Raport științific

PN-III-P1-1.1-TE-2021-1374

CGTFE: Classification, Game Theory and Feature Engineering: new models and applications

(Clasificare, teoria jocurilor și extragerea atributelor: noi modele si aplicații)

1 Sumar

Obiective și rezultate estimate

Etapa 1 - Modele bazate pe teoria jocurilor pentru clasificare de date cu număr mare de atribute

- 1.1 Recenzia abordărilor recente legate de teoria jocurilor, extragerea atributelor și clasificarea
- 1.2 Extinderea FROG pentru a face față unui număr mare de caracteristici din date. Explorați metode de boosting.
- 1.3 Ingineria atributelor folosind metode de clustering de retele și de teoriei jocurilor
- 1.4 Diseminarea rezultatelor
- 1.5 Management de proiect

Etapa 2 - Modele bazate pe teoria jocurilor pentru probleme de clasificare multiclasa cu un număr mare de atribute.

- 2.1 Explorarea folosirii tehnicilor de mechanism design pentru selectarea și clasificarea atributelor.
- 2.2 Clasificarea de date reale: aplicarea în finanțe și marketing.
- 2.3 Documentarea abordărilor recente legate de teoria jocurilor, ingineria caracteristicilor și clasificarea cu clase multiple.
- 2.4 Analiza provocărilor legate de extinderea modelelor propuse în O1 la clasificarea cu clase multiple; Extinderea FROG pentru clasificarea cu clase multiple pentru date cu numar mare de atribute.
- 2.5 Diseminarea rezultatelor
- 2.6 Management de proiect

Etapa 3 - Modele bazate pe teoria jocurilor și analiza de rețele pentru clasificare cu clase multiple

- 3.1 Metode de clasificare bazate utilizarea de rețele multipartite și a teoriei jocurilor
- 3.2 Clasificarea de date reale: aplicarea metodelor propuse pentru date din finanțe și marketing
- 3.3 Diseminarea rezultatelor

3.4 Management de proiect

Livrabile propuse

- 5 articole trimise spre publicare (indexate Web of Science)
- raport de cercetare
- pagina web a proiectului: `https://www.cs.ubbcluj.ro/~mihai-suciu/cgtfe/`

Rezultate obținute Toate obiectivele au fost abordate cu succes. Au fost trimise şi acceptate spre publicare cinci articole iar alte două fost trimise. Astfel:

- În lucrarea [P1] se propune o metodă de selectare a atributelor bazata de un arbore de decizie construit folosind conceptul de echilibru Nash;
- În lucrarea [P2] este propus un algoritm genetic pentru selecția atributelor, importanța precum și eficacitatea atributelor selectate de fiecare individ sunt evaluate prin utilizarea arborilor de decizie, arborele indus de cel mai bun individ din populație este utilizat pentru clasificare a datelor;
- În lucrarea [P3] se propune o strategie evolutivă pentru selecția atributelor, ponderile atributelor sunt evoluate cu arbori de decizie care utilizează conceptul de echilibru Nash pentru a împărți datele din noduri, arborii sunt menținuți până când variația probabilităților induse de ponderile atributelor stagnează;
- În lucrarea [P4] se propune un model *decision forest* bazat pe echilibrul Nash pentru a selecta atributele relevante în problema clasificării.
- În lucrarea [P5] se propune o hyper euristică pentru optimizare continua. În următorul pas aceasta este folosită pentru o problema de clasificare, transformând problem de clasificare într-o problema de optimizare continua.
- În lucrarea [P6] hyper-eurisitica propusa în P5 este îmbunătățita, este prezentată o aplicație pentru problema de clasificare, problema de clasificare este descris ca o funcție continua.
- Lucrarea [P7] propune un framework pentru rezolvarea problemei selecției atributelor bazat pe teoria cooperativa a jocurilor: problema este transformată într-un joc de tipul *weighted voting*, indexul Banzhaf este utilizat pentru selecția atributelor.

Lista de articole trimise spre publicare/acceptate:

P1 Suciu, M., Lung, R. I. (2023). A New Filter Feature Selection Method Based on a Game Theoretic Decision Tree. In A. Abraham, T.-P. Hong, K. Kotecha, K. Ma, P. Manghirmalani Mishra, & N. Gandhi (Eds.), Hybrid Intelligent Systems (pp. 556–565). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-27409-1_50

P2 Suciu, MA., Lung, R.I. (2023). Feature Selection Based on a Decision Tree Genetic Algorithm. In: García Bringas, P., et al. Hybrid Artificial Intelligent Systems. HAIS 2023. Lecture Notes in Computer Science(), vol 14001. Springer, Cham. https://doi.org/10.1007/978-3-031-40725-3_37 (articol publicat)

P3 Lung, R. I., Suciu, M.-A. (2024). An Evolutionary Approach to Feature Selection and Classification. Machine Learning, Optimization, and Data Science: 9th International Conference, LOD 2023, Grasmere, UK, September 22–26, 2023, Revised Selected Papers, Part I, 333–347. https://doi.org/10.1007/978-3-031-53969-5_25

P4 Mihai Suciu, Rodica Ioana Lung. A game theoretic decision forest for feature selection and classificatio. Logic Journal of the IGPL. (articol acceptat 2023)

P5 Bándi, N., Gaskó, N. (2024). Nested Markov chain hyper-heuristic (NMHH): A hybrid hyper-heuristic framework for single-objective continuous problems. PeerJ Computer Science, 10, e1785. https://doi.org/10.7717/peerj-cs.1785 (articol publicat)

P6 Nándor Bándi, Noémi Gaskó. Improved Nested Markov Chain Hyper-heurisitic framework. A clustering application. Expert Systems with Applications. (articol trimis)

P7 Noémi Gaskó. Feature selection based on Banzhaf index, an approach based on cooperative game theory. Neurocomputing (articol trimis)

2 Rezumat Executiv

În prima fază, proiectul se axează pe combinarea arborilor de decizie cu modele din teoria jocurilor pentru a rezolva problema selecției atributelor relevante pentru problema de clasificare binară. Problema selecției atributelor a devenit o activitate cheie în cadrul învățării automate. Pentru problemele de clasificare, se știe că reduce complexitatea de calcul a estimării parametrilor, dar adaugă și o contribuție importantă la aspectele de înțelegere și explicare a rezultatelor.

Un arbore de decizie bazat pe un model de joc este utilizat pentru selecția atributelor. În timpul fazei de inducție a arborelui, atributul utilizat la împărțirea datelor este ales pe baza unui joc între instanțe din aceeași clasă. Presupunerea abordării este: componenta de joc va indica cele mai importante atribute. O măsură pentru importanța unui atribut este calculată pe baza numărului de apariții a unui atribut și a adâncimii nodului în arbore. Rezultatele sunt comparabile și mai bune în unele cazuri decât cele raportate printr-o abordare standard bazată și pe arbori de decizie.

A doua abordare analizată propune un algoritm genetic pentru selecția atributelor. Importanța, precum și eficacitatea atributelor alese de fiecare individ, este evaluată prin utilizarea arborilor de decizie. Importanța atributului indicată de arborele de decizie este utilizată în faza de selecție și recombinare a algoritmului genetic. Arborele indus de cel mai bun individ din populație este folosit pentru clasificare. Experimentele numerice ilustrează comportamentul abordării.

Datorită legăturii structurale naturale dintre arborii de decizie și rețelele neuronale (NN), în ultimii ani s-au dezvoltat un tip de arbori de clasificare și regresie care folosesc pentru separarea datelor din noduri funcții împrumutate din NN, cum ar fi funcția sigmoid [32]. Urmând direcția folosirii arborilor de decizie pentru selecția de atribute, se studiază efectul folosirii FROG/probit pentru separarea datelor în construcția arborelui și eficiența acestei abordări în identificarea importanței atributelor. O altă direcție studiată constă în folosirea unui algoritm *random forest* pentru extragerea atributelor, prin agregarea datelor din frunze și folosirea unei variante extinse FROG pentru probleme de clasificare cu mai multe clase împreuna cu un mecanism de *boosting* pentru atribute.

Arborii de decizie pentru clasificare sunt în general construiți într-o manieră recursivă, ”top-down”, pornind de la nodul rădăcină. Într-o abordare opusă, construcția arborilor se poate face și ”bottom-up”, prin popularea inițială a frunzelor cu date și agregarea lor mergând către rădăcină. Această abordare folosește un algoritm de *clustering* pentru a crea frunzele, împreună cu un algoritm care reprezintă separarea sub forma unui hiper-plan, cum ar fi SVM pentru a agrega datele. În cadrul proiectului se explorează folosirea jocului propus în [P1] pentru construirea unui arbore de decizie de tip bottom-up pornind de la datele din frunze. Distanța minimă dintre clusterii din frunze (sau fiecare nivel de nod) este calculată și clusterii cei mai apropiați sunt agregați, generându-se o regulă de separare a datelor agregare folosind echilibrul Nash. Importanța atributelor se calculează pe baza arborelui astfel construit. Avantajul acestei abordări constă în controlul mai mare asupra mărimii arborelui și a purității datelor din frunze.

O altă direcție de cercetare se axează pe găsirea unor moduri optime de evaluare a valorii sau a contribuţiei atributelor bazate pe concepte de teoria jocurilor astfel încât acestea să reflecte cât mai eficient caracteristicile datelor. În acest sens s-a studiat efectul mai multor modele de evaluare şi s-au explorat mai multe variante de funcţii de câstig în vederea dezvoltării de algoritmi de selecţie de atribute în explicarea datelor.

Se studiază o metodă de clasificare bazată pe selecția atributelor care evoluează ponderile atributelor folosind arbori de decizie care utilizează conceptul de echilibru Nash pentru a-și împărți datele din noduri. Se propune o strategie evolutivă pentru selecția atributelor. Indivizii reprezintă vectori de importanță a atributelor, evoluați cu scopul de a identifica cele mai relevante atribute din setul de date care pot explica problema de clasificare. Un arbore de decizie care folosește echilibrul Nash este utilizat în scopuri de clasificare și evaluare. Abordarea propusă nu implică mecanisme de selecție, arborii sunt crescuți împreună și formează un ecosistem în care toți sunt implicați în sarcina de predicție. Un individ se oprește din evoluție atunci când nu mai există variații în probabilitățile pe care le oferă pentru selectarea atributelor pentru inducerea arborelui său.

O altă abordare analizează interpretarea informațiilor furnizate de arborii de decizie și metode *random forest* pentru clasificare și selecția atributelor. Un arbore de decizie bazat pe concepte din teoria jocurilor este utilizat pentru a construi o pădure care rezolvă problema clasificării și pentru a evalua importanța atributelor. La construirea pădurii, informațiile despre selecția anterioară a atributelor sunt folosite pentru a îmbunătăți căutarea. Arborele de decizie și *random forest* reprezintă date sub formă de subseturi: partiții în cazul arborilor de decizie și seturi de partiții în cazul *random forest*. Aceste seturi oferă informații locale despre datele care pot fi utilizate în continuare pentru a face predicții pentru datele de test care se potrivesc în acea regiune specială a spațiului de căutare. Datele locale furnizate de *random forest* sunt agregate, iar un clasificator este folosit pentru a face predicții pentru probleme de clasificare.

În continuarea demersului de a folosi elemente oferite de *mechanism design* în problema de clasificare s-au explorat variante de a selecta atribute bazate pe contribuţia marginală la valoarea unui indicator consacrat în această problemă. Deoarece în cele mai multe situaţii problema este de a găsi un compromis între mai multe valori (de ex. corelaţii între atribute/clase), o abordare bazată pe contribuţie marginală poate conduce la soluţii de echilibru cu valoare practică mai stabilă decât cele obţinute prin combinarea aritmetică indicatorilor utilizaţi. O astfel de abordare este testată folosind ca şi indicator meritul unei mulţimi de atribute şi un algoritm genetic standard. Rezultatele preliminare indică o acurateţe superioară raportata de varianta bazată pe contribuţii marginale.

