Raport științific

1 Sumar

Obiective și rezultate estimate:

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

Livrabile (propuse):

- 2 articole trimise spre publicare
- raport de cercetare
- pagina web a proiectului: `https://www.cs.ubbcluj.ro/~mihai-suciu/cgtfe/`

Rezultate obținute: În cursul acestei etape s-au atins toate obiectivele. Au trimise şi acceptate spre publicare trei articole iar alte două sunt în curs de pregătire. Astfel:

- În lucrarea [P1] 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 [P2] 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 [P3] se propune un model *decision forest* bazat pe echilibrul Nash pentru a selecta atributele relevante în problema clasificării.

Lista de articole trimise spre publicare (acceptate):

P1 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)

P2 Rodica Ioana Lung, Mihai Suciu. An Evolutionary Approach to Feature Selection and Classification. LOD 2023. (articol acceptat şi prezentat în cadrul conferinţei)

P3 Mihai Suciu, Rodica Ioana Lung. A game theoretic decision forest for feature selection and classificatio. Logic Journal of the IGPL. (articol acceptat)

2 Rezumat executiv

Direcția principală 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.

O primă abordare constă în combinarea arborilor de decizie cu modele din teoria jocurilor pentru a rezolva problema selecției atributelor relevante pentru problema de clasificare. Problema selecției atributelor aplicată în cadrul problemei se clasificare reduce complexitatea de calcul a estimării parametrilor, dar adaugă și o contribuție importantă la aspectele de înțelegere și explicare a rezultatelor.

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.

3 Descrierea științifică

Principalele rezultate trimise spre publicare și accetate sunt prezentate mai jos, obiectivele etapei II fiind îndeplinite.

3.1 O abordare evolutivă pentru selecția atributelor și clasificare

3.1.1 Introducere

Algoritmii evolutivi (*Evoluitonary Algorithms* - EA) au fost utilizați pe scară largă în problema selecției și clasificării atributelor [6, 25, 29], deoarece sunt flexibili și adaptabili la diferite medii de optimizare. Algoritmii genetici (GA) au fost prima alegere naturală, deoarece reprezentarea binară poate fi utilizată pentru această problemă în mod natural [9]. Multe exemple de algoritmi genetici sunt menționate în [29], și, în special, combinația cu arborii de decizie (DT) a fost atrăgătoare de la început [2], urmând multe variante, extinzându-se la algoritmi *random forest* [11] sau optimizare multi-obiectiv [28]. Exemple de EA pentru selecția atributelor și arborii decizionali pot fi găsite în [14, 15], aplicații în detectarea intruziunilor în rețea [22], chimie [11], recunoașterea vorbirii [16], etc.

Cu toate acestea, eficiența oricărei abordări evolutive depinde de alegerea adecvată a parametrilor și de mecanismele de evaluare a adecvării soluției. În cadrul EA, selecția este în principal responsabilă pentru ghidarea căutării, deoarece supraviețuirea indivizilor nou creați se bazează în cele din urmă pe valoarea funcției de adecvare (*fitness function*). Atunci când adecvarea este asociată cu rezultatele sarcinilor de clasificare și se bazează pe un anumit indicator de performanță (raportat pe eșantioane de date), se constată o variabilitate ridicată între diferite eșantioane, ceea ce face ca compararea rezultatelor să fie irelevantă în scopul selecției atributelor.

Paradigma luptei pentru supraviețuire este de obicei implementată în cadrul EA prin compararea indivizilor folosind valorile lor pentru funcția de adecvare și prin a decide, în funcție de mecanismul de selecție utilizat, care indivizi sunt păstrați și care sunt eliminați din procesul de evoluție. Se consideră că indivizii concurează pentru resurse, iar cei mai adaptați vor supraviețui, pentru a le accesa, exploata și explora în continuare.

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 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.

3.1.2 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.

