Raport Ştiinţific Final

Identificare:

Contract de finanţare nr. 34/2021
Contractor: Universitatea de Vest din Timişoara (UVT)
Cod: PN-III-P4-ID-PCE-2020-0407
Perioada: Ian 2020 - Dec 2023

Director proiect:

Prof. Dr. Dana Petcu

Membrii proiectului:

Dr. Gabriel Iuhasz
Dr. Silviu Panica
Dr. Adrian Florin Spătaru
Drd. Teodora Selea
Drd. Alexandru Munteanu

Pagină web: http://coco.hpc.uvt.ro

1. Obiectivul proiectului

Principalul obiectiv ştiinţific al CoCo este de a permite calculul continuu Cloud-to-Edge. În special, CoCo intenţionează să cupleze mecanismele de învăţare (inclusiv învăţarea federată) şi descoperirea inteligentă a resurselor pentru a realiza un mediu de găzduire adaptiv capabil să ruleze atât pe cloud, cât şi aproape de margine.

Principalele obiective tehnice sunt:

TO1. Crearea unui runtime colaborativ Cloud-to-Edge care poate fi implementat pe întregul continuum de calcul folosind generaţiile IaC (Infrastructure as a Code). Platforma implementată va fi capabilă de reconfigurare autonomă (tratarea evenimentelor şi descoperirii). Va fi dezvoltat un mediu sandbox pentru a testa comportamentul IaC şi pentru a simula diferite implementări şi scenarii.

TO2. Realizarea achiziţia şi analiza de date relevante pentru IaC. Ambele ar trebui să poată rula pe hardware eterogen (cloud şi edge). De asemenea, sunt vizate descoperirea şi adăugarea autonomă a resurselor edge şi cloud. Asigurându-vă că toate metodele de analiză dezvoltate pot rula în mod fiabil pe hardware cu specificaţii foarte diferite, natura lor transparentă va fi explorată.

Aceste obiective au fost îndeplinite integral,după cum se va prezenta în cele ce urmează.

2. Rezultatele proiectului

CoCo a intenţionat să cupleze învăţarea automată federată şi descoperirea inteligentă de resurse pentru a realiza un mediu de găzduire adaptabil, capabil să funcţioneze atât în Cloud cât şi în Edge. *Rezultatele sunt prezentate în cele ce urmează, demonstrând prin experimente viabilitatea cuplării propuse.*

Etapa 1: Cerinţe, arhitectură, monitorizare şi achiziţie a datelor

În contextul CoCo, una dintre problemele majore care trebuie abordate este detectarea evenimentelor care pot afecta generarea şi performanţa IaC. Impactul acestor evenimente este amplificat atunci când avem de-a face cu implementarea pe dispozitive Edge, astfel încât mecanismele de detectare ar trebui să fie implementate cât mai aproape posibil de dispozitivele de margine. Pentru a ajuta la implementarea modelelor predictive cât mai aproape de dispozitivele Edge/Fog, am propus includerea unor mecanisme de adaptare transparente care sunt capabile să se adapteze sau chiar să

genereze modele predictive cu un compromis cunoscut și funcțional între timpii de inferență, costul de calcul și precizie.

Accentul nostru principal a fost îmbunătățirea motorului de detectare a evenimentelor existent, cu funcții care permit adaptarea transparentă. Aceste caracteristici se ocupă nu numai de optimizarea hiper-parametrilor modelului ML, ci și de pregătirea și preprocesarea datelor. Aceste metode sunt, de asemenea, responsabile pentru o analiză mai aprofundată a cauzelor principale ale evenimentelor anormale, care pot fi utilizate ulterior de mecanisme autonome de adaptare.

IaC ridică provocări semnificative când vine vorba de monitorizare. În special anomaliile specifice IaC și anti-patternele sunt greu de detectat și, din păcate, și de simulat. Identificarea și testarea acestor tipuri de anomalii este încă o sarcină continuă. Pentru testare a fost folosit un set de date substituent care cuprinde un set de anomalii legate de hardware. În ambele cazuri, formularea problemei este aceeași, ocupându-se cu datele din serii de timp pentru identificarea anomaliilor, astfel toate metodele și procesele implementate și testate pe setul de date substituenți pot fi reutilizate în versiunea finală a soluției CoCo.

În prima etapă a proiectului COCO s-au desfășurat activități privind analiza cerințelor, definirea arhitecturii și monitorizarea și achiziția datelor cu privire la timpul de execuție distribuit. Activitățile au încheiat proiectarea arhitecturală pentru runtime distribuit și componentele asociate de monitorizare și achiziție de date. Una dintre activitățile principale ale etapei a fost identificarea clară care sunt cerințele pentru runtime distribuit care îndeplinește obiectivul tehnic T01. Runtime trebuie să fie implementat fără probleme pe întregul continuum de calcul (pentru mediul de resurse eterogene) și trebuie să aibă capacitatea de a reconfigura automat și de a se adapta la diverse aspecte legate de mediu.

Monitorizare și date

Pentru setul inițial de experimente am folosit o platformă de monitorizare bazată pe Prometheus[1]. Valorile disponibile din această soluție sunt ușor de extins prin intermediul clienților de monitorizare personalizați disponibili într-o mare varietate de legături de limbă[2]. Acest lucru permite instrumentarea ușoară a generării și monitorizării IaC.

După cum am menționat anterior, în prezent lucrăm cu un set de date de anomalii surogat care se ocupă de anomaliile hardware. Aceste anomalii au fost induse pe o implementare cu 4 noduri distribuite. Anomaliile sunt inductibile atât cu sau fără suprapunere (de exemplu, mai multe anomalii pot fi induse la un anumit interval de timp). Toate seturile de date sunt colectate la o rezoluție de 1 secundă per eveniment pentru un interval de timp total de 2 ore. Aceste seturi de date sunt concepute pentru a facilita crearea de modele predictive ML supravegheate folosind un set de date dezechilibrat, evenimente anormale reprezentând sub 1% din cantitatea totală de evenimente monitorizate. Instrumentul utilizat pentru inducerea acestor anomalii și crearea setului de date a fost dezvoltat în timpul mai multor proiecte de cercetare, în principal H2020 ASPIDE[3] și Serrano[4].

Învățare automată pentru detectarea evenimentelor

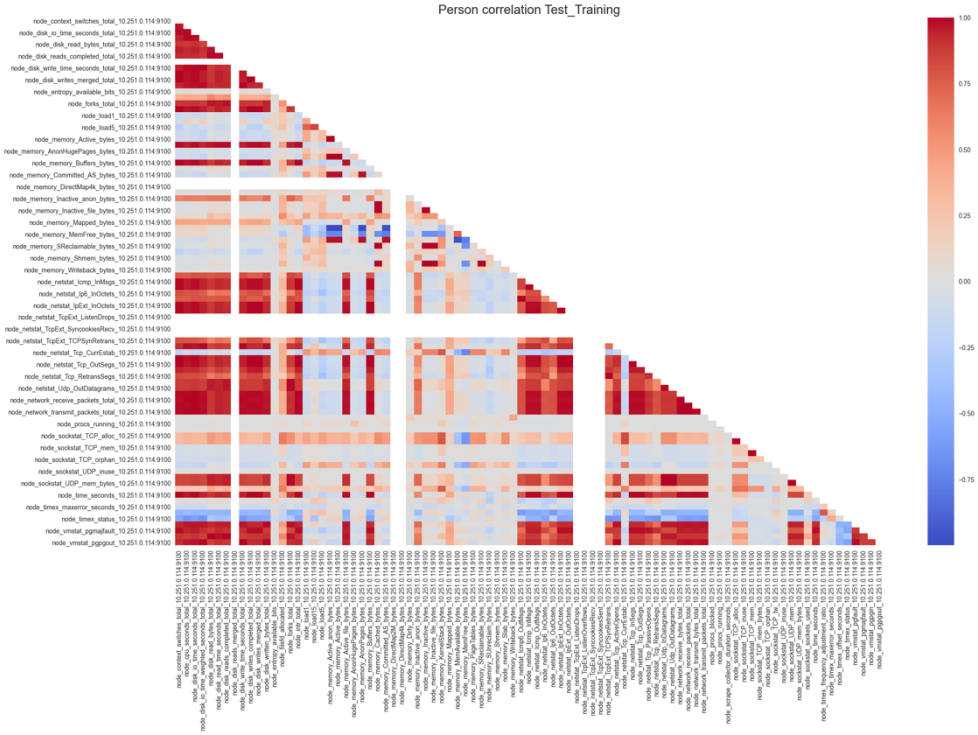
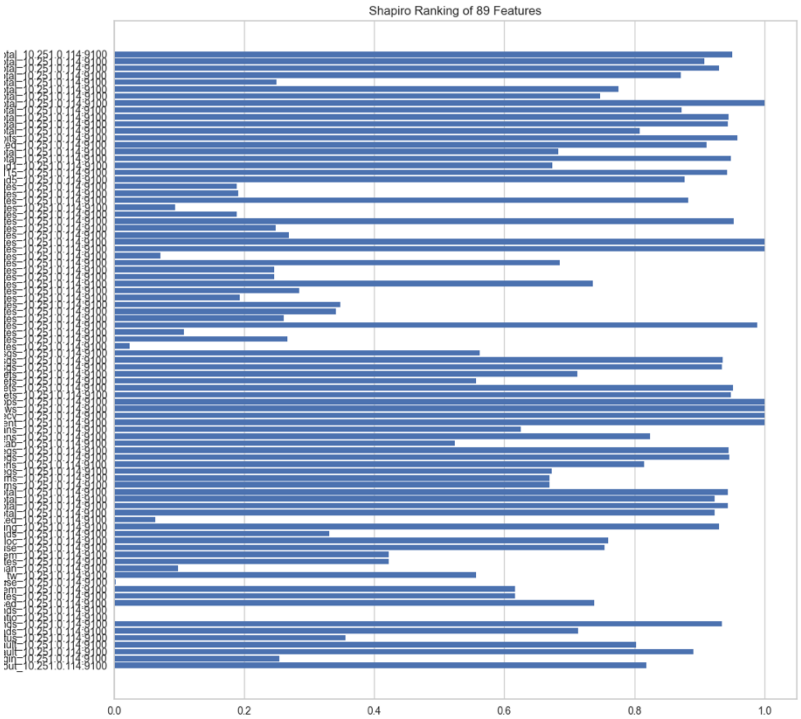
Componenta care se ocupă de detectarea evenimentelor (Event Detection Engine[5] -EDE) se bazează pe un instrument dezvoltat în cadrul proiectului H2020 ASPIDE. Adăugările făcute în timpul acestei etape pentru proiectul CoCo se concentrează în mare parte pe adaptarea transprecisă a metodelor de detectare a anomaliilor ML supravegheate.

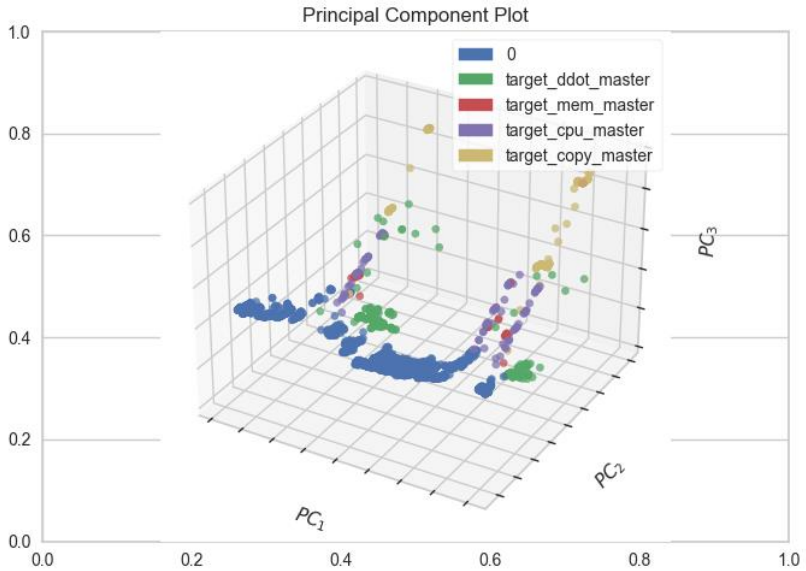
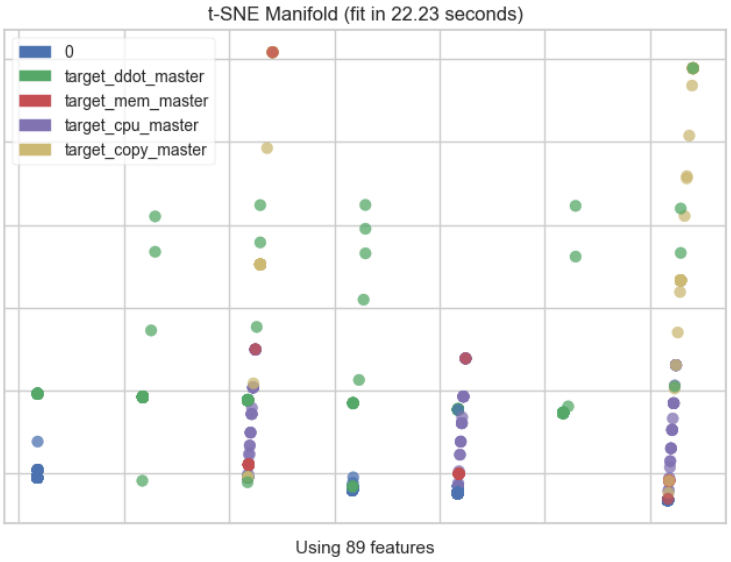
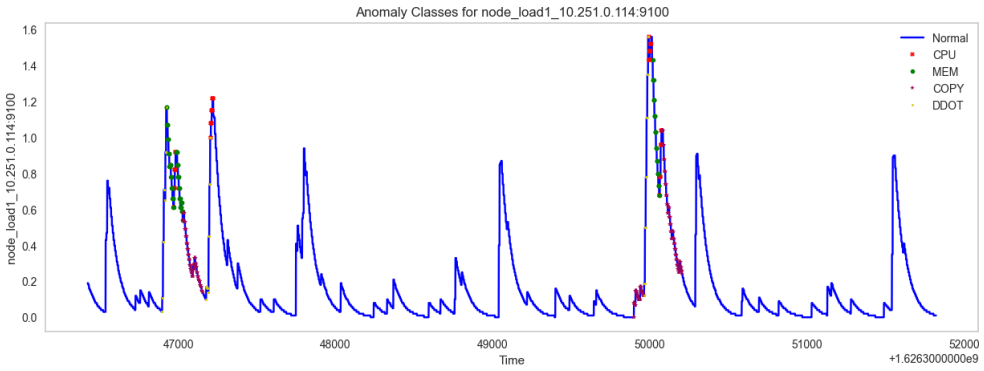
Metodele de preprocesare și analiză

