Raport Științific și Tehnic

Identificare:

Contract de finanțare nr. 34/2021
Contractor: Universitatea de Vest din Timișoara (UVT)
Cod: PN-III-P4-ID-PCE-2020-0407
Etapa 1: Ian - Dec 2021
Nume etapă: Cerințe, arhitectură, monitorizare și achiziție a datelor

Autori ai raportului și membrii ai proiectului:

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

Director proiect:

Prof. Dr. Dana Petcu

Obiectivul proiectului

Principalul obiectiv științific al CoCo este de a permite calculul continuu Cloud-to-Edge. În special, CoCo intenționează să cupleze mecanismele de învățare (inclusiv învățarea federată) și descoperirea inteligentă a resurselor pentru a realiza un mediu de găzduire adaptiv capabil să ruleze atât pe cloud, cât și 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 generațiile IaC (Infrastructure as a Code). Platforma implementată va fi capabilă de reconfigurare autonomă (tratarea evenimentelor și descoperirii). Va fi dezvoltat un mediu sandbox pentru a testa comportamentul IaC și pentru a simula diferite implementări și scenarii.

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

Rezultatele estimate ale proiectului

CoCo intenționează să cupleze învățarea automată federată și descoperirea inteligentă de resurse pentru a realiza un mediu de găzduire adaptabil, capabil să funcționeze atât în Cloud cât și în Edge. Se preconizează ca o platformă open-source ce demonstrează conceptul să fie rezultatul tangibil al proiectului CoCo.

Pagină web:

http://coco.hpc.uvt.ro

Rezumatul Etapei 1: Cerințe, arhitectură, monitorizare și achiziție a datelor

În contextul CoCo, una dintre problemele majore care trebuie abordate este detectarea evenimentelor care pot afecta generarea și performanța IaC. Impactul acestor evenimente este amplificat atunci când avem de-a face cu implementarea pe dispozitive de margine, 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 primul an de implementare 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.

Gradul de îndeplinire a obiectivelor etapei

Obiectivul științific și tehnic al etapei a fost îndeplinit în totalitate, cum vom prezenta în cele ce urmează.

Descriere științifică și tehnică

Reamintim pachetele de lucru și sarcinile stabilite pentru etapa 1 a proiectului în descrierea inițială:

WP1 Requirements, architecture and continuous integration
T1.1 Requirements analysis
T1.2 Architecture definition
T1.3 Integration and validation strategy
WP2 Federated analysis and discovery
T2.1 Monitoring and data acquisition
T2.2 Federated data analysis and Machine Learning
T2.3 Intelligent service/edge node discovery
WP3 Runtime and sandboxing
T3.1 Distributed/Collective Runtime

WP4 Management, dissemination, exploitation and communication
T4.1 Day-to-day management, quality assurance, periodic reporting
T4.2 Dissemination, communication and networking
T4.3 Exploitation strategy, sustainability & innovation management

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 necesare pentru Task 2.1.

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].

