Raport științific și tehnic

privind implementarea proiectului 287PED/2020 “Graphomaly” în 2022

0. Introducere

Proiectul Graphomaly are ca scop producerea unei biblioteci de programe pentru *detecția anomaliilor în grafuri* ce modelează tranzacții financiare; prin anomalii înțelegem în primul rând posibile tranzacții frauduloase cu caracter complex, cum ar fi spălarea de bani, care se manifestă prin structuri tipice în graful de tranzacții.

Anul 2022 este ultimul al proiectului și în cele patru luni rămase până la termenul final ne-am dedicat eforturile pentru realizarea într-o formă cât mai bună a obiectivelor majore, publicații și biblioteca Graphomaly. Prezentăm în acest raport:

1. Rezumatul etapei
2. Implementarea bibliotecii Graphomaly
3. Teste și rezultate finale
4. Publicații realizate și diseminare
5. Modul de îndeplinire a planului de realizare.

Raportul științific final, un document separat, conține sinteza rezultatelor obținute pe durata întregului proiect.

1. Rezumatul etapei

În perioada decembrie 2021—aprilie 2022, scopurile principale au fost:

- Terminarea implementării bibliotecii Graphomaly, cu modulele dezvoltate de-a lungul execuției proiectului, și a documentației aferente. Descrierea implementării se găsește în Sec.2. Biblioteca este acum publică.
- Testarea bibliotecii, pe date reale și artificiale, astfel încât să fie dovedită funcționarea corectă, cu indici buni de performanță în detectarea anomaliilor. Sec.3 detaliază modul de testare și prezintă unele rezultate.
- Publicarea de lucrări științifice rezultate din lucrul la proiect. Sec.4 conține rezumatele acestor lucrări, cea mai importantă fiind un articol în IEEE Access, la care am lucrat intens în lunile decembrie și ianuarie.

Toate aceste scopuri au fost atinse, așa cum este detaliat în Sec.5, unde am descris modul și gradul de îndeplinire a planului de realizare (ale cărui trei componente principale sunt cele din lista de mai sus).

2. Implementarea bibliotecii Graphomaly

Ultimele luni au fost folosite în special pentru a organiza programele existente într-o formă structurată pe modelul scikit-learn [SciLe11], foarte popular în comunitatea machine learning. Avantajul acestei abordări este integrarea facilă cu alte pachete dedicate detecției de anomalii, precum și un start rapid pentru cei din domeniu.

În acest moment

- biblioteca graphomaly este publică, la https://gitlab.com/unibuc/graphomaly/graphomaly;
- ea a fost publicată în Python Package Index (pypi.org) și deci poate fi folosită cu un simplu *import graphomaly*;
- documentația poate fi citită la https://unibuc.gitlab.io/graphomaly/graphomaly/index.html, ea fiind realizată cu ajutorul Sphinx, din fișiere dedicate și antetele funcțiilor implementate;
- biblioteca conține exemple simple de utilizare, pe date artificiale de dimensiuni rezonabile, ilustrând atât funcționalitatea cât și căi de eficientizare a calculelor, utile pentru date de dimensiuni mari;
- biblioteca are un caracter deschis, structura curentă permițând adăugarea facilă de metode noi de extragere de trăsături.

Urmând modelul scikit-learn, categoriile principale de operații pe care le-am implementat sau integrat sunt:

- Preprocesare, adică transformarea listei de tranzacții sau a grafului asociat
- Detecție de anomalii
- Metode de votare și grid search (pentru optimizare)

Discutăm mai jos despre fiecare. Clasa umbrelă pentru procesarea datelor din forma inițială de listă de tranzacții până la rezultatele detecției de anomalii, incluzând votare și grid search, este GraphomalyEstimator.

2.1. Preprocesare

În jargonul scikit-learn, preprocesarea se referă la transformarea datelor inițiale pentru învățarea automată ulterioară (detecție de anomalii în cazul nostru, clasificare, regresie). Având în vedere că multe din contribuțiile noastre se referă la calculul de trăsături, ele se regăsesc în această categorie.