S-au adăugat mai multe metode de analiză pentru explorarea și înțelegerea datelor. Preprocesarea a fost extinsă cu capacitatea de a utiliza expresii regulate pentru selectarea caracteristicilor. În cazul analizei datelor au fost adăugate mai multe metode. Ar trebui să menționăm că toate metodele acceptă și generarea de plot. Din metodele implementate amintim:

---

[1] https://prometheus.io/
[2] https://prometheus.io/docs/introduction/overview/
[3] http://aspide-project.eu
[4] https://ict-serrano.eu
[5] https://github.com/IeAT-ASPIDE/Event-Detection-Engine

Nume	Descriere
Pearson and Spearman correlation	O versiune îmbunătățită a corelației Pearson, precum și suport pentru corelația Spearman 
Shapiro ranking	S-a adăugat suport pentru clasarea caracteristicilor Shapiro: 
Principal Component analysis (PCA)	S-a adăugat suport pentru metoda de reducere a caracteristicilor PCA pentru fiecare clasă de anomalii:

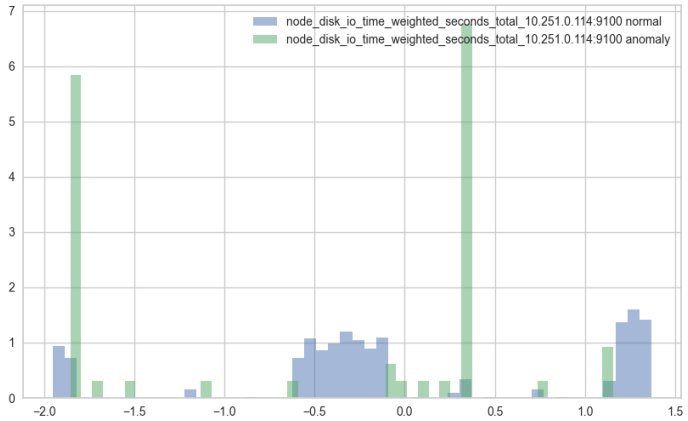
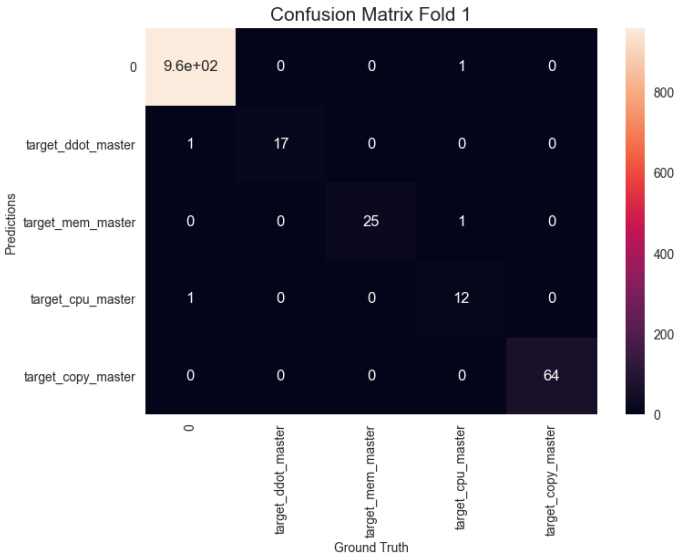
	
t-SNE Manifold embedding	S-a adăugat suport pentru reducerea caracteristicilor suplimentare (reproiectare la o dimensiune inferioară): 
Anomaly label plotting on features	S-au adăugat metode de trasare a claselor de anomalii pe caracteristicile selectate de utilizator: 

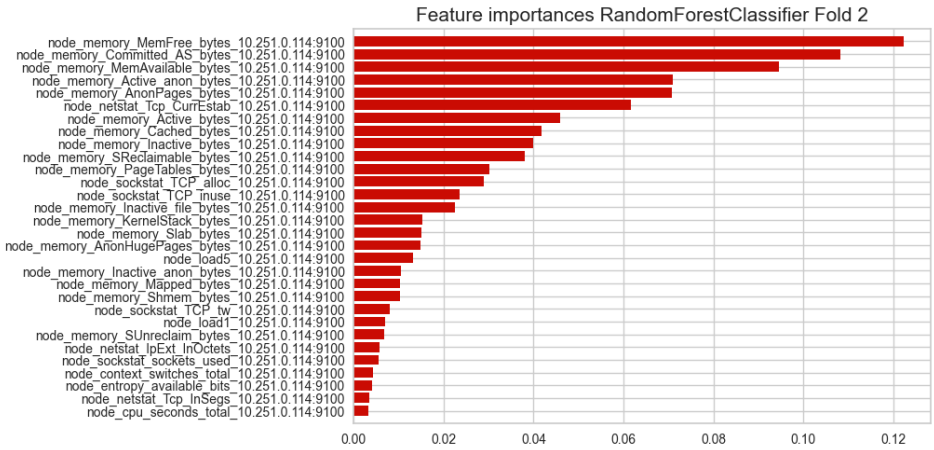
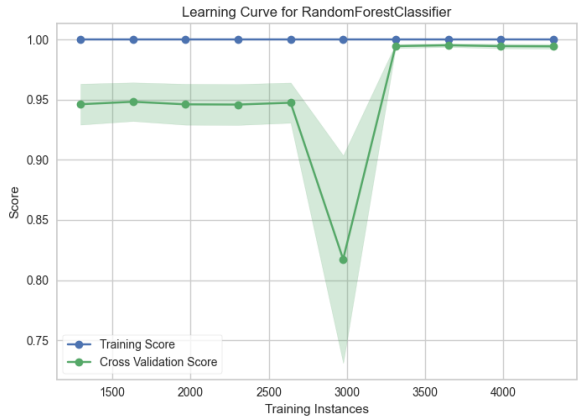
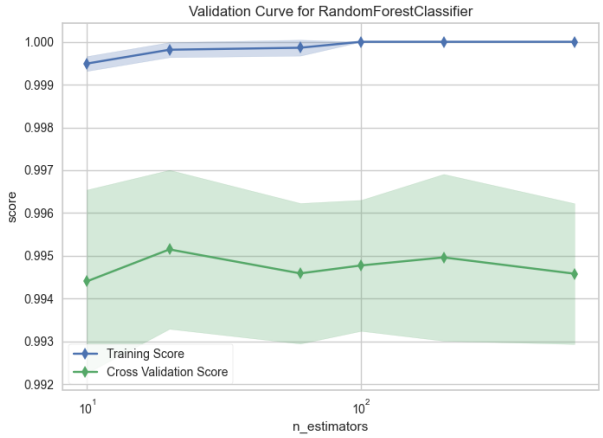
Optimizarea și raportarea învățării automate

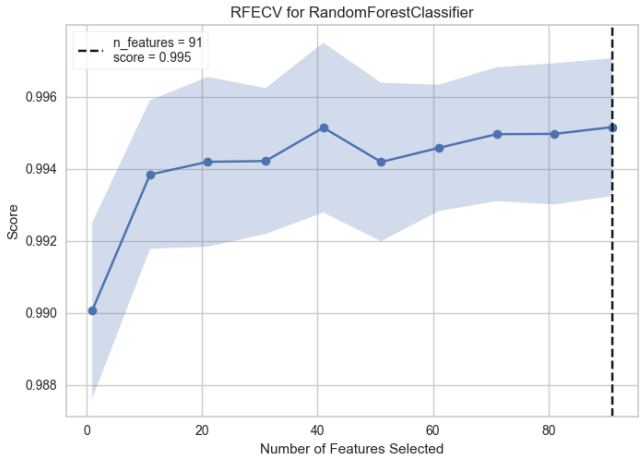
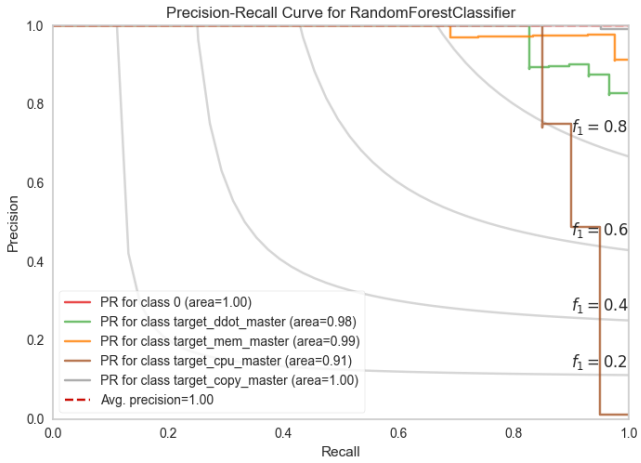
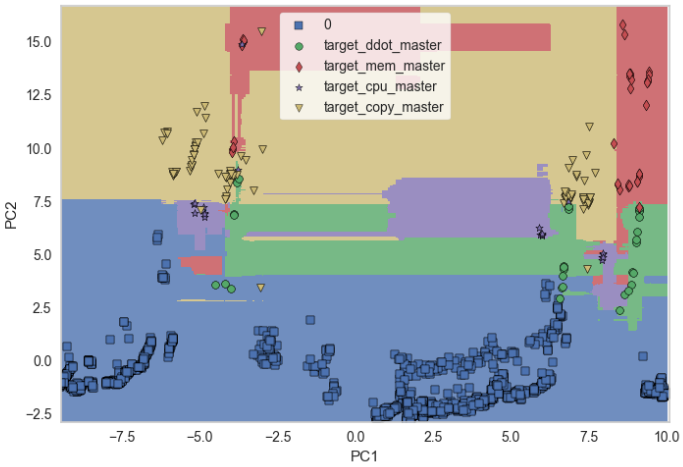
Principalul nostru obiectiv pentru metodele de optimizare a fost pe metodele supravegheate, deoarece acestea oferă o modalitate mai ușoară de a măsura performanțele lor predictive și de calcul. EDE oferă deja mecanism pentru optimizarea hiper-parametrică și metrici de evaluare personalizate, definite de utilizator, care sunt utilizate pentru adaptarea precisă a modelelor supravegheate. De remarcat este includerea unor metode ghidate de optimizare a hiper-parametrilor, cum ar fi cele bazate pe metode bayesiene și algoritmi genetici. Cu toate acestea, unele funcții necesare acolo lipsesc și au fost adăugate pentru a fi utilizate în CoCo.

Aceste caracteristici suplimentare sunt utilizate pentru o mai bună optimizare a modelelor rezultate pentru diferite cazuri de utilizare, inclusiv adaptarea precisă și detectarea evenimentelor IaC. Adaptarea autonomă și va necesita informații suplimentare din partea oricărui mecanism de detectare a evenimentelor bazat pe ML. Acest lucru ne-a condus la includerea metodologiilor preliminare explicabile AI care pot oferi teoretic informațiile suplimentare necesare.

Dintre caracteristicile implementate amintim:

Nume	Descriere
Graficul de separare a caracteristicilor	Trasează separarea claselor pe valorile metrice definite de utilizator: 
Matricea de confuzie	Trasează matricea de confuzie care detaliază nu numai predicția corectă, ci și false pozitive și false negative:  În cazul unor seturi de date dezechilibrate, aceste informații sunt extrem de valoroase. Am inclus, de asemenea, mecanisme de raportare suplimentare pentru valorile dezechilibrate, cum ar fi: precizie, reamintire, specificitate, scor F1, medie geometrică și acuratețe echilibrată cu indicele.