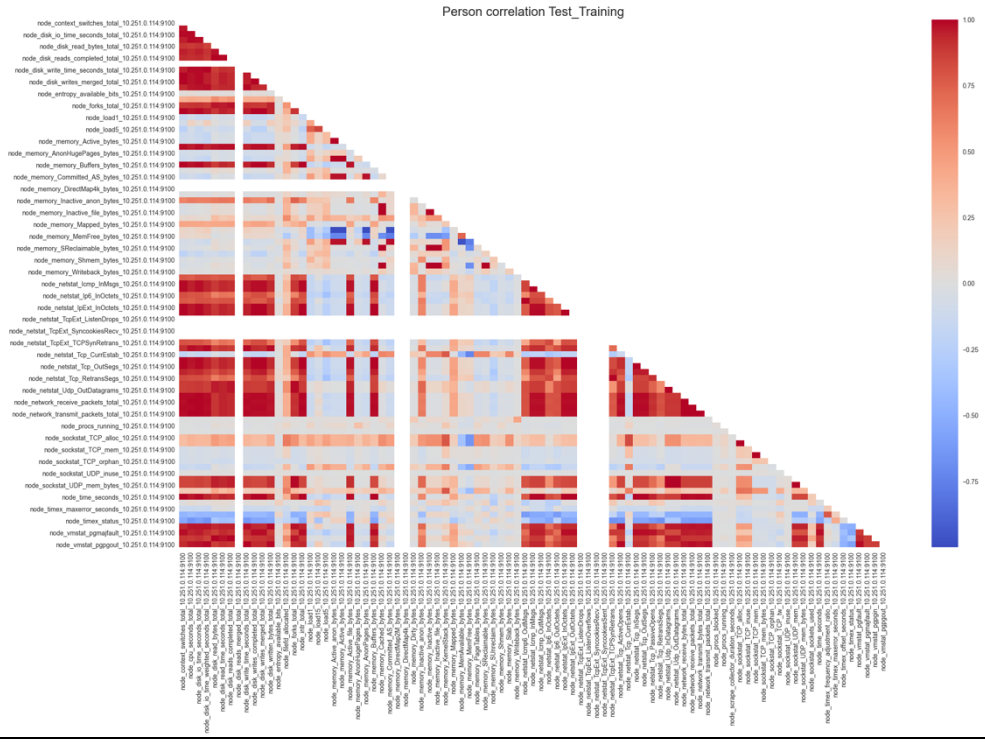
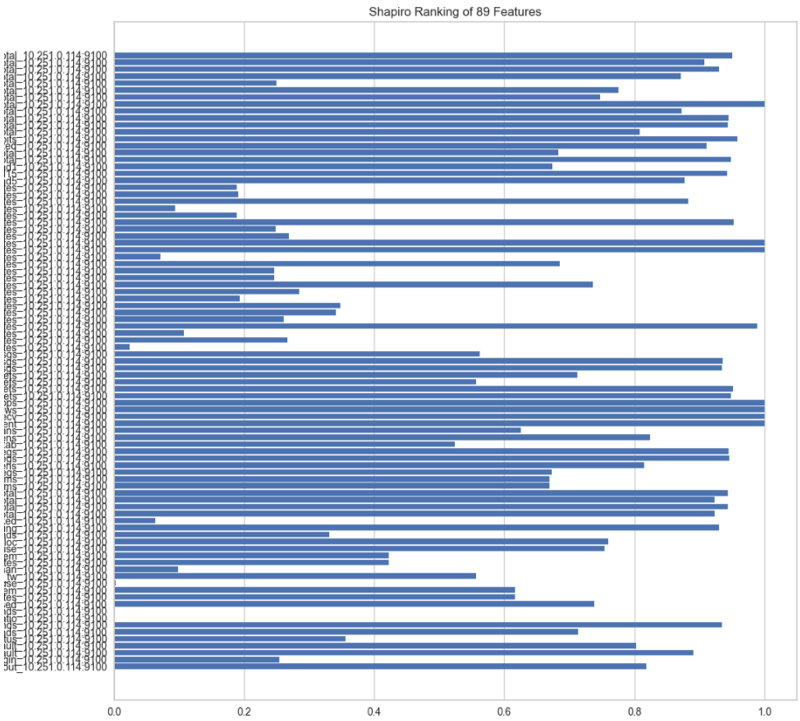
Învățare automată pentru detectarea evenimentelor

