UNIVERSITATEA DIN PITEȘTI
FACULTATEA DE MATEMATICĂ – INFORMATICĂ
ȘCOALA DOCTORALĂ – INFORMATICĂ

Teză de doctorat

Tehnici de clustering în mineritul datelor utilizate în auditul fraudei economico-financiare

Rezumat

Conducător științific:
Prof. univ. dr. Bălănescu Tudor

Doctorand:
Sabău Andrei Sorin

PITEȘTI, 2016

Cuprins

Lista tabelelor

Lista figurilor

Lista algoritmilor

1. Introducere

Realități economice precum gradul de globalizare în continuă creștere, presiunea concurenței, schimbări regulatorii frecvente forțează companiile să caute în mod activ modalități prin care să își îmbunătățească activitatea la toate nivelele, inclusiv în sfera controlului intern, atât în ceea ce privește detectarea cât și în ceea ce privește prevenirea fraudei economico-financiare. Frauda și corupția identificate la giganți precum WorldCom și Enron subliniază nu doar importanța auditului dar și efectele devastatoare pe care le poate avea frauda asupra companiilor și auditorilor implicați.

Potrivit raportului „Global Economic Crime Survey" redactat de PricewaterhouseCoopers în anul 2016, mai mult de o treime din organizații s-au confruntat cu cel puțin o forma de criminalitate economică în ultimele 24 luni. Acesta confirmă trendul ascendent înregistrat de activitățile fraudulente în ultima decadă.

Progresul înregistrat în tehnologia informației în această perioadă a condus la o creștere explozivă a activităților legate de generarea, stocarea și analiza datelor în cadrul companiilor. Generarea informației nu mai este limitată doar la indivizi introducând informație în sisteme; majoritatea informației este generată de către sisteme automate introducând informație în sisteme automate. Această creștere explozivă a informațiilor disponibile în cadrul unei companii marchează o tranziție a procesului de audit de la un proces manual la un proces semi-automat sau, în unele cazuri, complet automat. Metodele de data mining sunt folosite din ce în ce mai des, marcând un moment de cotitură în anul 2011 când, pentru prima dată, prin monitorizarea tranzacțiilor suspecte, au devenit soluția cea mai eficientă de detectare a fraudei.

Din metodele de data mining disponibile, clustering s-a dovedit a fi o soluție viabilă, constant aplicată în industria de specialitate în vederea detectării fraudei. Majoritatea algoritmilor de clustering folosiți sunt algoritmi clasici, partiționali (peste

70%) aplicați la nivelul unor seturi de date statice în cadrul unor misiuni de audit periodice. Acești algoritmi nu acoperă, și nu pot acoperi, procesul de audit continuu în care totalitatea tranzacțiilor este monitorizată în timp real.

Analiza continuă a unor tranzacții generate în mod constant la intervale scurte sau foarte scurte de timp, în vederea detectării acelor tranzacții suspecte fără o clasificare manuală prealabilă, poate fi făcută doar prin intermediul metodelor de data mining special concepute pentru parcurgerea, cu restricții stricte de memorie și timp de execuție, fluxurilor de date.

Pe parcursul cercetării sale științifice, autorul acestei teze a propus și publicat o serie de algoritmi de clustering, atât tradiționali cât și axați pe analiza fluxului de date, algoritmi ce pot fi utilizați fie pentru analiza generală a unui set/flux de date, fie pentru identificarea tranzacțiilor suspecte în vederea detectării fraudei economico-financiare.

Capitolul 1 justifică utilitatea algoritmilor de clustering asupra fluxurilor de date în detectarea anomaliilor în general, și a tranzacțiilor suspecte în vederea detectării fraudei economico-financiare în particular.

Capitolul 2 detaliază algoritmi de clustering tradiționali și algoritmi de clustering asupra fluxurilor de date. Ambele categorii cuprind algoritmi partiționali, bazați pe densitate și tip grid.

Capitolul 3 cuprinde analiza lucrărilor științifice în domeniul detectării fraudei financiare cu ajutorul algoritmilor de clustering. Sunt interogate o serie de baze de date conținând lucrări științifice folosind expresii regulate formate din termenul „cluster*” cumulat cu fiecare formă majoră de fraudă financiara. Dintre lucrările analizate, majoritatea folosesc algoritmi de clustering partiționali (72%) utilizați de sine stătător (41%) sau în tandem cu alte tehnici de data mining (41%).

Capitolul 4 introduce estimarea masei ca alternativă la estimarea densității. Este prezentat MaStream, un algoritm de clustering asupra fluxurilor de date bazat pe

estimarea masei. Cu ajutorul unui ansamblu de arbori h:d, acesta poate identifica clustere de forme arbitrare cu perturbații minore cauzate de zgomot sau valori extreme. MaStream este comparat cu alți trei algoritmi de clustering asupra fluxurilor de date: CluStream, DenStream și ClusTree. Experimentele pe o serie de seturi/fluxuri de date sintetice cât și reale dovedesc acuratețea si scalabilitatea algoritmului.

Capitolul 5 introduce concepte legate de structuri de date probabilistice și avantajele acestora în analiza fluxurilor de date. Este prezentat BloomStream, un algoritm ce analizează fluxuri de date folosind două astfel tipuri de structuri: filtrele bloom și structura count-min. Scalabilitatea și acuratețea algoritmului este dovedită prin comparația cu alți trei algoritmi de clustering asupra fluxurilor de date, CluStream, DenStream și D-Stream II, atât pe seturi/fluxuri de date reale cât și sintetice.

Capitolul 6 evidențiază contribuțiile aduse domeniului abordat prin introducerea si publicarea unei serii de algoritmi de clustering, atât tradiționali cât și axați pe analiza fluxului de date, ce pot fi utilizați fie pentru analiza generala a unui set/flux de date, fie pentru identificarea tranzacțiilor suspecte în vederea detectării fraudei economico-financiare. Lucrări ilustrând stadiul actual al cunoașterii și analize comparative privind eficienta algoritmilor de clustering în detectarea anomaliilor completează contribuțiile aduse domeniului de cercetare abordat.

2. Stadiul cunoașterii

Tehnica de data mining nesupervizată, clustering reprezintă procesul prin care instanțe de date sunt grupate în grupuri sau clustere pe baza similarității, disimilarității acestora. Deși nu există o definiție universal acceptată, majoritatea definițiilor clasice cuprind următoarele caracteristici:

- Instanțele de date aparținând aceluiași cluster sunt cât mai similare posibil.
- Instanțele de date aparținând unor clustere diferite sunt cât mai diferite posibil.
- Măsurile de similaritate, disimilaritate sunt clar definite.

Fie un set de date X conținând n instanțe d-dimensionale $x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \in A$, unde A reprezintă spațiul atributelor si $1 \leq i \leq n$. Fiecare atribut poate fii fie numeric, fie categorial. Acest format de date corespunde unei matrici $n \times d$, formatul cel mai des folosit în cadrul algoritmilor de clustering. În urma execuției unui algoritm de clustering rezultă o soluție de clustering $C = \{C_1, C_2, \dots, C_k\}$, cu fiecare cluster $C_j \subseteq X$, $1 \leq j \leq k$. Pe baza tipului de algoritm de clustering folosit, soluția de clustering rezultată prezintă următoarele caracteristici:

- Algoritmi hard clustering. Clusterele generate sunt disjuncte, nu au instanțe comune si reuniunea acestora formează setul inițial de date. Nu există clustere vide sau conținând toate instanțele de date.

$$\bigcup_{j=1}^{k} C_j = X$$

$$\bigcap_{j=1}^{k} C_j = \emptyset$$

$$\emptyset \subset C_j \subset X, 1 \leq j \leq k$$

- Algoritmi hard clustering cu anomalii. Clusterele generate sunt disjuncte, nu au

instanțe comune dar reuniunea acestora nu formează setul inițial de date. Anumite instanțe de date, catalogate drept valori extreme si/sau izolate, nu sunt alocate nici unui cluster, reprezentând un subset izolat $V \subset X$:

$$\bigcup_{j=1}^{k} C_j \cup V = X$$

$$\bigcap_{j=1}^{k} C_j = \emptyset$$

$$\emptyset \subset C_j \subset X, 1 \leq j \leq k$$

- Algoritmi fuzzy clustering. Fiecare cluster generat conține întreg setul X. Soluția generată conține în plus o matrice $W = w_{i,j} \in [0,1], 1 \leq i \leq n, 1 \leq j \leq k$ a gradului de apartenență a fiecărei instanțe de date x_i în cadrul fiecărui cluster C_j.

$$\bigcup_{j=1}^{k} C_j = X$$

$$\bigcap_{j=1}^{k} C_j = X$$

$$\sum_{j=1}^{k} w_{i,j} = 1, 1 \leq i \leq n$$

Diferitele abordări ale procesului de clustering pot fi descrise cu ajutorul Figurii nr. 1. La nivelul superior există o separare între abordările de tip partiţional şi cele de tip ierarhic. Metodele ierarhice de clustering produc o serie imbricată de partiţionări, pe când cele partiționale obțin o singură partiție.

