

Raport științific

privind implementarea proiectului PCE 6/2022 “AsyDiL” în 2023

0. Introducere

Proiectul “Învățarea asimetrică a dicționarelor” (Asymmetric Dictionary Learning—AsyDiL) are ca scop deducerea unor algoritmi pentru reprezentări rare și învățarea dicționarelor atunci când, în reprezentările rare aferente, atomii nu mai sunt simpli vectori ficși, ci sunt aleși dintr-o mulțime infinită, de exemplu un con. În acest fel cresc flexibilitatea și precizia reprezentării. Aplicația principală în anul 2023 este detecția de anomalii.

Conținutul raportului:

1. Descriere științifică, cu punerea în evidență a rezultatelor etapei anuale și gradul de realizare a obiectivelor.
2. Un sumar al progresului (livrabile realizate, indicatori de rezultat, diseminarea rezultatelor, justificare diferențe, dacă e cazul)
3. Un rezumat executiv al activităților realizate în perioada de implementare
4. Alte informații despre proiect
5. Concluzii
6. Bibliografie

1. Descriere științifică

Etapa 2023 a proiectului AsyDiL are cinci activități:

- Versiuni kernel ale reprezentărilor cu atomi-mulțime
- Învățarea dicționarelor, etapa de actualizare, cu atomi-mulțime (bazată pe optimizare)
- Biblioteci MATLAB și Python pentru algoritmi realizați în 2022
- Proiectare algoritmi și acordare hiper-parametri (pentru detecția de anomalii)
- Detecție de anomalii pe date de test publice

Vom prezenta în această secțiune rezultatele obținute pentru fiecare din activități.

1.1. Versiuni kernel ale reprezentărilor cu atomi-mulțime

Reamintim punerea problemei de reprezentare rară cu dicționare conținând atomi mulțime, în cele două variante pe care le-am explorat.

Atomi con. Fiecare atom al dicționarului este un con $\mathcal{C}(d, \rho)$, în care d , cu $\|d\| = 1$, este atomul central și ρ este raza conului, care conține toți vectorii a , cu $\|a\| = 1$, pentru care $\|a - d\| \leq \rho$. Pentru reprezentarea rară, atunci când un atom-con d este selectat, din el se utilizează atomul

efectiv a care este cel mai util în minimizarea erorii de reprezentare. Asociind un con fiecărui atom dintr-un dicționar $D \in R^{m \times n}$, pentru reprezentarea unui semnal y trebuie minimizată funcția obiectiv

$$\|y - \sum_{i=1}^n x_i a_i\|^2 \quad (1)$$

cu restricțiile $\|x\|_0 \leq s$ (numărul de coeficienți nenuli este cel mult s) și $a_i \in C(d_i, \rho_i)$ (atomul efectiv aparține conului asociat atomului central respectiv).

Atomi gaussieni. Asociem o probabilitate fiecărui potențial atom efectiv. Dacă d este un atom central, atunci un atom efectiv a aparține mulțimii $G(d, \sigma)$ cu probabilitatea

$$p(a, d) \sim \exp\left(-\frac{\|a - d\|^2}{2\sigma^2}\right)$$

Pentru un dicționar $D \in R^{m \times n}$, asociem fiecărui atom d_j o distribuție $G(d_j, \sigma_j)$. Pentru a calcula o reprezentare în acest context, maximizăm probabilitatea atomilor din reprezentare, asigurând în același timp și o eroare mică de reprezentare. Funcția obiectiv este

$$\sum_{i=1}^n \frac{1}{\sigma_i^2} \|a_i - d_i\|^2 + \lambda \|y - \sum_{i=1}^n x_i a_i\|^2$$

(2)

Primul termen reprezintă logaritmul negativ al probabilității (log-likelihood) atomilor efectivi $a_i \in G(d_i, \sigma_i)$, deci minimizarea lui maximizează probabilitatea. Al doilea termen, ponderat cu o constantă λ , este eroarea de reprezentare a semnalului y .

Versiuni kernel. Pentru ambele probleme de mai sus, ilustrăm modificările necesare pentru termenul de eroare. Extensia la neliniaritate care conduce la metodele kernel se face printr-o funcție φ ; acesteia îi este asociată o funcție kernel κ , cu proprietatea (Mercer) $\kappa(u, v) = \varphi(u)^T \varphi(v)$. Un exemplu tipic este RBF (Radial Basis Function), cu expresia $\kappa(u, v) = \exp(-\|u - v\|^2 / 2\zeta^2)$. Avantajul unei astfel de funcții kernel este modul rapid de calcul și lipsa necesității unei forme explicite a funcției φ .

În acest context, este naturală redefinirea noțiunii de con. Notăm acum $C(d, \rho)$ mulțimea vectorilor a pentru care $\|\varphi(a) - \varphi(d)\| \leq \rho$. Termenul de eroare din (2) poate fi scris pe scurt $\|y - Ax\|$, unde A este matricea atomilor efectivi și x vectorul de coeficienți asociat. Extensia la metode kernel se face prin definirea erorii sub forma $\|\varphi(y) - \varphi(A)x\|$, unde prin $\varphi(A)$ înțelegem aplicarea funcției φ asupra fiecărei coloane a matricei A .

În esență, metodele kernel modifică noțiunea de distanță, dar calculele implicând norme sunt directe. De exemplu, putem scrie

$$\begin{aligned} \|\varphi(y) - \varphi(A)x\|^2 &= \varphi(y)^T \varphi(y) - 2\varphi(y)^T \varphi(A)x + x^T \varphi(A)^T \varphi(A)x \\ &= \kappa(y, y) - 2\kappa(A, y)x + x^T \kappa(A, A)x \end{aligned}$$

Observăm că expresia din dreapta este tot o formă pătratică în x , a cărei optimizare se face similar cu cazul standard.

Metodele kernel se pot deci adapta destul de direct la dicționare cu atomi-mulțime, prin modificarea noțiunii de distanță, în funcție de funcția kernel aleasă. În cazul DL, structura dicționarului se complică [Dulr18], dimensiunea lui crescând datorită construcției acestuia plecând

de la semnalele disponibile pentru antrenare. Din păcate, deși am implementat algoritmi de reprezentare cu kernel, rezultatele obținute sunt dezamăgitoare. Este posibil ca o cauză să fie apariția unor parametri suplimentari, precum și timpul mai mare de execuție, care permite căutări mai puține, de unde să rezulte inabilitatea noastră de a găsi valori adecvate. Cercetarea în această direcție nu a fost abandonată, dar nu sunt semne de optimism în rezultatele de până acum.

Alte extensii. Lucrul cu metode kernel, în contextul de detecție de anomalii (subiect principal în acest proiect), a condus la lucrarea [IDR23]. Căutarea de metode de extragere de trăsături a condus natural la legătura cu metode de vizualizare, cum este t-SNE [Mahi08]. Contextul acestor metode este cel al reducerii dimensionale, care în t-SNE se efectuează cu ajutorul a două distribuții de probabilitate: cea rezultată din punctele existente (într-un spațiu cu multe dimensiuni) și cea dorită pentru punctele din spațiul cu puține dimensiuni. Corespondența dintre ele se realizează prin divergențe Kullback-Leibler. Minimizarea sumei acestora conduce la găsirea punctelor în spațiul redus dimensional.