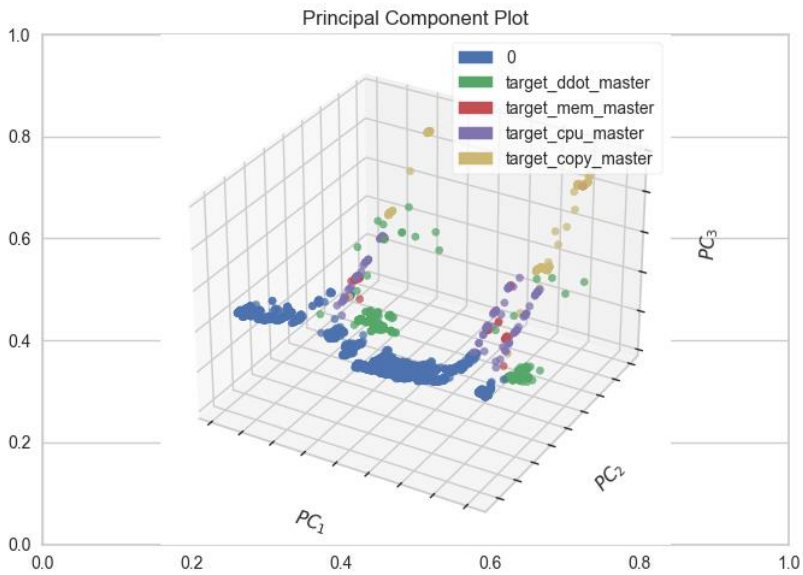
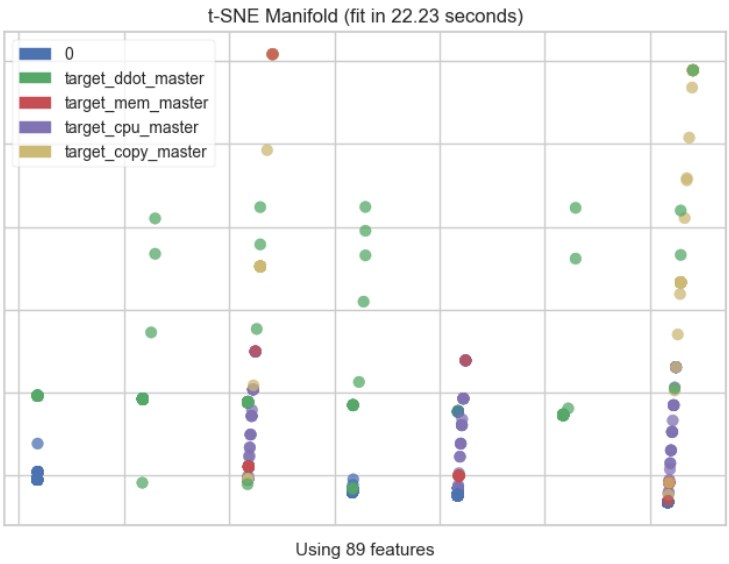
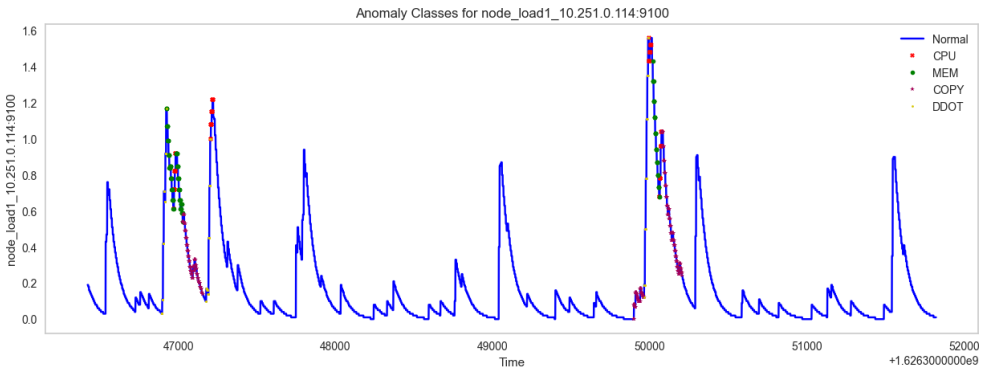
Componenta care se ocupă de detectarea evenimentelor (Event Detection Engine[5] -EDE) se bazează pe un instrument dezvoltat în cadrul proiectului H2020 ASPIDE. Adăugările făcute în timpul acestei etape pentru proiectul CoCo se concentrează în mare parte pe adaptarea transprecisă a metodelor de detectare a anomaliilor ML supravegheate. Mai exact, acoperă sarcinile T2.2 și T2.4 cu contribuții inițiale la T2.3.

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	S-a adăugat suport pentru metoda de reducere a caracteristicilor PCA pentru fiecare clasă de anomalii:

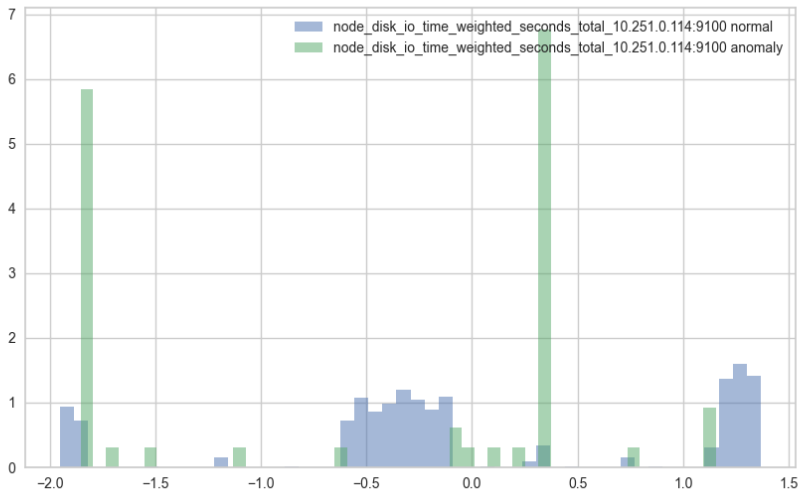
analysis (PCA)	
t-SNE Manifold embedding	S-a adăugat suport pentru reducerea caracteristicilor suplimentare (reproiectare la o dimensiune inferioară): 
Anomaly label plotting on features	S-au adăugat metode de trasare a claselor de anomalii pe caracteristicile selectate de utilizator: 