Prezentăm mai jos clasele care servesc implementării modulare a preprocesării, fiecare având metodele fit(), care în general stabilește sau calculează parametrii, și transform(), care implementează transformarea efectivă a datelor.

Transformări tranzacții → graf

Clasa Transactions2Graph permite transformarea unei liste de tranzacții care conține câmpurile tipice ID sursă, ID destinație, sumă transferată de la sursă la destinație (alte câmpuri sunt

procesate doar la indicația expresă a utilizatorului), într-un graf orientat ale cărui noduri sunt ID-uri iar arcele au două atribute: suma totală transferată și numărul de tranzacții.

Transformări graf → trăsături

În această categorie se află mai multe tipuri de transformări prin care se calculează trăsături asociate cu nodurile, clasele reprezentative fiind:

- EgonetFeatures, care permite calculul trăsăturilor de tip egonet și egonet redus; sunt 16 trăsături de bază, care se calculează împreună și pot fi salvate și reîncărcate ulterior pentru calcul rapid, și alte 16 trăsături derivate, al căror calcul este rapid dacă trăsăturile de bază sunt disponibile.
- RwalkFeatures, prin care se calculează trăsături de tip random walk (8 trăsături de bază și 2 derivate).
- SpectrumFeatures, prin care se pot calcula până la 4 trăsături constând în valorile singulare cele mai semnificative ale Laplacienilor asociați egoneților.

Transformări tranzacții → trăsături

În această categorie se găsesc transformări prin care se calculează trăsături asociate cu arcele grafului, adică cu tranzacțiile. Clasele reprezentative sunt:

- TimeFeatures, prin care se calculează 8 trăsături legate de câmpul “timp” al listei de tranzacții, ca de exemplu ziua din an, ziua din lună, ora, minutul, etc. Aceste trăsături pot folosi în a distinge elemente de regularitate în tranzacții, care de obicei semnalează tranzacții normale, sau, dimpotrivă, structuri de timp neobișnuite.
- GroupHistoricalTimeFeatures, prin care se calculează diferențele temporale între tranzacții succesive ale aceluiași client. Sunt 4 trăsături, care pot fi asociate oricăruia dintre ID-urile sursă sau destinație. Sunt posibile și alte grupări, nu doar după ID, ci în principiu după orice câmp care produce grupuri.
- GroupStatisticalFeatures, prin care se calculează 8 trăsături statistice (min, max, medie, percentile, etc.), peste valorile aferente unui câmp (de obicei suma tranzacției, dar orice câmp numeric poate fi ales), relativ la valori din trecut (la intervale de timp precizate) selectate după un anume criteriu (tipic, ID-ul).
- DifferenceFeatures, prin care se calculează diferențe între câmpuri cu valori numerice. Utilizarea tipică este pentru diferența dintre valoarea unui câmp (de exemplu suma tranzacției) și o statistică (de exemplu media) a aceluiași câmp pe o perioadă precedentă (o săptămână, o lună, etc.)

2.2. Detecție de anomalii

Majoritatea metodelor accesibile din Graphomaly sunt împrumutate din PyOD [ZNL19], o bibliotecă recentă dedicată detecției de anomalii. Am implementat o metodă bazată pe învățarea dicționarelor [DI18], care folosește programe din biblioteca dictlearn, implementată în Python în cadrul acestui proiect și disponibilă la https://gitlab.com/unibuc/graphomaly/dictionary-learning.

2.3. Metode de votare și grid search

Pentru alegerea celei mai probabile etichete pentru fiecare tranzacție, biblioteca include o serie de metode de voting, ce integrează metodele de voting existente în Sklearn cu cele din PyOD. Clasa relevantă pentru aceste operații este VotingClassifier.