Propunerea din [IDR23] este de a utiliza nu punctele din spațiul original, ci cele dintr-un spațiu extins cu ajutorul unei funcții kernel. Aceasta se poate face și pentru spațiul inițial și pentru cel redus dimensional, simultan sau separat, așa cum este ilustrat în Fig.1 (E2E este abrevierea pentru “end-to-end”). Algoritmul din [Mahi08], pe bază de gradient, se poate adapta la varianta kernel, iar reducerea complexității se poate face prin aproximări de tip Nystrom. Articolul [IDR23] prezintă rezultate și indicatori de performanță care arată că algoritmul propus poate separa mai bine clasele atunci când sunt vizualizate seturi de date cu clase cunoscute (Fig.2 prezintă un astfel de exemplu) și de asemenea poate oferi valori mai bune pentru parametrul încredere (trustworthiness)

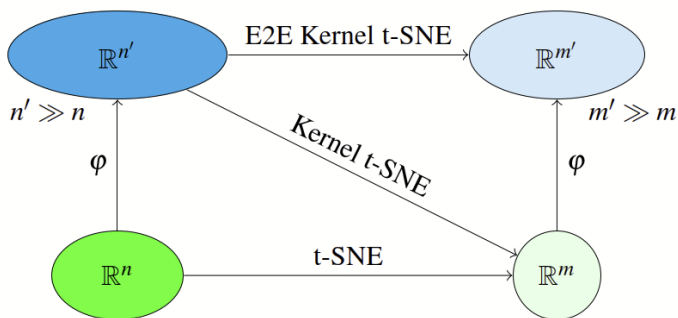


Fig.1 Schema transformărilor kernel t-SNE.

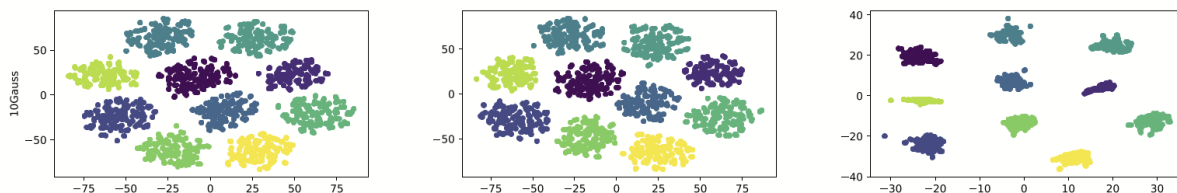


Fig.2 Rezultatele vizualizării setului de date 10Gauss pentru t-SNE (stânga), kernel t-SNE (mijloc) și E2E kernel t-SNE (dreapta).

Concluzie. Gradul de realizare a acestei activități: complet, în sensul explorării științifice, dar fără rezultate notabile. Vom căuta moduri alternative de punere a problemei.

1.2. Învățarea dicționarelor, etapa de actualizare, cu atomi-mulțime (bazată pe optimizare)

Conținutul acestei secțiuni se bazează pe [BID23], [IBD23a].

Problema antrenării de dicționare (dictionary learning—DL) pentru atomi mulțime se pune similar celei clasice, în care atomii sunt vectori. Se dau semnalele de antrenare, organizate într-o matrice $Y \in R^{m \times N}$ și trebuie găsit dicționarul care dă reprezentarea lor rară optimă, de obicei pentru un nivel de sparsitate s dat. Diferența e că, în principiu, trebuie antrenați și atomii centrali și razele ρ_j ale conurilor (în cazul atomilor con) sau varianțele σ_j (în cazul atomilor gaussiani). Pentru a simplifica problema, am presupus că doar atomii centrali sunt variabili, în timp ce razele sau varianțele sunt fixe.

Atomi con. Pentru atomi con, am găsit încă de anul trecut (în avans), o metodă de optimizare inspirată de algoritmul PAK-SVD (Parallel AK-SDV: varianta secvențială se găsește în [RZE08], cea paralelă în [Dulr18]). Algoritmul rezultat, numit Cone-DL [BID23], are structura iterativă obișnuită a algoritmilor DL. Fiecare iterație este formată din două etape: reprezentare rară, realizată cu Cone-OMP [IBD23], și actualizare a atomilor, realizată cu metoda de optimizare adaptată din PAK-SVD. Anul acesta am realizat implementări și teste, în special pentru detecția de anomalii, ce vor fi detaliate în Sec. 1.5. Prezentăm aici, în Tab.1, doar timpii de execuție pentru Cone-DL pe 30 de seturi de date din ADBench [HHH22], alături de cei ai metodei clasice AK-SVD (care utilizează OMP pentru reprezentare rară, într-o implementare foarte eficientă și pre-compilată). Se observă că Cone-DL este de doar câteva ori mai lent, deși rezolvă o problemă mult mai complexă. În tabel, parametrul $c = n/m$ este raportul dintre dimensiunea dicționarului și cea a semnalelor; grade de supradimensionare cu valori 2-4 sunt tipice în DL.

Algorithm	$s = 2$			$s = 3$	
	$c = 2$	$c = 3$	$c = 4$	$c = 2$	$c = 3$
AK-SVD + OMP	59.45	76.84	89.70	70.16	92.84
Cone-DL	156.66	166.18	167.16	310.78	314.34

Tab.1. Timpii de execuție pentru 30 de seturi de date din ADBench.

Fig.3 ilustrează evoluția erorii medii a Cone-DL pentru unul din seturile de date din ADBench, comportament destul de tipic; pentru ceilalți algoritmi este arătat doar nivelul erorii finale, printr-o linie orizontală. Cone-DL reușește atingerea unei erori mai mici decât AK-SVD urmat de Cone-

OMP, confirmând astfel că antrenarea are efect și dicționarul obținut este mai bun. Prin ‘swap’ este denumit algoritmul care, după terminarea fazei de antrenare, permută razele conurilor după numărul de semnale în care este utilizat atomul respectiv; atomii cei mai folosiți primesc raze mai mari; în acest exemplu, razele sunt aleator distribuite în intervalul [0.01,0.1]. Se observă că pentru Cone-DL, această realocare nu are efect benefic, ceea ce confirmă încă o dată că antrenarea este bine efectuată pe distribuția inițială a razelor. În schimb, pentru Cone-OMP și dicționarul preantrenat cu AK-SVD, eroarea scade ușor, ceea ce arată suboptimalitatea acestei metode.

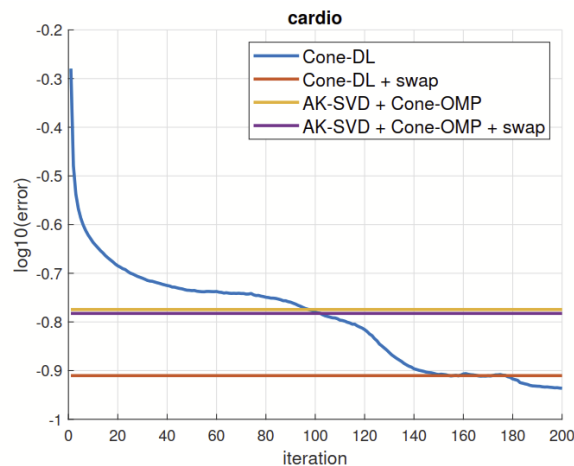


Fig.3 Evoluția erorii medii a Cone-DL pe setul de date *cardio*.