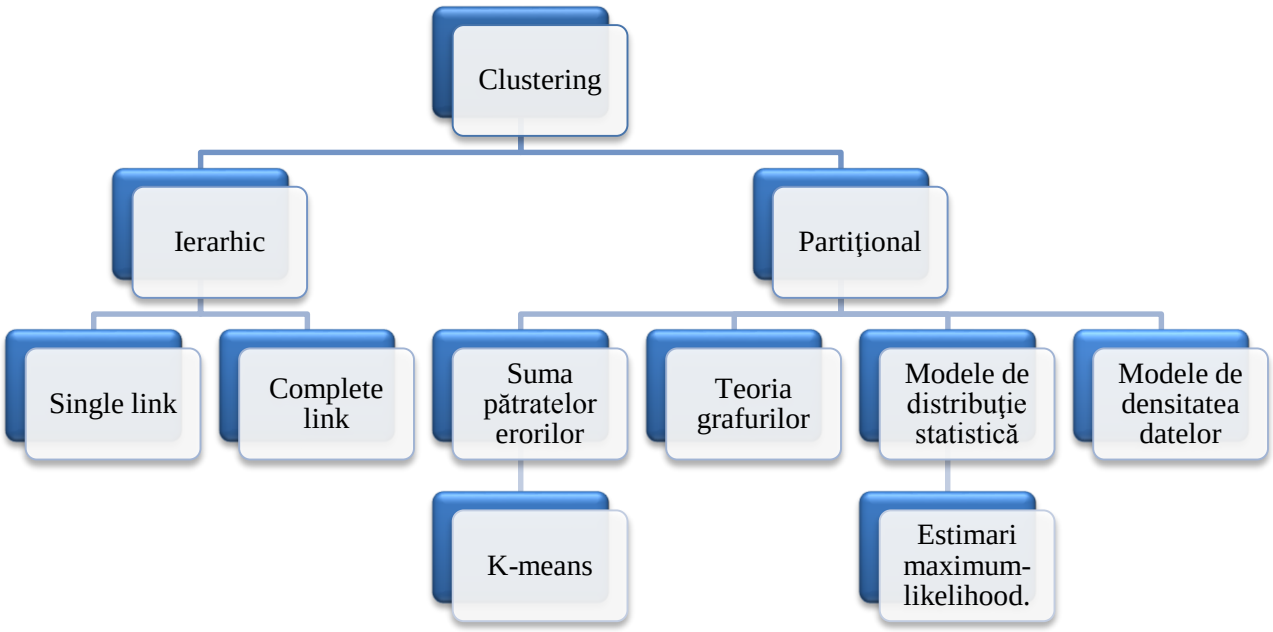


Figura 1. Taxonomia tehnicilor de clustering

Taxonomia ilustrată în Figura nr. 1, este suplimentată de o serie de aspecte/proiecţii ce afectează toate tehnicile de clustering indiferent de plasamentul acestora în taxonomie.

Tehnici aglomerative versus tehnici divisive - O tehnică aglomerativă începe cu fiecare entitate drept cluster distinct (singleton). Prin operaţii succesive, clustere sunt combinate împreună până când un criteriu de finalizare este îndeplinit. O tehnică divizivă este exact inversul unei tehnici aglomerative, începând cu toate entităţile într-un singur cluster. Prin operaţii succesive, clusterele sunt divizate până când un criteriu de finalizare este îndeplinit.

Tehnici monoteice versus tehnici politeice - O tehnică monoteică foloseşte un singur atribut la un moment dat, o tehnică politeică foloseşte toate atributele. Majoritatea algoritmilor sunt politeici: toate atributele intră în calcularea distanţelor între entităţi, deciziile fiind luate pe baza acestor distanţe.

Tehnici hard/crisp versus tehnici fuzzy - O tehnică hard/crisp alocă fiecare entitate unui singur cluster, atât pe durata rulării, cât şi ca rezultat final. O tehnica fuzzy alocă o entitate mai multor sau tuturor clusterelor pe baza unei distribuţii de probabilităţi. Tehnicile fuzzy pot fi transformate în tehnici crisp prin alocarea fiecărei entităţi acelui cluster cu probabilitatea cea mai mare.

Tehnici deterministe versus tehnici stohastice - Acest aspect este relevant în principal în cadrul tehnicilor partiţionale având drept obiectiv minimizarea sau maximizarea unei anumite funcţii. Aceasta optimizare poate fi implementată folosind tehnici tradiţionale sau cu ajutorul unei căutări aleatoare în spaţiul de soluţii.

Tehnici incrementale versus tehnici non incrementale - Acest aspect devine important în momentul în care setul de date vizat este suficient de mare pentru a impune constrângeri privitoare la timpul de execuţie sau memoria / resursele hardware folosite. Istoria timpurie a tehnicilor de clustering nu conţine multe exemple de algoritmi de clustering special gândiţi pentru seturi mari de date. Avansul general şi interesul crescut în data mining a condus într-o mare măsură la dezvoltarea unor algoritmi de dată mai recentă ce minimizează numărul de parcurgeri al setului de date, reduce numărul de entităţi examinate şi/sau reduce dimensiunea structurilor de date folosite în timpul execuţiei.

3. Clustering în detectarea fraudei

Ca parte a metodologiei de cercetare, au fost definite o serie de criterii de căutare și selectare a articolelor. În prima fază, bazele de date Thomson Reuters Web of Science, IEEE Transactions, ScienceDirect Freedom Collection și Springer-Link Contemporary au fost interogate folosind expresia regulată "cluster* fraud*" în sumarul articolului. De asemenea, bibliografia articolelor relevante, pană la două nivele de adâncime, a fost considerată pentru incluziune. în cea de a doua fază, termenul „clustering” cumulat cu fiecare forma majoră de fraudă financiara a fost interogat în cadrul Google Scholar cu primele 100 de rezultate fiind considerate pentru o posibila incluziune, din nou considerând lucrările prezente în bibliografie pana la două nivele de adâncime. în afara de relevanța la domeniul analizat, fiecare articol îndeplinește o serie de criterii adiționale. Întreg conținutul articolului este disponibil, acesta conține un studiu de caz și acest studiu de caz se folosește de un set de date real. A existat și un număr mic de excepții pentru studiile de caz folosind seturi de date sintetice, excepții justificate de valoarea științifică a lucrărilor. în marea majoritate a cazurilor au fost preferate seturile de date reale, pentru că în acest fel, cel puțin într-o anumita măsura, rezultatele sunt cuantificabile.

Lucrările analizate folosesc un spectru larg de tehnici clustering. Intr-o extremitate a spectrului întâlnim tehnici de clustering independente reprezentând întregul proces de data mining. În cealaltă extremitate, întâlnim tehnici hibride unde clustering reprezintă doar una din uneltele de data mining folosite, în una sau mai multe etape. De asemenea, sunt prezente și tehnici de clustering de vizualizare aplicate în detectarea fraudei financiare.