A fost dezvoltată o metodă de căutare a parametrilor optimi pentru fiecare algoritm implementat. Configurațiile parametrilor fiecărui algoritm se furnizează în format .yaml sau direct prin argumente ale constructorului clasei GridSearch. Comună tututor metodelor testate este rata de contaminare, reprezentând raportul de anomalii din baza de date, restul câmpurilor fiind specifice fiecărui algoritm. Metoda returnează numărul și rata de fals negative, de fals pozitive, numărul și rata de anomalii identificate corect, respectiv greșit, precum și acuratețea pentru fiecare configurație. De asemenea, pentru fiecare metodă testată, este identificată configurația optimă în ce privește acuratețea.

3. Testare

3.1. Testare algoritmi de detecție de anomalii în grafuri

Articolul [DBS22] cuprinde rezultatele semnificative în privința testării algoritmilor noștri, folosind implementări aflate acum în biblioteca Graphomaly, pe date reale și sintetice. Prezentăm aici doar cele mai semnificative rezultate.

O primă direcție de lucru a fost găsirea unor seturi de trăsături asociate nodurilor care să aibă atât coerență, cât și rezultate bune. În final, ne-am oprit la trei astfel de seturi.

Egonet. Trăsăturile sunt: grad (intrare și ieșire = I/O), sumă totală I/O, sumă medie I/O, densitate arce egonet asociat. Precizăm că, de exemplu, suma totală intrare este suma valorilor transferurilor către nodul respectiv, iar suma medie intrare este suma totală împărțită la numărul tranzacțiilor având ca destinație nodul respectiv. Amintim că egonetul asociat unui nod *k* (numit rădăcină sau centru) al unui graf *G* este subgraful *E(k)* al lui *G* generat de nodul *k* și toți vecinii lui direcți.

Egonet redus (egored). Reamintim că un egored este format dintr-un egonet prin eliminarea nodurilor care au un singur arc, cel cu rădăcina. Trăsăturile sunt cele de la egonet, la care se adaugă cele similare de la egored, cu precizarea că suma totală în egored este relativă, adică se împarte la suma totală în egonet (ceea ce reprezintă practic o normalizare).

Egonet + RW. Trăsăturile sunt cele de la egonet, la care se adaugă o singură trăsătură de tip random walk (RW), anume suma maximă care parcurge un RW care formează un ciclu (deci suma se întoarce în nodul de origine după ce trece prin mai multe noduri).

După obținerea valorilor trăsăturilor, detecția de anomalii se face cu Isolation Forest [LTZ08], un algoritm general de detecție de anomalii.

Rezultatele pe datele de la Libra Internet Bank au fost incluse în raportul pe 2021. Prezentăm aici câteva din rezultatele noi pentru date sintetice, obținute în 2022.

Ca și în raportul precedent, datele sintetice sunt generate utilizând un model de graf multidirecțional bloc stochastic cu 50000 de noduri. Am generat grafuri având gradul mediu 7, 12, 17 și 22. Sumele tranzacționate respectă o distribuție log-normală și sunt limitate în intervalul [1, 1000000]. Tranzacțiile din întreg graful se înscriu într-o perioadă de 10 luni.

În fiecare graf sunt introduse în locații aleatoare un număr de 25 de subgrafuri reprezentând structuri anormale, toate de dimensiune 10 noduri:

- 15 structuri de tip clică (de trei feluri, fiecare reprezentând aproximații sau forme speciale ale clicii ideale)
- 5 structuri de tip stea
- 5 structuri de tip inel

Pentru fiecare tip de graf au fost generate trei instanțe diferite, iar rezultatele sunt obținute prin mediere. Tabelul de mai jos prezintă valorile TPR AUC 1% (True Positive Rate Area under Curve) în 1% din totalul nodurilor analizate. Acesta este un indice care arată eficiența metodelor în zona de interes maxim, cea a primelor anomalii detectate. (Menționăm că sunt 250 de noduri cu anomalii, adică 0.5% din totalul nodurilor.) În afara metodelor noastre, prezentăm rezultate și pentru OddBall [AMF10], o metodă foarte cunoscută și eficientă pentru detectarea unor structuri speciale, îndeosebi clici. Rezultatele cu metoda din [Ell19] nu sunt prezentate, ele fiind slabe.