O aplicație practică care analizează clasificarea țărilor în grupe de venit pe baza indicatorilor de dezvoltare mondială prezintă o interpretare a abordării selecției atributelor este folosită pentru a ilustra metodele propuse şi a evidenţia caracterul explicativ al acestora.

Obiectivul de a automatiza procesul de generare de forme de jocuri ale căror echilibre conduc la clasificarea corectă a datelor este atins prin explorarea folosirii de metode ale programării genetice pentru a separa instanțe în clasificarea binară. Se folosește o funcție obiectiv care poate reprezenta funcția de câștig a unui joc corespunzător cu sumă nulă (sau constantă), astfel încât metoda propusă să modeleze jocuri în selecția de atribute. Folosirea soluțiilor oferite de teoria jocurilor crește robustețea metodei și poate îmbunătăți explicabilitatea abordării.

Pe o altă direcție se explorează folosirea mecanismelor oferite de teoria grafelor pentru identificarea atributelor importante dintr-un set de date pentru clasificare nesupervizată cu mai multe clase. Astfel, setul de date este codificat ca și rețea multipartită, în care fiecare nivel corespunde unui atribut. Nivelele cele mai importante au un grad mai mic; eliminarea nivelelor cu grad (total) mai ridicat conduce la o clasificare mai bună a datelor cu diferite metode

consacrate de clustering.

3 Descrierea științifică

În continuare sunt detaliate o selecție a rezultatelor obținute în cadrul proiectului legate de selecția de atribute folosind metode ale inteligenței computaționale și teoria jocurilor.

3.1 O nouă metodă de selecție a atributelor relevante bazată pe un arbore de decizie generat folosind concepte din teoria jocurilor

Un pas cheie în analiza datelor este reprezentat de selecția atributelor. Orice decizie luată pe baza rezultatelor unei analize trebuie să țină cont de limitările care reies în mod natural din date, precum și de metodele utilizate pentru a decide care sunt caracteristicile care sunt efectiv analizate. În timp ce selecția atributelor este obligatorie în contextul datelor mari, beneficiile acesteia pot fi avute în vedere și pe seturi de date mai mici, selecția reprezintă un prim pas în explicarea intuitivă a modelului de bază.

Metodele de selecție a atributelor [6] sunt, în general, clasificate în trei grupe: metode de filtrare, în care caracteristicile sunt selectate pe baza unor metrici care indică importanța lor [16], metode *wrapper* care iau în considerare subseturile setului de atribute evaluate prin potrivirea unui modelul de clasificare [15] și modelele încorporate care realizează intrinsec selecția caracteristicilor în timpul etapei de adaptare, de ex. arbori de decizie și păduri de arbori de decizie [28].

Arborii de decizie (*decision trees*) sunt, de asemenea, folosiți des pentru a valida metodele de selecție a atributelor [11]. Diverse aplicații din lumea reală folosesc arbori de decizie pentru testarea și validarea metodelor de selecție a atributeloe. De exemplu, în detectarea intruziunilor în rețea [23], predicția stocului [26], predicția azotului în stații de ape [2], *code smell* [14] sunt câteva din abordările unde modelele bazate pe arbori de decizie au prezentat o îmbunătățire semnificativă a performanței după selecția atributelor.

Cu toate acestea, selecția atributelor în sine bazată pe inducerea arborelui de decizie este una dintre cele mai intuitive abordări pentru a evalua importanța atributelor unui set de date. Deoarece arborii de decizie sunt construiți recursiv și, la fiecare nivel de nod, trebuie alese anumite atribute pe baza cărora se vor împărți datele, este firesc să ne gândim că atributele implicate în procesul de împărțire sunt importante în explicarea datelor. Cu toate acestea, majoritatea metodelor de selecție a atributelor care se bazează pe arbori de decizie utilizează în cele din urmă o formă de pădure aleatoare (*random forest*), adică arbori multipli induși pe date și atribute eșantionate, în diferite forme și pentru diferite aplicații [12, 22, 28].

Presupunem că mai este loc de explorat în utilizarea unui singur arbore de decizie pentru selecția atributelor unui set de date, deoarece performanța oricărei abordări depinde în mod natural de metoda de inducție a arborelui. Propunem utilizarea unui arbore de decizie care împarte datele pe baza unei abordări teoretice de joc pentru a calcula importanța unei caracteristici și a o utiliza pentru selecție. Comparăm abordarea propusă cu o metodă de filtrare pentru o abordare *random forest* pe seturi de date sintetice și reale.

Selecția atributelor pe baza unui arbore de decizie și un model de joc (G-DTfs) Fie un set de date $(\mathcal{X}, \mathcal{Y})$, cu $\mathcal{X} \subset \mathbb{R}^{n \times d}$ și $\mathcal{Y} \subset \{0, 1\}^n$, astfel încât fiecare instanță $x_i \in \mathcal{X}$ are eticheta $y_i \in \mathcal{Y}$. Dacă $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_d)$, cu $\mathbf{X}_j \in \mathbf{R}^n$, dorim să găsim o submulțime $\{\mathbf{X}_1, \ldots, \mathbf{X}_d\}$ care explică cel mai bine etichetele $\mathcal{Y}$.

Se propune utilizarea unui arbore de decizie bazat pe teoria jocurilor pentru a identifica caracteristicile/atributele lui X care sunt cele mai influente în separarea datelor în cele două clase. La fiecare nivel de nod, atributul folosit pentru a împărți datele este ales prin simularea unui joc între cele două clase. Arborele este construit recursiv, de sus în jos, începând cu întregul set de date (de antrenament) de la nodul rădăcină. Pașii descriși în continuare sunt utilizați pentru a împărți datele nodului curent (X, Y).

Verificarea datelor În primul rând se verifică dacă datele din nod trebuie să fie împărțite sau nu. Condiția folosită este: dacă toate instanțele au aceeași etichetă, sau dacă X conține o singură instanță, nodul devine o frunză.

Împărțirea datelor la nivel de nod pe baza jocului Dacă datele (X, Y) dintr-un nod trebuie împărțite, se calculează un hiperplan paralel al axei pentru fiecare atribut $j = \overline{1, d}$ din date în felul următor.

Jocul nodului Se ia în considerare următorul joc $\Gamma(X, Y|j)$ compus din:

- jocul are doi jucători, L și R, de corespund celor două sub-noduri (nodurile fiu ale nodului părinte din arborele binar), respectiv celor două clase;
- strategia fiecărui jucător este să aleagă un parametru pentru hiperplan, β: β_L și, respectiv, β_R;
- câștigul fiecărui jucător este calculat în felul următor:

$$u_L(\beta_L, \beta_R|j) = -n_0 \sum_{i=1}^{n} (\beta_{1|j} x_{ij} + \beta_{0|j})(1 - y_i),$$

și

$$u_R(\beta_L, \beta_R|j) = n_1 \sum_{i=1}^{n} (\beta_{1|j} x_{ij} + \beta_{0|j}) y_i,$$

unde $\beta = \frac{1}{2}(\beta_L + \beta_R)$, n_0 și n_1 reprezintă numărul de instanțe având etichetele 0 și, respectiv, 1.

Echilibrul Nash (*Nash Equilibria* - NE) al acestui joc este reprezentat de o valoare β care combină β_L și β_R în așa fel încât niciunul dintre jucători să nu își mai poată muta sumele de produse la stânga sau la dreapta, respectiv, în timp ce celălalt jucător își menține alegerea neschimbată. Echilibrul jocului poate fi aproximat folosind o imitație a procedurii de joc fictiv (*fictitious play*) [5].

Aproximarea echilibrului Nash Versiunea simplificată de joc fictiv folosită aici pentru a găsi o valoare potrivită pentru β este implementată după cum urmează: pentru un număr de η iterații, cel mai bun răspuns al fiecărui jucător față de strategia celuilalt jucător este calculat folosind un algoritm de optimizare. Deoarece ne propunem doar să aproximăm valorile β care împart datele într-un mod rezonabil, căutarea se oprește după ce a trecut numărul de iterații. În fiecare iterație, cel mai bun răspuns la media strategiilor celuilalt jucător în iterațiile anterioare este considerat drept cel fix.

Alegerea unui atribut pe baza echilibrului Nash Pentru a alege atributul folosit pentru a împărți datele nodurilor, NE pentru fiecare atribut $j = \overline{1, d}$ sunt aproximate și datele corespunzătoare ale sub-nodurilor sunt separate și evaluate în continuare pe baza câștigului de entropie. Oricare atribut care întoarce cel mai mare câștig de entropie este ales pentru împărțirea datelor și β_j corespunzător este utilizat pentru a defini hiperplanul de separare.

Atribuirea importanței unui atribut în problema selecției atributelor

Odată ce arborele a fost indus/generat, importanța fiecărui atribut în împărțirea datelor poate fi luată în considerare dacă caracteristica este utilizată pentru împărțire și de adâncimea nodului

care o folosește. Pentru fiecare atribut $j \in \{1 \dots d\}$ notăm cu $v_j = \{v_{jl}\}_{l \in I_j}$ mulțimea care conține nodurile care împart datele pe baza atributului j, cu I_j setul de indici corespunzători din arbore și fie $\delta(v_{jl})$ adâncimea nodului v_{jl} în arborele de decizie, cu valori care încep de la 1 (nodul rădăcină). Atunci, importanța $\phi(j)$ a atributului j poate fi calculată ca:

$$\phi(j) = \begin{cases} \sum_{l \in I_j} \frac{1}{\delta(v_{jl})}, & I_j \neq \emptyset \\ 0, & I_j = \emptyset \end{cases} . \tag{1}$$

Astfel, importanța unui atribut depinde de adâncimea nodului care îl folosește pentru a împărți datele. Presupunem că atributele care sunt utilizate la începutul inducției arborelui de decizie pot fi mai influente. De asemenea, un atribut care poate apărea pe mai multe noduri cu o adâncime mai mare poate fi influent, iar indicatorul $\phi()$ cuprinde și această situație.

Experimente numerice Experimentele numerice sunt efectuate pe seturi de date sintetice și reale cu diferite grade de dificultate pentru a ilustra stabilitatea și performanța abordării propuse.

Parametrii folosiți Pentru seturile de date sintetice folosim parametrii pentru funcția `make_classification`: numărul de instanțe (250, 500, 1000), numărul de atribute (50, 100, 150), seed (500), ponderea fiecărei etichete (0.5 - seturile de date sunt echilibrate) și separarea claselor (0.5 - există suprapunere intre instanțe din clase diferite). Creeăm seturi de date pentru teste folosind toate combinațiile ale parametrilor descriși anterior.

Pentru abordarea propusă (G-DTfs) testăm diferiți parametri: adâncimea maximă a unui arbore (5, 10, 15), numărul de iterații pentru jocul fictiv (5).

Împărțim fiecare set de date, sintetice sau reale, în $M = 10$ subseturi. Raportăm rezultatele G-DTfs și abordarea comparată pe 10 rulări independente pentru fiecare set de date utilizat.

Comparăm rezultatele obținute de G-DTfs cu atributele alese de un clasificator *Random Forest* (RF) [4]. Pentru clasificatorul RF setăm parametrii: numărul de estimatori (100), criteriul de împărțire dintr-un nod (indexul *gini*), adâncimea maximă a fiecărui estimator (acest parametru ia aceeași valoare ca adâncimea maximă a G-DTfs).