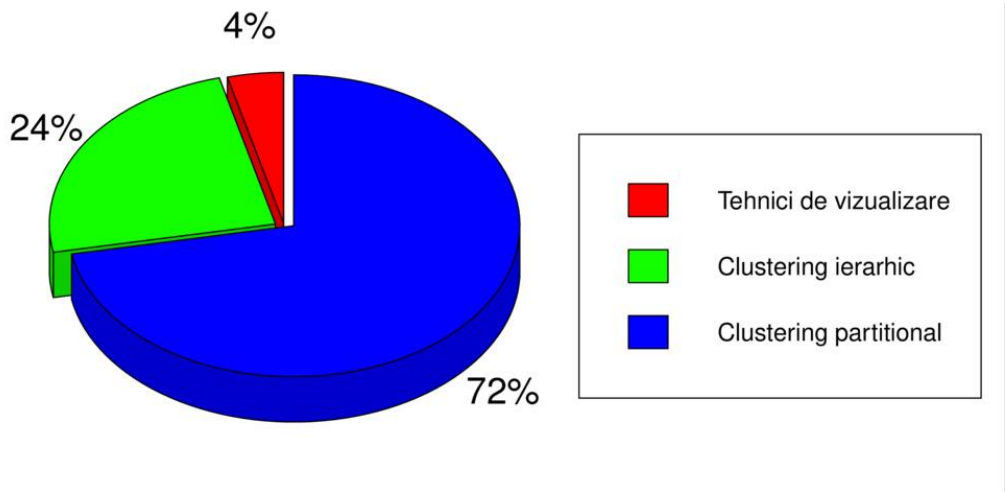


Figure 15. Tipuri de clustering folosite in detectarea fraudei

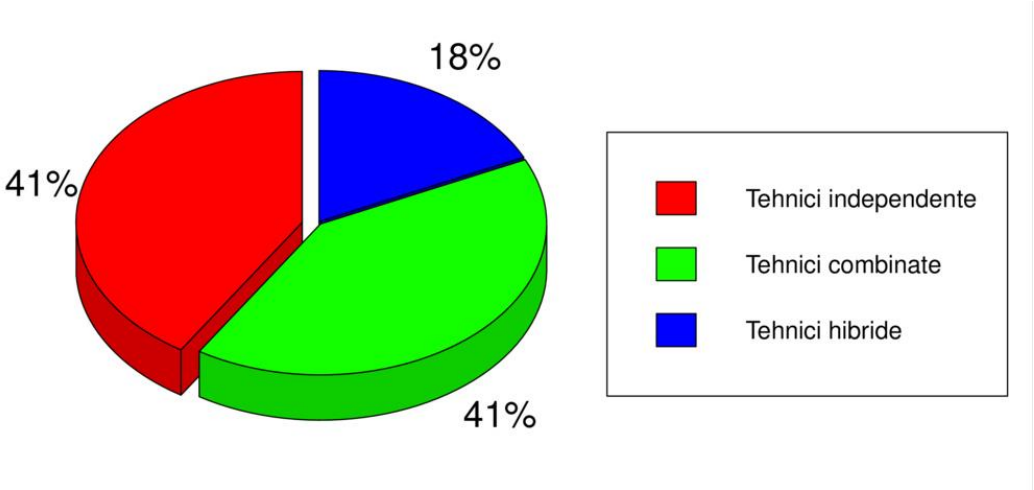


Figure 16. Combinatii clustering folosite in detectarea fraudei

În comparație cu alte domenii în care tehnicile de clustering sunt folosite pentru identificarea anomaliilor, atacurilor, etc.. tehnicile de clustering folosite în

identificarea fraudei tind sa utilizeze algoritmii tradiționali, verificați în timp. Tehnici relativ noi precum ansamble de clustering, clustering asupra fluxurilor de date, au o prezenta foarte scăzută în cadrul lucrărilor analizate.

Dintre lucrările analizate, aproape trei sferturi folosesc algoritmi de clustering partiționali – Figura 15. Unele lucrări au fost înregistrate ca folosind atât algoritmi partiționali cât și ierarhici având în vedere folosirea combinata a acestora. Dintre algoritmii partiționali, k-means și variantele acestuia folosind distanța euclidiană ca măsura de similaritate sunt cei mai des folosiți. Algoritmii ierarhici se situează pe locul doi fiind utilizați de un sfert din lucrările analizate. Tehnici de clustering interactive, de vizualizare sunt de asemenea folosite dar într-o foarte mică măsura.

În ceea ce priveste modul în care diferitele tehnici de clustering sunt combinate sau folosite impreună ca alte tehnici de data mining, lucrările analizate au fost clasificate continand tehnici independente cu un singur algoritm de clustering folosit fără alte tehnici de data mining, tehnici combinate continand doi sau mai multi algoritmi de clustering fără alte tehnici de data mining, tehnici hybrid continand atat algoritmi de clustering cât și alte tehnici de data mining – de cele mai multe ori clasificatori bazati pe arbori decizionali, retele neuronale și masini cu suport vectorial – Figura 16. Tehnicile independente și hybrid sunt cel mai des folosite, fiecare tehnică fiind prezentă în aproximativ 40% din totalul lucrărilor analizate. Imbunătătirea acuratetii solutiei de clustering prin combinarea mai multor algoritmi diferiti de clustering nu este percepută ca aducand beneficii signifiante, numai 18% din lucrările analizate folosind această abordare.

4. MaStream

Lucrarea curentă introduce un algoritm de clustering bazat pe estimarea masei. Aceasta alternativa la estimarea densității nu folosește măsuri de similaritate, oferind o adaptabilitate ridicată algoritmului în fluxuri cu dimensionalitate atât scăzută cât și ridicată. Prin intermediul unui ansamblu de arbori h:d, pot fi identificate clustere de forme arbitrare cu perturbații minore cauzate de zgomot sau valori extreme.

Parte a tehnicilor de data mining, tehnicile bazate pe estimarea masei au fost introduse de Ming et al., cuprinzând atât aspecte teoretice cât și o metodă de estimare pentru seturi de date unidimensionale. Comparativ cu estimarea densității, estimarea masei prezintă următoarele particularități:

- Estimarea masei rezultă intr-o ordonare a instanțelor de date de la instanțele centrale (core points) la cele periferice, cu instanțele periferice având o masa inferioara instanțelor centrale. Această ordonare a instanțelor de date nu este prezentă în estimarea densității.
- Estimarea masei este calculată cu ajutorul unor contoare simple, necesită doar o fracțiune a întregului set de date. Estimarea densității necesită un efort computațional mai mare.
- Masa poate fi interpretata ca o măsură de relevantă a unei instanțe de data, instanțele centrale având o importantă ridicată iar cele periferice o importantă scăzuta.

Ming, Wells generalizează estimarea unidimensională a masei la estimări multidimensionale. Lucrarea propune în premiera un algoritm de clustering bazat pe estimarea masei cu ajutorul unor arbori h:d fără a folosi masuri de similaritate bazate pe distante sau densități. O variantă a arborelui h:d este folosită și în cadrul algoritmului propus.

Fie $x_1 < x_2 < \cdots < x_{n-1} < x_n$ o serie de instante unidimensionale ordonate, $x_i \in R,\ n > 1$ şi s_i o segmentare binara intre x_i şi x_{i+1}. O estimare a masei de nivel h se calculează recursiv cu ajutorul a $h-1$ estimări de masă de nivel inferior.

$$mass(x,h) = \begin{cases} \sum_{i=1}^{n-1} mass_i(x, h-1)p(s_i), & h > 1 \\ \sum_{i=1}^{n-1} m_i(x)p(s_i), & h = 1 \end{cases}$$

Probabilitatea de selectare s_i ca segmentare binară aleatoare urmând o distribuție uniformă devine:

$$p(s_i) = \frac{x_{i+1} - x_i}{x_n - x_1} > 0$$

Estimarea recursivă a masei se oprește la nivelul $h = 1$ unde functia bazată pe masa $m_i(x)$ utilizează rezultatul segmentarii binare s_i unde L şi R reprezinta regiunea din stanga, respectiv din dreapta segmentării cu $m_i^L = n - m_i^R = i$. La acest nivel masa fiecărei regiuni este reprezentată de numărul total de instanțe de date din cadrul acesteia.

$$m_i(x) = \begin{cases} m_i^L, x \le s_i \\ m_i^R, x > s_i \end{cases}$$

Fiecare segmentare binară s_i la nivelul $h > 1$ produce două estimări de masă la nivel $(h-1)$ pentru regiunile din stânga și dreapta segmentării conținând $\{x_1, \dots, x_i\}$, respectiv $\{x_{i+1}, \dots, x_n\}$.

$$mass(x,h) = \begin{cases} mass_i^L(x, h-1), & x \le s_i \\ mass_i^R(x, h-1), & x > s_i \end{cases}$$

O estimare aproximativă a masei $\overline{mass}(x,h)$ poate fi calculată cu ajutorul subseturilor $D_i \subset D$ $(i = 1, \dots, t)$ extrase fără înlocuire din setul original de date *D*. Acuratețea aproximării este direct proporționala cu mărimea subseturilor folosite.

$$\overline{mass}(x,h) = \frac{1}{t}\sum_{i=1}^{t} mass(x,h|D_i)$$

Un factor aleator este introdus în ecuațiile din subsecțiunea precedenta înlocuind probabilitatea unei segmentări binare $p(s_i)$ cu o functie $T(.)$ capabilă a partitiona aleator R^d intr-o serie de subspatii, generalizand segmentarea binară în $2h$ segmentări unde h reprezintă nivelul de segmentare folosit.

Fie x o instanta de date în R^d, $T(x)$ subspatiile în care x este prezentă și m numărul total de instante din aceasta regiune. O funcție generalizată a estimării masei este definită ca:

$$m\big(T(x)\big) = \begin{cases} m, & dacă\ x\ apartine\ unui\ subspatiu\ definit\ de\ T \\ 0, & altfel \end{cases}$$

Estimarea aproximativă a unui set de date multidimensional, devine:

$$\overline{gmass}(x) = \frac{1}{t}\sum_{i=1}^{t}(T_i^h(x|D_i))$$

Funcția $T^h(x|D)$ responsabilă de partitionarea în subspații aleatoare poate fi implementata cu ajutorul unei structuri tip arbore partitionand R^d în 2^l subspații.

Urmând arhitectura CluStream, algoritmul propus MaStream definește parametrii de intrare doar pentru etapa online: t număr total de arbori h:d, adancimea maximă l a unui arbore, și ψ instante de date pentru construirea unui arbore. Etapa offline generează o solutie de clustering fără ajutorul unor parametrii definiti de catre utilizator și nici nu necesită definirea unei ferestre la nivelul fluxului de date. Etapa online actualizează în mod continuu modelul aferent pe măsura ce noi instanțe de date

sunt disponibile. Acest lucru permite declanșarea etapei offline, și implicit generarea unei soluții clustering, în orice moment pe durata de viața a fluxului de date.