Atomi gaussiani. Pentru acești tip de atomi, anul trecut am conceput un algoritm numit Gauss-OMP. Între timp am realizat implementarea lui și, mai ales, am demonstrat proprietăți care asigură convergența. De notat că optimalitatea este o problemă dificilă și că, în general, în reprezentări rare sunt extrem de puține rezultate de acest tip.

Anul acesta am găsit o metodă nouă de reprezentare, numită Gauss-L1, pe care o prezentăm în continuare. Ea se bazează pe optimizarea unei funcții obiectiv de tip relaxare convexă, de forma

$$\sum_{i=1}^n \frac{1}{\sigma_i^2} \|a_i - d_i\|^2 + \lambda \|y - \sum_{i=1}^n x_i a_i\|^2 + \gamma \|x\|_1$$

(3)

în care ultimul termen este adăugat la (2) pentru a încuraja apariția zerourilor în vectorul de coeficienți x , pentru a obține o reprezentare rară. Se renunță astfel la un număr de coeficienți s impus, lăsând flexibil nivelul de raritate, care este reglat prin ponderea γ . Detaliile modului de rezolvare, bazat pe coborâre pe coordonate, se găsesc în [IBD23a]. Prezentăm aici doar formulele rezultate. Presupunând toate variabilele fixate, mai puțin un atom efectiv a_j , optimizarea acestuia se efectuează cu

$$a_j = \frac{\sigma_j^2}{\lambda x_j^2 \sigma_j^2 + 1} (x_j \hat{y} + \frac{1}{\sigma_j^2} d_j)$$

unde $\hat{y} = y - \hat{A}\hat{x}$ este reziduul datorat celorlalți atom efectivi \hat{A} , din care lipsește atomul curent, cu reprezentările \hat{x} aferente. Ținând acum toate variabilele fixate, mai puțin coeficientul x_j , expresia optimă a acestuia este

$$x_j = \text{soft}(a^T_j \hat{y}, \frac{\gamma}{2\lambda})$$

unde funcția *soft* reprezintă operația “soft thresholding”.

După cum se vede, formulele de mai sus sunt simple, comparabile din punctul de vedere al complexității cu cele din AK-SVD, deși acolo atomii sunt vectori, nu mulțimi.

După ce am prezentat mai sus două metode de reprezentare rară cu atomi gaussieni, putem acum prezenta modul de actualizare a atomilor centrali, care este esențial pentru un algoritm DL și care se poate combina cu ambele moduri de reprezentare. Reamintim că avem o mulțime de semnale $Y \in R^{m \times N}$, din care dorim optimizarea dicționarului D al atomilor centrali. Făcând din nou trimitere la [IBD23a] pentru detalii, amintim aici doar rezultatul. În etapa de reprezentare dintr-o iterație a algoritmului DL, numit DL-Gauss-L1 sau DL-Gauss-OMP, după algoritmul de reprezentare folosit, pentru semnalul y_l sunt folosiți în reprezentare atomii efectivi a_{il} . Pe baza acestor atomi, noul atom central d_i are expresia optimă

$$d_i = \sum_{l, x_{il} \neq 0} a_{il}$$

adică o simplă mediere. Este esențial că această expresie poate fi calculată pe măsură ce se calculează reprezentările, fără a fi necesară memorarea atomilor efectivi.

Deși convergența algoritmilor propuși nu e garantată, comportarea practică este foarte bună. Fig.4 ilustrează convergența funcției obiectiv a algoritmului Gauss-L1 pentru 100 de semnale din setul *landsat* din ADBench [HHH22]. Se observă scăderea cvasi-permanentă a obiectivului. Comportamentul DL-Gauss-L1 este similar. DL-Gauss-OMP are o comportare mai haotică, dar tendința generală este clar descrescătoare.

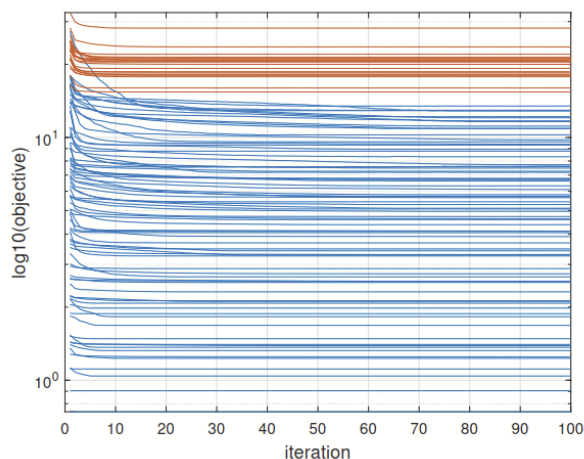


Fig.4 Evoluția funcției obiectiv a algoritmului Gauss-L1 pe 100 de semnale.

Alte rezultate. În explorarea metodelor de optimizare potrivite pentru problemele neconvexe, cum sunt cele din DL, am inclus și metode de optimizare globală. Articolul [MD23] folosește o meta-heuristică în care mai mulți agenți de tip Differential Evolution (DE) [StPr97] rezolvă problema de optimizare, parametrii lor fiind optimizați pe parcurs de un algoritm genetic. Articolul este numit

HyperDE. El este o extensie a lucrării de conferință [MD23c], în care agentul este Sparrow Search Algorithm (SSA) [XuSh20]. HyperDE este mult mai bun decât HyperSSA și de asemenea mai bun decât algoritmi consacrați de tip DE, ca de exemplu SHADE [TaFu13] și L-SHADE [TaFu14]. Testele au fost efectuate pe probleme de la competiții CEC foarte recente.

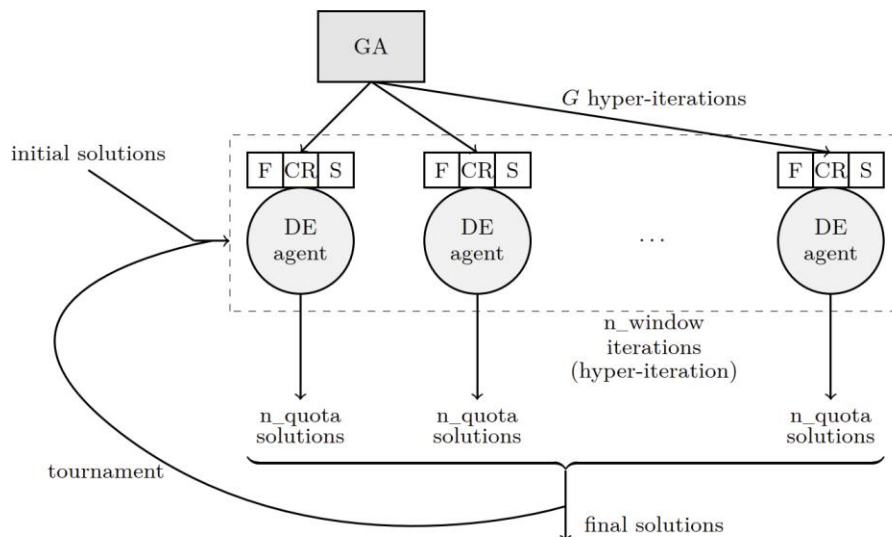


Fig.5 Schema bloc a algoritmului HyperDE.