Evaluarea performanței Pentru a evalua performanța G-DTfs, se folosește indicatorul de stabilitate, SC, [3, 18]. Indicatorul de stabilitate se bazează pe corelația Pearson între rezultatele raportate pe datele eșantionate și indică dacă metoda de selecție a atributelor este stabilă, adică cât de diferite/asemănătoare sunt atributelor alese pe baza unor eșantioane diferite din aceleași date. Deoarece aceasta este o caracteristică dorită a unei metode de selecție a atributelor, o folosim aici pentru a compara rezultatele raportate de G-DTfs cu o abordare standard *Random Forest* (RF) pentru selecția atributelor [19].

Pentru a calcula măsura stabilității, setul de date este împărțit în M subseturi prin utilizarea reeșantionării, iar metoda de selecție a atributelor este aplicată pe fiecare subset, rezultând M seturi de atribute, care sunt reprezentate ca vectori Z_i, $i = \overline{1, M}$, cu z_{ij} având valoarea 1 dacă atributul j a fost ales pe eșantionul i și 0 în caz contrar. Măsura stabilității face o medie a corelațiilor dintre toate perechile de vectori atribute, adică:

$$SC = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} Cor(Z_i, Z_j), \tag{2}$$

unde $Cor(Z_i, Z_j)$ arată corelația liniară dintre Z_i și Z_j. O corelație ridicată indică faptul că aceleași atribute sunt identificate ca fiind influente pentru toate eșantioanele, în timp ce o valoare de corelație apropiată de 0 ar indica aleatoriu în selecția atributelor. Dacă scorul este utilizat pentru a evalua metodele de selecție a atributelor, acesta indică care dintre ele este mai stabilă, cu cât scorul este mai mare, cu atât mai bine.

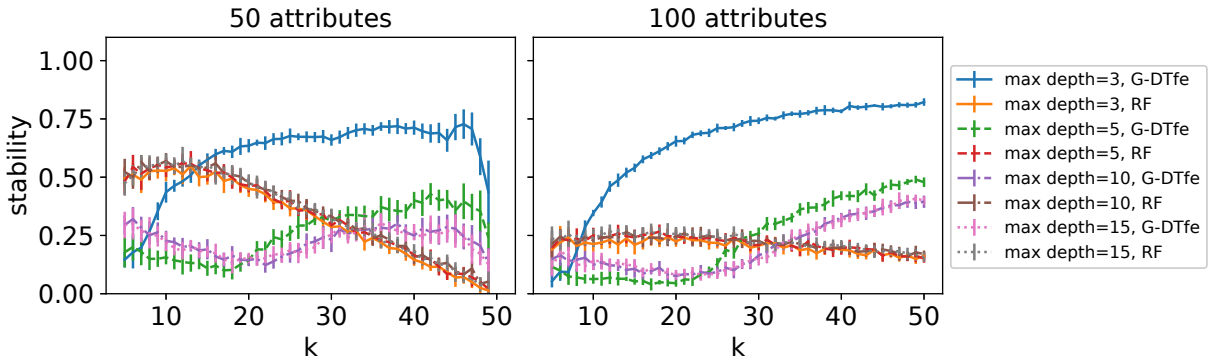

Figura 1: Efectul parametrului k asupra stabilității selecției atributelor pentru modelele G-DTfs și RF cu valori diferite pentru parametrul de adâncime maximă a arborelui de decizie $(3, 5, 10, 15)$ pe seturi de date sintetice cu 50 de atribute (stânga) și 150 de atribute (dreapta).

Rezultate numerice Rezultatele sunt prezentate ca medie și abatere standard a scorului SC raportat de cele două metode pentru diferitele combinații ale parametrilor testați. Rezultatele unui test t statistic care compară scorurile de stabilitate raportate de cele două metode însoțesc datele. Pentru seturile de date sintetice, observăm că în 14 setări (combinații diferite de parametrii) rezultatele obținute de G-DTfs sunt semnificativ mai bune. De asemenea, în majoritatea cazurilor, testul t este de prisos, deoarece diferențele indicate de valorile medii și deviației standard sunt evident semnificative. Acest lucru este totuși adevărat în ambele sensuri: ori de câte ori rezultatele RF sunt mai bune, diferența este, de asemenea, evident semnificativă.

Aceeași situație apare și în cazul datelor din lumea reală, cu observația suplimentară că creșterea numărului de atribute luate în considerare pare să scadă performanța RF și o crește pe cea a G-DTfs, în termeni de stabilitate. Deși este adevărat că se dorește un număr minim de atribute, trebuie luat în considerare și comportamentul unei metode atunci când se confruntă cu un număr mare de instanțe și atribute.

Efectul mărimii setului de atribute k asupra rezultatelor G-DTfs este ilustrat în Figura 1 pentru două seturi de date sintetice, cu 50 și 100 de atribute, comparativ cu cel al RF. Găsim măsuri de stabilitate mai mari pentru G-DTfs cu adâncime mică a arborelui (adâncime maximă de 3) și, de asemenea, tendința de scădere a măsurii de stabilitate pentru RF. Influența adâncimii arborelui asupra acelorași seturi de date este ilustrată în Figura 2 pentru diferite valori k. Rezultatele prezentate în aceste cazuri confirmă că scorul de stabilitate nu depinde de dimensiunea arborelui după un anumit prag, care, pentru aceste seturi de date, este în jur de 5.

Concluzii Problema identificării atributelor cheie care pot fi utilizate pentru a explica o caracteristică a datelor este una centrală în învățarea automată. Similar altor sarcini de învățare automată, eficiența și simplitatea sunt dorite din abordările practice. În această lucrare se folosește un arbore de decizie pentru a atribui o măsură de importanță atributelor care pot fi utilizate pentru filtrarea lor. Noutatea abordării constă în utilizarea unui mecanism de divizare teoretică a jocului pentru datele nodurilor în timpul inducției arborelui. Importanța unei caracteristici este atribuită în funcție de poziția nodului (nodurilor) care este utilizat pentru împărțirea datelor. În timp ce utilizarea unui singur arbore de decizie a dat rezultate comparabile și chiar mai bune decât o abordare standard de pădure aleatoare, o direcție deschisă de cercetare constă în explorarea unei păduri de arbori de decizie bazați pe teoretică a jocului pentru selecția atributelor.

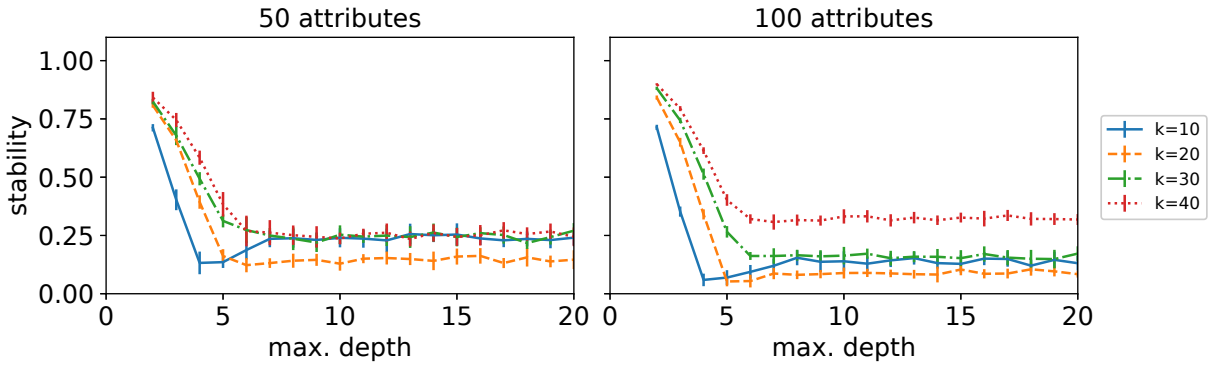

Figura 2: Efectul parametrului adâncime maximă a arborelui de decizie asupra stabilității selecției de atribute pentru G-DTfs pentru seturi de date sintetice cu 50 de atribute (stânga) și 150 de atribute (dreapta) și valori diferite pentru parametrul k $(10, 20, 30, 40)$.

3.2 Selecția atributelor bazată pe un arbore de decizie și un algoritm genetic

Introducere Explorarea diferitelor mecanisme de selecție a atributelor reprezintă un pas cheie în procesul de înțelegere a comportamentului modelelor de clasificare în prezența informațiilor redundante sau incomplete, contribuind, de asemenea, în mare măsură la înțelegerea modelului. Deoarece necesitatea metodelor de selecție a atributelor este evidentă în contextul datelor mari, din punctul de vedere al explicabilității și al atenției la detalii, poate adăuga valoare și atunci când este utilizat în contexte mai mici. În mod ideal, ar putea oferi o perspectivă asupra contribuției fiecărui atribut, sau grup de atribute, la rezultatele generale și să reducă complexitatea de calcul, crescând în același timp explicabilitatea.

Pentru abordarea acestei probleme se propune o metodă *wrapper* pe baza unui algoritm genetic (GA) hibridizat cu un mecanism încorporat. Metoda wrapper și mecanismul încorporat folosesc arbori de decizie care oferă populației GA valori de fitness, precum și importanțele atributelor care sunt utilizate în timpul procesului de recombinare și selecție.

fsGA-DT: selecția atributelor folosind algoritmi genetici și arbori de decizie Problema clasificării (binare) abordată constă în găsirea unei reguli care atribuie clase, sau etichete, datelor, pe baza informațiilor furnizate de un set de antrenament în care sunt cunoscute clase. Fie $X \subset \mathbb{R}^{n \times d}$ un set de date cu n instanțe și d atribute (sau caracteristici) și $Y \subset \{0, 1\}^n$ lor etichetele corespunzătoare. Problema de selecție a atributelor constă în găsirea subsetului de atribute care explică cel mai bine datele, adică care oferă o soluție bună la problema de clasificare. Minimizarea dimensiunii acestui set este, de asemenea, de dorit.

Arborii de decizie [27] sunt modele de clasificare și regresie care împart spațiul de date în regiuni cât mai pure posibil, adică conținând majoritatea instanțelor din aceeași clasă.

fsGA-DT evoluează o populație de indivizi care codifică seturi de caracteristici și utilizează informațiile furnizate de arbori de decizie pentru a ghida căutarea. Comunicarea dintre populația GA și DT este bidirecțională: informațiile circulă de la GA la DT sub formă de seturi de caracteristici și de la DT la GA sub forma evaluării performanței și a importanței caracteristicilor care trebuie utilizate în timpul procesului de evoluție. Figura 3 ilustrează etapele de comunicare în cadrul fsGA-DT. Detaliile fiecărei etape sunt explicate în cele ce urmează.

Codificarea fsGA-DT folosește codificare binară. Un individ x este reprezentat ca un șir de biți de dimensiunea d, unde valoarea 1 indică faptul că atributul corespunzător este ales și 0

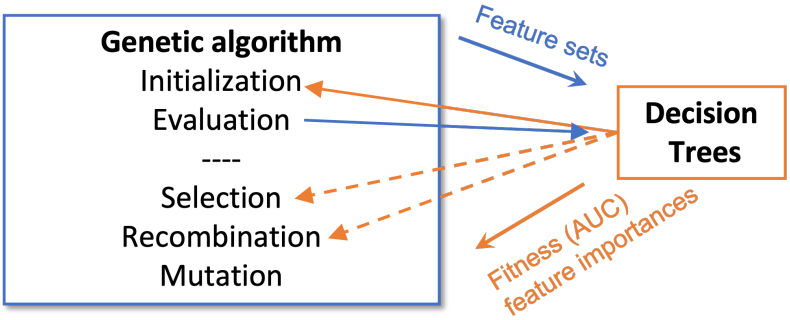

Figura 3: Comunicarea între GA și DT. Inițializarea se realizează pe baza informațiilor oferite exclusiv de DT. În timpul evaluării, seturile de atribute sunt trimise către DT și evaluate, primind o valoare de fitness și importanța atributelor care sunt utilizate în timpul selecției și recombinării.

că nu este. Pentru individul x notăm cu $FS(x) = \{j_1, \ldots, j_k\}$, $FS(x) \subset \{1, 2, \ldots, d\}$ setul de atribute care sunt selectate pe baza acestuia.