Pe parcursul tuturor experimentelor derulate, clusterele reale, corecte sunt cunoscute apriori. Din aceasta cauză, am putut alege puritatea ca măsură de evaluare externă în vederea comparării acurateței soluțiilor clustering generate. Pentru a calcula puritatea, fiecărui cluster ii este desemnată clasa cu cele mai multe instanțe din cadrul acestuia. Puritatea claselor generate de soluția clustering devine:

$$Puritate = \frac{\sum_{i=1}^{k} \frac{|C_i^d|}{|C_i|}}{k} \times 100$$

unde k reprezintă numărul total de clustere, $|C_i^d|$ reprezintă numărul de instanțe aparținând clasei dominante din clusterul i și $|C_i|$ reprezintă numărul total de instante aparținând clusterului i. Puritatea este calculată numai pe un număr limitat de instanțe aparținând unei singure ferestre din cadrul fluxului de date având în vedere că algoritmii cu care MaStream este comparat necesită o astfel de fereastră.

MaStream este implementat în Python cu extensii C++. CluStream, DenStream și ClusTree ruleaza în R apelând librarii Java. Sunt utilizate trei seturi de date din pachetul R mlbench: Smiley, Cassini și Spirals. Datele sintetice au fost generate o singură dată, salvate și ulterior servite fiecărui algoritm comparat. Fiecare flux de date conține 100 unități de timp, viteza acestuia fiind de 1000 instanțe per interval de timp. Etapa de procesare online pentru algoritmii CluStream, DenStream și ClusTree este realizată cu valorile implicite ale parametrilor de inițializare, valori sugerate de respectivii autori. O serie de ajustări au fost considerate asupra fiecărui parametru, dar puritatea soluțiilor clustering rezultate s-a dovedit a fi îmbunătățită sub 5%. Pentru același set de trei algoritmi, etapa de procesare offline folosește un clustering ierarhic single-link cu un număr total de clustere egal cu numărul real de clustere din cadrul fiecărui set de date.

În toate seturile de date sintetice, MaStream înregistrează valorile de puritate cele mai mari dar doar analiza purității nu reflectă neapărat și calitatea soluției de clustering având în vedere că pe măsura ce numărul de clustere creste, obținerea unei purități ridicate devine mai facila. Acest lucru este evitat în experimentele derulate. CluStream, DenStream și ClusTree sunt restricționați de componenta offline la generarea numărului corect de clustere, pe când MaStream nu are această restricție. în cadrul seturilor de date Smiley și Cassini, MaStream identifică numărul corect de clustere pe parcursul tuturor unităților de timp. în cadrul setului de date Spirals, MaStream identifică intre 3 și 5 clustere pe parcursul primei unități de timp și numărul corect de clustere în următoarele. Acest comportament este atribuit modului în care MaStream isi actualizează constant modelul de clustering. Pentru acest set de date, numai începând cu cea de a doua unitate de timp, au existat suficiente instanțe de date pentru a permite unificarea unor clustere prealabil separate. Din timpul total de execuție al MaStream, aproximativ 25% este consumat de generarea ansamblului inițial de arbori h:d, operațiunea cu cel mai mare consum de resurse.

Setul de date Forest CoverType conține habitate forestiere pe suprafețe de $30m \times 30m$. Sunt inregistrare observații despre 581,012 astfel de suprafețe, fiecare observație conținând 54 de atribute: 10 atribute cantitative, 4 atribute binare reprezentând tipul de zona sălbatica, 40 atribute binare reprezentând tipul de sol. Observațiile sunt clasificate pe tipul de habitat forestier existând 7 clase distincte. Pentru fiecare algoritm viteza fluxului de date a fost setată la 1000 instanțe per unitatea de timp. Numai cele 10 variabile cantitative au fost folosite. Majoritatea unităților de timp nu conțin toate cele 7 tipuri de habitate forestiere, de obicei conținând doar 2 sau 3 dintre acestea.

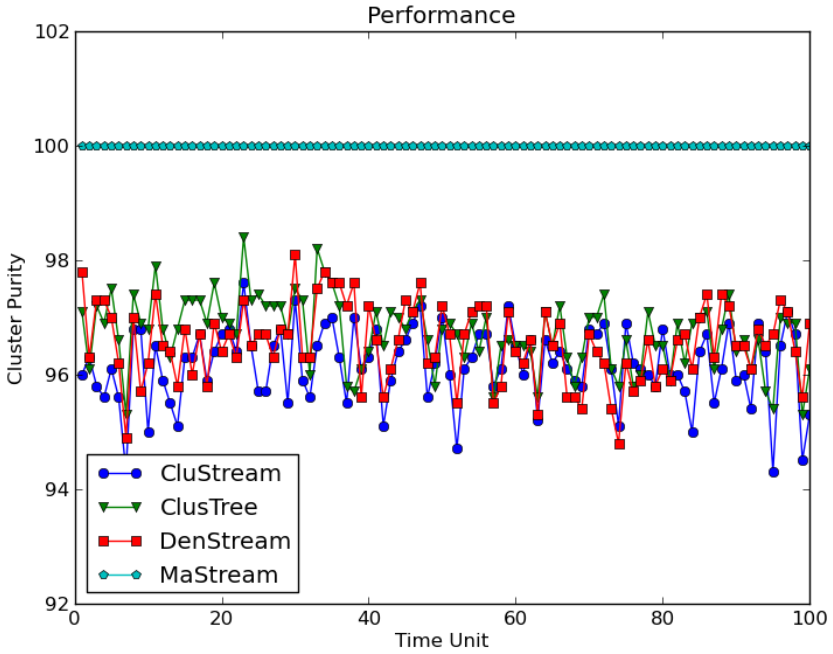


Figure 20. MaStream, puritate in cadrul setului de date Smiley.

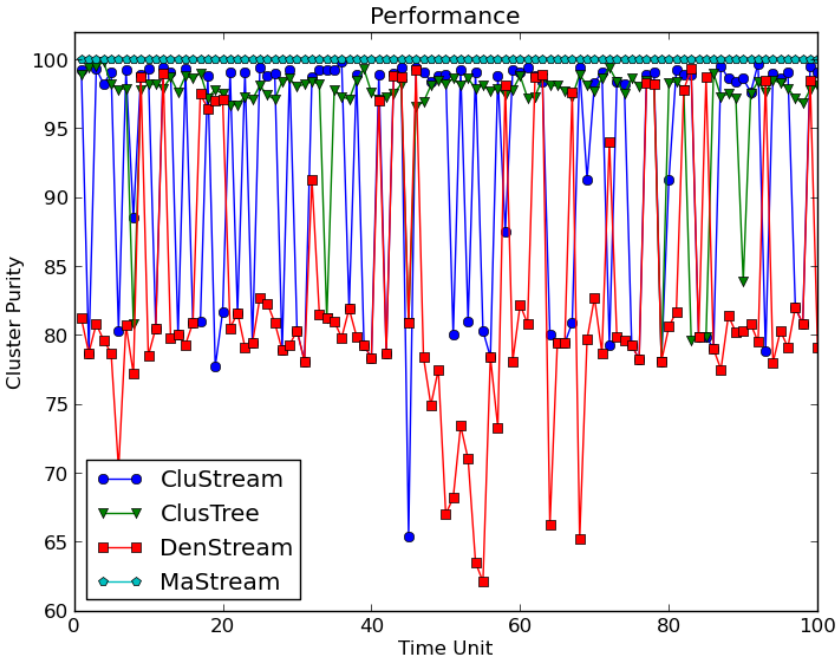


Figure 21. MaStream, puritate in cadrul setului de date Cassini.

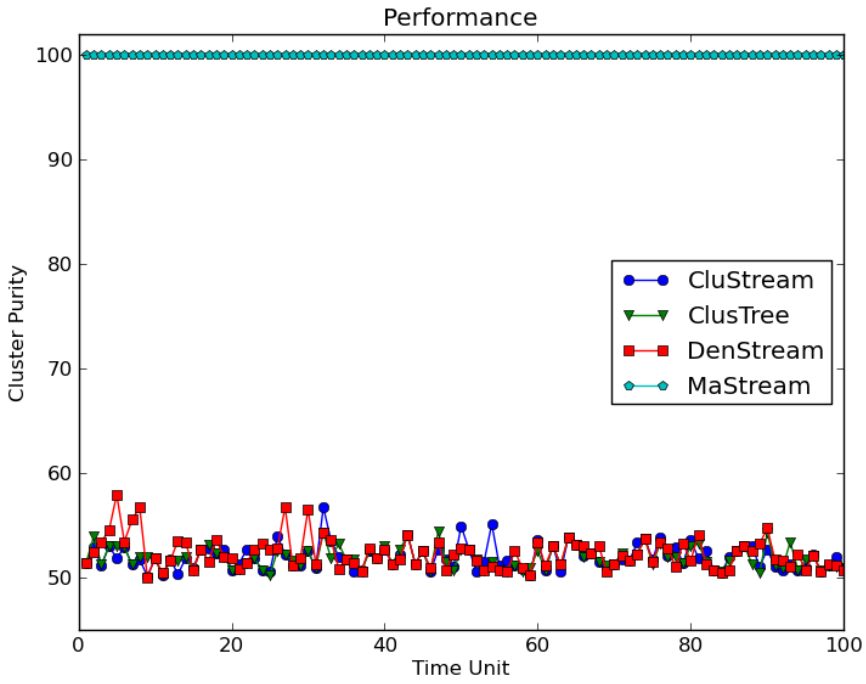


Figure 22. MaStream, puritate in cadrul setului de date Spirals.