Importanța caracteristicilor	Indică importanța pe care o au caracteristicile asupra rezultatului predicției:  Trebuie remarcat faptul că nu toate metodele ML acceptă extragerea importanței caracteristicilor.
Curbă de învățare	Reprezintă scoruri de instruire și de validare încrucișată definite de utilizator pe baza cantității de date de instruire (adică, instanțe de instruire) furnizate: 
Curba de validare	Reprezintă scoruri de instruire și de validare încrucișată definite de utilizator pe baza intervalelor de valori definite de utilizator pentru un hiper-parametru specific: 
Eliminarea caracteristicilor recursive	Reprezintă scorul de validare obținut prin adăugarea iterativă de caracteristici în timpul antrenamentului:

	
Curba de rechemare de precizie	Oferă o perspectivă mai profundă asupra performanței unui model predictiv dat: 
Limita de decizie	Reprezintă limita de decizie pentru toate clasele în cazul unui model predictiv:  Această reprezentare necesită un set de date cu 2 sau 3 dimensiuni, astfel încât în majoritatea scenariilor mecanismele de reducere a dimensionalității trebuie utilizate pentru a reprezenta cu succes granițele de decizie (exemplul de mai sus folosește PCA cu 2 componente principale). Trebuie remarcat faptul că aceste parcele tind să fie zgomotoase în cazul datelor cu un spațiu mare de caracteristici.

Obiectivele actuale pe termen scurt și mediu pot fi rezumate după cum urmează:

- Includerea unor metode AI explicabile suplimentare pentru o analiză îmbunătățită a cauzei principale (integrarea curentă numai cu SHAP [6]).
- Identificarea parametrilor specifici IaC și generarea de seturi de date privind anomaliile specifice IaC CoCo și antipattern.
- Implementarea modelelor de învățare federate (suportăm deja metodele de Deep Learning prin legături TensorFlow Keras)

Runtime distribuit

Un runtime distribuit creează un mediu de rulare colaborativ capabil să consume resurse eterogene atât din Cloud, cât și din Edge. În timp ce orchestrarea resurselor din spectrul Cloud este o problemă deja abordată, inclusiv resursele Edge vor pune provocări semnificative din cauza naturii resurselor Edge. În mare parte fiind instabilă, starea resurselor trebuie amenințată ca incertă, iar puterea de procesare și spațiul de stocare sunt în mod normal limitate. Pentru a rezolva aceste neajunsuri, faza de orchestrare trebuie adaptată în consecință prin combinarea acțiunilor proactive și reactive pentru a ajusta automat parametrii de execuție pentru a asigura în sfârșit un mediu de execuție normal pentru aplicație. Aceste activități sunt vag cuplate cu sarcini specifice pentru detectarea evenimentelor și monitorizarea datelor.

Monitorizare și date

Setul inițial de experimente a fost efectuat folosind datele de monitorizare colectate cu Promotheus[7], dar din cauza complexității acestei soluții ne imaginăm unele probleme la rularea clienților colector pe resursele Edge. După cum sa menționat deja, resursele Edge sunt limitate în ceea ce privește resursele hardware, prin urmare soluția de monitorizare trebuie adaptată pentru a remedia acest neajuns. În acest fel am investigat o soluție ușoară de monitorizare, Monit[8], care se potrivește bine resurselor Edge, dar îi lipsește un serviciu de colectare distribuit pentru monitorizare și colectare de date. Am dezvoltat o soluție, bazată pe MongoDB[9] și Python Flask[10], capabilă să ofere o magistrală de colectare a datelor de monitorizare distribuită în care datele de monitorizare sunt ingerate de clienții de monitorizare Edge și interogate în continuare de serviciile de monitorizare. Arhitectura propusă este prezentată în Figura 1.

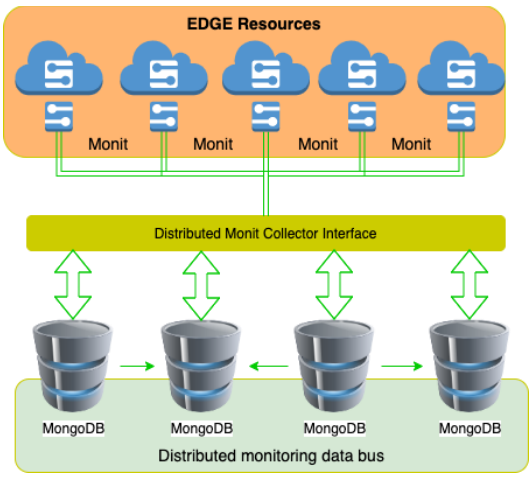


Figura 1: Arhitectura Monit distribuit

Folosind natura distribuită a MongoDB, rezolvăm, de asemenea, problema monitorizării agregării datelor, unde o bucată de date, distribuite pe diferite resurse (Cloud sau Edge, combinate),

[6] https://shap.readthedocs.io/en/latest/
[7] https://prometheus.io/
[8] https://mmonit.com/monit/
[9] https://www.mongodb.com
[10] https://flask.palletsprojects.com/en/2.0.x/

pot fi combinate cu ușurință pentru a îndeplini scopul monitorizării (de exemplu, detectarea anomaliilor).

Pentru a implementa întreaga stivă de monitorizare Edge, pentru fiecare componentă trebuie create șabloane de implementare pentru a gestiona aspectele de instalare, configurare și integrare a platformei.

Cerințele unui runtime distribuit și deployment

Runtime-ul distribuit ar trebui să satisfacă, în principal, următoarele cerințe: să fie scalabil prin utilizarea resurselor atât din sediul Cloud, cât și din cel Edge și să orchestreze aplicațiile distribuite peste resursele hibride Cloud și Edge. Soluția propusă pentru a satisface aceste cerințe este Kubernetes[11], un sistem open-source pentru automatizarea implementărilor de aplicații virtualizate și gestionarea scalabilității. În timp ce soluția de orchestrare se ocupă de stratul software, odată ce resursele sunt deja cunoscute, instalate și configurate. Ceea ce se întâmplă înainte ca stiva de orchestrație să fie implementată este un aspect provocator.

Pentru gestionarea resurselor hardware, o soluție este utilizarea Canonical MAAS[12], un software de bootstrapping rapid și de încredere (bare metal). În timp ce MAAS acceptă majoritatea resurselor hardware clasice (cum ar fi serverele normale stivuibile), nu are suport pentru dispozitivele Edge.

Pentru a susține dispozitivele Edge, trebuie create module specifice, module care se ocupă de operațiunile automate de descoperire și implementare a resurselor. În plus, după ce resursa hardware este configurată, în mod normal cu un sistem de operare de bază, stiva de software de rulare trebuie să fie implementată, pe baza profilului de implementare configurat. Pentru gestionarea acestor operațiuni JAAS[13], Canonical JUJU, este un candidat perfect deoarece permite definirea șabloanelor software speciale (juju charms) care se ocupă de întregul ciclu de viață al unei aplicații. Practic, vom folosi JUJU pentru a porni orchestratorul Kubernetes (și, dacă este necesar, alte stive de software care necesită rularea resurselor hibride Cloud și Edge).

Având Kubernetes implementat, aplicațiile distribuite sunt gestionate de Helm[14], un manager de pachete/aplicații pentru Kubernetes. Întregul concept proiectat este prezentat în Figura 2.

Pentru a valida această abordare, a fost aleasă o aplicație distribuită ca caz de utilizare pentru a fi implementată pe un grup de resurse hibride bootstrap. Aplicația este eWPS, un API modern OGC WPS 2.0/WPS – Processes[15] implementarea standard care a executat servicii geospațiale prin resurse distribuite.

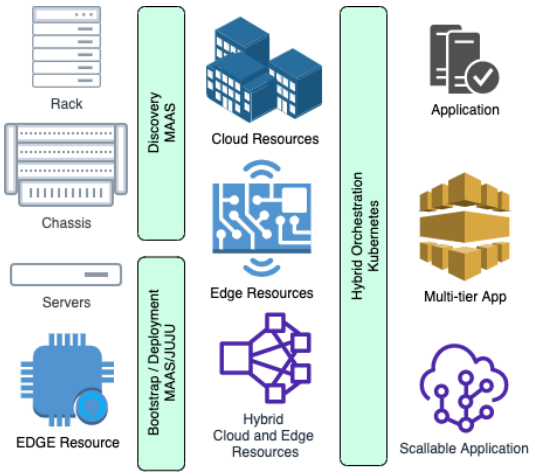


Figura 2: Imagine generală a Runtime-ului distribuit din COCO

[11] https://kubernetes.io/
[12] https://maas.io/
[13] https://jaas.ai
[14] https://helm.sh
[15] https://ogcapi.ogc.org/processes/

Aplicație: Învățare automată aplicată pla detecția vegetației în imagini satelitare

Categorizarea acoperirii și utilizării terenului (land cover/land use) prezintă un subiect de interes actual. Importanța cunoașterii și caracterizării acoperirii și utilizării terenului se extinde de la dezvoltarea planurilor de mediu, la monitorizarea efectelor produse de schimbări climatice, până la managementul dezastrelor naturale și prognozelor de mediu.

Metodele de caracterizare dezvoltate până în prezent se bazează în principal, pe metode de caracterizare manuală sau semi-automatizată. Acest lucru este costisitor din punctul de vedere al resurselor umane cât și a timpului de efectuare al caracterizărilor.

Dezvoltarea unor soluții complet automate pentru a crea inventarieri ale caracteristicilor ce acoperă suprafața Pământului sunt necesare. O abordare folosită în cadrul proiectului presupune antrenarea rețelelor neuronale complet convoluționale (FCN) pentru rezolvarea problemei. Au fost antrenate diverse topologii prezente în literatura de specialitate în domeniul Computer Vision, ale căror utilitate a fost demonstrată în contextul datelor satelitare.

Antrenarea acestor topologii a fost realizată folosind mediul de dezvoltare Hugin[16]. Setul de date folosit pentru antrenare este creat folosind date din programul ESA Copernicus[17] care au fost preluate de satelitul Sentinel 2 MSI[18] (Multi-Spectral Imagery) și inventarul Corine Land Cover[19] 2018. Pentru dezvoltarea setului de date au fost folosite imagini satelitare din 2017 până în 2021 acoperind suprafața României (selectată ca suprafață de interes).

Etapa 2: Analiza federată a datelor și învățare automata, descoperirea resurselor, sistem de execuție distribuită, adaptare automată

Scopul principal al proiectului CoCo este de a permite generarea IaC, adaptabilitatea și performanța fiind cele două caracteristici principale de ghidare. Pe baza muncii depuse în etapa 1, s-a continuat dezvoltarea componentei de detectare a evenimentelor, care permite detectarea oricăror și a tuturor evenimentelor anormale care pot avea un impact negativ asupra generației și performanței IaC. Componenta dispozitivului Edge/Fog a soluției CoCo este una dintre cerințele principale, deoarece informează o mare parte din activitatea de dezvoltare și experimentală efectuată în etapa 2. Impactul evenimentelor anormale este amplificat atunci când avem de-a face cu implementarea pe dispozitive Edge, astfel încât mecanismele de detectare ar trebui să fie implementate cât mai aproape de dispozitivele de margine. Pentru a ajuta la implementarea modelelor predictive cât mai aproape de dispozitivele Edge/Fog, am dezvoltat în continuare mecanisme de adaptare cu transprecizie care sunt capabile să se adapteze sau chiar să genereze modele predictive cu un compromis cunoscut și funcțional între timpii de inferență, costul de calcul și precizie. In etapa 2 s-a pus accent pe crearea de experimente cu privire la metodele de detectare a anomaliilor care urmează să fie incluse în motorul de detectare a evenimentelor (EDE, prezentat in etapa I) cuplate cu mecanisme de adaptare cu transprecizie. Experimentele au fost concepute și executate, care au inclus nu numai optimizarea hiper-parametrului modelului de învățare automată (ML, machine learning), ci și pregătirea și preprocesarea datelor. Testele de analiză a cauzei principale au fost, de asemenea, executate folosind calculul valorii Shapely pentru a fi ulterior inclus în mecanismele autonome de adaptare.

După cum s-a menționat anterior, IaC prezintă provocări semnificative când vine vorba de monitorizare. Anomaliile și anti-modele specifice IaC sunt greu de definit. Am ajuns la concluzia că acestea vor trebui induse pentru a evalua orice mecanisme de detectare. În urma analizelor și cercetărilor ulterioare, am ajuns și la concluzia că setul de date propus inițial în ultima perioadă de raportare este potrivit pentru această metodă inițială de evaluare comparativă. S-a ajuns la această concluzie prin simplul fapt că anomaliile utilizate sunt toate anomalii legate de hardware care sunt un numitor comun pentru orice mecanism de adaptare bazat pe calcul.

---

[16] https://pypi.org/project/hugin/
[17] https://www.esa.int/Applications/Observing_the_Earth/Copernicus
[18] https://sentinel.esa.int/web/sentinel/missions/sentinel-2
[19] https://land.copernicus.eu/pan-european/corine-land-cover

Experimente

Pe baza muncii depuse în etapa anterioară, au fost concepute o serie de experimente pentru a evalua ce este nevoie pentru a crea un flux de lucru care are ca rezultat un model predictiv bazat pe ML capabil să detecteze diverse evenimente anormale într-o varietate de scenarii din lumea reală, inclusiv considerând transprecizia.