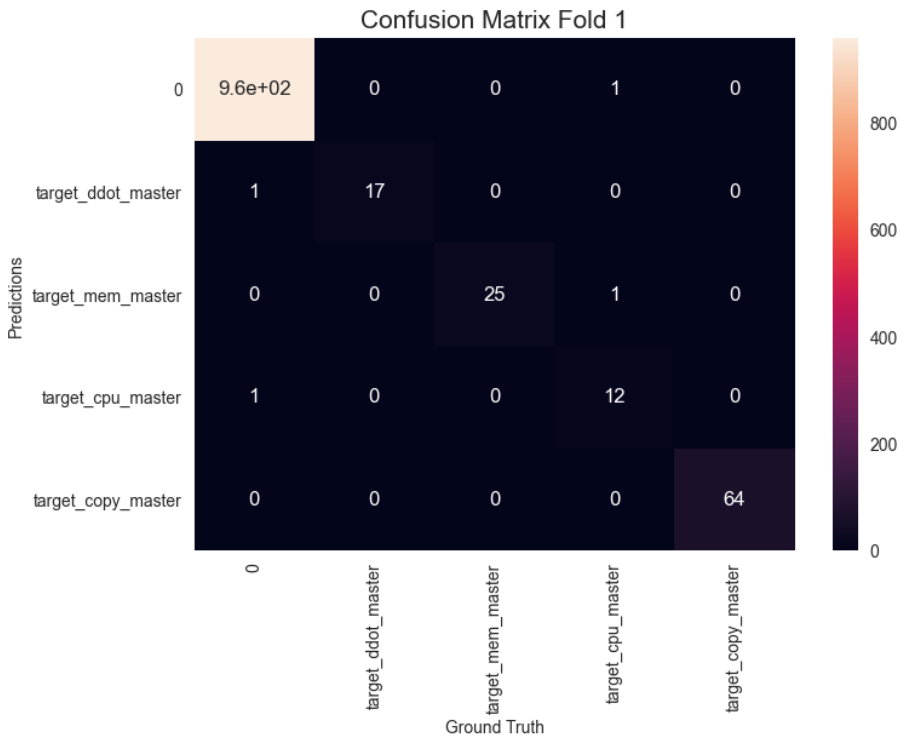
Optimizarea și raportarea învățării automate

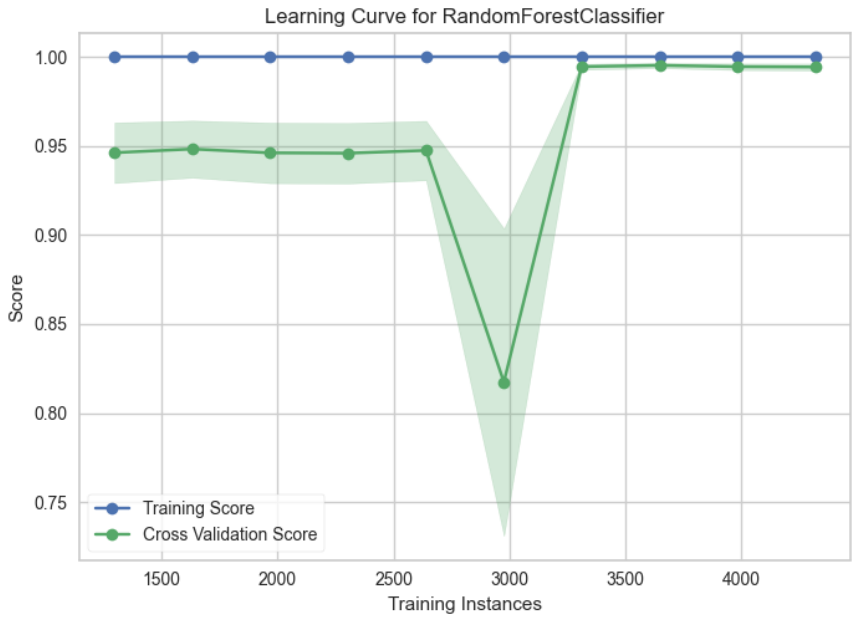
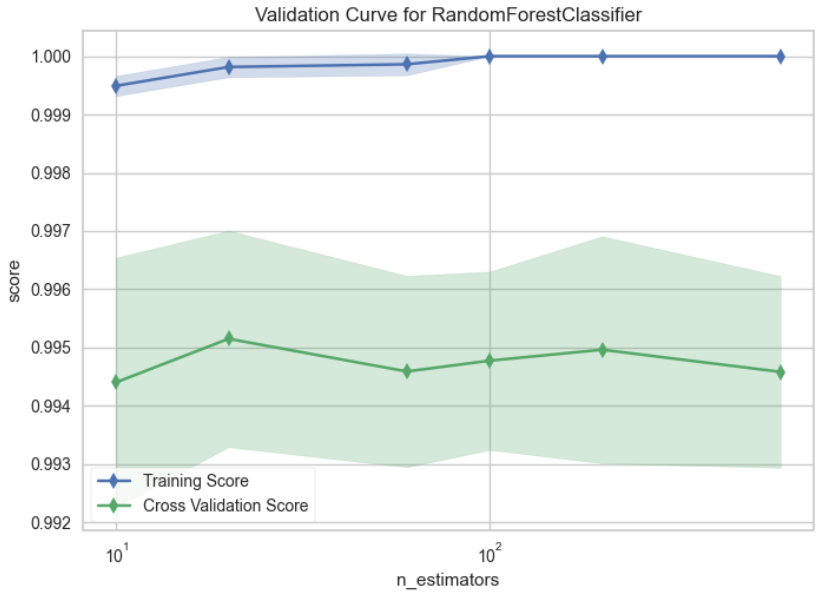
Principalul nostru obiectiv pentru metodele de optimizare a fost pe metodele supravegheate, deoarece acestea oferă o modalitate mai ușoară de a măsura performanțele lor predictive și de calcul. EDE oferă deja mecanism pentru optimizarea hiper-parametrică și metrici de evaluare personalizate, definite 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 în timpul sarcinilor T2.2 și lucrărilor inițiale pe T2.4.