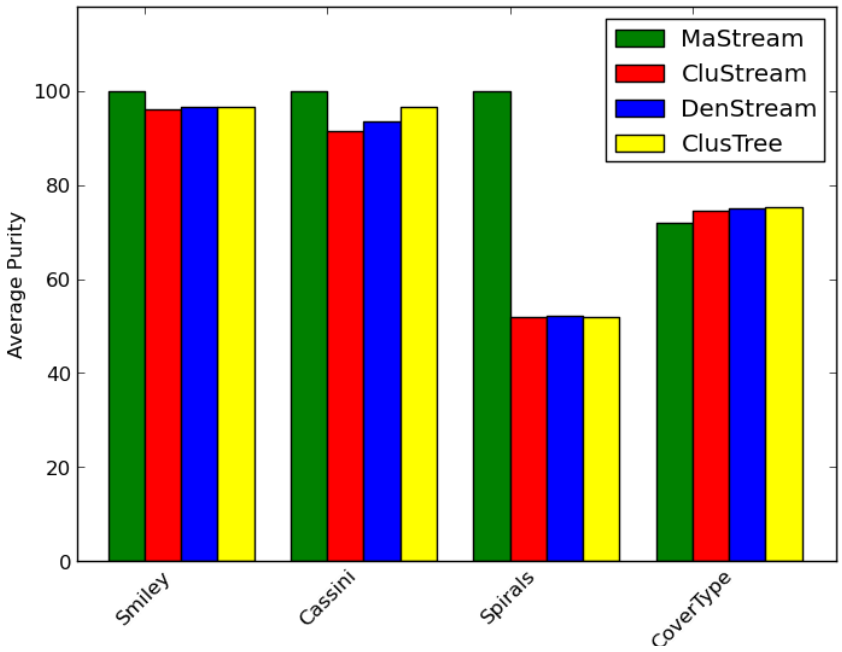


Figure 23. MaStream, puritate medie in cadrul seturile de date analizate.

5. BloomStream

BloomStream, algoritmul propus bazat pe densitate, se folosește de structuri de date probabilistice pentru a determina atât densitatea fluxului de date cât și clusterele din cadrul acestuia. O structura Count-Min folosind un model de tip damped window estimează densitatea fluxului de date. Filtre Bloom utilizând o variație a buffering-ului activ-activ estimează apartenența la clustere. Ambele structuri folosesc același set de funcții hash generate din combinații liniare a doar două funcții hash independente și uniform aleatoare fără a înregistra repercusiuni negative asupra probabilității de eroare asimptotice.

Filtrul bloom standard suporta interogări privind apartenenta aproximativa la un set. Cu toate ca eficiența spațiala ridicată este obținuta cu costul detectării de fals pozitive, în cele mai multe cazuri un compromis avantajos poate fi obținut ținând probabilitatea de eroare la un nivel suficient de scăzut.

Fie U universul elementelor posibile intr-un set și $S \subset U$ un set cu $n = |S|$ elemente. S este reprezentat de un vector de biți B, conținând $m = O(n)$ biti, inițial setați pe 0. Maparea elementelor din U în B se face prin intermediul a k funcții hash independente, uniform aleatoare $h_1, \dots, h_k : U \to \{1, \dots, m\}$.

Pentru fiecare element $x \in S$, k biți din B sunt setați la valoarea 1 pe pozițiile $h_1(x), \dots, h_k(x)$. Dat fiind un element $q \in U$, daca $\forall 1 \leq i \leq k,\ B\big(h_i(x)\big) = 1$, elementul este probabil conținut în set, altfel cu siguranța nu este conținut. Condiția de mai sus poate fi îndeplinita chiar daca $q \notin S$ în cazul în care elemente din S sunt mapate pe aceleași poziții bit cu q. Aceasta rata de fals pozitive sau eroare bloom depinde de selecția parametrilor m, k precum și de cardinalitatea setului S. Probabilitatea ca un bit sa conțină în continuare valoarea 0 după introducerea întregului set S în filtrul bloom este:

$$\left(1-\frac{1}{m}\right)^{kn} \approx e^{-\frac{kn}{m}}$$

Probabilitatea de fals pozitive *fp* având toți cei *k* biți setați pe 1, determinând în mod eronat $q \in S$, devine:

$$fp = \left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1-e^{-\frac{kn}{m}}\right)^{k}$$

Generarea a *k* funcții hash independente și calcularea valorilor hash aferente fiecărui element ce urmează a fi inserat este prohibitiv de scump în cadrul analizei unui flux de date. Din fericire, cerința de independenta a funcțiilor hash poate fi relaxata. Combinații liniare de doar doua funcții hash independente, uniform aleatoare pot fi folosite la generarea celorlalte funcții, păstrând aceeași rata de fals pozitive:

$$g_i(x) = h_1(x) + ih_2(x) \; mod \; p$$

Afirmația de mai sus se verifica atunci când tabelele de dispersie a fiecărei funcții hash sunt separate. Dat fiind un vector de $m = kp$ biți, fiecărei funcții hash $h_1, \dots, h_k: U \to \{1, \dots, p\}$ îi sunt alocate m/k poziții bit consecutive ca intervale disjuncte. Asimptotic, probabilitatea ca un bit sa conțină în continuare valoarea 0 după introducerea întregului set *S* în filtrul bloom rămâne aceeași:

$$\left(1-\frac{k}{m}\right)^{n} \approx e^{-\frac{kn}{m}}$$

Structura count-min suportă interogări privind multiplicitatea aproximativa în cadrul unui multiset. Aceasta extinde vectorul de biți din cadrul filtrului bloom într-o matrice bidimensionala în vederea calculului frecventelor de apariție.

Fie U universul elementelor posibile într-un multiset cu cardinalitate $n = |U|$ şi $M \subset U$ un multiset prezentat iterativ sub forma unei serii de actualizări (i_t, c_t) cu $i_t \in M$ şi c_t frecvența aferentă. Deși structura count-min suportă frecvențe de orice valoare, în cazul de față restricționam c_t la valoarea 1 având în vedere ca prezentăm

M element cu element. *M* poate fi vizualizat ca un vector de frecvențe $a(T) = [a_1(T), a_2(T), \dots, a_n(T)]$ unde, pentru $t \leq T, i_t = j$, $a_j(T) = \sum c_t$ denotă multiplicitatea lui *j* în *M* la momentul *T*. în ceea ce urmează, renunțam la *T* și ne referim doar la starea actuala a *M*.

Pe baza unei marje de eroare ε și probabilități de eroare δ, *M* este reprezentat de o matrice bidimensionala *C* de $d = \lceil \ln 1/\delta \rceil$ linii și $w = \lceil e/\varepsilon \rceil$ coloane cu toate intrările setate inițial la valoarea 0. Maparea elementelor din *U* în *C* este realizată cu ajutorul a *d* funcții hash independente două câte două $h_1, \dots, h_d : U \to \{1, \dots, w\}$.

Pentru fiecare element $x \in M$ și $\forall 1 \leq j \leq d$, *d* contoare aparținând *C* sunt incrementate cu valoarea $c_t = 1$:

$$C\left[j, h_j(a_x)\right] = C\left[j, h_j(a_x)\right] + 1$$

Dintr-o perspectivă multiset, suntem interesați doar de capabilitățile structurii count-min de a estima frecventele de apariție. Pentru un element dat $q \in U$, multiplicitatea acestuia în *M* este estimată de:

$$\hat{a}_q = \min_{1 \leq j \leq d} C\left[j, h_j(a_q)\right]$$

Având în vedere ca frecventele de apariție sunt strict pozitive ($c_t = 1$), marja de eroare cauzată de coliziunile la nivelul tabelei de dispersie deplasează valorile intr-un singur sens, si, cu o probabilitate de cel putin $1 - \delta$:

$$\hat{a}_q \leq a_q + \varepsilon \|a\|_1$$

Folosind același procedeu detaliat în secțiunea 5.3.1, condiția utilizării a *d* funcții hash independente două cate două poate fi relaxată. Combinații liniare de

$$2\lceil (ln\, 1/\delta) \,/\, (ln\, 1/\varepsilon) \rceil$$

funcții hash independente două cate două sunt folosite pentru a le genera pe restul, menținând aceeași marja și probabilitate de eroare.

La un nivel de bază, BloomStream folosește o tehnica de clustering tip grid. O astfel de tehnică discretizează spațiul în intervale disjuncte de-a lungul fiecărei dimensiuni transformând fluxul nelimitat de instanțe de date intr-un număr finit de celule grid. Numărul de instanțe din cadrul unei celule determină densitatea acesteia, celulele dense adiacente fiind agregate ulterior în clustere.

