contoarele tradiționale și le integrează în contextul Smart Grid.

Abstract (English)

The purpose of the PhD thesis *"Contributions to the Implementation of Artificial Intelligence in Instrumentation Networks"* is to consolidate, enhance and expand instrumentation networks by introducing Artificial Intelligence (AI). The one suitable for AI is the organization of state-driven control that underlies telecommunications standards - in particular industrial communications. In the instrumentation networks, that are the subject of research, I have shown that the level of SCE (Service Creation Environment) as an IN (Intelligent Networks) layer - the first in the "top-down" approach of these networks - is the level where AI can be introduced through the Service Building Blocks (SSB). Implementation of AI can be done both "horizontally" by instantiating the input parameters (produced by neural networks) and on the "vertical of the commands" by mediating (with classifiers) the "events computing". Thus, the best transition from the optimal state trajectory can be estimated - the evolutionary "strategy" of the finite state automaton. I created an extension "Cloud for Instrumentation" for the system with 6 "server blades" Dell 1850 "Power Edge" at the Department of Electronics and Computers of the University "Transilvania", by installing an Instrumental Informatics System focused on software solutions from National Instruments. I extended the concept of Cloud Computing / Edge Computing with the concept of Cloud AI / Edge AI. This was illustrated by the implementation of two demonstrators, one focusing on AI centralized in a cognitive radio scenario and another demonstrator analyzing several alternatives to the implementation of AI at the Edge level for virtual sensors. In a case study developed with IBM Bluemix and Apache OpenWhisk, I integrated a series of modern Cloud Computing techniques ("Serverless / Functionless - Computing") into a Cloud optimization mechanism for serving routines on the principle: the event that takes place at the Edge of the network triggers an "interrupt serving routine" that is brought from the Cloud, always optimized. This is done through AI at the level of parameterization and / or decision through QoE (Quality of Experience) mechanisms. We proposed an interfacing solution between the operator and the instrumentation network which, through conversational agents, "humanizes" the network response and contributes to the assisted decision. I have implemented two electrical / non-electrical "Cloud metering" solutions: The first solution is Cloud-native and uses the direct transformation of the acquired voltage and current values. The second solution is focused on "utility" meters (water, natural gas, heat, etc.): based on OCR ("Optical Character Recognition") technologies - it reads non-invasively the consumption from traditional meters and integrates them in the context of Smart Grids.