A fost dezvoltat un set de date de anomalii surogat care se ocupă de anomalii hardware. Aceste anomalii au fost induse pe o implementare cu 4 noduri distribuite. Anomaliile sunt inductibile atât cu sau fără suprapunere (de exemplu, mai multe anomalii pot fi induse la orice interval de timp). Toate seturile de date sunt colectate la o rezoluție de 1 secundă per eveniment pentru un interval de timp total de 2 ore. Aceste seturi de date sunt concepute pentru a facilita crearea de modele predictive ML supravegheate folosind un set de date dezechilibrat, evenimente anormale reprezentând sub 1% din cantitatea totală de evenimente monitorizate.

Au fost selectate o serie de metode ML supravegheate capabile să se ocupe de clase minoritare care sunt o necesitate în scenariul actual. Aceste metode pot fi găsite în tabelul 1.

Tabel 1 Descrierea metodelor

Numele metodei	**Descriere**
AdaBoost	Boostingul adaptiv este un meta-algoritm de clasificare statistică. Aici predicțiile cursanților mai slabi sunt combinate folosind o sumă ponderată ca rezultat al arborelui îmbunătățit final. S-a demonstrat că în anumite circumstanțe este mai puțin susceptibil la supraadaptare decât multe alte metode „clasice”. Astfel, este folosit în cea mai mare parte ca metodă ML „out of box” de bază.
XGBoost	Este o metodă ML optimizată de creștere a gradientului distribuit. Are mai multe caracteristici care îl fac potrivit pentru cazul nostru de utilizare. Este similar cu AdaBoost, deoarece se bazează și pe arbori de decizie și asamblare. În multe benchmark-uri și competiții online, XGBoost este o componentă cheie a celor mai performante soluții.
LightGBM	Este foarte asemănător cu XGBoost, incluzând unele dintre aceleași caracteristici. Diferența majoră este în modul în care crește arborii de decizie. În loc să crească rând cu rând (la nivel de arbore), selectează un nod frunză despre care crede că va produce cea mai mare scădere a pierderii.
CatBoost	Este o altă metodă de creștere a gradientului destul de similară cu ultimele două metode. Una dintre diferențele sale majore este că generează arbori simetrici, ceea ce în unele cazuri reduce atât supraadaptarea, cât și timpul de calcul. De asemenea, are metode specifice care facilitează tratarea datelor categorice.
RandomForest	Este una dintre cele mai populare metode de clasificare a ansamblurilor bazate pe arbore de decizie. În mod similar, pentru AdaBoost este utilizat pe scară largă ca metodă de bază „out-of-box”. Cu toate acestea, este notoriu dificil să se acorde anumite domenii cu probleme și se comportă destul de prost cu caracteristici categorice.
DNN	Aceasta reprezintă o rețea neuronală bazată pe Deep Learning implementată folosind Tensorflow. Topologia DNN dezvoltată pentru aceste experimente este în mare măsură rezultatul mecanismelor de optimizare a hiper-parametrului.

Fiecare dintre metodele descrise în tabelul 1 trece prin același flux de lucru experimental:

1. Mai întâi se încarcă setul de date și sunt executați câțiva pași de preprocesare. Acești operatori se asigură că datele de instruire sunt de înaltă calitate și valide. Toate caracteristicile de date lipsă sunt gestionate în funcție de necesitățile fiecărei metode ML. Unele metode, cum ar fi XGBoost, pot gestiona tipuri de date N/A, în timp ce altele nu. În sfârșit, scalarea datelor este aplicată pentru a aduce toate punctele de date în intervalul [0,1].
2. Fiecare metodă va suferi o optimizare hiper-parametrică. Metoda aleasă pentru a realiza acest pas de optimizare se bazează pe un algoritm genetic implementat folosind biblioteca DEAP[20]. Am stabilit parametrii algoritmului genetic după cum urmează: dimensiunea populației 40, generațiile

[20] https://deap.readthedocs.io/en/master/

30, probabilitatea de mutație 0,2, probabilitatea de încrucișare 0,5. Funcția de scor este F1, iar validarea încrucișată este setată la Stratified Kfold cu numărul de divizări setat la 4. Aceste setări ne permit să evaluăm în mod adecvat spațiul hiper-parametru. Mărimea reală a acestui spațiu de căutare diferă în funcție de metoda ML optimizată. De obicei variază de la 4000 la 9000 de soluții candidate. Deoarece aceasta este o metodă de optimizare ghidată, nu toate soluțiile candidate vor fi evaluate, deoarece nu este nevoie să faceți acest lucru.

3. Odată ce este identificată cea mai performantă configurație de hiper-parametru, antrenăm un model separat folosind acești parametri. Apoi se fac pași suplimentari de optimizare, rezultând curbe de antrenament și validare care descriu mai bine performanța fiecărui model. Aceste informații pot fi apoi utilizate pentru crearea unui depozit de modele predictive ale căror timpi de inferență, amprenta de calcul și performanță predictivă sunt cunoscute.

Din motive de concizie, în următoarele paragrafe vom descrie rezultatele unui singur model ML antrenat în acest mod. Cu toate acestea, este de reținut faptul că avem aceleași rezultate pentru toate modelele ML enumerate în această secțiune. Calculele de valoare Shapely nu sunt prezentate aici, deoarece acestea nu sunt relevante pentru experimentele curente. Experimentele viitoare vor include acest pas de analiză post antrenament.

Rezultatele obținute pentru AdaBoost vor fi detaliate în paragrafele următoare. Pentru metoda HPO am selectat o serie de parametri ai metodei pentru tuning; numărul de estimatori, rata de învățare și algoritmul. Odată ce toate configurațiile viabile au fost explorate, optimizarea este oprită automat.

Figura 3 prezintă matricea de confuzie pentru cel mai performant model AdaBoost. Putem vedea că, deși algoritmul a funcționat bine. Numărul de clasificări greșite este încă relativ mare.

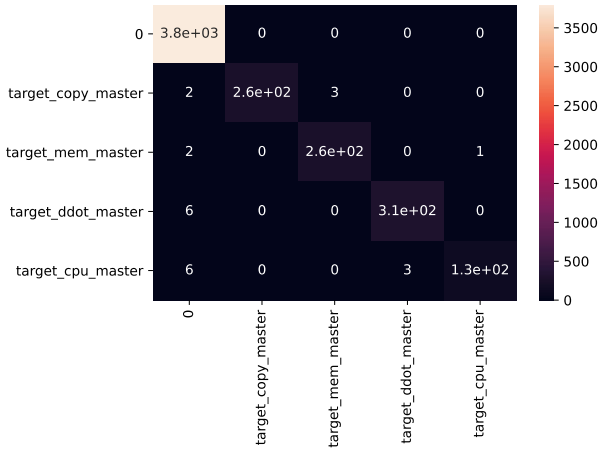


Figura 3 Confusion Matrix AdaBoost după HPO

Performanța modelului este de asemenea vizibilă în Tabelul 2. Aceste scoruri au fost obținute utilizând un clasificator AdaBoost cu 200 de estimatori folosind algoritmul de amplificare „SAMME” și o rată de învățare de 0,1. A fost nevoie de metoda HPO 852 de iterații pentru a găsi cea mai bună soluție candidată.

Timpul total de execuție HPO este puțin mai dificil de cuantificat, deoarece depinde în mare măsură de numărul de estimatori și rata de învățare stabilită pentru fiecare candidat, împreună cu faptul că se efectuează o validare încrucișată de 4 ori pentru toate soluțiile candidate rezultate. În cea mai performantă soluție candidată, instruirea și validarea încrucișată de patru ori au durat 11,7 s.

Selectăm acest model cu cele mai bune performanțe pentru analiză și optimizare ulterioară. Acest lucru va presupune o inginerie a caracteristicilor și o ajustare suplimentară a hiper-parametrilor. Candidatul inițial este un punct de plecare foarte bun și este adesea greu de optimizat în continuare. Cu toate acestea, deoarece dorim să descriem complet performanța modelului luând în considerare nu doar performanța predictivă, ci și cerințele de calcul, acest pas este necesar.

Tabel 2 Raport de clasificare a dezechilibrelor

Precision	Recall	Specificity	f1	Geometric Mean	Index Balance Accuracy	Support
0.99581	1.00000	0.98395	0.99790	0.99194	0.98553	3803.00000
1.00000	0.98141	1.00000	0.99062	0.99066	0.97959	269.00000
0.98881	0.98881	0.99934	0.98881	0.99406	0.98711	268.00000
0.99045	0.98107	0.99933	0.98574	0.99016	0.97863	317.00000
0.99259	0.93706	0.99979	0.96403	0.96792	0.93099	143.00000
0.99520	0.99520	0.99520	0.99520	0.99520	0.99520	0.99520
0.99521	0.99521	0.99521	0.99521	0.99521	0.99521	0.99521
0.98720	0.98720	0.98720	0.98720	0.98720	0.98720	0.98720
0.99517	0.99517	0.99517	0.99517	0.99517	0.99517	0.99517
0.99116	0.99116	0.99116	0.99116	0.99116	0.99116	0.99116
0.98321	0.98321	0.98321	0.98321	0.98321	0.98321	0.98321
4800.00000	4800.00000	4800.00000	4800.00000	4800.00000	4800.00000	4800.00000

Figura 4 arată matricea de confuzie și importanța caracteristicilor pentru cea mai performantă ori în timpul celei de-a doua etape de antrenament pentru modelul AdaBoost. Putem face câteva observații interesante. În primul rând, numărul de caracteristici care influențează performanța predictivă a modelului este destul de limitat, cel puțin cu cele 4 anomalii prezente în setul de date. În al doilea rând, matricea de confuzie arată că există mult mai puține clasificări greșite. Acest lucru se datorează în mare parte faptului că trasăm matricea de confuzie numai pe un set de rezistență care nu este utilizat în timpul antrenamentului sau chiar ca set de validare.

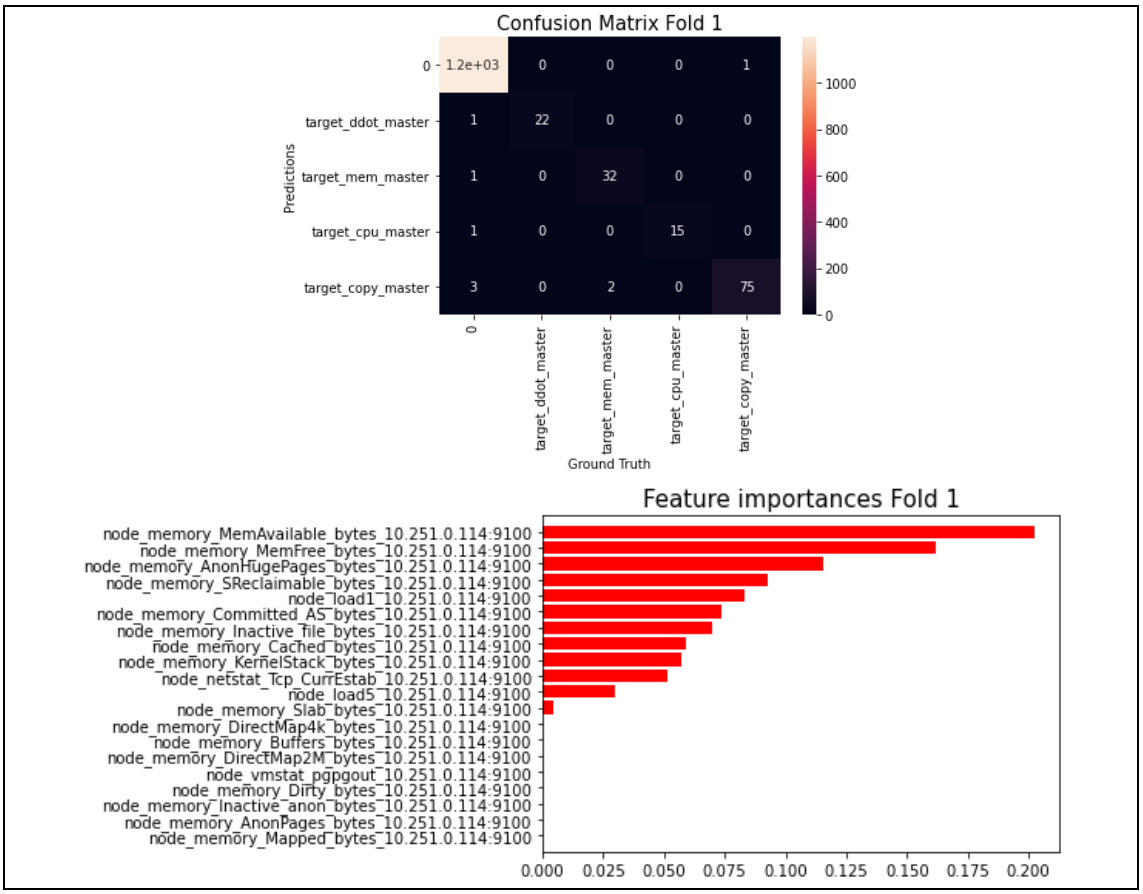


Figura 4 Cele mai bune performanțe ale modelului AdaBoost

Au urmat mai multe experimente de validare încrucișată de fiecare dată când am schimbat o valoare a parametrului. Acest lucru ne permite să vedem cum acești parametri influențează performanțele modelului.