Structuri precum tabele hash sau arbori permit o reprezentare explicită și acces direct la celulele dense cu costul managementului unui număr de celule exponențial cu dimensionalitatea fluxului de date. Aproximând suplimentar fluxul de date cu ajutorul unor structuri probabilistice poate fi obținut un alt tip de compromis. Cerințele de spațiu nu mai depind de dimensionalitatea fluxului de date ci de parametrii optimi ai filtrului bloom detaliați în subsecțiune 5.3.3. O astfel de complexitate spațiala scăzuta este obținuta cu costul imposibilității de a accesa direct celulele dense din cadrul structurii count-min, cel puțin nu fără a fi forțați a interoga individual fiecare posibil set de coordonate al celulelor grid.

Această abordare marchează o direcție noua în tehnicile de clustering la nivelul fluxului de date, părăsind abordarea în două etape, având în vedere lipsa capabilității de a accesa direct informația privind densitatea fluxului în etapa offline. Fiecare instanța a fluxului de date actualizează structura count-min si, în cazul în care celula grid corespunzătoare este identificată a fi densă, actualizează soluția clustering curentă. Atât celulele din vecinătatea celulei dense curente cât și clusterele existente sunt stocate cu ajutorul filtrelor bloom. O serie de intersecții și reuniuni asupra acestor filtre, creează noi clustere, unesc sau elimina clustere existente.

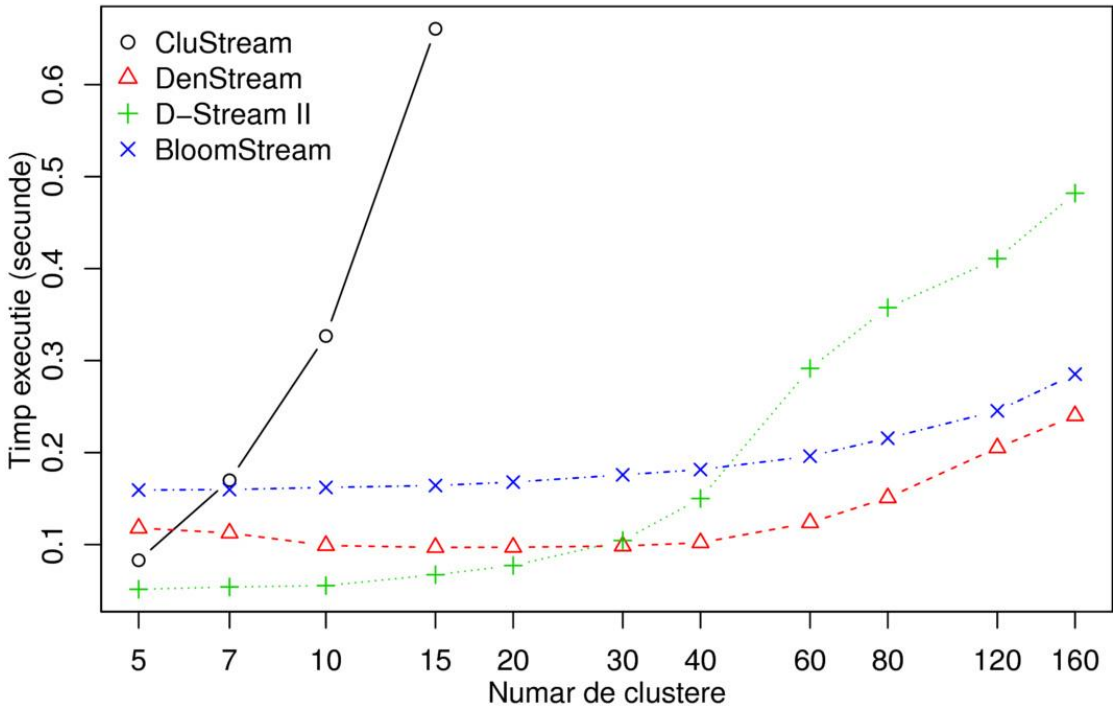
BloomStream este implementat în C++ cu o interfața R pentru pachetul Stream R, asemenea celorlalți algoritmi comparati: algoritmilor CluStream, DenStream și D-Stream II. Algoritmul propus nu diferențiază intre etapele de clustering online și offline tipice și nici nu colectează statistici bazate pe distante la nivelul fiecărui cluster. Acest lucru face inutilizabile toate măsurile de evaluare la nivel de micro-

cluster și toate măsurile bazate pe distante la nivel de macro-cluster. Din măsurile externe de validare la nivel de macro-cluster, am selectat puritatea pentru a cuantifica acuratețea soluției de clustering în comparație cu clasificarea corectă, realizata manual.

Testele de scalabilitate sunt realizate asupra unor fluxuri de date sintetice conținând o serie de clustere gaussiene cu 10% zgomot uniform distribuit. Clusterele sunt clar separate cu o distanța minima intre clustere de cel puțin 4. Experimentele pornesc de la un flux de date conținând 5 clustere în 5 dimensiuni în cadrul unei ferestre conținând 2000 instanțe de date. Separat, numărul de clustere și dimensionalitatea fluxului de date sunt mărite pana la 160. Orizontul de evaluare este egal cu mărimea ferestrei și setat de asemenea la 2000 instanțe de date.

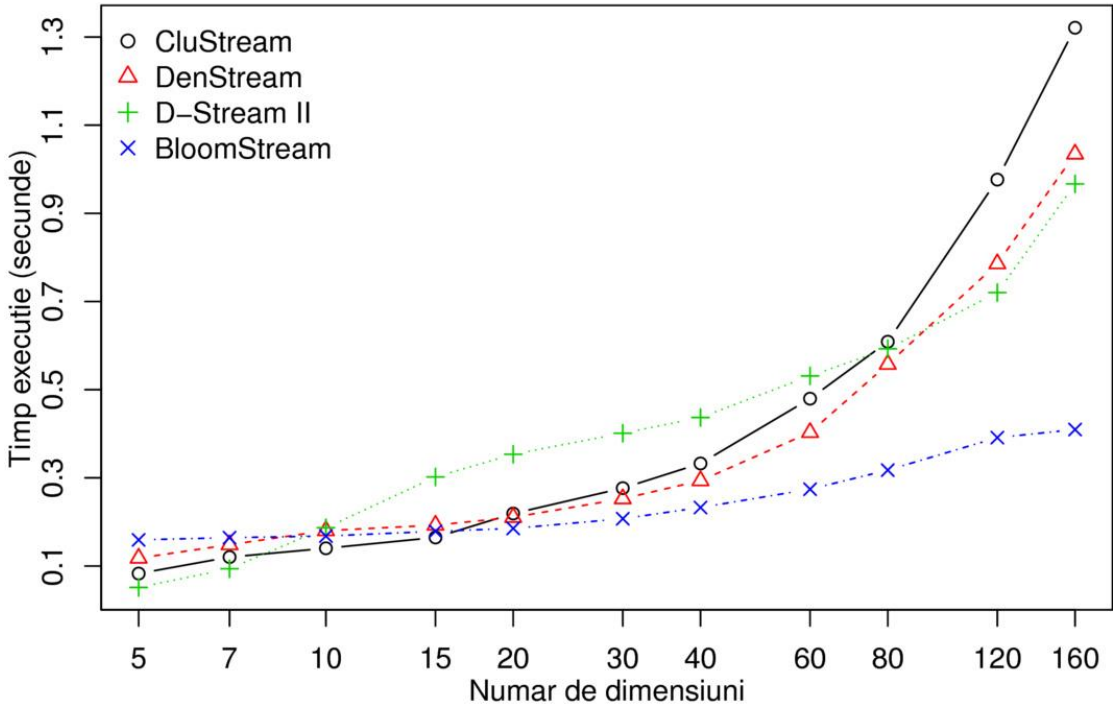
Testele de acuratețe sunt realizate asupra setului de date KDD CUP'99 Network Intrusion Detection ce conține aproximativ 5 milioane de conexiuni TCP, cu fiecare conexiune conținând 42 de atribute. în total sunt prezente 25 de clustere reprezentând 24 de tipuri de atacuri plus conexiunea normala. Datorita naturii sporadice a atacurilor, fluxul de date aferent evoluează rapid cu un număr variat de clustere prezente la orice moment în timp. Din aceasta cauză, lungimea intervalului de evaluare a fost setată la 1000 instanțe date. Sunt folosite toate cele 34 de atribute continue, normalizate la [0, 1].

Cu ajutorul structurii count-min, BloomStream determină daca o nouă instanța de date aparține unei celule dense în timp *O(k)*. Dacă celula este identificată a fi densă, un filtru bloom din fragmentul cluster aferent finalizează o actualizare a soluției clustering în timp *O(m)*. Timpul total de execuție depinde de numărul celulelor dense și mărimea fragmentului cluster. Evaluarea unei instanțe de date pentru determinarea etichetei se face în timp *O(m)*, timpul necesar pentru a testa dacă semnătura celulei grid aferente este conținuta de către una din semnăturile cluster existente. Aceasta evaluare nu depinde de numărul de clustere sau dimensionalitatea fluxului de date.