Arbori de decizie și *random forests* Arborii de decizie sunt unele dintre cele mai populare tehnici de învățare automată [20, 27] datorită eficienței și ușurinței cu care pot fi explicate rezultatele. Pentru problema clasificării ei împart recursiv spațiul de date în regiuni separate, cu scopul de a găsi zone cât mai pure posibil. Arborii mari tind să se potrivească prea mult pe datele pe care au fost antrenați, iar arborii mici s-ar putea să nu împartă suficient datele. O modalitate de a depăși aceste dezavantaje este de a utiliza ansambluri de arbori, de exemplu sub forma de *random forest* [4], și de a agrega rezultatele obținute într-o anumită formă. Arborii de decizie pot fi, de asemenea, utilizați pentru a evalua importanța atributelor pe baza structurii arborelui și a purității datelor divizate în fiecare nod [26].

În cele ce urmează, luăm în considerare problema clasificării binare: având un set de date $\mathcal{X} \subset \mathbb{R}^{N \times d}$ care conține N instanțe $x_i \in \mathbb{R}^d$, $i = 1, \ldots, N$ și $\mathcal{Y}$ etichetele corespunzătoare, cu $y_i \in \{0, 1\}$ eticheta x_i, scopul este de a găsi o regulă care să prezică cel mai bine etichetele $\hat{y}$ pentru instanțele x care provin din aceeași distribuție ca $\mathcal{X}$.

Arbori de decizie bazați pe conceptul de echilibru Majoritatea arborilor decizionali 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 propunem utilizarea unui arbore de decizie care calculează parametrii hiperplanului prin aproximarea echilibrului unui joc necooperativ [23]. 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

Algorithm 1 Aproximarea echilibrului Nash

Input: X, Y - data to be split by the node; j - attribute evaluated
Output: $X_{L|j}, y_{L|j}, X_{R|j}, y_{R|j}$, and β_j to define the split rule for the node based on attribute j;
Initialize β_L, β_R at random (standard normal distribution)
for η iterations: **do**
  Find $\beta_L = \underset{b}{\operatorname{argmin}}\, u_L(b, \beta_R)$;
  Find $\beta_R = \underset{b}{\operatorname{argmin}}\, u_R(\beta_L, b)$;
end for
$\beta_j = \frac{1}{2}(\beta_L + \beta_R)$
$X_{L|j} = \{x \in X | x_j^T \beta \leq 0\}$, $y_{L|j} = \{y_i \in y | x_i \in X_{L|j}\}$
$X_{R|j} = \{x \in X | x_j^T \beta > 0\}$, $y_R = \{y_i \in y | x_i \in X_{R|j}\}$

epuizat numărul de iterații. În fiecare iterație, cel mai bun răspuns la media strategiilor celuilalt jucător din cele anterioare este considerat cel fix. Procedura este prezentată în algoritmul 1.

Pentru fiecare atribut $j \in \{1, \ldots, d\}$, datele sunt împărțite folosind algoritmul 1; 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". Este rezonabil să presupunem că poziția nodului în arbore (nivelul acestuia) indică și importanța atributului în clasificarea datelor și o măsură a importanței poate fi derivată pe baza structurii arborelui. Astfel, pentru fiecare entitate $j \in \{1 \ldots d\}$ notăm cu $v_j = \{v_{jl}\}_{l \in I_j}$, setul care conține nodurile care împart datele pe baza atributului j, cu I_j mulțimea indicilor corespunzători din arbore și fie $\delta(v_{jl})$ adâncimea nodului v_{jl} în arbore, cu valori începând de la unu la nodul rădăcină. Importanța $\phi(j)$ a atributului j poate fi calculată ca:

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

Formula (1) se bazează pe presupunerea că atributele care divizează datele la primele niveluri ale arborelui pot fi mai influente. De asemenea, aparițiile multiple ale unui atribut în noduri cu adâncimi mai mari pot indica importanța acestuia și sunt numărate de $\phi()$.

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:

$$P(j|w) = \frac{w_j}{\sum_{k=1}^{d} w_k}. \tag{2}$$

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:

$$w_j \leftarrow w_j + \phi(j) \cdot \alpha, j = 1, \ldots, d, \tag{3}$$

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. (2). Scopul căutării este de a găsi o distribuție pe setul de atribute: dacă în mai multe iterații de aplicare a eq. (3) 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 $t + 1$:

$$\frac{\sigma(P(\cdot|w_t)}{\sigma(P(\cdot|w_{t+1})} < \epsilon \tag{4}$$

unde σ reprezintă abaterea standard și $P(\cdot|w)$ vectorul probabilităților în eq. (2) luat pentru toate atributele j. Dacă condiția 4 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 (Algoritmul 2, linia 6) și un pas de predicție (Algoritmul 2, linia 17).