Schema algoritmului este prezentată în Fig.5. Algoritmul genetic (GA), aflat la nivelul cel mai înalt, modifică valorile parametrilor algoritmilor DE, care produc n_quota soluții ale problemei de rezolvat, operând timp de n_window iterații. Algoritmii DE au trei parametri (F, CR, S), dar metoda este generică, putând lucra cu orice agent parametrizat. GA operează pe baza soluțiilor produse de agenți, încercând să ajusteze parametrii astfel încât soluțiile obținute să se îmbunătățească.

Concluzie. Gradul de realizare a acestei activități: complet.

1.3. Biblioteci MATLAB și Python pentru algoritmi realizați în 2022

Am implementat un repository dedicat și public (<https://gitlab.cs.pub.ro/asydil>) în care am introdus sub forma unor proiecte codul sursă necesar reproducerii rezultatelor incluse în majoritatea lucrărilor publicate. Acest repository reprezintă un punct de plecare pentru buna gestionare a codului sursă. De asemenea, programele sunt disponibile pe site-ul proiectului, <http://asydil.upb.ro/>.

Sunt implementați toți algoritmi descriși în lucrări, în formă modulară. De asemenea, există programe de test care pot fi utilizate direct pentru a reproduce rezultatele din lucrare sau, cu modificări minore, rezultate conexe. Fiecare proiect este însoțit de un fișier *readme* care descrie modul de utilizare.

Până la sfârșitul anului vom grupa programele create într-o formă unitară, rezultatul final fiind o bibliotecă dedicată sarcinilor de detecție a anomaliilor. Biblioteca va fi disponibilă în Matlab și Python. Deocamdată fiecare problemă este rezolvată într-un singur limbaj. De notat că nu este vorba doar de algoritmi realizați în 2022, unii dintre ei rafinați sau corecți anul acesta, dar și de algoritmi corespunzând articolelor din 2023, ceea ce constituie un avans prevăzut în planul de realizare din 2024.

Concluzie. Gradul de realizare a acestei activități: satisfăcător în acest moment (1 decembrie), cu premise foarte bune de terminare până la sfârșitul anului.

1.4. Proiectare algoritmi și acordare hiper-parametri (pentru detecția de anomalii)

Detecția de anomalii cu DL se face de obicei [AEH15, YMW19, PBT20, HZS21] în două etape, în mod nesupervizat. În prima se antrenează un dicționar cu toate datele disponibile pentru antrenare, indiferent de tipul lor (semnale normale sau anomalii). În a doua etapă se face ordonarea semnalelor după scorul de anomalie ales, care este de obicei eroarea de reprezentare; pentru cazul dicționarelor cu atomi mulțime, aceasta are expresia (1); în cazul dicționarelor standard, în care atomii sunt vectori, eroarea are o expresie similară, în care apar atomii (în locul atomilor efectivi). Deoarece se presupune că semnalele normale sunt multe și similare între ele, procesul de optimizare asociat DL tinde să reprezinte bine aceste semnale, în detrimentul anomaliilor, care sunt puține. DL produce mulți atomi care sunt bine specializați pentru semnalele normale, dar alocă puțini atomi anomaliilor, pentru care o eroare mai mare nu afectează prea mult funcția obiectiv a DL. Așadar, se consideră anomalii semnalele reprezentate cu eroare mare. De exemplu, în Fig.4, anomaliile sunt reprezentate cu roșu și semnalele normale cu albastru; este un caz fericit, în care toate anomaliile reale au într-adevăr o eroare mai mare decât a semnalelor normale.

În cazul dicționarelor cu atomi conuri, vom folosi în continuare eroarea (1) ca scor de anomalie. În cazul atomilor gaussieni, sunt mai multe posibilități:

- eroarea de reprezentare (1);
- funcția obiectiv (2), pentru Gauss-OMP și DL-Gauss-OMP, respectiv (3) pentru Gauss-L1 și DL-Gauss-L1;
- probabilitatea reprezentării, asociată în cazul L1 cu distanța dintre atomii efectivi și cei centrali

$$\sum_{i=1}^n \frac{1}{\sigma_i^2} \|a_i - d_i\|^2;$$

pentru cazul OMP, suma de mai sus conține doar termenii pentru care coeficienții x_i sunt nenuli și se adaugă și $m \log \sigma_i$, pentru a ține seama de varianța asociată atomilor utilizați (în varianta L1, toți atomii pot fi utilizați).

Hiperparametri. Metodele propuse conțin mai mulți hiperparametri. Pentru a fi utile, este bine ca mulți dintre aceștia să aibă valori fixe sau să fie date rețete de alegere a valorilor în funcție de context. Prezentăm aici câteva concluzii utile în detecția de anomalii.

În algoritmul Cone-DL [BID23]:

- Razele conurilor trebuie să aibă valori mici. Pentru raze egale, valoarea $\rho = 0.05$ este o alegere robustă. Pentru raze variabile, valori mai mici decât 0.1 (de exemplu uniforme distribuite) par a fi convenabile. Pentru raze mai mari, rezultatele devin mai slabe, probabil din cauză că eroarea de reprezentare scade foarte mult, atât pentru semnalele normale cât și pentru anomalii.
- Gradul de supradimensionare a dicționarelor $c = n/m$ are valorile uzuale, de exemplu 2 sau 3.
- Nivelul de raritate s are valori mici, de exemplu 2 sau 3. Valori mai mari conduc la specializare mai slabă a atomilor și deci la reprezentări (indezirabil de) bune ale anomaliilor.
- Cone-DL conține un parametru care specifică distanța minimă între conuri, astfel încât să nu existe suprapuneri. Am obținut rezultate foarte bune cu $\delta_0 = 0.01$. Valori mai mari cresc incoerența, ceea ce în principiu este mai bine pentru găsirea reprezentării optime, dar nu neapărat și pentru detecția de anomalii.
- În Cone-OMP, după adăugarea unui atom la suport, e necesară recalcularea atomilor efectivi deja calculați printr-o metodă de optimizare succesivă: fiecare atom rezultă din proiecția reziduului curent (fără aportul atomului) pe conul respectiv. Am constatat că 10 iterații de recalculare sunt suficiente, chiar dacă nu întotdeauna se obține soluția optimă.

În algoritmul Gauss-L1 [IBD23a]:

- Pentru alegerea deviațiilor standard σ_i din (3), se aplică aceleași observații ca pentru razele conurilor în Cone-DL.
- Idem, pentru gradul de supradimensionare a dicționarelor
- Parametrii λ și γ din (3) se pot lua egali, ponderând astfel în mod egal eroarea și termenul de încurajare a rarității. Valorile lor absolute, adică modul de ponderare a erorii față de probabilitate (primul termen în (3)), pot acoperi un domeniu larg, după cum se va vedea în Sec. 1.5.1.
- Algoritmul Gauss-L1 folosește FISTA [BT09] pentru inițializare. Parametrii FISTA folosiți sunt: 50 de iterații și pas $t = 0.0001$.

În plus, algoritmul Gauss-OMP folosește algoritmul secțiunii de aur pentru găsirea unui atom efectiv atunci când toți ceilalți sau dați (sau fixați). Parametrii sunt cei implicați.

Rezultatele descrise mai sus nu au condus direct la publicații, dar stau la baza multor experimente și simulări descrise în secțiunea următoare. Ele au fost utilizate în [BID23], [IBD23a]. Prezentăm mai departe alte rezultate utilizând DL în detecția de anomalii.