Figura 5 prezintă curba de validare pentru mai multe valori ale ratei de învățare. Punctele reprezintă scorul median obținut în cazul fiecăruia dintre etapele de validare încrucișată de 5 ori. Putem observa că cea mai bună performanță a modelului se obține cu o rată de învățare de 0,1 la fel ca cea calculată prin metoda HPO. Ratele de învățare mai mari pot produce scoruri mai mari în cazul unora dintre faldurile de validare încrucișată, totuși, în medie, acestea sunt mai mici. Acest hiper-parametru afectează de obicei timpul de antrenament în mod semnificativ.

Figura 6 prezintă curba de validare pentru numărul de estimatori. Putem observa cu ușurință că performanța este slabă atât pentru numărul mic, cât și pentru cel mare de estimatori. De fapt, cea mai bună performanță a fost obținută în intervalul de la 100 la 1000 de estimatori.

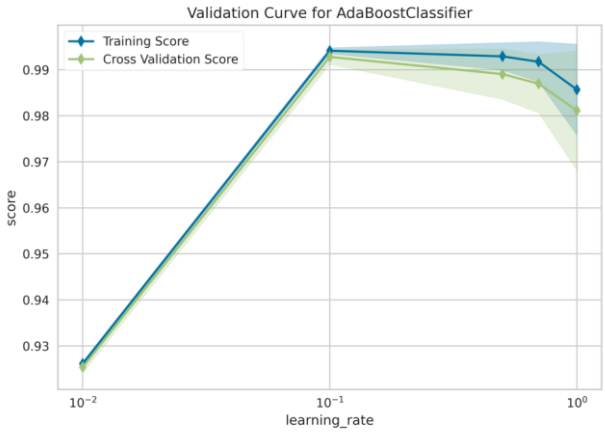


Figura 5 Curba de validare pentru rata de învățare

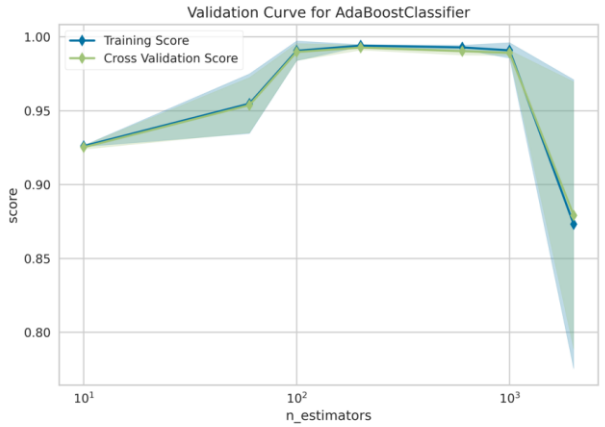


Figura 6 Curba de validare a numărului de estimatori

Ar trebui să remarcăm că numărul de estimatori afectează direct cerințele de calcul ale modelului. Un număr mai mare de estimatori generează un model mai mare, care este mai greu de antrenat și durează mai mult pentru a instanția. Cu toate acestea, există un compromis de luat în considerare. Numărul scăzut de estimatori poate produce performanțe slabe, deoarece modelul nu poate învăța pe deplin modelele prezente în date. Numărul mare de estimatori poate duce la supraadaptarea modelului, ceea ce duce la o performanță slabă a eșantionului.

Figura 7 arată curba Precision-Recall care este utilizată pentru a evalua calitatea ieșirii clasificatorului pentru toate cele 5 clase posibile. Precizia este o măsură a relevanței rezultatelor, în timp ce reamintirea este o măsură a câte rezultate cu adevărat relevante sunt returnate. O zonă mare

de sub curbă reprezintă atât o reamintire ridicată, cât și o precizie ridicată. Precizia ridicată indică rate scăzute de fals pozitive, în timp ce reamintirea ridicată indică rate scăzute de fals pozitive. În cazul nostru, vedem că există unele probleme de identificare a evenimentelor anormale legate de CPU. Acest lucru poate fi explicat prin faptul că alte anomalii induse, cum ar fi ALU și anomalii de memorie, au o componentă CPU.

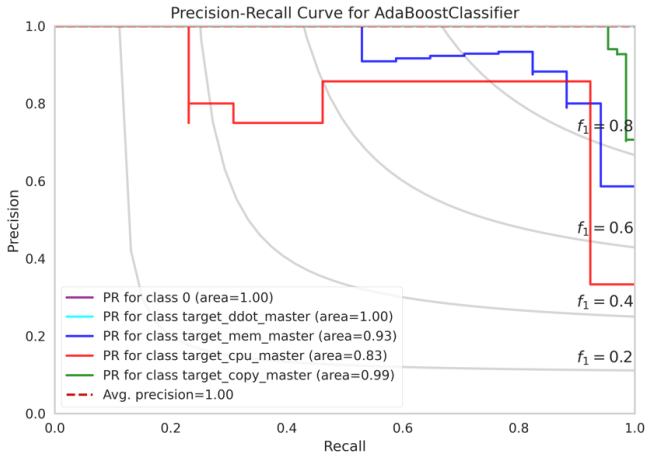


Figura 7 Curba Precision-Recall pentru AdaBoost

În continuare, am dorit să cuantificăm efectele pe care datele le au asupra performanței predictive a modelului. Luăm cei mai performanti hiper-parametri ai modelului și antrenăm diferite instanțe de model predictiv pe diferite cantități de date din setul de antrenament. Figura 8 arată efectul pe care acest lucru îl are asupra modelelor AdaBoost. Numărul redus de instanțe de antrenament generează modele instabile. Cu toate acestea, după ce se atinge aproximativ jumătate din instanțele de antrenament inițiale, performanța modelului se stabilizează. Acest lucru ne conduce la concluzia că performanța predictivă acceptabilă a modelului poate fi atinsă cu un set de antrenament inițial relativ mic.

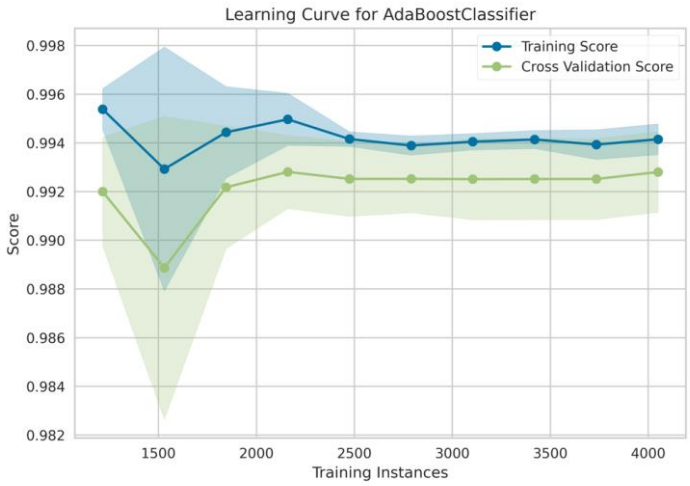


Figura 8 Instanțe de antrenament pentru curba de învățare

Figura 9 arată rezultatul eliminării caracteristicilor recursive. Aici luăm importanța caracteristicii calculată de AdaBoost și clasificăm fiecare caracteristică în consecință. În continuare, antrenăm un model AdaBoost cu caracteristica cel mai bine clasată, apoi antrenăm un al doilea model cu cele două caracteristici cel mai bine clasate și așa mai departe, până când ne antrenăm cu întreg spațiul de caracteristici. Putem vedea că este necesar un număr relativ scăzut de caracteristici pentru a obține o performanță predictivă bună (12 caracteristici).

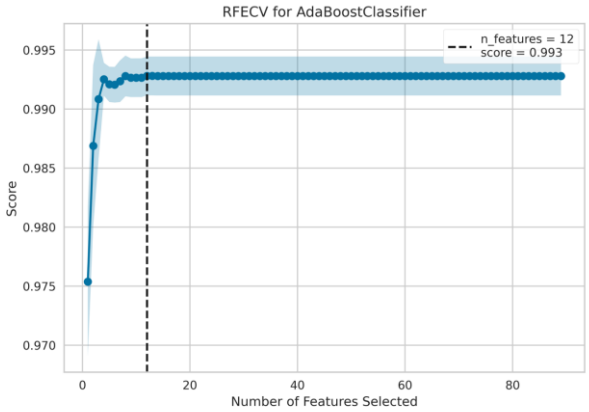


Figura 9 Eliminarea caracteristicilor AdaBoost

Unele metode de detectare ML nu au capacitatea de a returna importanța caracteristicilor după antrenament. Pentru aceste scenarii pot fi aplicate alte metode pentru a aproxima acest lucru. După cum s-a menționat anterior, valorile Shapely pot fi utilizate pentru a cuantifica importanța caracteristicilor pentru un anumit model predictiv. Poate fi folosit chiar și pe bază de predicție, calculând importanța caracteristicii doar pentru o singură predicție. Deși această metodă este o bună aproximare, este extrem de intensivă în resurse și necesită un model pregătit în prealabil pentru a calcula.

Alte metode de selectare a caracteristicilor pot fi utilizate direct pe setul de date, care încearcă să extragă importanța caracteristicilor din datele în sine. În multe dintre aceste metode, definiția motivului pentru care o anumită caracteristică este importantă este drastic diferită de ceea ce ar putea necesita o metodă ML. Chiar și așa, metode precum Kurtosis, Diferența medie absolută sunt ușor de aplicat în funcție de caracteristică. În experimentele noastre actuale, am folosit valori Shapely pentru a clasifica caracteristicile pentru modelul nostru DNN, care este singurul care nu este capabil să calculeze direct importanța caracteristicilor.

Este important de reținut că în majoritatea scenariilor reducerea caracteristicilor nu este la fel de evidentă ca în acest caz. Avem doar 5 clase posibile fiecare având un efect clar asupra unui număr mic de metrici. În plus, o astfel de eliminare a caracteristicilor agresive nu este recomandată. Este de preferat o abordare mai conservatoare.

În schimb, un spațiu de caracteristici cu dimensionalitate scăzută necesită, de asemenea, mai puține resurse de calcul atât pentru antrenament, cât și pentru inferență.

Discutie

Experimentele prezentate reprezintă doar o mică parte din munca totală depusă în această perioadă de raportare. Tabelul 1 enumeră toate metodele ML utilizate în timpul experimentelor. Rezultatele arată că toate metodele testate sunt potrivite pentru detectarea evenimentelor anormale în seturi de date extrem de dezechilibrate. În plus, am obținut un depozit mare de modele predictive preantrenate care pot fi utilizate/desfășurate într-un mediu Edge/Fog.

Cele mai performante modele au fost obținute de XGBoost, CatBoost și DNN. Modelele bazate pe arborele de decizie sunt, în general, mai ușor de antrenat, deși pot suferi de capacități de generalizare slabe, în timp ce modelul bazat pe DNN durează mai mult pentru antrenament, dar au o capacitate de generalizare foarte bună.

Experimente viitoare pot include metode nesupravegheate, deoarece acestea nu necesită un set de date etichetat pentru a fi instruit. Acest lucru este benefic în majoritatea scenariilor din lumea reală în care evenimentele anormale sunt destul de rare și etichetarea datelor este o întreprindere costisitoare. În cazul metodelor nesupravegheate, metodele AI explicabile sunt cruciale, deoarece cauza principală pentru care un anumit punct de date este considerat anormal nu este imediat evidentă. Pentru metodele supravegheate putem extrage importanța caracteristicilor care ne spune

care este metoda predictivă folosită pentru a face alegerea utilizând valori Shapely, putem crea un clasament similar pentru toate metodele predictive, inclusiv cele nesupravegheate.

Odată ce avem un flux de lucru viabil de detectare a anomaliilor, îl putem folosi pentru adaptare autonomă. Această componentă de adaptare va fi modelată după paradigma Credință-Dorință-Intenție (BDI, Belief-Desire-Intention) din AI.

Credințele (belief) sunt fapte cunoscute despre mediul care este modelat. În cazul nostru, mediul este reprezentat de infrastructura IaC și orice aplicație care rulează pe aceasta. Modelarea acestui mediu se face folosind datele din soluția noastră de monitorizare și evenimentele anormale detectate.

Dorința (desire) este modelată ca obiectiv general al sistemului, fie că este legat de performanță, securitate sau fiabilitate. Dacă convingerile actuale nu sunt în concordanță cu dorințele sistemelor, este necesară adaptarea stării. Această adaptare se realizează folosind intenții care sunt modelate ca planuri compuse din acțiuni individuale pe care sistemul le poate pune în aplicare pentru a-și atinge dorințele declarate. Pentru IaC, intențiile sunt ușor de implementat, deoarece are control direct asupra infrastructurii și aplicației monitorizate (într-o oarecare măsură).

Pentru a realiza pe deplin un model BDI trebuie să formalizăm un set de acțiuni atomice care pot fi folosite pentru a construi planuri de remediere. Odată ce aceasta este pusă în aplicare, ne putem concentra pe modelarea dorințelor generale ale sistemului. Acesta va fi punctul central al muncii noastre viitoare.