Metoda	G7	G12	G17	G22
egonet	0.6546	0.6433	0.6062	0.5739
egonet+egored	0.8205	**0.8084**	**0.7918**	**0.7790**
egonet+RW	0.8011	0.7880	0.7662	0.7640
OddBall	**0.8214**	0.7769	0.7616	0.7733

Din aceste rezultate putem remarca următoarele:

- Contribuțiile noastre, care constau în adăugarea trăsăturilor de tip egored și RW, îmbunătățesc semnificativ performanța celor clasice de tip egonet.
- Metoda egonet+egored este superioară OddBall atunci când grafurile au grade mai mari și deci detecția anomaliilor (care au dimensiune constantă) este mai dificilă.

Pentru detaliere, prezentăm în Figura 1 comportarea metodelor pe anomalii de tip clică și inel, pe un graf cu grad mediu 22. OddBall se comportă aproape perfect în detectarea clicilor, fiind binecunoscută pentru această calitate. Egonet+egored și egonet+RW sunt totuși destul de aproape de OddBall și evident mai bune decât egonet singur. La inel, situația se schimbă, metoda egonet+RW fiind de regulă cea mai bună, deoarece urmărește în mod direct ciclurile cu sume mari. Egonet+egored nu este departe și amândouă sunt mai bune decât OddBall și mult mai bune decât egonet singur.

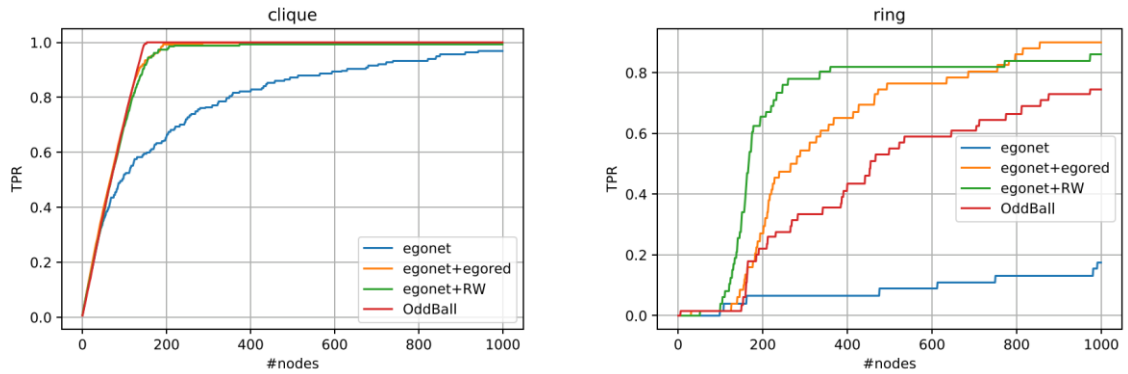

Figura 1. TPR pentru clică și inel.

Coroborând cele de mai sus cu rezultatele pe datele Libra [DBS22], unde OddBall este mult inferioară, iar metoda [Ell19] este apropiată ca rezultate de metodele noastre, putem trage concluzia că algoritmul bazat pe trăsături egonet+egored este eficient și robust, având rezultatele cele mai bune pe toate datele, atât reale cât și sintetice.

3.2. Testare algoritmi de detecție de anomalii în lista de tranzacții

Pentru această etapă, BRD a pus la dispoziție o nouă bază de date similară cu cea trimisă anterior, în care entitățile relevante au fost denumite cu scop de anonimizare KID, BOX1 și BOX2. Variabilele bazei de date diferă însă de cea utilizată în etapa precedentă, astfel că metodele utilizate în testele anterioare (descrise în raportul precedent) au fost modificate pentru a putea fi utilizate cu noul tip de date. Detaliem în continuare principalele trăsături ale noii baze de date.