DL incoerent. În [ID23a] am propus o nouă abordare pentru problema învățării de dicționare incoerente (IDL), utilizând o funcție de barieră a distanței care promovează incoerența. Această funcție poate fi utilizată atât în structuri locale, cât și globale. Această strategie obține rezultate mai bune în ceea ce privește eroarea de reprezentare și incoerența dicționarului în comparație cu problema standard, demonstrând îmbunătățirea performanței dicționarelor în probleme de

clasificare. Lucrarea propune două formulări ale problemei IDL: prima este o modificare simplă a problemei IDL prin introducerea unei bariere în spiritul metodei "contrastive loss"; a doua este inspirată de funcția "triplet loss" din învățarea automată. Ambele formulări sunt în folosite în practică pentru promovarea discriminării între clase și spații de reprezentare diferite. În această abordare, folosim cele două funcții pentru a discrimina dicționarele din clase diferite, promovând incoerența între clase. Experimentele au fost efectuate pe mai multe seturi de date și au demonstrat că metodele propuse IDB și ITDB au performanțe generale mai bune decât problema standard IDL și alte metode cunoscute în literatură, obținând o precizie superioară, dar cu un timp de rulare mai mare.

DL pentru ABOD. În [ID23b] am propus o modificare a metodei Angle-Based Outlier Detection (ABOD) [KSZ08] pentru detecția de anomalii. Strategia constă într-o extensie a algoritmului ABOD, prin combinarea ei cu algoritmi DL. Metoda ABOD este îmbunătățită prin construirea unui spațiu metric de reprezentare a datelor normale cu atomii obținuți din problema DL. Anomaliile sunt detectate mult mai ușor prin raportarea semnalelor la baza de reprezentare a atomilor din dicționare. Demonstrăm prin experimente practice că această strategie este capabilă să izoleze mult mai bine anomaliile. Metoda ABOD este invariantă la seturile de date care conțin anomalii grupate sub forma unor clustere. Prin raportarea la problema DL, metoda ABOD este capabilă să identifice și genul acesta de anomalii. În general, atomii pot captura direcția semnalelor normale, izolând astfel mai eficient anomaliile. Această abordare oferă o nouă perspectivă în detecția anomaliilor, îmbunătățind capacitatea de a distinge între datele normale și abateri.

Concluzie. Gradul de realizare a acestei activități: complet.

1.5. Detecție de anomalii pe date de test publice

În raportarea performanțelor utilizăm scorul numit aria de sub curba Receiver Operating Characteristic (ROC AUC), utilizat adesea în detecția de anomalii. Curba ROC ilustrează relația dintre rata de fals pozitive și rata de adevărat pozitive. Indicatorul ROC AUC raportează aria de sub curba ROC la pătratul unitate, astfel că o valoare de 1 a indicatorului semnalează o identificare perfectă a anomaliilor, în timp ce valoarea 0.5 corespunde unei preziceri aleatoare.

1.5.1 Teste pe datele ADBench

Atât algoritmi Cone-DL [BID23], cât și algoritmi propuși pentru antrenarea dicționarelor cu atomi probabilistici (Gauss-OMP și Gauss-L1, pentru calculul reprezentării și DL-Gauss-L1, DL-Gauss-OMP pentru antrenarea dicționarelor) [IBD23a], au fost testați pe unul dintre cele mai noi și probabil cel mai important *benchmark* în domeniul detecției de anomalii la momentul actual, ADBench [HHH22]. ADBench conține 57 baze de date, din domenii diverse (medicină, fizică, sociologie, finanțe, lingvistică ș.a.) și având proprietăți diferite (raport diferit între numărul de semnale și dimensiunea acestora, proporții diferite de anomalii). De asemenea, conține 14 metode de detecție nesupervizată de anomalii, atât consacrate, cât și noi (inclusiv rețele neuronale).

Autorii propun o taxonomie a anomaliilor în funcție de similaritatea acestora cu semnalele normale: anomaliile *global*, *locale*, *dependency* și *cluster*. Pentru fiecare tip, ADBench conține câte o funcție pentru a genera sintetic anomaliile, pornind de la semnalele normale din fiecare bază de date.

Din cele 57 baze de date am selectat 30 pe care am desfășurat în continuare testele. Într-o primă fază, am verificat performanțele algoritmilor propuși pe fiecare dintre cele 4 tipuri de anomalii. Când bazele de date conțin anomalii de tip *global*, problema este ușor de rezolvat, atât de soluțiile propuse, cât și de metodele din *benchmark*: pentru multe baze de date se obține un scor de detecție perfect. Anomaliile de tip *local* și *cluster* nu se pretează abordării cu algoritmi DL atunci când scorul de anomalie este bazat pe eroarea de reprezentare. Pentru a putea folosi DL în aceste cazuri sunt necesare alte metode de a eticheta semnalele. Motivul, în cazul anomaliilor *local*, este faptul că acestea sunt foarte similare cu multe semnale normale, ducând la erori mici de reprezentare. Asemănător, în cazul anomaliilor *cluster*, gruparea acestora conduce la o bună reprezentare. În baza concluziilor de mai sus am desfășurat în continuare teste suplimentare pentru anomaliile de tip *dependency*. De notat observația din [HHH22] că nicio metodă nu are performanțe ridicate pentru toate tipurile de anomalii.

Rezultatele metodelor care utilizează atomi conuri (Cone-DL)

În trei dintre bazele de date, Cone-DL obține un scor de detecție perfect (ROC AUC =1), în timp ce nicio metodă din ADBench nu atinge această performanță (pe niciuna din cele 30 baze de date). În general, rezultatele Cone-DL sunt corelate cu dimensiunea semnalelor: cu cât dimensiunea acestora este mai mare, cu atât predicția este mai bună. Tabelele 2 și 3 ilustrează această tendință. Tabelul 2 prezintă rangul (pozițiile medii ale) algoritmilor DL între cele 14 metode din ADBench pentru toate cele 30 baze de date. Sunt evaluate metoda DL clasică (AK-SVD+OMP), algoritmul Cone-DL propus, cât și varianta AK-SVD+Cone-OMP, anume în care sunt utilizați atomi conici doar în calculul reprezentării, în timp ce antrenarea dicționarului se realizează clasic, cu AK-SVD. Valorile parametrilor sunt cele descrise în Secțiunea 1.4. Testele sunt realizate atât utilizând raze constante, cât și raze distribuite uniform. Un rang mai mic de 2.5 înseamnă că metodele DL sunt mai bune decât toți algoritmi ADBench. Cea mai bună metodă din ADBench are rangul 2.6.

Tabelul 3 prezintă același clasament, de data aceasta luând în considerare doar primele 12 baze de date (ordonate descrescător după dimensiunea semnalelor), anume pentru care $m > 20$.