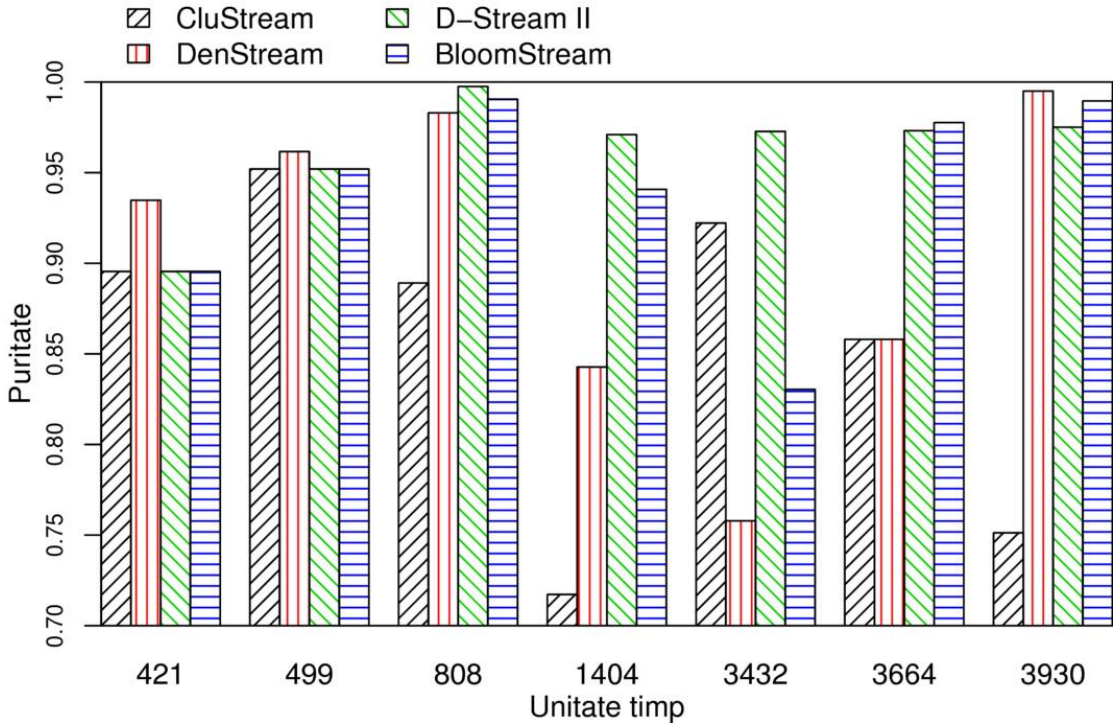
Figură 24. BloomStream, scalabilitate privind numărul de clustere

Scalabilitatea BloomStream în raport cu numărul de celule dense și mărimea fragmentului cluster este măsurata variind numărul de clustere (Figura 15), respectiv numărul de dimensiuni (Figura 16). Axa *x* pentru ambele grafice este logaritmica. Timpul de execuție cuprinde atât timpul de actualizare a soluției clustering cât și timpul de etichetarea al instanțelor de date. în ambele experimente, timpul de etichetare BloomStream este constant și reprezintă majoritatea timpului total de execuție atunci când fie numărul de clustere, fie numărul de dimensiuni este scăzut. Acest lucru explică timpul aproximativ constant de execuție atunci când dimensionalitatea fluxului este sub 20 dimensiuni, sau când acesta conține sub 40 de clustere. în aceste scenarii, timpul de execuție al celorlalți algoritmi este în general mai mic deoarece, în cazul acestora, procesele de actualizare al soluției clustering și etichetare scalează cu caracteristicile fluxului, fără costuri adiționale inițiale.



Figură 25. BloomStream, scalabilitate privind numărul de dimensiuni

Scalabilitatea D-Stream II este afectată de faptul ca este nevoit să creeze un număr mare de celule grid sporadice sau tranziționale cauzate de zgomot sau noi clustere în formare. Ca factor adițional, pragurile sale de densitate, pentru celulele sporadice și tranziționale, sunt calculate pe baza densității medii a unui celule grid, densitate ce se apropie de 0 pe măsura ce numărul de dimensiuni creste. Acest lucru duce la un număr mare de micro-clustere. în comparație, BloomStream nu prezintă aceste probleme, scaland superior pe măsura ce numărul de clustere sau dimensionalitatea fluxului de date creste.



Figură 26. BloomStream, puritate in setul de date KDDCUP 99 Network Intrusion

6. Concluzii

6.1. Contribuții originale

Pe parcursul cercetării sale științifice, autorul acestei teze a propus și publicat o serie de algoritmi de clustering, atât tradiționali cât și axați pe analiza fluxului de date, ce pot fi utilizați fie pentru analiza generala a unui set/flux de date, fie pentru identificarea tranzacțiilor suspecte în vederea detectării fraudei economico-financiare. Lucrări sumarizand stadiul actual al cunoașterii și analize comparative privind eficienta algoritmilor de clustering în detectarea anomaliilor completează contribuțiile aduse în domeniu. În ceea ce urmează sunt trecuți în revista algoritmii propuși de clustering asupra fluxurilor de date.

Algoritmul MaStream. Acest algoritm folosește conceptul de „mass estimation” ca alternativă la estimarea densității fără a folosi o măsura de similaritate bazată pe distante, fapt ce il face atractiv în analiza fluxurilor de date de dimensionalitate ridicată. Prin intermediul unei suite de arbori h:d, algoritmul permite identificarea clusterelor atât convexe cât și non-convexe, precum și a anomaliilor, printr-o singură scanare a datelor.

Algoritmul BloomStream. Discretizând fluxul de date în celule grid, algoritmii de clustering tip grid sunt nevoiți sa mentină un număr de celule grid exponențial în dimensionalitatea fluxului. Algoritmii existenți încearcă să minimizeze această problemă, cu mai mult sau mai puțin succes, implementând o serie de condiții și structuri specifice în vederea tratării celulelor sporadice și tranziționale cauzate de zgomot, respectiv clustere emergente. Algoritmul propus, BloomStream, nu suferă de aceste probleme deoarece structura count-min folosită în aproximarea densității celulelor grid oferă un timp de execuție și consum de memorie constant, independent de numărul de celule grid sau dimensionalitatea fluxului de date.

6.2.Lista lucrărilor originale

A. Lucrări publicate / în curs de publicare (acceptare) în reviste / jurnale de specialitate de circulaţie internaţională recunoscute (cotate / indexate ISI Thomson Reuters, sau indexate în alte Baze de Date Internaţionale - BDI specifice domeniului, care fac un proces de selecţie a revistelor pe baza unor criterii de performanţă)

1. Sabău, Andrei Sorin. "Stream Clustering using Probabilistic Data Structures", *Pattern Recognition Letters*, (pending), Scor relativ de influenţă: 1.632, ISSN: 0167-8655, indexată în: ACM Computing Reviews, CompuScience, Cambridge Scientific Abstracts, Computer Abstracts, Current Contents/Engineering, Computing & Technology, Engineering Index, Geographical Abstracts, INSPEC Information Services, SCISEARCH, Science Citation Index, Zentralblatt MATH, Scopus;
   http://www.journals.elsevier.com/pattern-recognition-letters
   Categorie jurnal: A, **Punctaj**: 8.

2. Sabău, Andrei Sorin. "Survey of clustering based financial fraud detection research." *Informatică Economică*, vol. 16, no. 1/2012, pp. 110-122, ISSN: 1453-1305, EISSN: 1842-8088, indexată în: CNCSIS B+, Directory of Open Access Journal, Cabell's Directories of Publishing Opportunities, EBSCO (Business Source Complete), ICAAP, Index Copernicus, Index of Information Systems Journals, Inspec, Open J-Gate, ProQuest Computing, RePEc, Ulrich's Periodicals Directory;

   http://revistaie.ase.ro/content/61/10%20-%20Sabau.pdf
   Categorie jurnal: D, **Punctaj**: 1.

B. Lucrări susţinute la conferinţe internaţionale şi publicate în volumele manifestărilor ştiinţifice internaţionale recunoscute, organizate în ţară şi străinătate, indexate ISI Thomson Reuters sau indexate în alte Baze de Date Internaţionale - BDI specifice domeniului, care fac un proces de selecţie a publicaţiilor pe baza unor criterii de performanţă

1. Sabău, Andrei Sorin. "Clustering Data Streams Using Mass Estimation." *15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2013, Publications IEEE Symbolic and Numeric Algorithms for Scientific Computing Symposium on, pp. 289-295, ISBN-13: 978-1-4799-3035-7, indexată în: IEEE Xplore, ISI Thomson Reuters Services.
   http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?filter%3DAND%28p_IS_Number%3A6821113%29&refinements=4224690034&pageNumber=1&resultAction=REFINE
   Categorie conferința: C, **Punctaj**: 2.