Aceasta conține 65535 de intrări, fiecare având 50 de variabile. Dintre acestea, 3 sunt ID-uri, respectiv KID, BOX1 și BOX2 amintite anterior. Din restul variabilelor, 18 se referă la proprietăți ce caracterizează câmpul KID, 9 se referă la câmpul BOX1, iar 20 la câmpul BOX2. BRD nu a furnizat informații despre natura sau semnificația variabilelor, prin urmare putem rezuma doar tipul datelor. Dintre variabile (exceptând IDurile), 4 au valori reale, restul fiind întregi, majoritatea putând fi atribuite unor variabile categoriale de cardinalitate mică. O singură variabilă dintre cele cu valori întregi conține numere negative. În ce privește etichetele, acestea nu au fost puse la dispoziție de către BRD, care a verificat în particular rezultatele furnizate de echipa Graphomaly. BRD a furnizat, însă, informații generice cu privire la numărul etichetelor și natura acestora.

Concret, există 3 tipuri de etichete, respectiv pentru fiecare dintre câmpurile KID, BOX1 și BOX2. În total există 410 anomalii pentru câmpul KID, 13 anomalii pentru câmpul BOX1 și 54 pentru câmpul BOX2. De asemenea, numărul de tranzacții pentru care etichetele tuturor celor 3 câmpuri sunt pozitive este de 7 tranzacții. Numărul de tranzacții pentru care cel puțin o etichetă este pozitivă este de 433.

Au fost testați următorii algoritmi: Angle-base Outlier Detection (ABOD), Copula Based Outlier Detector (COPOD), Histogram- based outlier detection (HBOS), Isolation Forest (IForest), Lightweight on-line detector of anomalies (LODA), Minimum Covariance Determinant (MCD), One-Class SVM (OCSVM). Testele au fost realizate pentru rate de contaminare de 0.07%, 0.08%, 0.1%, 0.15% și 0.2%, cu excepția OCSVM, pentru care s-au testat ratele de 0.07%, 0.1% și 0.2% . Rata de 0.07% reprezintă proporția reală de anomalii. Am considerat o anomalie acea tranzacție pentru care cel puțin una din etichete este pozitivă. În evaluarea performanțelor, BRD a verificat rata de fals pozitive (FPR), anomalii identificate corect (TPR) și rata de tranzacții normale identificate corect (TNR). În testare s-a utilizat funcția de Grid Search implementată în bibliotecă pentru a verifica diferite configurații de parametrii pentru fiecare metodă în parte. În total, au fost realizate 1906 teste. Tabelul de mai jos rezumă rezultatele obținute.

Metoda	**Rata de Contaminare**	**Acuratețe**	**TPR**	**TNR**
IForest	0.2	0.5470	0.2933	0.8006
KNN	0.2	0.5040	0.2079	0.8001
COPOD	0.2	0.5226	0.2448	0.8003
LODA	0.2	**0.6530**	**0.5035**	0.8026
ABOD	0.07	0.5020	0.0739	0.9300
HBOS	0.2	0.5586	0.3164	0.8008
MCD	0.07	0.5310	0.1316	**0.9304**
OCSVM	0.2	0.5721	0.3418	0.8023

Cu excepția a doi algoritmi, ABOD și MCD, toate metodele prezintă performanțe mai bune atunci când rata de contaminare este semnificativ mai mare decât cea reală, însă în detrimentul identificării corecte de anomalii (cele două metode au și performanțele cele mai slabe în ce privește TPR. Cele mai bune performanțe s-au obținut cu algoritmul LODA, urmat de OCSVM.

4. Publicații realizate și diseminare

În etapa curentă au fost finalizate și trimise spre publicare următoarele articole.

[DBS22] B.Dumitrescu, A.Băltoiu, Ş.Budulan, "Anomaly Detection in Graphs of Bank Transactions for Anti Money Laundering Applications", IEEE Access, 2022.