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.

3.1.3 Rezultate numerice

Experimentele numerice testează și ilustra performanța ESDF, compară rezultatele ESDF cu alte modele de clasificare de ultimă generație. Această secțiune este împărțită în două părți principale: prima prezintă rezultatele obținute pe date sintetice și reale cu diferite grade de dificultate utilizate pentru clasificare, iar a doua parte este o aplicație de date reale care implică clasificarea grupurilor de venituri ale țărilor.

Date de test sintetice și din lumea reală Pentru datele de test generate sintetic, pentru a asigura reproductibilitatea și a controla dificultatea setului de date rezultat, folosim funcția `make_classification` din biblioteca `scikit-learn`[1] din biblioteca Python [19]. Gradul de dificultate este controlat de parametrii funcției generatoare: numărul de instanțe, numărul de atribute / entități, gradul de suprapunere între instanțele din diferite clase, numărul *seed* utilizat pentru a genera datele de test și dezechilibrul clasei. Pentru experimentele noastre, folosim următoarele: numărul de instanțe $(100, 200, 500, 1000, 2500)$, numărul de atribute $(20, 50)$, numărul *seed* folosit folosit pentru a genera datele (500), gradul de suprapunere între instanțe de clase diferite $(0.1, 0.5)$ și toate seturile de date generate sunt echilibrate. Generăm seturi de date de test pentru toate combinațiile parametrilor de mai sus. Pentru a evalua mecanismul de selecție a atributelor, doar jumătate din entitățile din fiecare set de date sunt generate utilizând funcția `make_classification`, iar cealaltă jumătate aste generată aleator urmând o distribuție uniformă.

Pentru datele de test din lumea reală, folosim următoarele seturi de date din baza de date UCI Machine Learning Repository [7]: setul de date iris (R1) din care am eliminat instanțele *setosa* pentru a obține o problemă liniară de clasificare binară neseparabilă, Pima Indians Diabetes

[1] version `1.1.1`

Algorithm 2 ES-DF: Evolution Strategy Decision Forest

input: training set $\mathcal{X}, \mathcal{Y}$,
parameters:: - pop_size - population size;
- p - the proportion of attributes used for a tree;
- μ - maximum tree depth;
- $MaxGen$ - maximum number of generations;
output: predictions C for a (test) set T; Feature weights ω;
t=0;
Initialize population W_0 with $w_{0,ij} = 1/d, i = 1, \ldots, pop\_size, j = 1, \ldots, d$;
Step 1: Evolution
while $t < MaxGen$ **or not** all trees have reached maturity **do**
  $X_t \leftarrow$ sample of size N with replacement from $\mathcal{X}$;
  for each individual w_t **do**
    $\overline{X}_{t,w} \leftarrow$ sample proportion p of attributes from X_t using probabilities P in eq. (2);
    $DT_{t,w} \leftarrow$ game based decision tree based on $\overline{X}_{t,w}, \mu$;
    Update w_{t+1} using eq. (3);
    Check maturity using condition (4); if (4) holds, mark individual as mature and stop its update;
  end for
  $t \leftarrow t + 1$;
end while
Step 2: Prediction
for each $x_t \in T$ **do**
  $RF(x_t) = \cup_{w,t} DT(x_t)$;
  Fit LR on $RF(x_t)$;
  Assign c_t to x_t - probability that x_t has class 1, based on LR;
end for
return
- $C = (c_1, c_2, \ldots, c(x_{|T|^1}))$
- ω - average feature weights over the entire population.

[1] $|\cdot|$ denotes the cardinality of a set.

(R2), Connectionist Bench (Sonar, Mines vs. Rocks) (R3), acute inflamations (R4), heart disease (R5), Somerville Happiness Survey (R6), appendicitis (R7), blogger (R8), bupa (R9), monks (R10), thoracic-surgery(R11), vertebra-column-2c (R12), wholesale-channel (R13), și setul wdbc (R14).