2. Sgârdea, Florinel Marian, Sabău, Andrei Sorin. "A Comparative Analysis of Semi-Supervised Clustering Performance în Financial Fraud Detection", *12th International Conference On Informatics în Economy (IE)*, 2013, Proceedings of the 12th International Conference on Informatics în Economy, pp. 624-631, ISSN: 2284-7472, indexată în: ISI Thomson Reuters Services, Directory of Open Access Journal, Cabell's Directories of Publishing Opportunities, EBSCO (Business Source Complete), ICAAP, Index Copernicus, Index of Information Systems Journals, Inspec, Open J-Gate, ProQuest Computing, Ulrich's Periodicals Directory, INFOREC Association.
   http://apps.webofknowledge.com/summary.do?SID=U2PuKFGC5ry78RsmWap&product=UA&qid=4&search_mode=GeneralSearch
   Categorie conferința: D, **Punctaj**: 0.5

3. Sabău, Andrei Sorin. "Variable density based genetic clustering", *14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2012, Publications IEEE Symbolic and Numeric Algorithms for Scientific Computing Symposium on, pp. 200-206, ISBN-978-0-7695-4934-7, indexată în: IEEE Xplore, ISI Thomson Reuters Services; http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?&filter%3DAND%28p_IS_Number%3A6480995%29&searchWithin=sabau&pageNumber=1&resultAction=REFINE
   Categorie conferința: C, **Punctaj**: 2.

4. Sabău, Andrei Sorin. "OpenCL Parallel GPU Programming Performance Optimization Strategies for Data Mining Clustering Techniques", *4th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2011, Proceedings of the International Conference on Electronics, Computers and Artificial Intelligence – ECAI – 2011, Vol.4-No.4, pp. 13-16, ISSN 1843-2115, indexată în: ISI Thomson Reuters Services; http://ecai.ro/Arhiva/ECAI-2011.pdf
   Categorie conferința: D, **Punctaj**: 1

5. Sabău, Andrei Sorin, Sgârdea, Florinel Marian, Sabău, Elena Monica, Budacia, Lucian Constantin Gabriel. "Fundamentals of continuous auditing and monitoring în enterprise resource planning system", 12th *WSEAS International Conference on Mathematics and Computers în Biology, Business and Acoustics*, Procedings of the world multiconference on Mathematics and Computers în Biology, Business and Acoustics, pp 45-48, ISBN 978-960-474-293-6, indexată în: ISI Thomson Reuters Services, http://dl.acm.org/citation.cfm?id=1991155

Categorie conferința: -, **Punctaj**: -

6. Sabău, Andrei Sorin, Sgârdea, Florinel Marian, Sabău, Elena Monica, Budacia, Lucian Constantin Gabriel. "Data mining life cycle în fraud auditing", 12th *WSEAS International Conference on Mathematics and Computers în Biology, Business and Acoustics*, Procedings of the world multiconference on Mathematics and Computers în Biology, Business and Acoustics, pp 55-59, ISBN 978-960-474-293-6, indexată în: ISI Thomson Reuters Services, http://dl.acm.org/citation.cfm?id=1991155

   Categorie conferința: -, **Punctaj**: -

Total punctaj pentru lucrările publicate / în curs de publicare: 14.5.

C. Lucrări elaborate și nepublicate (rapoarte de cercetare și prezentări în cadrul unor manifestări ştiinţifice academice)

1. Raport de cercetare nr. 1 "Realizarea studiului comparativ privind aplicarea diverselor tehnici de data mining în procesele de audit"
2. Raport de cercetare nr. 2 "Tehnici de clustering în baze de date relationale"
3. Raport de cercetare nr. 3 "Tehnici de clustering în stream mining"

D. Citări ale lucrărilor publicate (impactul lucrărilor ştiinţifice)

Articol citat:

Sabău, Andrei Sorin. "Survey of clustering based financial fraud detection research." Informatică Economică, vol. 16, no. 1/2012, pp. 110-122, ISSN: 1453-1305, EISSN: 1842-8088, Număr citări: 14;

Articole care citează:

- Ahmed, Mohiuddin, Abdun Naser Mahmood, and Md Rafiqul Islam. "A survey of anomaly detection techniques în financial domain." *Future Generation Computer Systems* 55 (2016): 278-288.
  Categorie A, Punctaj 8 / 1 = 8;
- Alexandre, Claudio, and João Balsa. "Integrating Client Profiling în an Anti-money Laundering Multi-agent Based System." *New Advances în Information Systems and Technologies*. Springer International Publishing, 2016. 931-941.
  Categorie D, Punctaj 1 / 1 = 1;
- Baracaldo, Nathalie. *TACKLING INSIDER THREATS USING RISK-AND-TRUST AWARE ACCESS CONTROL APPROACHES*. Diss. University of Pittsburgh, 2016.
  Categorie D, Punctaj 1 / 1 = 1;
- Fashoto, Stephen G., et al. "Development of improved K-means clustering for health insurance claims." *Computer Science & Telecommunications* 47.1 (2016).
  Categorie D, Punctaj 1 / 1 = 1;
- Hilton, Ross P., Nicoleta Serban, and Richard Y. Zheng. "Uncovering Longitudinal Healthcare Utilization from Patient-Level Medical Claims Data."*arXiv preprint arXiv:1603.00896* (2016).
  Categorie D, Punctaj 1 / 1 = 1;
- Hilton, Ross P. "Model-based data mining methods for identifying patterns în biomedical and health data." (2015). Hilton, Ross P. "Model-based data mining methods for identifying patterns în biomedical and health data." (2015).
  Categorie D, Punctaj 1 / 1 = 1;
- ALMENDRA, VINICIUS, and DENIS ENACHESCU. "Using Self-organizing Maps for Binary Classification with Highly Imbalanced

Datasets." *International Journal of Numerical Analysis and Modeling, series b* 5.3 (2014): 238-254.

Categorie D, Punctaj 1 / 1 = 1;

- Agarwa, Surbhi. "A Fast Fraud Detection A Clustering Based." *Journal of Basic and Applied Engineering Research* 10.1 (2014): 33-37.

  Categorie D, Punctaj 1 / 14 = 0.07;
- Minhas, Saliha, and Amir Hussain. "From Spin to Swindle: Identifying Falsification în Financial Text." *Cognitive Computation* (2014): 1-17.

  Categorie D, Punctaj 1 / 1 = 1;
- Aziz, N. H. A., N. B. Zakaria, and I. S. Mohamed. "Financial fraud: Data mining application and detection." *Innovation, Communication and Engineering* (2013): 341.

  Categorie D, Punctaj 1 / 1 = 1
- da Silva Almendra, Vinicius, and Denis Enachescu. "Using self-organizing maps for fraud prediction at online auction sites." *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2013 15th International Symposium on*. IEEE, 2013.

  Categorie C, Punctaj 2 / 1 = 2;
- Santoso, Halim Budi. *KLASTERISASI HARGA SAHAM DAN KOMODITAS MENGGUNAKAN METODE HYBRID KLASTERISASI*. Diss. UAJY, 2013.

  Categorie D, Punctaj 1 / 1 = 1;
- Yu, Zhang, Yu Guang, and Jin Zi-qi. "Violations detection of listed companies based on decision tree and K-nearest neighbor." *Management Science and Engineering (ICMSE), 2013 International Conference on*. IEEE, 2013.

  Categorie C, Punctaj 2 / 1 = 2
- Wang, John, and James GS Yang. "APPLICATIONS OF BUSINESS

ANALYTICS în CONTEMPORARY BUSINESS ISSUES." *JOURNAL OF BUSINESS ISSUES (2012)*: 35.
Categorie D, Punctaj 1 / 1 = 1;

Articol citat:

Sabău, Andrei Sorin. "Variable density based genetic clustering", 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012, Publications IEEE Symbolic and Numeric Algorithms for Scientific Computing Symposium on, pp. 200-206, ISBN-978-0-7695-4934-7; Număr citări: 2;

Articole care citează:

- Yang, Shuling, et al. "An Optimized Clustering Algorithm Using Improved Gene Expression Programming." *International Symposium on Intelligence Computation and Applications*. Springer Singapore, 2015.
  Categorie D, Punctaj 1 / 1 = 1;
- Cobo Rodríguez, Germán. "Parameter-free agglomerative hierarchical clustering to model learners' activity în online discussion forums." (2014).
  Categorie D, Punctaj 1 / 1 = 1;

Total punctaj pentru impactul științific al lucrărilor publicate: 22

Punctaj lucrări publicate / în curs de publicare	**14.5**	**Număr lucrări publicate / în curs de publicare**
Punctaj lucrări publicate Punctaj lucrări în curs de publicare	6.5 8	Categoria A: 1 articol (in curs de publicare) Categoria C: 2 articole Categoria D: 3 articole Categoria articole nepunctate: 2 articole

Punctaj citari	**22**	2 articole citate din care: 1 articol: 14 citări 1 articol: 2 citări

Tabel 8. Punctaj lucrări și citări