Modelare

Specificația TOSCA a fost extinsă de echipa CoCo pentru a trata dispozitivele Edge, resurse Fog, acceleratoare hardware și diferite modelele de implementare pe care le folosesc (de exemplu, containere, software HPC). Extensia a fost numită TUFA, TOSCA Using Flexible Accelerators). Un repositoriu GitHub[21] stochează documentația pentru specificația TUFA și exemple de utilizare.

TOSCA (Topology Specification for Cloud Applications)[22] este o specificație OASIS pentru modelarea unei stive de aplicație și automatizarea implementării și gestionării acesteia în medii Cloud. Portabilitatea este esențială pentru TOSCA. O topologie TOSCA constă dintr-un set de nodurile și relațiile dintre ele. Relațiile între noduri sunt definite din punct de vedere al cerințelor nodurilor și capacități. O topologie validă are cerințele de fiecare nod legat de capacitățile expuse de alții. O topologie TOSCA poate fi implementată automat folosind diferite medii de execuție (de exemplu, Clouds, local VM). Un dezavantaj major al specificației actuale TOSCA este lipsa definițiilor pentru noile modele de implementare precum Containers
și resurse specializate, cum ar fi GPU-uri și clustere HPC.

Specificaţia TUFA permite modelarea infrastructurii (VM, containere, servere tradiționale, hardware Acceleratoare), servicii și relațiile lor.

Specificația TUFA a fost realizată de sub-echipa CoCo care lucrează si la H2020 SERRANO - Transparent Application Deployment in a Secure, Accelerated and Cognitive Cloud Continuum[23] și este folosită în contextul aplicațiilor proiectului SERRANO.

In plus față de specificația TUFA (comună CoCo și SERRANO), au fost rezilizate în CoCo:

- proiectarea și implementarea procesului Serviciului de Optimizare care îmbunătățește flexibilitatea atât a utilizatorului și furnizorii de resurse
- două pluginuri Alien4Cloud care se ocupă de procesul Serviciului de Optimizare, implementează aplicația și monitorizează implementarea.

Pentru a reduce presiunea asupra utilizatorului final al Cloudui, a fost introdus conceptul de optimizare a aplicațiilor și serviciilor abstracte. O aplicație abstractă este o topologie TOSCA care conține cel puțin un serviciu abstract. Procesul serviciului de optimizarea înlocuiește serviciile

[21] https://github.com/adispataru/tufa

[22] T. Binz, U. Breitenb¨ucher, O. Kopp, and F. Leymann, "Tosca: portable automated deployment and management of cloud applications," in *Advanced Web Services*. Springer, 2014, pp. 527–549.

[23] https://ict-serrano.eu/

abstracte, selectând o implementare adecvată pe baza unor constrângeri stabilite de utilizator și obiectivele de afaceri ale furnizorilor de resurse.

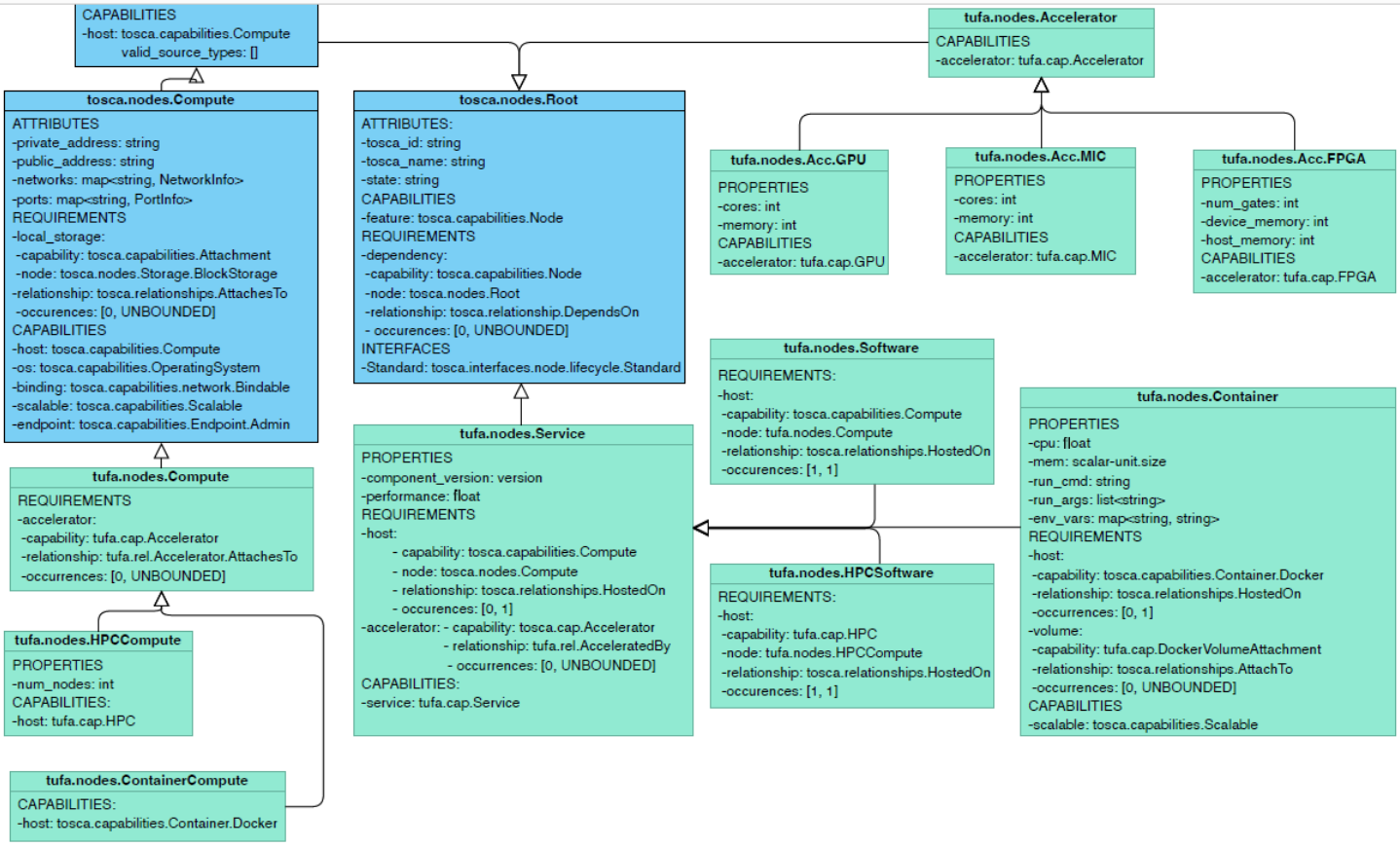


Figura 10 Diagrama de clasă pentru nodurile TUFA și relația lor cu specificația TOSCA (din doi: 10.1109/COMPSAC54236.2022.00185)

Aplicații

Experimentele ML care au fost expuse mai sus. au inspirat extinderea comparației tehnicilor ML pentru un domeniu practic: segmentarea imaginii prin teledetecție. Sub-echipa CoCo care lucrează la un asemenea topic a realizat o comparare a 4 topologii de învățare profundă de ultimă generație utilizate pentru segmentarea semantică a imaginilor. Detalii sunt prezentate in preprintul disponibil la doi:10.20944/preprints202211.0226.v1. Prezentarea informală a rezultatelor s-a facut la un workshop GRASS de la FOSS4G 2022 (Free and Open Source Software for Geospatial).

Un caz particular al IaCului este Data as a Code. Data as Code reprezintă un nou concept de vizualizare a datelor inspirat de DevOps, unde în ceea ce privește modificările de cod făcute des, automatizarea și colaborarea sunt cheia. Datele pot fi versionate împreună cu modelul și metadatele de învățare automată corespunzătoare. Datele permit inginerilor să revină la un anumit punct în timp pentru actualizări și revizuire a setului de date. Aplicația vizată de lucrarea de dizertație „Data as Code in the context of Cloud Continuum" îndrumată de directorul proiectului CoCo, se referă la recunoașterea activității umane, este o problemă de identificare a mișcării, acțiunea oamenilor si implică metode de clasificare care se bazează pe un proces care extrage caracteristici din datele brute pentru a alimenta un model de învăţare automată. Scopul este de a identifica mișcările umane cum ar fi mersul, șezutul, stând, mergând sus, coborând pentru a putea anticipa în continuare nevoile umane și îmbunătățirea calității vieții, de exemplu controlând lumina și temperatura camerei, fitness personal și monitorizarea și asistența sănătății. Un pipeline orchestrează un flux continuu de date de la margine din cloud, reprezentat de smartphone-ul care alimentează în continuare datele colectate de la utilizările sale într-un model de recunoaștere a activității umane privind achiziția de semnale bazate pe senzori. Prin pipeline de date se urmează pașii: achiziția datelor, preprocesare, segmentare,

extragere și clasificare a caracteristicilor. În scopul de a examina diferite soluții pentru a examina datele colectate de utilizator, caracteristică extrage și prezice, teza propune o abordare a învățării profunde folosind recurente rețele neuronale pentru recunoașterea activității umane și o automatizare dedicată pentru acesta, bazat pe integrarea continuă a codului, capabil să ruleze mai multe experimente în paralel deodată.

Etapa 3: Transprecizie in calcul continuu, mediu izolat și sigur de execuție, lansare și validare

Având în vedere că obiectivul principal al proiectelor CoCos este de a permite generarea IaC folosind metode care permit un grad mare de adaptabilitate și nu în ultimul rând performanță, am extins în continuare cercetările prezentate până acum. Această secțiune detaliază unele dintre progresele realizate în etapa 3 la componenta Detectare evenimente. Această componentă permite detectarea oricăror evenimente anormale care pot afecta negativ generarea și performanța IaC. Având în vedere că una dintre principalele ținte pentru CoCo sunt dispozitivele Edge/Fog, este evident că această țintă a focusat majoritatea cercetărilor și experimentelor efectuate până în acest moment. Configurația și designul experimental general este influențată în mare măsură de faptul că evenimentele anormale vor avea un impact mai mare asupra dispozitivelor Edge/Fog, astfel încât mecanismele de detectare ar trebui să fie implementate cât mai aproape de dispozitivele de margine. În ultimul raport am detaliat o parte din munca depusă în aplicarea mecanismelor de adaptare transparente care pot adapta sau chiar genera modele predictive cu un compromis cunoscut și realizabil între timpii de inferență, costul de calcul și precizie. În acest raport, explorăm obiective suplimentare de optimizare pe lângă performanța predictivă directă.

Accentul nostru principal a fost crearea de experimente folosind aceleași metode de detectare a anomaliilor incluse în Motorul de detectare a evenimentelor, împreună cu mecanisme de adaptare transparente, adăugând în același timp capacitatea de a măsura consumul de energie nu numai timpii de inferență. Acest lucru ne-a permis să îmbunătățim mecanismele de optimizare a hiper-parametrilor dezvoltate în ultima perioadă de raportare.

După cum sa menționat în raportul anterior, IaC ridică provocări semnificative pentru orice serviciu de analiză și monitorizare. Pe baza concluziilor descrise în raportul anterior, noile noastre experimente sunt concepute pentru a genera un depozit de modele predictive cu un compromis cunoscut între performanța predictivă și consumul de energie. Consumul de energie al fiecărei metode poate fi legat de complexitatea computațională, cel puțin pentru unele modele.

Design

Vom descrie în cele ce urmează designul general și scopul experimentelor noastre privind adaptarea algoritmică transprecisă. Am executat mai multe experimente al căror obiectiv principal a fost să găsim, folosind Hyper-Parameter Optimization (HPO) ghidat, cei mai buni hiper-parametri pentru fiecare model, în scopul de a maximiza performanța predictivă. Odată ce am avut cei mai performanti hiper-parametri pentru fiecare metodă ML, am proiectat un set de experimente pentru a evalua practic impactul energetic al diferitelor scenarii de antrenament și inferență pentru fiecare model. Pentru măsurarea utilizării energiei am folosit interfața RAPL (Running Average Power Limit) dezvoltată de Intel. Este utilizat pentru raportarea consumului de energie acumulat pentru diferite domenii de putere. Sunt acceptate procesoarele de calitate server, în mare parte din familia Intel Xeon (arhitectura Sandy Bridge după 2010). Aceste domenii depind în mare măsură de CPU utilizat. Figura 11 arată cum sunt organizate fiecare domeniu RAPL monitorizat. Domeniul pachetului conține informații despre nucleele CPU, memoria cache și GPU-ul integrat (dacă este disponibil). Domeniul DRAM adună informații despre memoria de lucru. Măsurătorile CPU și GPU integrate pot fi preluate folosind domeniile Core și Uncore în mod respectabil. Un domeniu suplimentar privind GPU-urile Nvidia poate fi obținut (Nvidia Volta 2018) folosind Biblioteca de management NVIDIA (NVML).