Algorithm	Radii	$s = 2$			$s = 3$	
		$c = 2$	$c = 3$	$c = 4$	$c = 2$	$c = 3$
AK-SVD + OMP	-	2.13	1.93	2.07	2.27	2.43
Cone-DL	$\rho = 0.05$	2.03	1.87	1.93	3.23	3.13
AK-SVD + Cone-OMP		1.87	1.80	1.80	2.17	2.47
Cone-DL		2.07	1.93	1.97	4.30	3.60
AK-SVD + Cone-OMP	uniform	1.97	1.87	1.83	2.50	2.53
Cone-DL + swap	$\rho \in [0.01, 0.1]$	1.90	1.73	1.80	3.37	3.13
AK-SVD + Cone-OMP + swap		1.73	1.67	1.67	2.10	2.23

Tab. 2. Clasamentul metodelor DL între cele 14 metodele ADBench pe toate cele 30 baze de date testate

Algorithm	Radii	$s = 2$			$s = 3$	
		$c = 2$	$c = 3$	$c = 4$	$c = 2$	$c = 3$
AK-SVD + OMP	-	2.08	1.92	1.83	1.75	1.75
Cone-DL	$\rho = 0.05$	1.83	1.67	1.67	2.00	1.50
AK-SVD + Cone-OMP		1.92	1.67	1.67	1.58	1.58
Cone-DL		1.83	1.67	1.75	4.00	2.33
AK-SVD + Cone-OMP	uniform	1.83	1.50	1.67	1.58	1.58
Cone-DL + swap	$\rho \in [0.01, 0.1]$	1.75	1.58	1.75	3.08	2.08
AK-SVD + Cone-OMP + swap		1.67	1.58	1.75	1.50	1.50

Tab. 3. Clasamentul metodelor DL între cele 14 metodele ADBench pe 12 baze de date, în care $m > 20$

În lucrarea [BID23] prezentăm în detaliu rezultatele, inclusiv luând în considerare mediile indicatorului ROC AUC, iar în varianta extinsă a acesteia și valorile ROC AUC obținute de algoritmi DL și metodele ADBench pentru fiecare bază de date în parte. Acestea confirmă concluziile clasamentului prezentat mai sus, anume faptul că metodele propuse sunt superioare soluțiilor *state-of-the-art*.

Rezultatele metodelor care folosesc atomi-mulțimi probabiliști

Tabelul 4 prezintă rezultatele Gauss-OMP și DL-Gauss-OMP pentru cele 30 de baze de date, sub forma clasamentului în metodele ADBench și indicatorului ROC AUC. Fiind vorba de aceleași condiții de testare pentru algoritmi din *benchmark*, se păstrează observația precedentă: algoritmi DL sunt mai buni decât cei din ADBench acolo unde rangul este mai mic decât 2.5.

parameters	score	s	λ	σ	Gauss-OMP			DL-Gauss-OMP		
					rank	ROC	AUC	rank	ROC	AUC
err	2	1		0.05	4.07	0.8676	3.13	0.8967		
err	2	1		$\in [0.01, 1]$	6.77	0.8092	4.10	0.8817		
obj	2	10		0.05	4.27	0.8643	2.20	0.9271		
obj	2	10		$\in [0.01, 1]$	6.70	0.8176	3.37	0.9192		
prob	2	100		0.05	3.77	0.8742	3.57	0.8802		
prob	2	100		$\in [0.01, 1]$	5.67	0.8267	2.50	0.9212		
prob	2	1000		0.05	3.70	0.8685	1.93	0.9401		
prob	2	1000		$\in [0.01, 1]$	6.80	0.8042	2.63	0.9357		
prob	3	1000		0.05	4.40	0.8464	2.10	0.9283		
prob	3	1000		$\in [0.01, 1]$	6.60	0.8110	2.47	0.9344		

Tab. 4. Rezultatele metodelor Gauss-OMP

Tabelul 5 prezintă rezultatele pentru metodele ce folosesc relaxarea L1. Spre comparație, cea mai bună metodă din ADBench obține o medie a ROC AUC de 0.9274. Se observă că toate variantele de alegere a parametrilor pentru Gauss-L1 și DL-Gauss-L1 conduc la rezultate mai bune.

parameters	σ	Gauss-L1			DL-Gauss-L1		
		rank	ROC	AUC	rank	ROC	AUC
$\lambda = \gamma = 0.001$	0.05	1.77	0.9479	1.77	0.9479		
	$\in [0.01, 1]$	1.73	0.9503	1.73	0.9503		
$\lambda = \gamma = 0.01$	0.05	1.80	0.9453	1.90	0.9454		
	$\in [0.01, 1]$	1.80	0.9457	1.80	0.9458		
$\lambda = \gamma = 0.1$	0.05	1.83	0.9465	1.83	0.9468		
	$\in [0.01, 1]$	1.73	0.9471	1.60	0.9476		
$\lambda = \gamma = 1$	0.05	1.73	0.9492	1.70	0.9513		
	$\in [0.01, 1]$	1.70	0.9498	1.63	0.9519		

Tab. 5. Rezultatele metodelor L1

1.5.2. Teste pe date din domeniul bancar

O direcție importantă a acestui proiect este detecția de anomalii pe date provenite din tranzacții bancare și organizate sub formă de graf; nodurile sunt clienți (mai multe conturi pot fi asociate cu aceeași persoană fizică sau juridică) și arcele conțin informații despre tranzacțiile dintre clienți, mai precis suma totală tranzacționată și numărul de tranzacții efectuate. Datele utilizate sunt descrise în [DBB22] și sunt de două feluri: i) date reale, provenite de la Libra Internet Bank, ii) date simulate prin injecția unor anomalii standard (clici, inele și stele, cu unele variații aleatoare) în grafuri aleatoare cu densități prestabilite.

În [BD23] este propus un scor de anomalie numit Community-Augmented Local Link Intensity (CALLI). Metoda utilizează comunități găsite cu algoritmul Louvain și intensități bazate pe medii geometrice, calculând raportul dintre suma intensităților în raport cu toate comunitățile și intensitatea în comunitatea proprie.

Un exemplu de rezultate este prezentat în Tab.6, pe un graf sintetic cu grad mediu aproximativ 17. CALLI, care este un simplu scor, are True Positive Rate (TPR) mult mai bună decât GAW, care este un vector de scoruri din [ECL19], cu o complexitate similară, bazat tot pe medii geometrice. De asemenea, adăugând scorul CALLI la metoda din [DBB22] (numită EGO în tabel) se observă o ușoară îmbunătățire a rezultatelor, deja foarte bune.

Method	TPR 0.1%	TPR 0.2%	TPR 0.5%	TPR 1%	AUC 1%	AN 0.1%	AN 0.2%	AN 0.5%	AN 1%
CALLI	0.2686	0.5047	0.6984	0.7267	0.6012	45	87	129	141
GAW	0.0058	0.0293	0.0445	0.0581	0.0412	1	5	9	12
EGO	0.3115	0.5984	0.9414	0.9670	0.7927	50	100	188	206
EGO & CALLI	0.3147	0.5995	0.9435	0.9670	0.7950	50	100	190	207
GAW & CALLI	0.1497	0.2288	0.3236	0.3885	0.2858	25	39	56	71

Tab.6. Rezultate CALLI.

În [ZD23] am prezentat un algoritm online de actualizare a trăsăturilor pe care se bazează metoda din [DBB22]. Trăsăturile sunt de tip egonet (subgraful format dintr-un nod și toți vecinii lui direcți) și egonet redus. Pentru calculul periodic, de exemplu zilnic, metoda din [DBB22] are o complexitate convenabilă, de aproximativ o oră în cazul datelor Libra. Pentru calcul online, recalcularea egoneților implică o complexitate nepractică; pentru unii egoneți sunt necesare chiar și zeci de secunde. Algoritmul online propus reușește să actualizeze exact trăsăturile fără a recalcula egoneții, folosind doar vecinii comuni a două noduri, care sunt mult mai ușor de calculat. Creșterea de viteză este semnificativă. Fig.6 prezintă o histogramă a creșterilor de viteză a algoritmului online față de cel bazat pe recalcularea egoneților. Pentru nodurile cu puțini vecini, care sunt majoritatea, creșterea de viteză are valori între 3 și 5. Pentru nodurile cu mulți vecini, creșterea de viteză are valori mult mai mari, chiar și peste 300. Deci algoritmul online este într-adevăr practic, timpul maxim de execuție fiind aproximativ 0.5 secunde.