[IADu22a] D.C.Ilie-Ablachim, B.Dumitrescu, "Anomaly Detection with Selective Dictionary Learning", 8th Int.Conf. Control, Decision and Information Technology (CoDIT), Istanbul, Turkey, May 2022.

[IADu22b] D.C.Ilie-Ablachim, B.Dumitrescu, "Reduced Kernel Dictionary Learning", 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, Aug. 2022.

Primele două au fost și acceptate. De menționat că revista IEEE Access este în categoria Q2 și are un factor de impact de 3.35; articolul a fost acceptat cu câteva zile înainte de terminarea proiectului, deci încă nu avem date referitoare la volum, pagină, etc. În plus, a mai fost acceptat unul din articolele trimise anul trecut [IRP21].

Prezentăm acum în câteva cuvinte conținutul articolelor realizate în acest an.

Articolul [DBS22] conține cele mai importante rezultate obținute în tematica propriu-zisă a proiectului, anume detecția de anomalii în grafuri. Este articolul cel mai reprezentativ pentru proiect. Algoritmii și bună parte din rezultatele pe datele Libra și sintetice au fost incluse în raportul din 2021. Nu mai reluăm aici modul de construcție a trăsăturilor de tip egonet redus (egored) și random walk. Progresul de anul acesta a fost obținut prin combinații noi de trăsături, care a condus la rezultate superioare. De asemenea, am extins testarea; rezultatele prezentate în sec.3 a acestui raport sunt toate noi. Articolul a fost acceptat pe 21 aprilie și a fost încărcat în formă finală, dar deocamdată este vizibil doar în secțiunea “early access” pe ieeexplore.

Lucrarea [IADu22a] conține un algoritm de detecție de anomalii bazat pe învățarea dicționarelor. Scorul de anormalitate este chiar eroarea de reprezentare. Ideea centrală este de a favoriza, prin selecție aleatoare în faza de reprezentare rară și prin selecție a semnalelor cu erori mici în faza de actualizare a dicționarului, reprezentarea mai bună a semnalelor normale în timpul învățării. Avem în vedere faptul că, prin selecție aleatoare, ignorarea unui semnal normal nu are ca efect înrăutățirea semnificativă a reprezentării lui, deoarece există alte semnale în apropiere. Invers, ignorarea unei anomalii are efecte mai severe în privința reprezentării. Teste în mediul PyOD

[ZNL19] arată rezultate satisfăcătoare, metoda propusă fiind competitivă și întrecând metodele standard în anumite probleme de detecție de anomalii.

Lucrarea [IADu22b] dezvoltă o idee din [IADu22a], anume alegerea nucleului pentru Kernel Dictionary Learning, pe direcția naturală: i) o submulțime a semnalelor, ii) un dicționar pentru reprezentarea rară a semnalelor, iii) un dicționar optimizat. Etapa iii) este cea de interes aici, algoritmii de optimizare având o formă matematică nebanală. Se dovedește experimental că astfel se pot obține reprezentări mai bune.

Pe lângă publicații, diseminarea a fost asigurată prin deschiderea accesului public la biblioteca și documentația Graphomaly, așa cum este explicat în sec.2.

Site-ul proiectului, http://graphomaly.upb.ro/, este actualizat și conține publicațiile și programele realizate, datele utilizate (reale și sintetice), precum și alte informații despre proiect.

5. Îndeplinirea planului de realizare

Enumerăm mai jos activitățile din 2022 (conform planului de realizare) și principalele rezultate obținute în cadrul lor.

Act. 3.1. Redactare manual de utilizare bibliotecă, raport, publicații

Articolul [DBS22] conține versiunea cea mai bună a algoritmilor propuși până acum pentru extragerea trăsăturilor din graful de tranzacții. De asemenea, am realizat două articole de conferință, unul acceptat [IADu22a], altul trimis [IADu22b]. Manualul de utilizare a bibliotecii Graphomaly se găsește la https://unibuc.gitlab.io/graphomaly/graphomaly/. El este realizat cu Sphinx și descrie toate operațiile implementate.