Evaluare Un individ x este evaluat prin inducerea unui arbore de decizie pe setul $FS(x)$ de atribute alese de acesta. Setul de date utilizat pentru selectarea atributelor este împărțit aleatoriu într-un set de antrenare care conține 70% din instanțe și un set de validare de 30% din instanțe, DT-ul este evaluat folosind setul de validare.

Importanța atributelor în timpul evaluării În timpul inducției arborelui, se ia o decizie la fiecare nivel de nod cu privire la atributul utilizat pentru a împărți datele nodului. Pe baza structurii arborelui, fiecărui atribut din setul de date i se atribuie o măsură de importanță. Aceste valori sunt normalizate, aparțin $[0, 1]$ și sunt transmise indivizilor din populație în timpul fazei de evaluare. Astfel, fiecărui individ x i se atribuie un vector care conține importanța atributului $fs(x) \in \mathbb{R}^d$. $fs(x)$ va avea valori pozitive care se adaugă la 1 pentru atributele nodurilor atribuite în DT de evaluare și 0 pentru celelalte poziții.

Inițializarea Pentru a începe căutarea cu o populație de soluții bune, inițializarea inversează procesul de evaluare: sunt crescuți un număr de arbori de decizie egal cu dimensiunea populației, iar indivizii din populația inițială sunt creați din fiecare arbore alegând la 1 toate atributele care au o importanță pozitivă în arbore. Arborii utilizați în această etapă sunt introduși/crescuți până când toate frunzele sunt pure pentru a obține o imagine de ansamblu asupra atributelor potențial utile încă de la începutul căutării.

Cei mai buni indivizi În fiecare iterație, cel mai bun individ din populație este reevaluat prin media rezultatelor raportate de 10 arbori de decizie pentru a reduce variabilitatea valorilor AUC. fsGA-DT păstrează cel mai bun individ peste toate iterațiile, evaluate în acest mod.

Selecția Selecția turnir este utilizată pe baza valorii de fitness. Cu toate acestea, dacă doi indivizi au aceeași valoare fitness (AUC), este selectat cel care are valoarea medie a importanței caracteristicii pozitive mai mare. O motivație din spatele acestei scheme este de a conduce căutarea către un număr mai mic de caracteristici, deoarece pentru fiecare individ, suma $fs(x)$ se adaugă la 1, iar o medie mai mare indică un număr mai mic de valori. Un altul este de a spori diversitatea, deoarece pentru unele probleme, multe setări distincte pot produce aceleași valori AUC, dar cu valori diferite de importanță a caracteristicilor. Scopul este de a păstra caracteristicile cu importanță mai mare și, de asemenea, de a utiliza valori de importanță fs în timpul variației.

Încrucișare pe baza atributelor Operatorul de încrucișare creează doi descendenți o_1 și o_2 din doi părinți x_1 și x_2 folosind următoarea abordare: pentru fiecare caracteristică (bit), valoarea de la x_1 sau x_2 cu cea mai mare importanță caracteristică este copiată în o_1 și este copiată și în

o_2 dar cu o probabilitate egală cu probabilitatea de încrucișare p_{cross}. Alte caracteristici din o_2 sunt setate aleatoriu, cu o probabilitate de 0,5.

Mutația fsGA-DT folosește mutația bitflip și este aplicată cu rata de mutație pe o_1 și o_2 prin inversarea fiecărui bit cu o probabilitate de mutație.

Selecția pentru supraviețuire Pentru a spori diversitatea, descendenții înlocuiesc toți părinții, în timp ce cel mai bun individ din întreaga căutare, notat cu Ψ_{best}, este păstrat separat.

Rezultate și evaluarea performanței Algoritmul produce cel mai bun $\Psi_{celmaibun}$ individ. Atributele indicate de acesta sunt folosite pentru a antrena un DT și apoi un set de testare este utilizat pentru a evalua performanța DT pe atributele furnizate de fsGA-DT.

Experimente numerice Sunt efectuate experimente numerice pentru a testa performanța fsGA-DT. Efectuăm experimente pe seturi de date generate sintetic cu diferite grade de dificultate și seturi de date din lumea reală. Toate seturile de date reprezintă o problemă de clasificare binară. Comparăm rezultatele obținute de fsGA-DT cu un clasificator Decision Tree care are incorporat o metodă de selecție a atributelor încorporată.

Evaluarea performanței Pentru a evalua performanța fiecărui clasificator pe fiecare problemă de test, datele au fost împărțite în instanțe de antrenare și instanțe de testare. Datele de antrenare constă în 70% din instanțele de date ale problemei de testare (alese aleatoriu), iar restul de 30% de date compun datele de test. Antrenăm modelele pe datele de antrenare și raportăm AUC (aria sub curba ROC) [8, 21] pentru datele de testare.

Sunt efectuate 100 de rulări independente pentru fiecare problemă testată, datele sunt împărțite aleator în date de antrenare și date de test. Seturile de date sunt apoi grupate pe baza unui parametru al setului de date (număr de instanțe, număr de atribute, suprapunere) și parametri ai algoritmului genetic (dimensiunea populației, numărul de generații, probabilitatea de mutație pentru fiecare bit) și sunt generate figuri ECDF (*empirical cumulative distribution function*) pentru a ilustra distribuția valorilor AUC raportate de cele două metode. Graficele arată proporția elementelor care sunt mai mici sau egale cu un anumit punct din grafic; atunci când se compară mai multe curbe pe același figură, cea din dreapta indică o probabilitate mai mică de a lua valori mai mici și poate fi considerată mai bună.

Parametrii folosiți Parametrii utilizați pentru fsGA-DT sunt dimensiunea populației (50, 100), numărul de generații (200, 300), probabilitatea de încrucișare (0.7), probabilitatea de mutație (0.2), probabilitatea de mutare a unui bit (0.03, 0, 05) și dimensiunea turnirului (3, 5)). Pentru Arborele de Decizie, criteriul de împărțire este indexul *gini*, iar adâncimea maximă a arborelui este de 5.

Rezultate experimentale Figura 4 prezintă diagrame ECDF pe seturile de date sintetice pentru diferiți parametri și valorile acestora. Tabelul 1 prezintă rezultate semnificative pentru un test care compară valorile AUC raportate de ambele metode pe cele 100 de rulări independente. Ipoteza nulă testată este că AUC medie raportată de fsGA-DT este mai mică decât cea raportată de DT; respingerea acesteia indică faptul că putem afirma că diferențele de rezultate sunt semnificative, iar rezultatul fsGA-DT poate fi considerat mai bun decât cel obținut de DT. Constatăm că pentru majoritatea combinațiilor de parametri diferențele între rezultate sunt semnificative. Cele mai multe rezultate pozitive sunt obținute pentru $d_3 = 1000$ instanțe și $p_2 = 1$, parametri care se suprapun. Rezultatele raportate pentru seturile de date din lumea reală sunt similare.

Concluzii Este explorată o abordare hibridă a clasificării atributelor. Un algoritm genetic este conceput pentru a dezvolta seturi de atribute evaluate prin utilizarea arborilor de decizie. Deoarece arborii de decizie realizează intrinsec selecția atributelor și, prin urmare, reprezintă

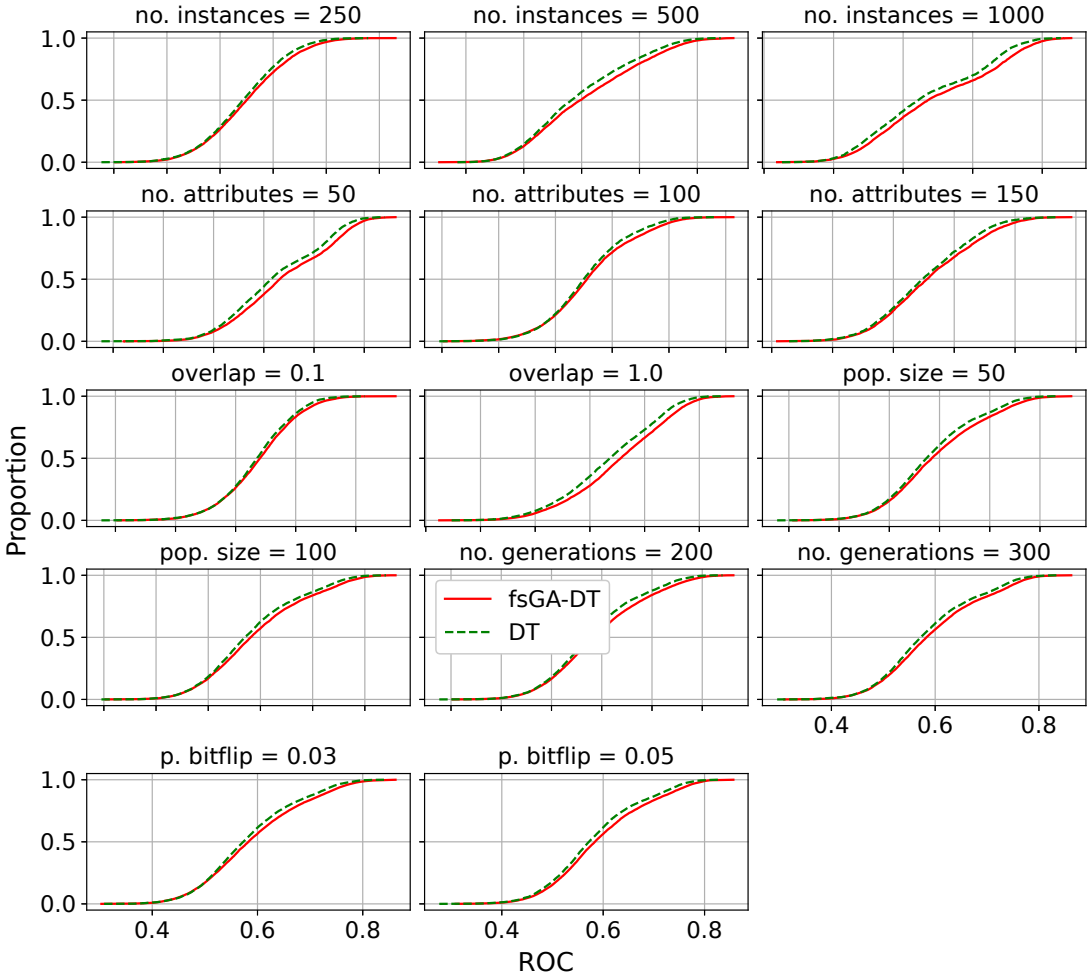

Figura 4: Grafice ECDF pentru rezultatele obținute de fsGA-DT și DT pe seturile de date sintetice pentru diferiți parametri. Rezultatele sunt raportate pentru 100 de rulări independente pe datele de test.