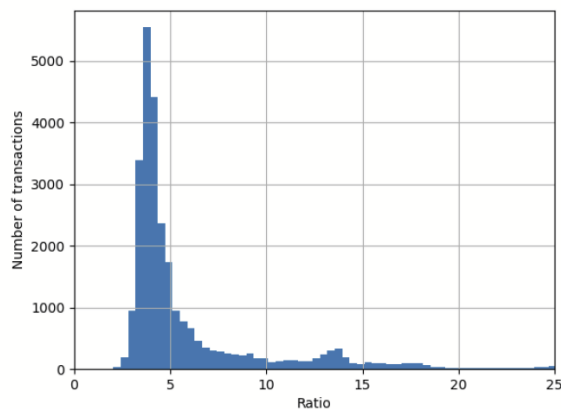


Fig.6 Histograma raporturilor dintre timpii de recalculare a trăsăturilor și timpii online.

De asemenea, am comparat performanța între detecția de anomalii online și cea efectuată la sfârșitul zilei; prima folosește un detector antrenat în ziua precedentă, a doua face antrenarea chiar înainte de detecție. Am observat că diferențele sunt ne semnificative, ceea ce dă și mai mare importanță metodei online propuse.

Concluzie. Gradul de realizare a acestei activități: complet.

2. Sumarul progresului

În anul 2023, până la data de 1 decembrie, am reușit următoarele publicații:

- 2 articole publicate în reviste [ID23], [MD23]. Primul a fost realizat în bună măsură în 2022.
- 6 articole acceptate (5 deja prezentate, cu lucrări vizibile pe ieeexplore) la conferințe indexate WoS [IBD23], [ID23a], [ID23b], [BD23], [MD23c], [ZB23]. Doar primul a fost trimis anul trecut în versiune inițială, revizuit însă anul acesta.
- Două articole trimise spre publicare, unul la revista Signal Processing (unde a trecut de primul rând de recenzii) [BID23], altul la IEEE Open Journal of Signal Processing [IBD23a].
- Un articol publicat pe arXiv [IDR23], cu rezultate utile, pentru care vom căuta un loc mai bun de publicare.

Articolele pot fi găsite pe site-ul proiectului, <http://asydil.upb.ro/>. Tot acolo se află și programele aferente lor.

Rezultatele descrise mai sus sunt peste cele prevăzute în planul de realizare, atât la publicații, cât și la algoritmi.

Prezentăm în continuare sumarul progresului pentru fiecare dintre cele cinci activități ale proiectului din anul 2023.

2.1. Versiuni kernel ale reprezentărilor cu atomi-mulțime

Am studiat versiunile kernel ale atomilor mulțime și dedus algoritmi pentru reprezentări rare cu ajutorul lor. Rezultatele obținute nu sunt mai bune decât pentru versiunile standard. Am obținut rezultate conexe [IDR23] sub forma unor versiuni kernel a algoritmului t-SNE de reducere dimensională. Per total, rezultatele la această activitate sunt sub așteptări.

2.2. Învățarea dicționarelor, etapa de actualizare, cu atomi-mulțime (bazată pe optimizare)

Această activitate, centrală pentru tema proiectului, a produs algoritmi de învățarea dicționarelor pentru atomi con și atomi gaussieni și s-a concretizat în articolele [BID23], [IBD23a]. Algoritmii sunt originali, au proprietăți bune de convergență, inclusiv unele susținute de rezultate teoretice. Utilizarea algoritmilor în detecția de anomalii a avut succes. Rezultate conexe au fost obținute în domeniul optimizării globale, publicate în [MD23]. Considerăm că această activitate este complet realizată, cu depășirea numărului de articole prevăzut.

2.3. Biblioteci MATLAB și Python pentru algoritmi realizați în 2022

Am implementat toți algoritmi propuși și i-am făcut publici pe gitlab, împreună cu programe de test și documentație primară. Până la sfârșitul anului ne vom asigura că toți algoritmi sunt implementați în ambele limbaje, inclusiv pentru algoritmi realizați în 2023, încheind cu succes această activitate.

2.4. Proiectare algoritmi și acordare hiper-parametri (pentru detecția de anomalii)

Am studiat modul în care se poate calcula scorul de anomalie în algoritmi cu atomi mulțime din [BID23], [IBD23a]. De asemenea, am studiat experimental modul de alegere a hiperparametrilor din acești algoritmi, găsim pentru majoritatea valori care se pot utiliza robust în mai multe aplicații. Am propus metode noi de clasificare [ID23a] și detecție de anomalii [ID23b], bazate pe învățarea dicționarelor. Apreciem că această activitate a fost îndeplinită complet, cu rezultate foarte bune.

2.5. Detecție de anomalii pe date de test publice

Am utilizat algoritmi de reprezentare rară și învățarea dicționarelor pentru detecția de anomalii pe seturi de date dintr-un banc de teste foarte recent [HHH22], concurând cu cei mai buni algoritmi actuali. Folosind dicționare cu atomi mulțime, am obținut cele mai bune rezultate pe unul din cele patru tipuri de bază de anomalii, numit 'dependency'. Am câștigat nu doar în fața metodelor în [HHH22], dar și în fața metodelor standard de învățare a dicționarelor. Rezultatele se găsesc în [BID23], [IBD23a]. De asemenea, am progresat în detecția de anomalii în date de tip graf, propunând algoritmi noi, cu rezultate bune, în [BD23], [ZD23]. Considerăm că această activitate a fost îndeplinită complet, cu rezultate excelente.

3. Rezumat executiv

Proiectul "Învățarea asimetrică a dicționarelor" (Asymmetric Dictionary Learning—AsyDiL) are ca scop deducerea unor algoritmi pentru reprezentări rare și învățarea dicționarelor atunci când, în reprezentările rare aferente, atomii nu mai sunt simpli vectori ficși, ci sunt aleși dintr-o mulțime infinită. Am studiat cazurile în care această mulțime este un con în jurul atomului central (atomi con) sau este caracterizată de o distribuție de probabilitate normală în jurul atomului central (atomi gaussieni). Am propus algoritmi noi (față de anul 2022) de reprezentare rară cu atomi gaussieni și algoritmi de învățare a dicționarelor. Aplicația principală a fost detecția de anomalii, unde au fost obținute rezultate foarte bune.

În etapa 2023 am obținut următoarele rezultate semnificative:

- Utilizarea algoritmului pentru învățarea dicționarelor cu atomi con din 2022, cu unele îmbunătățiri, la detecția de anomalii pentru tipul de anomalii numit 'dependency' din ADBench. Rezultatele sunt mai bune decât cele ale tuturor metodelor din ADBench, care este o bibliotecă implementând algoritmi de top. De asemenea, sunt clar mai bune decât cele obținute cu algoritmi standard de învățare a dicționarelor. A fost trimis un articol la revista Signal Processing, care a trecut de primul rând de recenzii.