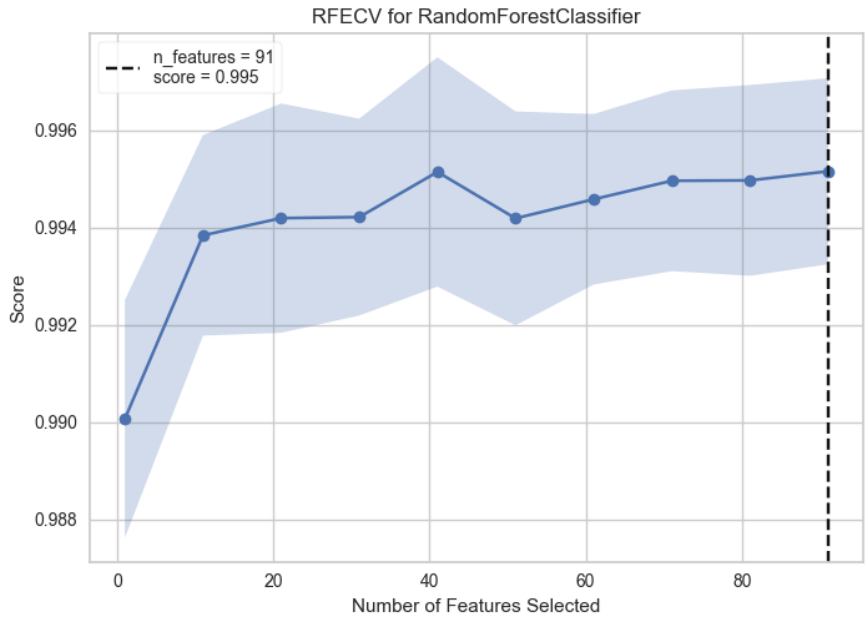
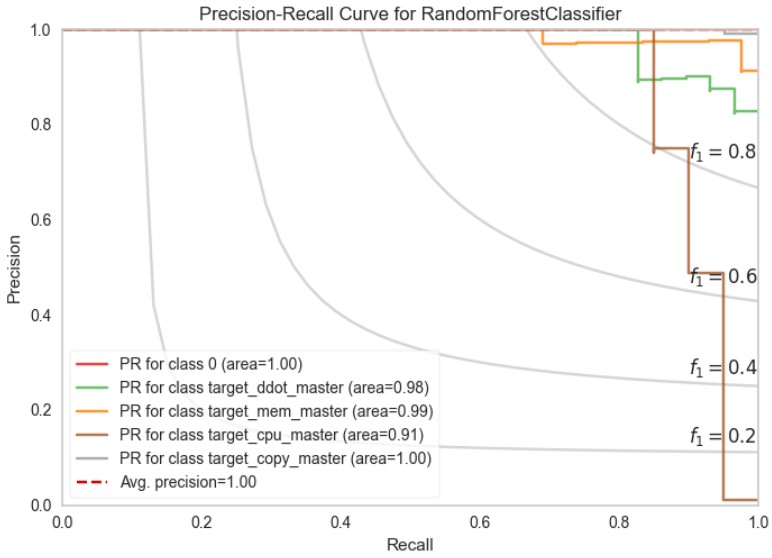
Aceste caracteristici suplimentare sunt utilizate pentru o mai bună optimizare a modelelor rezultate pentru diferite cazuri de utilizare, inclusiv adaptarea precisă și detectarea evenimentelor IaC. Sarcina T3.2 se va ocupa de adaptarea autonomă și va necesita informații suplimentare din partea oricărui 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.