și modele de selecție a atributelor încorporate, ei oferă, de asemenea, o măsură a importanței atributelor care poate fi utilizată pentru a ghida căutarea algoritmului genetic. Astfel, comunicarea dintre populația GA și componenta DT este bidirecțională: atributele sunt trimise către arborii de decizie pentru a fi evaluate, iar DT, pe lângă faptul că oferă o măsură de fitness, indică importanța atributului care este utilizată în selecție și recombinare. Pentru a evita *overfitting*, cel mai bun individ este evaluat prin utilizarea mai multor copaci crescuți pe date împărțite aleator în date de antrenare și de validare. Deși există multe abordări bazate pe algoritmi genetici pentru selecția atributelor, rezultatele arată că există încă loc pentru a explora diferite mecanisme de comunicare și noi operatori genetici care utilizează informații specifice de la clasificatori în timpul evoluției. Problema de selecție a atributelor este una complexă și fiecare pas mic către îmbunătățirea metodelor existente poate adăuga valoare și poate contribui la înțelegerea problemei.

3.3 O abordare evolutivă pentru selecția atributelor și clasificare

În această etapă se introduce o metodă de clasificare bazată pe selecția atributelor, care evoluează ponderile atributelor prin utilizarea unor arbori de decizie bazați pe concepte de echilibru din teoria jocurilor. Indivizii din populația algoritmului evolutiv reprezintă vectori de importanță a

Tabela 1: Comparații statistice între rezultatele obținute de fsGA-DT și DT pe seturile de date din lumea reală (R_1, R_2, și R_3) pentru diferite valori ale parametrilor algoritmului genetic: dimensiunea populației ($p_1 : 50, 100$), numărul de generații ($p_2 : 200, 300$) și probabilitatea de mutație a inversării biților ($p_3 : 0,03, 0,05$). Un simbol ($\triangle$) indică faptul că fsGA-DT raportează rezultate mai bune din punct de vedere statistic, un simbol ($\times$) indică faptul că abordarea DT obține rezultate mai bune din punct de vedere statistic, iar un simbol ($-$) indică nicio diferență semnificativă între rezultate.

p_1	p_2	p_3	R_1	R_2	R_3
50	200	0.03	-	$\times$	-
50	200	0.05	$\triangle$	$\times$	$\triangle$
50	300	0.03	$\triangle$	-	$\triangle$
50	300	0.05	-	$\triangle$	$\triangle$
100	200	0.03	$\triangle$	-	-
100	200	0.05	-	-	-
100	300	0.03	$\triangle$	$\triangle$	-
100	300	0.05	$\triangle$	-	$\triangle$

atributelor, evoluați cu scopul de a identifica cele mai relevante atribute din setul de date pentru problema clasificării. Un arbore de decizie modificat este utilizat în scopuri de clasificare și evaluare. Metoda propusă nu folosește un mecanism de selecție, arborii sunt crescuți împreună și formează un ecosistem în care toți sunt implicați în sarcina de predicție. Un individ încetează să evolueze atunci când nu mai există variații în probabilitățile pe care le oferă pentru selectarea atributelor folosite în inducția arborelui său. Se propune o aplicație practică care analizează clasificarea grupurilor de venituri ale țărilor pe baza indicatorilor de dezvoltare furnizați de *World Bank*, se prezintă o interpretare a abordării selecției atributelor.

Modelul propus - Evolution Strategy Decision Forest (ESDF) ESDF evoluează indivizi care reprezintă ponderile atributelor pentru a-i identifica pe cei care explică cel mai bine datele. Mecanismul strategiei de evoluție, precum și arborii de decizie utilizați pentru clasificare, sunt prezentați în cele ce urmează.

Arborii de decizie sunt unele dintre cele mai populare tehnici de învățare automată [20, 29] datorită eficienței și ușurinței cu care pot fi explicate rezultatele. În cele ce urmează, luăm în considerare problema clasificării binare.

Majoritatea arborilor de decizie sunt construiți de sus în jos, începând cu întregul set de date la rădăcina arborelui. Arbori diferiți împart datele în moduri diferite, utilizând fie hiperplane obținute prin axe paralele, oblice sau neliniare [1, 13, 17], calculând parametrii lor utilizând anumiți indicatori de puritate care evaluează datele sub-nodurilor, de exemplu indicele gini, entropia etc. [30]. La fiecare nivel al arborelui, are loc un proces de optimizare care implică fie parametrii hiperplanului, atributele de utilizat pentru divizare sau ambele.

Se propune utilizarea unui arbore de decizie care calculează parametrii hiperplanului prin aproximarea echilibrului unui joc necooperativ [24]. Echilibrul jocului își propune să găsească parametri astfel încât fiecare nod copil din arbore să "primească" date cât mai pure posibil prin deplasarea instanțelor cu etichete diferite la stânga/dreapta hiperplanului. Astfel, pentru a împărți datele nodului X, Y pe baza unui atribut j, folosim următorul joc necooperativ $\Gamma(X, Y|j)$:

- jucătorii, L și R corespund celor două sub-noduri și, respectiv, celor două clase;
- strategia fiecărui jucător este de a alege un parametru al hiperplan: β_L și, respectiv, β_R;

- câștigul fiecărui jucător este calculat în felul următor:

$$u_L(\beta_L, \beta_R|j) = -n_0 \sum_{i=1}^{n} (\beta_{1|j} x_{ij} + \beta_{0|j})(1 - y_i),$$

și

$$u_R(\beta_L, \beta_R|j) = n_1 \sum_{i=1}^{n} (\beta_{1|j} x_{ij} + \beta_{0|j}) y_i,$$

unde $\beta = \frac{1}{2}(\beta_L + \beta_R)$ și n_0 și n_1 reprezintă numărul de instanțe care au etichete 0 și, respectiv, 1.

Conceptul de echilibru Nash pentru acest joc reprezintă o soluție astfel încât niciunul dintre jucători să nu poată găsi o abatere unilaterală care să-i îmbunătățească câștigul, adică niciunul dintre jucători nu poate schimba mai multe date pentru a obține un câștig mai bun. O aproximare a unui echilibru poate fi obținută folosind o versiune stilizată a jocului fictiv [5] în felul următor. Pentru un număr de iterații (η), cel mai bun răspuns al fiecărui jucător împotriva strategiei celuilalt jucător este calculat folosind un algoritm de optimizare. Deoarece ne propunem doar să aproximăm valorile β care împart în mod rezonabil datele, căutarea se oprește după ce s-au epuizat numărul de iterații. În fiecare iterație, cel mai bun răspuns la media strategiilor celuilalt jucător din cele anterioare este considerat cel fix.

Pentru fiecare atribut $j \in \{1, \ldots, d\}$, datele sunt împărțite folosind algoritmul propus; atributul care este de fapt utilizat pentru a împărți datele este ales pe baza indicelui Gini [30]. Arborele astfel construit împarte datele în acest mod, recursiv, până când datele nodului devin pure (toate instanțele aparțin aceleiași clase) sau a fost atinsă o adâncime maximă a arborelui.

Predicția Un DT furnizează o partiție pentru datele de antrenare. Pentru a prezice eticheta pentru o instanță testată x, este identificată regiunea corespunzătoare a spațiului, adică frunza sa. Decizia se ia pe baza proporției de etichete din frunza respectivă. Fie DT un arbore de decizie bazat pe un set de date $\mathcal{X}$ și x o valoare testată. DT a împărțit $\mathcal{X}$ în date găsite în frunzele sale, notate cu $DT_1, \ldots, DT_m$, unde m este numărul de frunze din DT. Fie $DT(x)$ setul de date corespunzător regiunii frunzei lui x, $DT(x) \subset \mathcal{X}$. De obicei, modelul ar atribui lui x eticheta y cu o probabilitate egală cu proporția elementelor cu clasa y în $DT(x)$.

Importanța atributelor Mecanismul de divizare al arborelui bazat pe joc indică pentru fiecare nod atributul care împarte datele "cel mai bine" și poate fi determinată cu (1).

Evolution Strategy Decision Forest Algoritmul propus evoluează o populație de ponderi a atributelor pentru a identifica importanța acestora pentru problema de clasificare. Indivizii din populația finală indică importanța atributelor, în timp ce, în general, strategia de evoluție efectuează clasificarea folosind ponderile atributelor evoluate.

Codificare Indivizii din populație w sunt codificați ca vectori reali, cu valori pozitive, de lungime d, unde w_j reprezintă importanța atributului j, $j = 1, \ldots, d$.

Inițializare Toți indivizii sunt inițializați cu ponderi egale de $1/d$. ESDF menține o populație de pop_size indivizi.

Evaluare Nu există un mecanism explicit de atribuire a adecvării unui individ în cadrul ESDF. Indivizii evoluează indiferent de performanța lor pe baza informațiilor primite din mediul înconjurător. Motivația din spatele acestei abordări poate fi exprimată în două moduri: pe de o parte, evaluarea ponderilor atributelor poate fi efectuată folosind un algoritm de clasificare bazat pe performanța sa. Cu toate acestea, nu există un indicator de performanță universal acceptat care să poată fi utilizat pentru a compara rezultatele într-un mod fiabil. Pe de altă parte, dintr-un punct de vedere inspirat din natură, o paradigmă de agregare de tipul *forest* nu necesită o

concurență directă pentru resurse. Arborii cresc și se adaptează unul la celălalt. Unii pot înceta să crească din cauza lipsei de resurse, dar nu se înlocuiesc reciproc în fiecare generație. Astfel, toți arborii sunt adăugați pădurii și evaluarea are loc pe întreaga pădure la sfârșitul căutării.

Evoluția Procesul de evoluție are loc iterativ până când se atinge un număr maxim de generații sau până când toți indivizii ajung la maturitate.

Mecanismul de actualizare În fiecare iterație, un eșantion din date este utilizat pentru a induce un arbore de decizie descris mai sus pentru fiecare individ din populație. Atributele pentru fiecare arbore sunt selectate cu o probabilitate proporțională cu ponderile corespunzătoare în individ. Astfel, pentru individul w reprezentând ponderile atributelor, probabilitatea de a selecta j este:

$$P(j|w) = \frac{w_j}{\sum_{k=1}^{d} w_k}. \tag{3}$$

În prima iterație, probabilitățile sunt toate egale. Cu toate acestea, în generațiile următoare, importanța atributelor raportată de fiecare arbore este utilizată pentru a actualiza indivizii corespunzători:

$$w_j \leftarrow w_j + \phi(j) \cdot \alpha, j = 1, \ldots, d, \tag{4}$$

unde $\phi(j)$ reprezintă importanța atributului j raportată de arborele indus utilizând individul w (eq. (1)), iar α este un parametru care controlează magnitudinea actualizării.

În acest fel, ponderile atributelor care sunt considerate importante de către arbore sunt crescute, crescând și probabilitatea ca acestea să fie selectate în următoarele iterații. Deși aparent, acest lucru poate duce la supra adaptarea (*overfitting*) atributelor, faptul că în fiecare iterație, un eșantion diferit din date este utilizat pentru inducerea arborilor, căutarea unui individ se oprește atunci când ajunge la maturitate și, de asemenea, că există mai mulți indivizi menținuți pe aceleași date promovează diversitatea.

Astfel, rolul arborelui este de a atribui importanța atributelor pentru mecanismul de actualizare. În afară de aceasta, fiecare arbore este, de asemenea, conservat și utilizat în continuare în predicția pentru clasificare.