Deoarece majoritatea cluster-ului nostru HPC disponibil se bazează pe procesoare AMD (în mare parte AMD EPYC 7702), nu toate domeniile RAPL sunt acceptate. De fapt, pentru a utiliza măsurătorile RAPL, au trebuit făcute unele modificări la kernel-ul Linux. Cu toate acestea, chiar și așa

am reușit să culegem date de consum de energie doar pentru domeniul pachetului. Dacă există două socluri CPU pe hardware-ul fizic, atunci există două domenii de pachete care trebuie monitorizate (PKG 0, PKG 1). În cazul nostru, avem un singur soclu pentru serverul nostru de testare. Specificațiile complete ale infrastructurii de testare pot fi găsite mai jos (Tabelul 3).

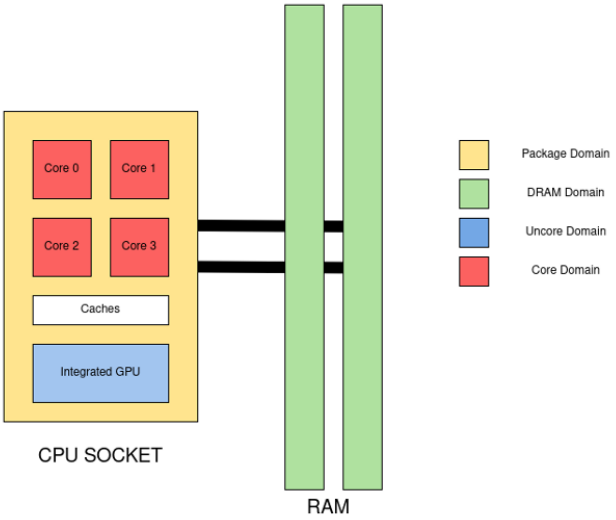


Figura 13 Domenii RAPL

Tabelul 3 Infrastructura hardware 16 x HPE Proliant DL385 Gen10

Specification	Description
CPU	128x EPYC 7702 2,0 Ghz/core
RAM	1024 GB
Storage	2x 480 GB SSD
Inter-connect (storage and communications)	2x25GbE adaptors
Operating system	Ubuntu 22.04 LTS (custom kernel)

Experimentele inițiale au fost efectuate folosind instrumentul perf Linux utilizat pentru analiza performanței. Acceptă contoare de performanță hardware, puncte de urmărire, contoare de performanță software și sonde dinamice. Un exemplu de scenariu de utilizare poate fi găsit mai jos:

```
perf stat -e ,power/energy-gpu/,power/energy-pkg/ ede.py -f config.yaml
```

Listing 1 Scenariul de utilizare perf

Exemplul anterior colectează consumurile de energie (în Jouli) pentru domeniile GPU și pachet. În timp ce utilizăm perf pentru măsurătorile de energie, am descoperit că oferă capabilități limitate de măsurare pentru cazul nostru de utilizare. De exemplu, atunci când antrenăm/validăm un model predictiv, de obicei executăm o formă de validare încrucișată, în timp ce utilizând perf ar fi necesar să împărțim fiecare divizare în apeluri de perf separate, nu există o metodă fiabilă pentru a împărți măsurătorile altfel. Acesta este motivul principal pentru care am decis să instrumentăm codul nostru pentru a oferi măsurare cât mai precisă posibil. Pentru aceasta am folosit pyJoules. Această bibliotecă ne permite să măsurăm anumite fragmente de cod. În cazul nostru am dorit să măsurăm doar antrenamentul și predicția modelului, ignorând operatorii de preprocesare. Pentru fiecare plica de validare încrucișată, colectăm antrenament, consumul de energie de inferință și durata. Măsurăm în continuare consumul total de energie folosind perf ca formă de verificare a stării de spirit.

Experimente și rezultate

În special pentru aceste experiențe, am folosit cei mai buni parametri de metodă obținuți după o optimizare extensivă, ghidată, hiper-parametrică a fiecărei metode de detectare a metodei descrise în perioada de raportare anterioară. Experimentele actuale pot fi împărțite în 3 faze distincte. Fiecare dintre aceste faze este concepută pentru a descrie pe deplin compromisul dintre consumul de energie și performanța predictivă.

În prima fază, executăm pentru fiecare algoritm o validare încrucișată de 10 ori folosind metoda stratificată pentru împărțirea seturilor de antrenament și validare. Acest lucru ne permite să menținem aceeași distribuție de clasă pentru aceste seturi ca și setul original. Dacă am împărți datele aleatoriu, este posibil ca unele clase să fie subreprezentate sau chiar să lipsească din seturile de instruire sau validare. Acest lucru va duce la o evaluare a performanței distorsionate. Scopul principal al acestei faze este de a obține o linie de bază a performanței, văzând că cei mai buni hiper-parametri sunt utilizați în acest pas.

În a doua fază, executăm o eliminare recursivă a caracteristicilor pentru fiecare metodă. Antrenarea modelelor predictive inițial pe o caracteristică, apoi adăugând la fiecare iterație o caracteristică suplimentară până când obținem spațiul complet de caracteristici. Acest lucru ne permite să verificăm modul în care dimensiunea datelor de intrare afectează atât consumul de energie, cât și performanța predictivă.

Figura 12 și Figura 13 arată rezultatele privind consumul de energie în timpul antrenamentului și, respectiv, inferența. Putem vedea că AdaBoost are o creștere aproape liniară a consumului de energie, în timp ce RandomForest și într-o oarecare măsură CatBoost prezintă aproape aceeași utilizare a datelor, independent de spațiul de intrare. XGBoost și LightGBM au un consum de energie relativ zgomotos. Dintre toate metodele, cel mai mare consum de energie a fost obținut de XGBoost, în timp ce cel mai scăzut este CatBoost. Trebuie menționat că pentru fiecare iterație se folosește aceeași metodologie de validare încrucișată de 10 ori, ceea ce înseamnă că în total au fost antrenate 10 * 89 de modele în timpul acestui proces. Figura 14 prezintă scorurile obținute pentru fiecare iterație. Putem vedea că ambele metode funcționează bine cu un spațiu de caracteristici relativ mic, LightGBM performând de fapt mai bine în comparație cu modelul CNN. Acest lucru este de așteptat, deoarece modelele Deep Lerning sunt renumite pentru că necesită o cantitate substanțială de date de antrenament.

În faza 3 luăm spațiul de caracteristici definit în timpul HPO și vedem ce impact are fiecare valoare a parametrului asupra performanței predictive și a consumului de energie. Figura 15 arată impactul valorilor parametrilor asupra consumului de energie. Putem vedea clar că unii parametri au un impact mult mai mare decât alții, un model clar poate fi observat pentru toate metodele. Aparent, modelele de învățare profundă sunt valori aberante. Acest lucru poate fi explicat oarecum din cauza inițializării GPU și a grupării datelor din memoria de lucru în GPU.

Din motive de concizie și simplitate, nu vom enumera rezultatele pentru fiecare metodă și parametrii acesteia. În schimb ne vom concentra pe LightGBM din simplul motiv că are o performanță destul de robustă atât din punct de vedere al predicției, cât și din punct de vedere al consumului de energie.

Ar trebui să menționăm în acest moment că toate experimentele au fost efectuate folosind o sămânță aleatorie fixă pentru a ajuta la reproductibilitatea experimentului. Această sămânță fixă a fost folosită pentru generarea diviziunilor de instruire și validare pentru validarea încrucișată și pentru inițializarea metodei. Un efect secundar interesant este modul în care unele perechi de seturi de date de antrenament și validare în timpul validării încrucișate provoacă creșteri ale consumului de energie. Acest lucru are loc în special în timpul antrenamentului și este legat de unele caracteristici de bază ale datelor. Cauza exactă nu este încă pe deplin înțeleasă și necesită cercetări suplimentare. În acest moment, cea mai probabilă cauză a acestui lucru ar putea fi legată de evenimente specifice (adică rânduri) și distribuția datelor și/sau entropia.

Figura 14 Eliminarea caracteristicilor recursive - Antrenament

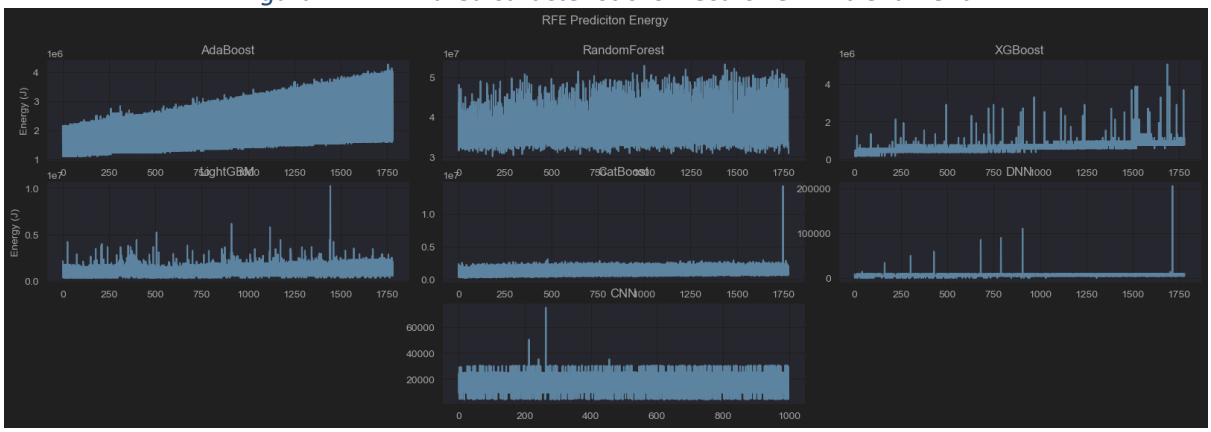


Figura 15 Eliminarea caracteristicilor recursive - Inferență

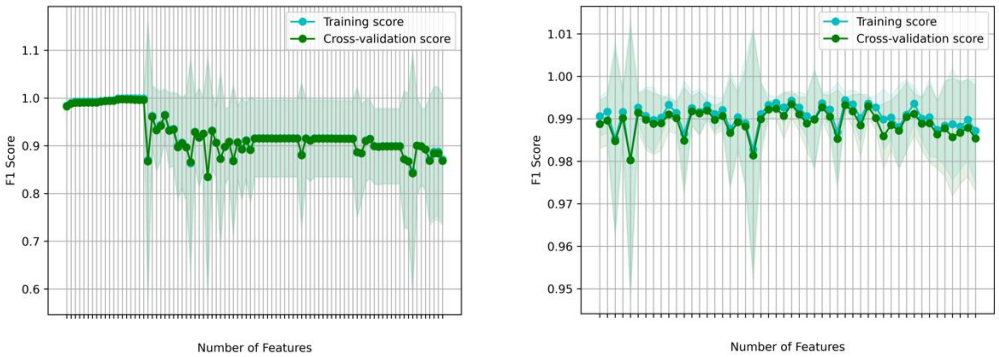


Figura 16 Scoruri RFE pentru LightGBM și CNN

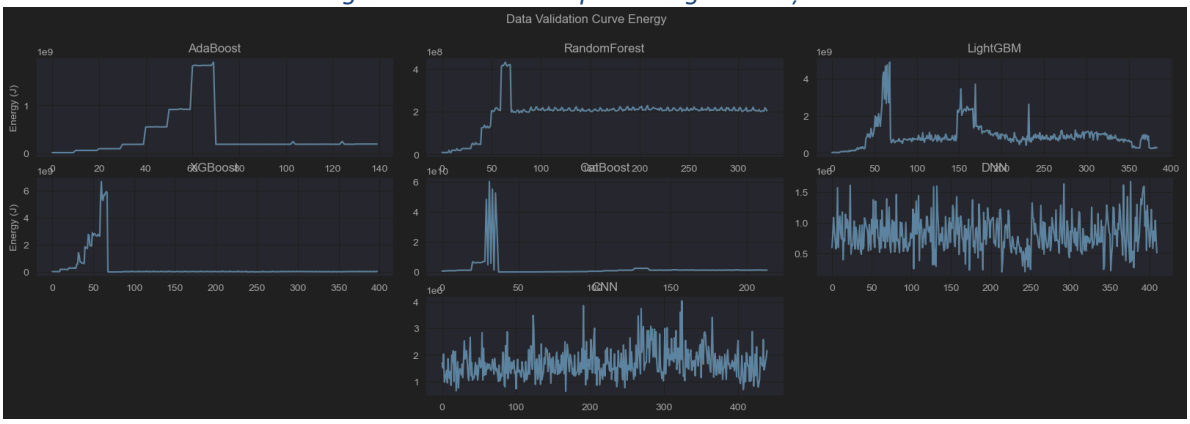


Figura 17 Curbele de validare pentru toate valorile hiper-parametrului

Figura 16 arată consumul de energie (partea stângă) și performanța predictivă folosind scorul F1 (partea dreaptă). Tabelul 4 prezintă un raport de clasificare în cazul LightGBM, acesta reprezintă o defalcare detaliată folosind mai multe măsuri de punctare, inclusiv scorul F1 separat pentru fiecare clasă parțial. Coloana „suport” arată, de asemenea, numărul de apariții ale fiecărei clase în datele de validare. Aceste tipuri de rapoarte sunt generate pentru fiecare rând de antrenament în timpul validării încrucișate.