Grad de realizare: complet. Mici retușuri și completări vor fi cu siguranță necesare în documentație, dar aceasta descrie complet clasele și funcțiile implementate.

Act. 3.2. Optimizare finală, retușuri

Combinațiile optime de trăsături extrase din graf au fost căutate așa cum e descris în [DBS22], vezi și sec.3 a acestui raport. Căutarea a necesitat un efort de calcul semnificativ, ea având loc pe datele Libra și pe mai multe seturi de date sintetice. O căutare exhaustivă este imposibilă, datorită numărului mare de trăsături, de aceea au fost alese în special combinații structurate, câștigătoare fiind chiar una dintre ele, care conține perechi de trăsături similare de tip egonet și egored. Avem acum o înțelegere bună a importanței fiecărei trăsături și a beneficiilor specifice pe care le poate aduce în detecția de anomalii.

Grad de realizare: foarte satisfăcător. Nu putem pretende că am ajuns la un algoritm optim, dar suntem competitivi la vârf, așa cum este dovedit în [DBS22].

Act. 3.3. Testare finală

Testarea finală a algoritmilor incluși în biblioteca Graphomaly a avut loc pe datele Libra și pe mai multe instanțe de date sintetice, cu diverse tipuri de anomalii. Considerăm că acest scenariu de testare acoperă foarte bine cerințele de funcționare. De asemenea, am comparat metoda noastră cu metode clasice [AMF10,Ell19], obținând rezultate comparabile cu metodele specializate pe anumite tipuri de date și rezultate mult mai bune pe tipuri de date care nu erau vizate special de acele metode. Putem deci afirma că metodele noastre au un comportament robust.

De asemenea, am testat bibliotecile Graphomaly și dictlearn în mai multe scenarii de funcționare, descoperind și corectând unele erori și inadvertențe. În plus, am efectuat teste de detecție de anomalii pe data furnizate de BRD, așa cum e descris în sec. 3.2.

Grad de realizare: foarte satisfăcător. Putem spune aproape cu certitudine că algoritmii sunt implementați corect și funcționează bine pe date suficient de generale.

6. Concluzii

Rezultatele obținute pe durata întregului proiect (vezi raportul științific final) și definitivate în această etapă ne permit să afirmăm că toate obiectivele majore al proiectului au fost îndeplinite. Bibliotecile graphomaly și dictlearn sunt funcționale și deschise tuturor utilizatorilor. Publicațiile au depășit planul inițial (planificat: 1 articol revista Q1/Q2, 3 articole de conferință; realizat: 1+5; în evaluare: 1+2) iar nivelul de calitate propus a fost atins.

Bibliografie

[AMF10] L. Akoglu, M. McGlohon, C. Faloutsos. “Oddball: Spotting anomalies in weighted graphs”. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 410–421, 2010.

[DI18] B.Dumitrescu, P.Irofti - Dictionary Learning Algorithms and Applications, Springer, 2018.

[Ell19] A. Elliott et al. Anomaly detection in networks with application to financial transaction networks. *arXiv:1901.00402*, 2019.

[LTZ08] F.T. Liu, K.M. Ting, Z.H. Zhou, Isolation Forest. In *8th IEEE International Conference on Data Mining*, pp. 413-422, Dec. 2008.

[IRP21] P.Irofti, C.Rusu, A.Pătrașcu, “Dictionary Learning with Uniform Sparse Representations for Anomaly Detection, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

[SciLe11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas. Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, 12, 2825-2830, 2011.

[ZNL19] Y. Zhao, Z. Nasrullah and Z. Li, (2019). *PyOD: A Python Toolbox for Scalable Outlier Detection*. Journal of Machine Learning Research (JMLR), 20(96), pp.1-7.

Director proiect,
Prof. Bogdan Dumitrescu