Maturitate Indivizii sunt folosiți pentru a alege atribute pentru antrenament folosind probabilitățile în eq. (3). Scopul căutării este de a găsi o distribuție pe setul de atribute: dacă în mai multe iterații de aplicare a eq. (4) importanța atributelor raportate de arbore nu se modifică semnificativ, abaterea standard a probabilităților $P()$ nu va varia. ESDF consideră că un individ a ajuns la maturitate și încetează să evolueze și să inducă arbori folosindu-l dacă nu există nicio modificare a variației probabilităților corespunzătoare. Următoarea condiție este utilizată pentru a compara evoluția unui individ de la generația t la $t + 1$:

$$\frac{\sigma(P(\cdot|w_t)}{\sigma(P(\cdot|w_{t+1})} < \epsilon \tag{5}$$

unde σ reprezintă abaterea standard și $P(\cdot|w)$ vectorul probabilităților în eq. (3) luat pentru toate atributele j. Dacă condiția 5 este valabilă, individul este considerat a fi ajuns la maturitate și nu mai este actualizat, adică nu mai sunt induși arbori pe baza lui. Nu toți indivizii ajung la maturitate în același timp, ceea ce înseamnă că mărimea populației scade în timpul căutării, reducând complexitatea metodei.

Clasificare Fiecare individ induce mai mulți arbori până când ajunge la maturitate. Toți acești arbori formează o pădure care poate fi folosită pentru a face predicții pentru problema clasificării. Acesta este ultimul pas al algoritmului și poate fi folosit pentru a valida rezultatele. Arborii sunt induși utilizând diferite eșantioane de date și atribute diferite. Selectarea atributelor fără nicio măsură de adecvare poate oferi (sau nu) arbori de clasificare buni. Pentru a evita supra

adaptarea (*overfitting*) sau utilizarea arborilor înșelători, predicția nu se face prin luarea în considerare a etichetelor din frunzele copacilor, ci prin agregarea datelor din frunze corespunzătoare instanțelor testate și prin aplicarea în continuare a algoritmului *logistic regression* (LR) pentru a face predicții. Fiecare arbore oferă o vecinătate pentru instanța testată. Agregarea tuturor acestor regiuni va oferi un set de instanțe relevante, permițând algoritmului să facă o predicție informată.

Schița ESDF ESDF are două etape principale: un pas de evoluție și un pas de predicție.

În timpul etapei de evoluție, o populație de ponderi este actualizată în mai multe iterații până când nu există nicio variație a probabilităților pe care le oferă pentru selectarea atributelor pentru inducția arborelui. Predicția se realizează pentru fiecare instanță testată prin agregarea datelor corespunzătoare frunzelor sale în toți arborii induși și aplicarea algoritmului *logistic regression* asupra setului de date rezultat.

Rezultatul ESDF constă în probabilități de predicție pentru datele testate care pot fi utilizate pentru a evalua întreaga abordare și ponderile medii ale atributelor pe întreaga populație.

Rezultate numerice Experimentele numerice testează și ilustra performanța ESDF, compară rezultatele ESDF cu alte modele de clasificare de ultimă generație.

Parametrii folosiți O strategie *Stratified k-fold Cross-Validation* [10] este utilizată pentru a estima eroarea de predicție așteptată. Setul de date este împărțit în $k = 10$ seturi echilibrate, dintre care nouă sunt utilizate pentru a antrena modelul, iar al zecelea set (set de testare) este utilizată pentru a evalua modelul. Partea de antrenare și partea de testare se repetă $k = 10$ ori, de fiecare dată când se utilizează un set diferit ca set de testare. Repetăm validarea încrucișată k-fold de patru ori, de fiecare dată când se utilizează un număr *seed* diferit pentru a împărți datele (folosim ca *seed* valorile $1, 2, 3, 4$), rezultând 40 de valori ale indicatorului care sunt comparate.

Pentru fiecare test al unui set de date, raportăm indicatori de performanță pe baza cărora comparăm performanța ESDF cu cea a altor clasificatori de ultimă generație. Instruim fiecare clasificator comparat pe aceleași date de tren ca ESDF pentru fiecare set și comparăm rezultatele ESDF cu cele raportate de modelele comparate pe setul de testare.

Valorile de performanță utilizate pentru comparație sunt: AUC (aria de sub curba ROC) [8, 21], scorul F_1 [31], acuratețea ACC și scorul log-loss [10].

Rezultatele ESDF sunt comparate cu alți clasificatori bazați pe arbore de decizie și, deoarece utilizează *Logistic Regression* în etapa de predicție, comparăm rezultatele și cu această metodă. De asemenea, comparăm performanța ESDF cu alte modele de clasificare bine-cunoscute.

Pentru fiecare set de date, fiecare metodă raportează 40 de valori ale indicatorilor de performanță corespunzătoare celor zece seturi generate de patru ori cu generatoare seed cu numere aleatorii diferite. Pentru a evalua diferența dintre rezultate, aceste valori sunt comparate folosind un *paired t-test*, cu ipoteza nulă că rezultatele furnizate de ESDF sunt mai slabe decât cele raportate de cealaltă metodă. Respingerea acestei ipoteze, cu o valoare p mai mică decât 0.05, indică faptul că putem considera rezultatele ESDF semnificativ mai bune decât cele obținute de modelele comparate.

Parametrii testați ESDF sunt: dimensiunea populației 5, numărul maxim de generații 20, adâncimea maximă a arborilor 5 și 10 și valorile α 0.3, 0.8 și 1. Toate experimentele se desfășoară cu o combinație a acestor parametri.

Rezultate În cazul AUC, ESDF obține cele mai bune rezultate pentru toate seturile de date. Pentru indicatorul de acuratețe, ESDF oferă în mod constant rezultate mai bune, iar atunci când sunt disponibile mai multe instanțe în setul de date, clasificatorii care agregă mai mulți clasificatori și logistic regression raportează, pentru câteva seturi de date, rezultate la fel de bune ca cele raportate de ESDF. Acest lucru este valabil și pentru indicatorii F1 și Log-loss.

Selecția atributelor O modalitate posibilă de a evalua eficiența mecanismului de selecție a

Tabela 2: Mean and standard deviation for the AUC and Log-loss indicators in the case of real-world data sets for ESDF and the best performing compared classifier (best M). A (⋆) symbol highlights the ESDF results that can be considered statistically better than the other method.

data set	AUC (ESDF)	AUC (best M)	Log-loss (ESDF)	Log-loss (best M)
R1	0.99±0.03⋆	M0: 0.95±0.06	0.15±0.14⋆	M3: 0.89±0.08
R2	0.83±0.05⋆	M6: 0.73±0.05	0.65±0.17	M5: 0.69±0.06
R3	0.92±0.06⋆	M2: 0.86±0.08	0.56±0.35⋆	M8: 0.72±0.10
R4	1.00±0.00	M0: 1.00±0.00	0.00±0.00⋆	M4: 0.83±0.08
R5	0.88±0.07⋆	M4: 0.84±0.07	0.73±0.53	M7: 0.72±0.07
R6	0.66±0.13⋆	M11: 0.62±0.12	0.94±0.30	M5: 0.53±0.10⋆
R7	0.83±0.16⋆	M3: 0.79±0.16	1.34±1.64	M8: 0.69±0.18⋆
R8	0.92±0.11⋆	M1: 0.82±0.14	0.62±1.05⋆	M3: 0.64±0.15
R9	0.77±0.08⋆	M9: 0.72±0.08	0.66±0.14⋆	M3: 0.70±0.07
R10	1.00±0.01	M8: 0.99±0.02	0.13±0.05⋆	M13: 0.76±0.05
R11	0.63±0.09⋆	M7: 0.56±0.09	0.64±0.22	M7: 0.56±0.09⋆
R12	0.94±0.04⋆	M6: 0.84±0.07	0.39±0.25⋆	M4: 0.80±0.06
R13	0.96±0.03⋆	M9: 0.91±0.05	0.36±0.28⋆	M8: 0.85±0.05
R14	1.00±0.01⋆	M1: 0.98±0.02	0.12±0.20⋆	M7: 0.92±0.04

atributelor este de a calcula indicatorul de stabilitate SC [3, 18] peste cele zece seturi (*folds*). Valorile indicatorului de stabilitate se bazează pe corelații medii: un SC apropiat de 1 indică faptul că metoda de selectare a entităților selectează aceleași entități în mai multe runde pe eșantioane diferite ale setului de date, indicând stabilitatea. Valorile scorului SC la selectarea a jumătate din entități pe baza ponderii lor pentru seturile de date sintetice variază între 0,7 și 1, indicând stabilitatea abordării. Pentru seturile de date cu 20 de atribute, intervalul de încredere pentru SC este (0,97; 0,99), iar pentru cele cu 50 de atribute este (0,88; 0,93).

Tabelul 2 prezintă rezultatele obținute de ESDF în raport cu cel mai bun rezultat raportat de metodele comparate pe seturile de date din lumea reală. Sunt prezentate deviația medie și standard pentru indicatorii AUC și log-loss. Rezultatele mai bune din punct de vedere statistic sunt evidențiate. Se poate observa că ESDF oferă în mod constant rezultate mai bune. Atunci când se compară valorile ASC, se poate observa că ESDF fie oferă rezultate statistic mai bune, fie este indiferent în comparație cu clasificatorul comparat cu cele mai performante.

În ceea ce privește diferitele valori ale parametrilor ESDF, nu am identificat diferențe semnificative între diferitele valori, nici pentru datele sintetice, nici pentru datele de test reale. Valoarea α nu influențează rezultatele, deoarece nu este utilizată direct pentru inducerea arborilor.

Clasificarea țărilor cu venituri mici pe baza indicatorilor de dezvoltare mondială: o aplicație Banca Mondială clasifică țările în patru grupe de venituri: venituri mari, venituri medii superioare, venituri medii inferioare și venituri mici anual, pe baza venitului național brut (VNB) pe cap de locuitor în valori USD, folosind metodologia Atlas [1]. Lista de clasificare pentru 2022 se bazează pe datele din 2021. În 2022, VNB-ul pe cap de locuitor este influențat de factori precum creșterea economică, inflația, cursurile de schimb și creșterea populației. Clasifi-

[1]https://datahelpdesk.worldbank.org, accesată ultima dată în ianuarie 2023

carea se bazează pe intervalele VNB[2]. Banca Mondială oferă, de asemenea, date referitoare la o varietate de alți indicatori. Setul de date al Indicatorului Dezvoltării Mondiale conține informații cu privire la diverși indicatori financiari care pot fi utilizați pentru a explica clasificarea grupului de venituri al unei țări. Pentru a testa această ipoteză, precum și eficiența ESDF pe o aplicație din lumea reală, am folosit aceste date pentru a clasifica țările cu venituri mici (mici și mici-medii) și pentru a identifica caracteristicile din lista indicatorilor de dezvoltare mondială care explică cel mai mult clasificarea.

Prelucrarea datelor Setul de date privind indicatorii de dezvoltare mondială (pentru anul 2021) conține 108 indicatori pentru 218 țări pentru care este atribuită și o categorie de venit. Cu toate acestea, nu toți indicatorii au valori pentru toate țările. Toți indicatorii cu valori pentru mai puțin de jumătate din numărul de țări au fost eliminați, rezultând un set de date cu 218 țări și 40 de indicatori. În plus, eliminarea tuturor țărilor cu mai puțin de jumătate din valorile indicatorilor a dus la un set de date care conține 138 de țări și 40 de indicatori. În acest set de date, am găsit 13,35% valori lipsă care au fost înlocuite, pentru fiecare indicator, cu valoarea medie a regiunii țării sale, care face parte din setul de date. Țărilor cu venituri medii inferioare și inferioare li s-a atribuit eticheta 1, iar celelalte 0, rezultând un set de date ușor dezechilibrat, cu 37% cazuri având clasa 1. În cele ce urmează, vom numi acest set de date setul de date al indicatorilor de venit ai Băncii Mondiale (WBII).