- Un algoritm pentru învățarea dicționarelor cu atomi gaussieni. Algoritmul se bazează pe relaxare convexă și poate fi aplicat la detecție de anomalii. Se obțin rezultate mai bune decât cele de la punctul anterior. A fost realizat un articol trimis la o revistă bună.
- Doi algoritmi de detecție de anomalii în grafuri, dintre care unul online, publicați în lucrări de conferință.
- Un algoritm de optimizare globală bazat pe meta-euristici, cu rezultate de top pe seturi de date din competiții recente, publicat într-un articol de revistă.

Site-ul proiectului, <http://asydil.upb.ro/>, conține articolele realizate și seturi de programe complete care rezolvă problemele studiate, cu exemple relevante de funcționare, în majoritatea cazurilor pe date publice, unele reale, altele prelucrate din date reale.

4. Alte informații despre proiect

Echipa din 2022 (Bogdan Dumitrescu, Andra Băltoiu, Denis Ilie-Ablachim, Cristian Zica, Theodor Badea) a rămas stabilă. Un alt doctorand al lui BD, Alexandru Mănescu, a colaborat ocazional, fără a fi inclus oficial în echipă. Au avut loc întâlniri periodice, uneori online. Aceste întâlniri au fost deseori între directorul de proiect și unul sau doi membri ai echipei, deoarece orarul încărcat al cursurilor a îngreunat posibilitatea întâlnirilor întregului grup.

Nu au fost întâmpinate dificultăți administrative semnificative, deși unele achiziții au fost amânate din cauza transformării UPB după unirea cu Univ. Pitești, ceea ce a produs o perioadă de blocaj.

5. Concluzii

Din punctul de vedere al publicațiilor reușite, rezultatele sunt satisfăcătoare. Dacă vor fi acceptate cele două articole trimise în a doua jumătate a anului, care conțin rezultate centrale în tematica proiectului, rezultatele vor fi foarte bune. Algoritmii propuși de reprezentare rară și învățare a dicționarelor au dat rezultate foarte bune în detecția de anomalii, fiind mai buni decât cei dintr-o bibliotecă recentă cu algoritmi de top, pentru un tip fundamental de anomalii. Mare parte din rezultatele prevăzute la începutul proiectului au fost obținute, deși nu neapărat în forma imaginată inițial. Continuând în același ritm, putem încheia cu rezultate peste cele din propunerea de proiect.

Articole realizate

[BD23] T.A. Badea, B.Dumitrescu, "Community-Augmented Local-Link Intensity: a score for anomaly detection in graphs", Proc. 9th. Int. Conf. on Control, Decision and Information Technology (CoDIT), Rome, Italy, pp.1936-1941, July 2023. [ieeexplore paper](#)

[BID23] A.Băltoiu, D.C. Ilie-Ablachim, B.Dumitrescu, "Dictionary learning with cone atoms and application to anomaly detection", submitted, 2023. [versiune extinsa](#)

[IBD23] D.C. Ilie-Ablachim, A.Băltoiu, B.Dumitrescu, "Sparse representations with cone atoms", Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Rodos, Greece, June 2023. [ieeexplore paper](#)

[IBD23a] D.C. Ilie-Ablachim, A.Băltoiu, B.Dumitrescu, "Sparse representations with Gaussian atoms", submitted 2023.

[ID23] D.C. Ilie-Ablachim, B.Dumitrescu, "Incoherent frames design and dictionary learning using a distance barrier", Signal Processing, vol.209, art. 109019, Aug. 2023. WOS:000964485800001

[ID23a] D.C. Ilie-Ablachim, B.Dumitrescu, "Classification with dictionary learning and a distance barrier promoting incoherence", Proc. IEEE Int. Workshop Machine Learning in Signal Processing (MLSP), Rome, Italy, Sep. 2023. [ieeexplore paper](#)

[ID23b] D.C. Ilie-Ablachim, B.Dumitrescu, Angle-Based Dictionary Learning for Outlier Detection, Proc. IEEE Int. Conf. Signal, Control and Communication (SCC), Hammamet, Tunisia, Dec. 2023.

[IDR23] D.C. Ilie-Ablachim, B.Dumitrescu, C.Rusu, Kernel t-distributed stochastic neighbor embedding, arXiv, July 2023. <https://arxiv.org/abs/2307.07081>

[MD23] A.R. Manescu, B.Dumitrescu, HyperDE: An Adaptive Hyper-Heuristic for Global Optimization, Algorithms, vol. 16, no. 9, art. 451, Sep. 2023. WOS:001072482800001

[MD23c] A.R. Manescu, B.Dumitrescu, "Evolutionary hyper-heuristics for improving global optimization algorithms", Proc. 24th. Int. Conf. on Control Systems and Computer Science, Bucharest, Romania, May 2023. [ieeexplore paper](#)

[ZD23] C.E. Zica, B.Dumitrescu, "Online Computation of Reduced Egonet Features for Anomaly Detection in Bank Transactions Graphs", Proc. IEEE Int. Workshop Machine Learning in Signal Processing (MLSP), Rome, Italy, Sep. 2023. [ieeexplore paper](#)

Bibliografie

[AEH15] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "Sparse coding with anomaly detection," Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

[BT09] A. Beck and M. Teboulle. Fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci., 2(1):183–202, 2009.

[DBB22] B. Dumitrescu, A. Băltoiu, Ș. Budulan. Anomaly detection in graphs of bank transactions for anti money laundering applications. IEEE Access, 10:47699–47714, 2022.

[DuIr18] B. Dumitrescu, P. Irofti. Dictionary Learning Algorithms and Applications. Springer, 2018.

- [ECL19] A. Elliott, M. Cucuringu, M.M. Luaces, P. Reidy, G. Reinert. Anomaly detection in networks with application to financial transaction networks, arXiv, 2019.
- [HHH22] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao. ADBench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 35:32142–32159, 2022.
- [HZZ21] X. Han, H. Zhang, W. Sun, Spectral anomaly detection based on dictionary learning for sea surfaces, *IEEE Geoscience and Remote Sensing Letters* vol.19, pp. 1–5, 2021.
- [KSZ08] H.P. Kriegel, M. Schubert, A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 444–452, 2008.
- [Mahi08] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.
- [PBT20] B. Pilastre, L. Boussouf, S. d’Escrivan, J. Tourneret, Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning, *Signal Processing*, vol. 168, art. 107320, 2020.
- [RZE08] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. Technical Report CS-2008-08, Technion Univ., Haifa, Israel, 2008.
- [StPr97] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.
- [TaFu13] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution. In *Proceedings of the 2013 IEEE congress on evolutionary computation (CEC)*, pp. 71–78, 2013.
- [TaFu14] R. Tanabe, A. Fukunaga, Improving the search performance of SHADE using linear population size reduction. In *Proceedings of the 2014 IEEE congress on evolutionary computation (CEC)*, pp. 1658–1665, 2014.
- [YMW19] Y. Yuan, D. Ma, Q. Wang, Hyperspectral anomaly detection via sparse dictionary learning method of capped norm, *IEEE Access*, vol.7, pp.16132–16144, 2019.
- [XuSh20] J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering*, vol.8, pp.22–34, 2020.