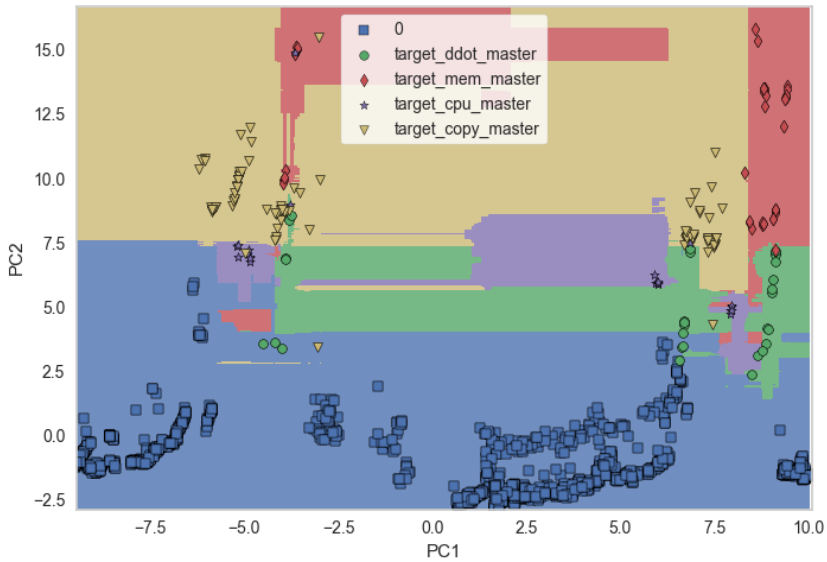
Dintre caracteristicile implementate amintim:

Nume	Descriere
Feature separation plot	Trasează separarea claselor pe valorile metrice definite de utilizator: 
Confusion matrix	Trasează matricea de confuzie care detaliază nu numai predicția corectă, ci și false pozitive și false negative:

<table>
<tr>
<td></td>
<td>


	
Validation Curve	Reprezintă scoruri de instruire și de validare încrucișată definite de utilizator pe baza intervalelor de valori definite de utilizator pentru un hiper-parametru specific: 
Recursive feature eliminatio n	Reprezintă scorul de validare obținut prin adăugarea iterativă de caracteristici în timpul antrenamentului:

	
Precision Recall Curve	Oferă o perspectivă mai profundă asupra performanței unui model predictiv dat: 
Decision Boundary	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)

Este important de menționat că în prezent lucrăm la un articol extins de jurnal care se ocupă de detectarea anomaliilor în sistemele distribuite. Caracteristicile prezentate aici servesc drept „coloana vertebrală” pentru experimentele realizate pentru această lucrare viitoare.

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

[6] https://shap.readthedocs.io/en/latest/

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 WP2 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 (T2.1). 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. 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), pot fi combinate cu ușurință pentru a îndeplini scopul monitorizării (de exemplu, detectarea anomaliilor).