Experimente Aceeași metodologie utilizată pentru testarea ESDF pe baza datelor de test sintetice si reale a fost utilizată și pentru setul de date WBII. Validarea încrucișată de 10 ori a fost aplicată de patru ori cu numere seed diferite pentru generatorul de numere aleatoare, iar cei patru indicatori au fost utilizați pentru a evalua rezultatele. Parametrii ESDF au fost $\alpha = 0.8$, adâncimea maximă a arborelui utilizată a fost 10, iar mărimea populației a fost 5.

Rezultate - clasificare Rezultatele numerice pentru clasificare raportate prin toate metodele pentru setul de date WBII sunt prezentate în tabelul 3. Se observă că rezultatele raportate de ESDF sunt la fel de bune sau chiar mai bune decât cele raportate prin celelalte metode. În special, valorile log-loss sunt semnificativ mai bune decât toate celelalte metode.

Rezultate - selecția atributelor Pentru a ilustra o posibilă interpretare practică a atributelor alese, figura 5 reprezintă ponderile atributelor raportate de ESDF pe cele 10 seturi (*folds*) utilizate pentru validarea încrucișată. Scorul de stabilitate corespunzător este de 0,74, indicând o corelație puternică între caracteristicile selectate (când jumătate dintre ele sunt alese pe baza valorii greutății lor). Atributele cu cele mai mari ponderi sunt:

1. GFDD.AI.11: Received wages: into a financial institution account (% age 15+)
2. GFDD.AI.05: Financial institution account (% age 15+)
3. GFDD.AI.21: Debit card ownership (% age 15+) and GFDD.AI.20: Credit card ownership (% age 15+)
4. GFDD.EI.01: Bank net interest margin (%)
5. GFDD.AI.06: Saved at a financial institution (% age 15+)
6. GFDD.AI.10: Received domestic remittances: through a financial institution (% age 15+)

Această listă indică faptul că activitățile bancare individuale pot fi considerate indicatori pentru grupul de venituri al unei țări. Deși nu există nicio cauzalitate implicată aici, rezultatele indică o relație între acești indicatori și grupul de venituri.

Concluzii Modelul ESDF pentru selectarea și clasificarea atributelor propus prezintă câteva aspecte originale: indivizii sunt evoluați fără o adecvare explicită; un mecanism de actualizare, imitând mutația, crește întotdeauna fiecare componentă; conversia valorilor în probabilități, atunci când este necesar, scade efectul aditiv; căutarea se oprește pentru fiecare individ atunci

[2]https://blogs.worldbank.org, accesată în ianuarie 2023

Tabela 3: WBII data set: mean and standard deviation values for the four indicators reported by all methods. We find results reported by ESDF better or as good as the others for all indicator values.

Method	AUC	ACC	F1	Log-loss
ESDF	0.90 ± 0.08	0.85 ± 0.09	0.79 ± 0.12	0.99 ± 1.28
M0	0.78 ± 0.11	0.80 ± 0.11	0.72 ± 0.16	7.02 ± 3.67
M1	0.85 ± 0.09	0.86 ± 0.08	0.81 ± 0.12	4.77 ± 2.93
M2	0.81 ± 0.09	0.84 ± 0.08	0.75 ± 0.13	5.40 ± 2.71
M3	0.81 ± 0.10	0.83 ± 0.10	0.76 ± 0.16	5.71 ± 3.32
M4	0.78 ± 0.11	0.76 ± 0.11	0.72 ± 0.13	8.19 ± 3.75
M5	0.75 ± 0.12	0.76 ± 0.11	0.67 ± 0.17	8.20 ± 3.97
M6	0.84 ± 0.10	0.85 ± 0.10	0.79 ± 0.13	5.20 ± 3.30
M7	0.81 ± 0.08	0.83 ± 0.08	0.76 ± 0.12	6.06 ± 2.73
M8_5	0.80 ± 0.08	0.82 ± 0.07	0.74 ± 0.11	6.38 ± 2.62
M8_10	0.79 ± 0.09	0.81 ± 0.08	0.72 ± 0.13	6.90 ± 2.91
M9	0.86 ± 0.09	0.87 ± 0.08	0.82 ± 0.11	4.58 ± 2.81
M10_5	0.85 ± 0.10	0.87 ± 0.09	0.81 ± 0.13	4.76 ± 3.12
M11_5	0.87 ± 0.09	0.88 ± 0.08	0.83 ± 0.11	4.37 ± 2.93
M12_5	0.87 ± 0.09	0.88 ± 0.08	0.83 ± 0.11	4.44 ± 2.80
M10_10	0.84 ± 0.09	0.86 ± 0.08	0.79 ± 0.12	5.16 ± 2.82
M11_10	0.86 ± 0.08	0.88 ± 0.08	0.83 ± 0.11	4.44 ± 2.80
M12_10	0.86 ± 0.08	0.88 ± 0.08	0.83 ± 0.11	4.44 ± 2.75
M13	0.81 ± 0.10	0.82 ± 0.10	0.75 ± 0.14	6.41 ± 3.54

când nu mai există variații ale valorilor probabilității; evaluarea are loc la sfârșitul căutării, în timpul fazei de predicție pentru clasificare, când datele sunt colectate din frunzele în care se găsesc instanțe testate de la toți arborii și se aplică *logistic regression* (dar se poate utiliza orice metodă de clasificare) pentru predicție.

3.4 O metodă de tipul *decision forest* bazată pe concepte din teoria jocurilor pentru selecția atributelor

În această secțiune se propune un model *decision forest* bazat pe concepte din teoria jocurilor pentru a aborda problema selecției atributelor relevante. Arborii de decizie care compun pădurea folosesc un mecanism de împărțire a instanțelor în nod bazat pe conceptul de echilibru Nash. O măsură a importanței atributelor este calculată după ce fiecare arbore este construit. Selecția atributelor pentru următorii arbori se bazează pe informațiile furnizate de această măsură. Experimentele numerice ilustrează eficiența abordării. Este prezentat un exemplu real de date care studiază grupurile de venituri ale țărilor și indicatorii de dezvoltare mondială utilizând abordarea propusă.

Game-theoretic Decision Forests - GT-DF GT-DF primește ca intrare parametrii obișnuiți ai unui algoritm *random forest*: setul de date de antrenare $(\mathcal{X}, \mathcal{Y})$, numărul de arbori de decizie K, procentul din numărul de atribute eșantionate pentru împărțirea arborilor p, adâncimea maximă a unui arbore μ și nivelul minim de puritate π. GT-DF utilizează arbori de decizie care împart datele folosind abordarea teoretică a jocului prezentată în subsecțiunea 3.3.

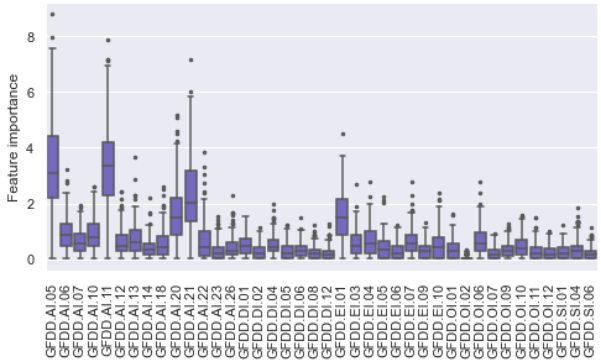

Figura 5: Example of distribution of feature weights for one run on the WBII data set.

GT-DF are două etape principale. În primul pas pădurea crește pe eșantioane de date *bootstrapped* și folosind un mecanism *boosted* îmbunătățit de selecție a atributelor bazat pe importanța ϕ a atributelor atribuite de arborii anteriori. Astfel, după construirea fiecărui arbore k, un vector al ponderilor atributelor $\omega \subset \mathbb{R}^d$ este actualizat folosind valorile $\phi_k(j)$, calculate conform eq. (1). Ponderile ω sunt inițializate cu $1/d$, iar după ce fiecare arbore este adăugat în pădure, valorile lor sunt actualizate prin:

$$\omega_j \leftarrow \omega_j + \phi_k(j), j = 1, \ldots, d. \tag{6}$$

Selecția atributelor pentru arborii ulteriori se face proporțional cu ponderile ω. Astfel, fiecare atribut j este ales cu probabilitatea P_j, cu

$$P_j = \frac{\omega_j}{\omega_1 + \omega_2 + \ldots + \omega_d}. \tag{7}$$

În al doilea pas pentru a face o predicție pentru unele date de testare, GT-DF colectează datele din toate frunzele în care acea instanță s-ar potrivi în toți copacii din pădure, iar predicția se face utilizând *logistic regression* (LR) [10]. În acest sens, GT-DF acționează similar cu metoda celui mai apropiat vecin. O abordare similară poate fi găsită în [25].

Experimentele numerice efectuate pe date de test sintetice și reale ilustrează potențialul abordării, GT-DF alegând atributele importante din seturile de date pentru a obține rezultate bune ale clasificării.

Concluzii Este propusă o pădure de decizie de selecție a atributelor și clasificare bazată pe teoria jocurilor. Pădurea este creată folosind un arbore de decizie bazat pe echilibrul Nash pentru împărțirea datelor nodurilor. Predicția se face prin agregarea datelor din frunzele copacilor care conțin datele testate și aplicând *logistic regression* la acele date locale. O măsură a importanței atributelor este derivată din fiecare arbore și utilizată pentru a construi un vector de ponderi, utilizat în continuare pentru a selecta atributele ulterioare. Vectorul ponderi poate fi folosit pentru a identifica cele mai influente atribute din date.

Performanța metodei este ilustrată prin experimente numerice efectuate pe date sintetice și din lumea reală. Mecanismul de atribuire a importanței atributelor este ilustrat prin utilizarea unei aplicații de date reale în care țările cu venituri mai mici sunt identificate pe baza indicatorilor de dezvoltare de la Banca Mondială. Constatăm că atributele identificate cel mai mult ca fiind importante sunt cele legate de activitatea bancară individuală, de exemplu, procentul de persoane care au un cont bancar, își primesc salariul într-un cont bancar sau dețin un card de credit sau de debit.

3.5 Alte rezultate

3.5.1 Selecția atributelor pe baza corelație

Selecția atributelor pe baza corelație [9] este o metodă de filtrare care are ca scop selectarea seturilor de entități care au o corelație scăzută între ele și o corelație ridicată în raport cu clasa. Acest lucru se realizează prin introducerea meritului $M(s)$ al unui subset s de atribute care au elemente k:

$$M(s) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \tag{8}$$

unde:

- k este dimensiunea setului de atribute s;
- $\overline{r_{cf}}$ este corelația medie între atribut și clasa;
- $\overline{r_{ff}}$ este corelația medie atribut-atribut pentru atributele din setul s.

Un set de atribute cu un merit mai mare este considerat mai bun, se dă un algoritm greedy pentru determinarea setului de atribute cu cel mai mare merit [9].

Cu toate acestea, pot apărea mai multe probleme: pot exista mai multe seturi cu cea mai mare valoare de merit; cea mai mare valoare de merit poate fi obținută prin diferite compromisuri între valorile $\overline{r_{ff}}$ și $\overline{r_{cf}}$. Soluția propusă în cadrul acestui proiect se bazează pe conceptul teoretic de joc utilizat al contribuției marginale a unui jucător la valoarea unui joc/coaliție. Pentru fiecare element/atribut f_i din s contribuția marginală la meritul setului de atribute calculată ca diferența dintre meritul setului de atribute și meritul setului de atribute cu caracteristica f_i eliminată din set:

$$m(s, i) = M(s) - M(s_{-i}), \tag{9}$$

unde $s_{-i} = s \backslash \{f_i\}$.