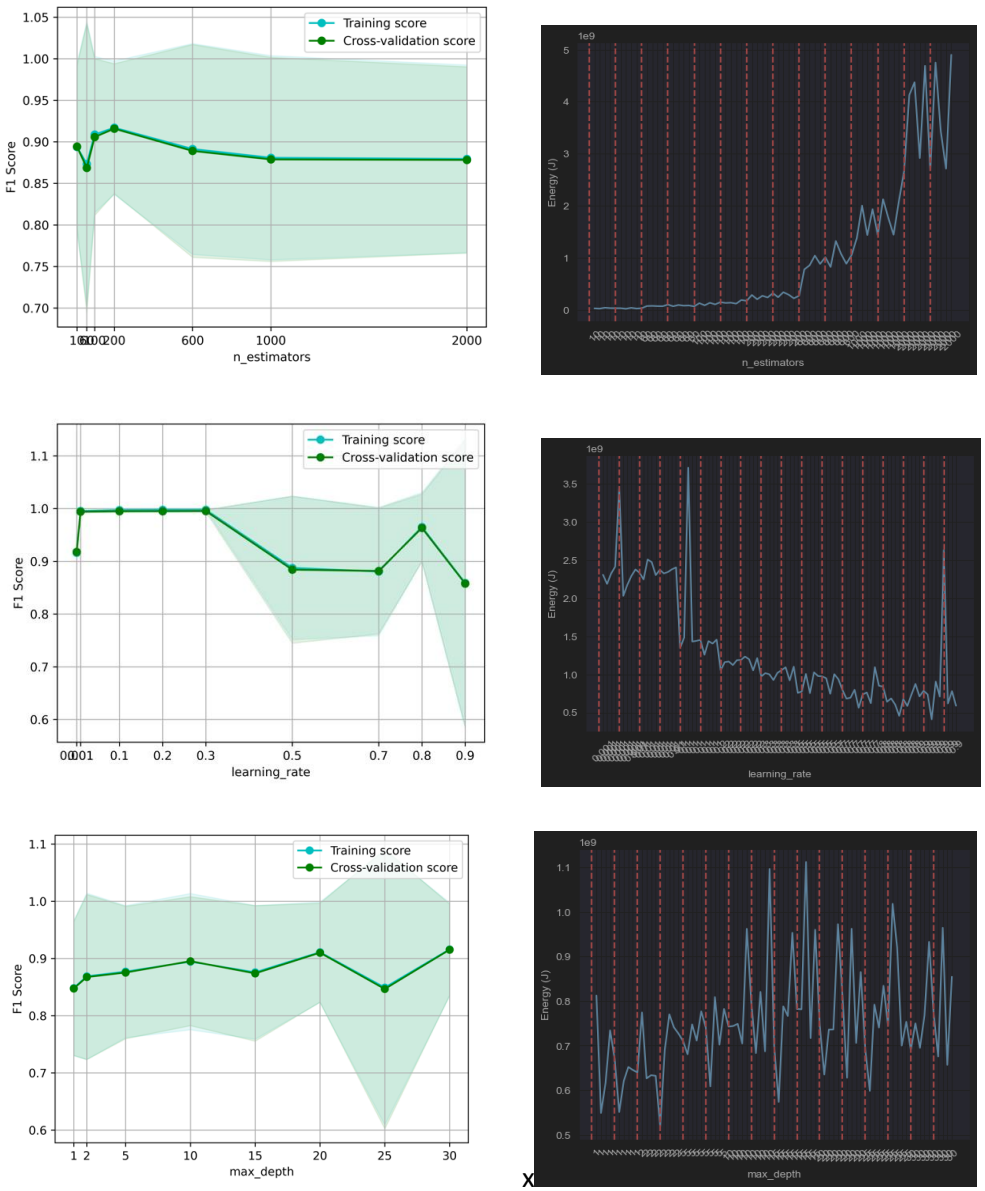


Figura 18 Consumul de energie LightGBM pe valoarea parametrului și Scorul F1 pe parametru

Tabel 4 Raport de clasificare pentru LightGBM (Fold 6)

classes	precision	recall	specificity	F1	geo	iba	support
normal	0.998	0.998	0.991	0.998	0.995	0.991	959.0
ddot	1.0	1.0	1.0	1.0	1.0	1.0	18.0
mem	1.0	0.923	1.0	0.960	0.960	0.915	26.0
cpu	0.928	1.0	0.999	0.962	0.999	0.99	13.0
copy	0.984	1.0	0.999	0.992	0.999	0.99	64.0
avg/total	0.997	0.997	0.997	0.997	0.997	0.997	0.997

Discuție

Rezultatele prezentate în această secțiune ne permit să creăm un depozit atât de modele predictive pre-antrenate, cât și un set de hiper-parametri viabili, care oferă un compromis posibil între performanța predictivă și consumul de energie. În plus, am observat o corelație puternică între timpii de antrenament și de inferență în ceea ce privește consumul de energie. Acest lucru ne conduce la concluzia preliminară că cel puțin când vine vorba de metodele pe care le-am testat, un consum mai mic de energie duce la timpi de inferență mai mici. Cu toate acestea, aceasta este doar o ipoteză în această etapă, deoarece am folosit doar o singură configurație hardware pentru testare. Trebuie efectuat un experiment extins care să ia în considerare toate arhitecturile majore actuale ale CPU, X86_64, ARM, Power. În cazul modelelor bazate pe Deep Learning, optimizarea transprecisă suplimentară după antrenament poate fi aplicată și sub formă de cuantificare a greutății, grupare de greutate. Aceste optimizări nu au fost aplicate în această perioadă de raportare și vor necesita cercetări și experimente suplimentare. Aplicarea calculului valorii Shapely pentru fiecare predicție oferă o clasare a cât de influentă este fiecare caracteristică în determinarea/detectarea oricăror evenimente anormale. Deși acest lucru nu este neapărat în conformitate cu paradigma inițială Belief Desire Intention (BDI), pe baza experienței noastre empirice, considerăm că este suficient pentru planificarea unor acțiuni semnificative de remediere a codului IaC.

Aplicație

Setul de date AgriSen-COG propus subliniază volumul semnificativ de date de observare a Pământului și provocările în procesarea lor eficientă, Crearea unui set de date pe scară largă, multi-temporale și multi-țari pentru cartografierea culturilor prezintă capacitatea tehnică care se aliniază cu obiectivul științific al proiectului COCO de a permite calcularea eficientă a continuumului cloud-to-edge. Abordarea de a crea un set de date de referință și de a folosi metode de detectare a anomaliilor contribuie cu informații valoroase la proiectul COCO în gestionarea eterogenității resurselor și orchestrarea infrastructurii. Investigația privind detectarea culturilor folosind modele de învățare profundă din articol are implicații de anvergură pentru politicile agricole și cadrele socio-economice. Acest lucru rezonează cu scopul proiectului COCO de a influența domeniile științifice și socio-economice. Diseminarea constatărilor articolului și a setului de date poate servi drept model pentru COCO în angajarea cu comunitatea științifică mai largă. Metodologia utilizată pentru crearea setului de date AgriSen-COG a presupus colectarea datelor, procesarea și detectarea anomaliilor.

3.Impactul estimat al rezultatelor obținute

Indicatorii pentru măsurarea impactul sunt legați de rezultatele diseminării, colaborările cu alte echipe și transferul de cunostiințe către generațiile de cercetători tineri.

Articole în reviste

Trei articole au fost înainte ca rezultat direct a proiectului către revista open-access Scalable Computing: Practice and Experience (SCPE, ISSN 1895-1767), indexată de Clarivate Analytics în baza de date Emerging Sources Citation Index (ESCI), listată prima dată în iunie 2021 în Q4 în Journal Citation Reports și care în iunie 2023 a primit pentru prima dată factor de impact Clarivate (1,1):

1. *D. Petcu*, Service Deployment Challenges in Cloud-to-Edge Continuum, SCPE, vol. 22, no. 3 (2021), 313–320, doi: 10.12694/scpe.v22i3.1941, WOS: 000721262900004 [4 citări]
2. *A. Spătaru*, A Review of Blockchain-enabled Fog Computing in the Cloud Continuum Context, SCPE, vol. 22, no. 4 (2021), doi: 10.12694/scpe.v22i4.1947, WOS: 000731705800005 [2 citări]
3. *A. Spătaru*, J. Aperribay, Configuration of Container deployments on the Compute Continuum using Alien4Cloud, SCPE, vol. 25, no. 1 (2024), doi: 10.12694/scpe.v25i1.2937 (in evaluare)

Un alt articol a fost înaintat la revista open-access Frontiers in Big Data, care este indexată de Clarivate Analytics în ESCI, cu ISSN 2624-909X, din iunie 2023 cu factor de impact Clarivate (3,1):

4. Kimovski D., Mathá R., *Iuhasz G.*, Marozzo F., *Petcu D.*, Prodan R., Autotuning of Exascale Applications With Anomalies Detection, Frontiers in Big Data, vol. 4, Noiembrie 2021, 101, doi: 10.3389/fdata.2021.657218, article 657218, WOS: 000891537300001

Aplicații în reviste indexate ISI în prima categorie, cu factor de impact Clarivate (5):

5. *T. Selea*, 2023. AgriSen-COG, a Multicountry, Multitemporal Large-Scale Sentinel-2 Benchmark Dataset for Crop Mapping Using Deep Learning, *Remote Sensing* 15, no. 12: 2980. https://doi.org/10.3390/rs15122980, WOS: 001018506700001

Articole în volume de conferinte

Rezultatele activităților proiectului au fost publicate în volume ale unor conferințe:

6. *A. Spătaru*, *G. Iuhasz, S. Panica*, TUFA: A TOSCA extension for the specification of accelerator-aware applications in the Cloud Continuum, 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), doi: 10.1109/COMPSAC54236.2022.00185 (B în CORE 2021), WOS: 000855983300177 [1 citare]
7. *D. Petcu*, Cognitive Cloud Continuum : Plenary Talk, 2023 IEEE 17th Inter. Symposium on Applied Computational Intelligence and Informatics (SACI), doi:10.1109/SACI58269.2023. 10158559.
8. E. Chondrogiannis, E. Karanastasis, V. Andronikou, *A. Spătaru*, A. Nanos, A. Kretsis, P. Kokkinos (2023). Intent-Based AI-Enhanced Service Orchestration for Application Deployment and Execution in the Cloud Continuum. In: Papadopoulos, G.A., et al (eds) Service-Oriented and Cloud Computing. ESOCC 2023. LNCS 14183. doi: 10.1007/978-3-031-46235-1_17
9. M. Neagul, I. Nedelcu, *A. Munteanu*, Building a National Spatio-Temporal Datacube, IGARSS 2023: IEEE International Geoscience and Remote Sensing Symposium, IEEE, doi: 10.1109/ IGARSS52108.2023.10281405
10. A. Tuns, *A. Spătaru*, Cloud Service Failure Prediction on Google's Borg Cluster Traces Using Traditional Machine Learning, 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, in print
11. *G. Iuhasz*, F. Fortis, *S. Panica*, A. Duma, Distributed platform for Cycle detection and analysis in Production Cyber-Physical Systems, AINA 2024, in evaluation

Alte prezentări la conferinte

12. *D. Petcu*, Extreme Data Processing in Exascale Systems, Invited Keynote, 16th Intern. Symposium on International Symposium on High-Level Parallel Programming and Applications (HLPP 2023)
13. *G. Iuhasz*, Performance and Compliance Anomaly Detection, Tutorial, 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing SYNASC 2023
14. *T. Selea, G. Iuhasz*, M.Neagul, A study of deep learning model performance for remote sensing image segmentation, doi: 10.20944/preprints202211.0226.v1. Workshop GRASS @ FOSS4G 2022

Teze

15. *A. Munteanu*, Semantic Segmentation of Vegetation in Remote Sensing Imagery Using Deep Learning, Teză de dizertație, Septembrie 2021
16. L. Popa, Data as Code in the context of Cloud Continuum, Teză de dizertație, Universitatea de Vest din Timișoara, Septembrie 2022, coordonator: *D. Petcu*

Organizare de evenimente științifice

Au fost organizate trei evenimente științifice legate de tema proiectului CoCo:

17. FiCloud 2022: The 9th International Conference on Future Internet of Things and Cloud, 22-24 August 2022, Roma, chairs: *Dana Petcu* & Irfan Awan
18. ECIS 2022: 2022 European Conference of Information Systems, 18-24 Iunie 2022, Timisoara, chairs: Roman Beck & *Dana Petcu* & Marin Fotache. Postproceedings R. Beck, *D. Petcu*, M. Fotache, S. Matook, 30th European Conference on Information Systems - New Horizons in Digitally United Societies, ECIS 2022, AIS, ISBN: 978-1-958200-02-5
19. DBB 2023: The 4th Joint International Conference on Deep Learning, Big Data and Blockchain, August 2023, Marrakesh, chairs: *Dana Petcu* & Abdeslam En-Nouaary. Postproceedings: M. Younas, I. Awan, S. Benbernou, *D. Petcu*, The 4th Joint International Conference on Deep Learning, Big Data and Blockchain, Springer, Lecture Notes in Networks and Systems (LNNS), volume 768, eBook ISBN 978-3-031-42317-8, doi: 10.1007/978-3-031-42317-8