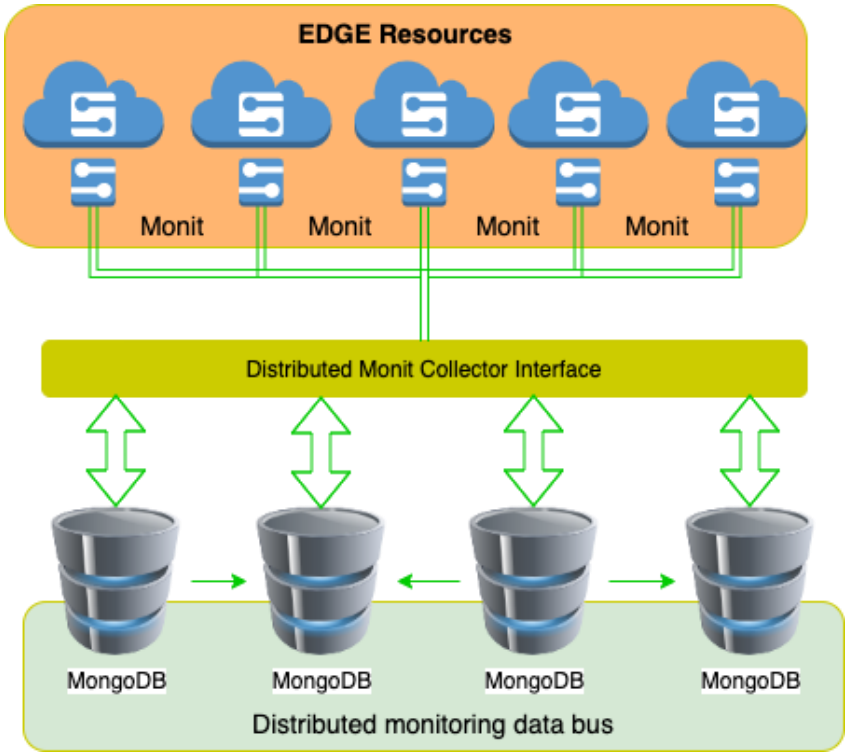


Figura 1: Arhitectura Monit distribuit

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

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 (T1.3, T3.1).

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 (T1.1). În sarcina T1.2, în colaborare cu T3.1, 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 (T1.3) această abordare, a fost aleasă o aplicație distribuită ca caz de utilizare pentru a fi implementată pe un grup de resurse hibride bootstrap cu Platforma COCO. Aplicația este eWPS, un API modern OGC WPS 2.0/WPS – Processes[15] implementarea standard care a executat servicii geospațiale prin resurse distribuite.

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

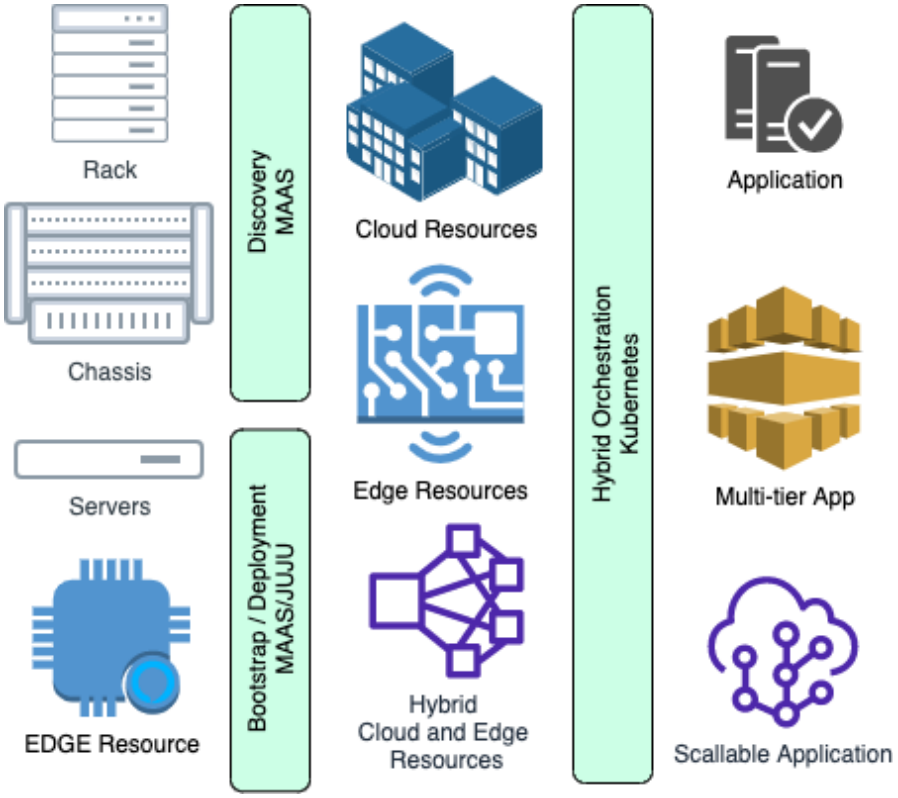


Figura 2: Imagine generală a Runtime-ului distribuit din COCO

O lucrare care descrie aplicația eWPS a fost depusă spre publicare în revista SCPE (www.scpe.org): „OGC API - Proceseses implementation for dynamically allocate resources in cloud continuum" – Silviu Panica, Marian Neagul, Teodora Selea.