Suma contribuțiilor marginale ale fiecărui atribut la setul de atribute poate fi considerată ca o funcție alternativă de adecvare care trebuie maximizată, deoarece maximizează meritul setului de atribute s, precum și contribuția fiecărui atribut la meritul general al setului. Dacă un atribut se corelează foarte mult cu alte atribute din set, corelația medie atribut-atribut va crește, scăzând meritul general al setului. Cu toate acestea, corelația medie atribut-atribut a setului la eliminarea acestuia va scădea, ducând, în unele cazuri, la un merit crescut pentru s_{-i}. Astfel, contribuțiile marginale ale atributelor pot contribui la meritul general al setului de atribute, deși mai complexe, pot oferi un compromis mai bun între cele două corelații (clasă-atribute și atribut-atribut).

În aceste situații, *meritul marginal* $MM(s)$ al setului de atribute s este:

$$MM(s) = \sum_{i=1}^{k} m(s, i). \tag{10}$$

$MM(s)$ poate fi utilizat ca o funcție obiectiv pentru orice euristică utilizată. Algoritmii genetici care folosesc codificarea binară sunt adecvați pentru rezolvarea problemei selecției atributelor.

De exemplu, se poate lua în considerare următoarea aplicație bazată pe date reale preluate din baza de date a Băncii Mondiale (secțiunea 3.3). Maximizarea sumei meritului marginal folosind un algoritm genetic standard coduce la identificarea următoarelor atribute în explicarea clasificării ţărilor:

- GFDD.OI.06 5-bank asset concentration

- GFDD.AI.05 Financial institution account (% age 15+)
- GFDD.OI.01 Bank concentration (%)
- GFDD.SI.01 Bank Z-score
- GFDD.OI.11 External loans and deposits of reporting banks vis-à-vis the nonbanking sectors (% of domestic bank deposits)
- GFDD.AI.23 Paid utility bills: using a mobile phone (% age 15+)
- GFDD.AI.13 Saved using a savings club or a person outside the family (% age 15+)

Pentru aceste date, folosind toate atributele, un SVM [10] raportează folosind 10-fold cross-validation 79.09% acurateţe. Folosind metoda greedy bazată pe merit [9] acurateţea creşte la 87.69%. Algoritmul genetic oferă o acurateţe de 89.84%.

3.5.2 Selecție de atribute folosind programarea genetică

Programarea genetică, ca și subdomeniu al inteligenței computaționale, se referă la generarea și evoluarea de expresii sau programe particularizate pentru anumite probleme [7]. Pentru problemele de clasificare, programarea genetică oferă o modalitate intrinsecă de filtrare de atribute, alegând pentru construcția modelului de clasificare doar atributele care conduc la optimizarea funcției obiectiv. În cadrul proiectului s-a propus folosirea unei funcții obiectiv derivată din teoria jocurilor, care să genereze o expresie de separare a două clase pentru optimizare binară în modul următor:

$$F(X,Y) = |\{(x,y) \in (X,Y) | GP(x)y - GP(x)(1-y) \leq 0\}|, \tag{11}$$

where $GP(x)$ is the expression evolved by the genetic programming algorithm to minimize the function F for the training data set (X,Y). Predictions are made by assigning label 0 to instances x for which $GP(x) \leq 0$ and 1 otherwise. The values of $GP()$ can be converted in probabilities in various manners (e.g. passing them through the sigmoid function). Exemplu de expresie evaluată pe tabelul *Iris*, convertit pentru clasificare binara, în format LISP:

```
GP(x) : (+)(x2, (-)(x4, (-)((*)(x4, 0.5779776453361087), (-)
    ((*)(2, x2), (-)((*)((*)(x4, (-)((*)(x4, x3), x1)), (*)(x3
    , x4)), x1))))).
```

unde `x1-4` reprezintă atributele lui `x`. Abordarea aceasta conduce la o acuratețe medie de 0.95 (folosind 10-fold cross-validation) , similară cu cea raportată de alte metode standard de clasificare folosite pe aceleași date (e.g. 0.94 pentru un random forest). De asemenea, atributele indicate de algoritmul de programare genetică arată că în acest caz toate contribuie la modelul de clasificare, ceea ce este de asemenea un rezultat important. Testarea modelului pe date generate sintetic cu număr mare de atribute arată ca expresia generate folosind funcția obiectiv din (11) selectează atributele relevante. Împărțirea lui F în doi termeni corespunzând celor două clase conduce la construcția unui joc necooperativ în care fiecare clasă își selectează atributele relevante. O negociere între cei doi jucători conduce la filtrarea globală. La momentul redactării raportului se efectuează experimente numeric extinse pentru validarea abordării. Implementarea a fost făcută în limbajul de programare *Julia* pentru o mai bună eficiență în timp și spațiu, implementările anterioare suferind de ineficiența limbajului *python*.

3.5.3 Modele bazate pe teoria jocurilor și analiza de rețele pentru clasificare cu clase multiple

În problemele de clasificare nesupervizată cu clase multiple provocarea este de asemenea de a identifica atributele care conduc la o separare a datelor, în lipsa informației legate de etichete.

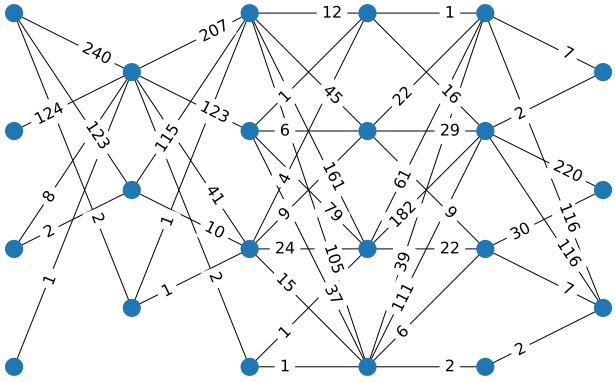

Figura 6: Exemplu de rețea construită dintr-un set de date cu 500 de instanțe, 6 atribute, 4 clase.

Se propune convertirea setului de date de d atribute, pentru care se caută k clase, într-o rețea multipartită cu d straturi și maxim $k \times d$ noduri în felul următor: fiecare atribut corespunde unul nivel al rețelei; pentru fiecare nivel datele corespunzătoare se împart în k intervale egale, respectiv k noduri; fiecare nod se populează cu instanțele care aparțin intervalului respectiv; nodurile care conțin aceeași instanță sunt unite și ponderea legăturii este crescută cu 1 (Exemplu în Figura 6). Dintr-o astfel de rețea se pot *citi* clusterele de date în diferite moduri. Mai mult, atributele importante pot fi identificate ca reprezentând nivelele cu cel mai mic grad (neponderat), respectiv cele cu grad mai ridicat nu pot contribui la clasificare deoarece probabil datele din acele niveluri sunt uniform distribuite. Ca și exemplu numeric, eliminarea nivelelor cu gradul cel mai mare a nodurilor conduce la o creștere a NMI (normalized mutual indicator, indicator de evaluare a performanței metodelor de clustering) de la o medie de 0.88 la 0.97 pentru o serie de algoritmi recunoscuți (Kmeans, Gaussian Mixture, Affinity Propagation, si Birch). Aceste rezultate arată că direcția aleasă în cadrul proiectului poate fi explorată în continuare, oferind o multitudine de posibilități de aplicare.

Referințe

[1] Satyabrata Aich, Kim Younga, Kueh Lee Hui, Ahmed Abdulhakim Al-Absi, and Mangal Sain. A nonlinear decision tree based classification approach to predict the parkinson's disease using different feature sets of voice data. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 638–642, 2018.

[2] Faramarz Bagherzadeh, Mohamad-Javad Mehrani, Milad Basirifard, and Javad Roostaei. Comparative study on total nitrogen prediction in wastewater treatment plant and effect of various feature selection methods on machine learning algorithms performance. *Journal of Water Process Engineering*, 41:102033, 2021.

[3] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143:106839, 2020.

[4] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[5] George W Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

[6] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[7] B de la Iglesia. Evolutionary computation for feature selection in classification problems. *WILEY INTERDISCIPLINARY REVIEWS-DATA MINING AND KNOWLEDGE DISCOVERY*, 3(6):381–407, November 2013.

[8] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[9] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[11] Nazrul Hoque, Mihir Singh, and Dhruba K. Bhattacharyya. EFS-MI: an ensemble feature selection method for classification. *Complex & Intelligent Systems*, 4(2):105–118, June 2018.

[12] Mia Huljanah, Zuherman Rustam, Suarsih Utama, and Titin Siswantining. Feature Selection using Random Forest Classifier for Predicting Prostate Cancer. *IOP Conference Series: Materials Science and Engineering*, 546(5):052031, June 2019. Publisher: IOP Publishing.

[13] Ozan Irsoy, Olcay Taner Yıldız, and Ethem Alpaydın. Soft decision trees. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 1819–1822. IEEE, 2012.

[14] Shivani Jain and Anju Saha. Rank-based univariate feature selection methods on machine learning classifiers for code smell detection. *Evolutionary Intelligence*, 15(1):609–638, March 2022.

[15] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.

[16] Cosmin Lazar, Jonatan Taminau, Stijn Meganck, David Steenhoff, Alain Coletta, Colin Molter, Virginie de Schaetzen, Robin Duque, Hugues Bersini, and Ann Nowe. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(4):1106–1119, 2012.

[17] Sreerama K Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of artificial intelligence research*, 2:1–32, 1994.

[18] Sarah Nogueira and Gavin Brown. Measuring the stability of feature selection. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 442–457, Cham, 2016. Springer International Publishing.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn - machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1986.

[21] Saharon Rosset. Model selection via the auc. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 89, New York, NY, USA, 2004. Association for Computing Machinery.

[22] Mukesh Saraswat and K. V. Arya. Feature selection and classification of leukocytes using random forest. *Medical & Biological Engineering & Computing*, 52(12):1041–1052, December 2014.

[23] Shina Sheen and R. Rajesh. Network intrusion detection using feature selection and decision tree classifier. In *TENCON 2008 - 2008 IEEE Region 10 Conference*, pages 1–4, 2008.

[24] Mihai Suciu and Rodica Ioana Lung. A new filter feature selection method based on a game theoretic decision tree. In Ajith Abraham, Tzung-Pei Hong, Ketan Kotecha, Kun Ma, Pooja Manghirmalani Mishra, and Niketa Gandhi, editors, *Hybrid Intelligent Systems*, pages 556–565, Cham, 2023. Springer Nature Switzerland.

[25] Mihai-Alexandru Suciu and Rodica Ioana Lung. A new game theoretic based random forest for binary classification. In Pablo et al. García Bringas, editor, *Hybrid Artificial Intelligent Systems*, pages 123–132, Cham, 2022. Springer International Publishing.

[26] Chih-Fong Tsai and Yu-Chieh Hsiao. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, 50(1):258–269, 2010.

[27] Paul E. Utgoff. Incremental Induction of Decision Trees. *Machine Learning*, 4(2):161–186, November 1989.

[28] Suhang Wang, Jiliang Tang, and Huan Liu. Embedded Unsupervised Feature Selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), February 2015.

[29] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, January 2008.

[30] Mohammed J. Zaki and Wagner Meira, Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press, 2 edition, 2020.

[31] A.P. Zijdenbos, B.M. Dawant, R.A. Margolin, and A.C. Palmer. Morphometric analysis of white matter lesions in mr images: method and validation. *IEEE Transactions on Medical Imaging*, 13(4):716–724, 1994.

[32] Ozan İrsoy, Olcay Taner Yıldız, and Ethem Alpaydın. Soft decision trees. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1819–1822, 2012.

Director Proiect,
Conf. univ. dr. Mihai Suciu