Pentru următoarea perioadă de execuție a proiectului ne vom concentra pe descoperirea resurselor nodurilor inteligente și pe adaptarea platformei autonome. Aceste funcții vor îmbogăți timpul de execuție distribuit în principal la nivel hardware/software și îl vor ajuta pe orchestratorul hibrid, Kubernetes, să gestioneze corect evenimentele neprevăzute, cum ar fi disponibilitatea resurselor, și să gestioneze mai bine aplicațiile distribuite găzduite.

Aplicație: Învățare automată aplicată pentru detecția vegetației în imagini satelitare

Categorizarea acoperirii și utilizării terenului (land cover/land use) prezintă un actual subiect de interes. 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).

Diseminarea rezultatelor

1. Articole

Trei articole au fost înainte ca rezultat direct a proiectului către revista open-access Scalable Computing: Practice and Experience (SCPE, ISSN 1895-1767, www.scpe.org) care este indexată de Clarivate Analytics în baza de date Emerging Sources Citation Index (ESCI) și care în iunie 2021 a fost listată prima dată ca fiind în Q4 în Journal Citation Reports de către Clarivate Analytics (motiv pentru care a fost aleasă pentru publicații):

- *Dana Petcu*, Service Deployment Challenges in Cloud-to-Edge Continuum, Scalable Computing: Practice and Experience, vol. 22, no. 3 (2021), 313–320, doi: 10.12694/scpe.v22i3.1941, WOS: 000721262900004
- *Silviu Panica*, Marian Neagul, *Teodora Selea*, OGC API - Proceseses implementation for dynamically allocated resources in cloud continuum, Scalable Computing: Practice and Experience, vol. 22, no. 4 (2021), doi: 10.12694/scpe.v22i3.1950
- *Adrian Spătaru*, A Review of Blockchain-enabled Fog Computing in the Cloud Continuum Context, Scalable Computing: Practice and Experience, vol. 22, no. 4 (2021), doi: 10.12694/scpe.v22i3.1947

Un alt articol a fost înaintat la revista open-access Frontiers in Big Data, care este indexată de Clarivate Analytics în repozitoriu Emerging Sources Citation Index (ESCI), with ISSN 2624-909X:

- Kimovski Dragi, Mathá Roland, *Iuhasz Gabriel*, Marozzo Fabrizio, *Petcu Dana*, Prodan Radu, Autotuning of Exascale Applications With Anomalies Detection, Frontiers in Big Data, vol. 4, Noiembrie 2021, 101, doi: 10.3389/fdata.2021.657218, article 657218

care este un follow-up a proiectului H2020 ASPIDE finalizat în mai 2021, dar care include o parte dintre noile contribuții aduse la EDE de către CoCo în lunile după finalizarea ASPIDE.

[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

2. Conferințe

A 23 ediție a simpozionului SYNASC (http://synasc.ro/2021) a fost organizată în perioada 7-10 decembrie 2021. Doi dintre membrii proiectului CoCo (D. Petcu și S. Panica) sunt persoane cheie ale evenimentului (https://synasc.ro/2021/committees/). Publicația pre-proceeding în IEEE CSP este așteptată să fie indexată ISI proceedings în 2022. Programul evenimentului organizat este disponibil la: https://easychair.org/smart-program/SYNASC2021/. În mod special, track-ul de Distributed Computing a fost aliniat cu tematicile CoCo. Trei dintre lucrările aceptate spre publicare sunt elaborate de doctoranzii directorului local al CoCo și sunt lucrări colaterale:

- Sebastian-Aurelian Stefanigă, Methods of acceleration for feature extractions in medical imaging using GPU processing
- Sorin Valcan, Convolutional Neural Network Training System For Eye Location On Infrared Driver Recordings Using Automatically Generated Ground Truth Data
- Alexandru Ionascu, Extracting football players video sprites from broadcast video

3. Teze

Teza de dizertatie

- *Alexandru Munteanu*, Semantic Segmantation of Vegetation in Remote Sensing Imagery Using Deep Learning, Septembrie 2021

a fost elaborată folosind o parte din prototipurile software construite în acest proiect (menționată prin „aplicație” în secțiunea anterioară).

Buget

Costurile de personal sunt conforme cu efortul depus și estimările inițiale. Participările la conferințe internaționale (planificate în propunerea de proiect) au fost afectate și înlocuite cu publicații în reviste și conferința internațională organizată SYNASC. Costul auditului a fost redus conform rezultatului licitației.

Concluzii

Obiectivele etapei au fost îndeplinite în totalitate.
Implementarea conceptelor este în faza incipientă.
Abaterile de la buget nu sunt reflectate în calitatea rezultatelor.
Rezultatele etapei au fost parțial publicate în reviste de specialitate.