Parametrii folosiți O strategie *Stratified k-fold Cross-Validation* [12] 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 [12].

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 clasificatoare bine-cunoscute. Lista clasificatorilor comparați este: M0 - Support Vector Machine cu linear kernel, M1 - Support Vector Machine cu a radial kernel, M2 - $k-$nearest-neighbour classifier $k = 3$, M3 - AdaBoost classifier, M4 - Gaussian Naive Bayes, M5 - stochastic gradient descent, M6 - Gaussian process classification, M7 - decision tree classifier which splits nodes until its leaves contain only instances of one class, M8 - a decision tree cu adâncimea maximă ca și ESDF, M9 - a random forest classifier pentru care fiecare estimator împarte nodurile până când frunzele sunt pure, M10 - a random forest classifier cu 10 astimatori, M11 - a random forest classifier cu 50 astimatori, M12 - a random forest classifier cu 100 estimatori (for M10, M11 and M12 each estimator has a maximum depth equal to ESDF), and M13 - logistic regression classifier. Pentru reproductibilitate și control, folosim implementarea lor din biblioteca software *scikit-learn* [19].

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 Figurile 1 și 2 prezintă indicatorii de performanță obținuți de toate metodele pe seturile de date sintetice. Figura 1 prezintă indicatorii AUC și acuratețe, iar figura 2 indicatorii F1 și log-loss. Un pătrat din harta arată valoarea obținută de un clasificator pentru un anumit set de parametri utilizați pentru a genera setul de date (număr instanțe - $100, 200, 500, 1000, 2500$, număr atribute - $20, 50$ și grad de suprapunere - $0.1, 0.5$). Fiecare rând prezintă rezultatele obținute de un clasificator diferit. Pentru abordarea noastră, raportăm, de asemenea, numărul de cazuri în care ESDF obține rezultate semnificativ mai bune decât metodele comparate conform unui test t. 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 (Figura 2).

Selecția atributelor O modalitate posibilă de a evalua eficiența mecanismului de selecție a 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 1 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

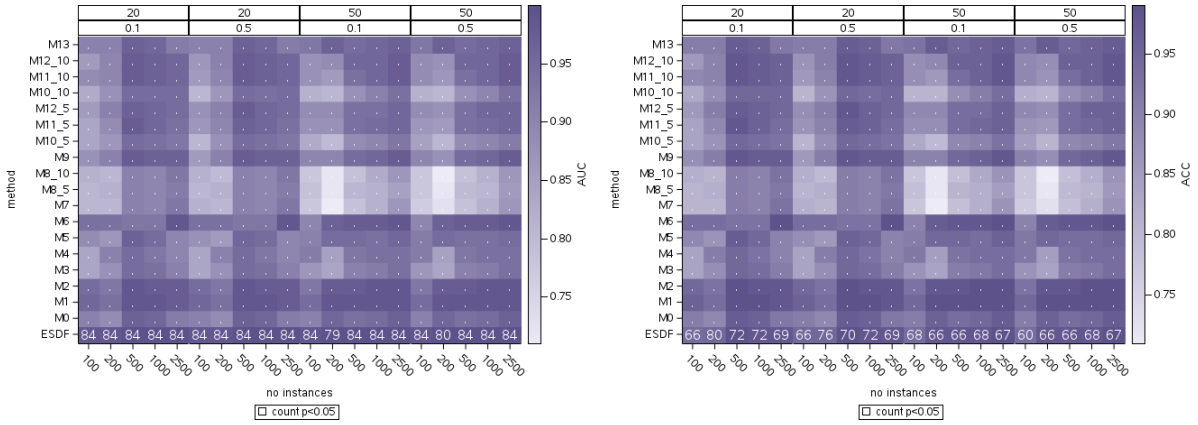

Figura 1: Heatmaps for the AUC and ACC indicators for the synthetic data sets. A higher (darker) value is desirable. The last line also presents the number of times ESDF results were significantly better than the other method based on the p values of the t test comparing results reported for all folds (out of 120). The first line of the heading indicates the number of attributes and the second line the class separator for the data sets.

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 [2]. 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. Clasificarea se bazează pe intervalele VNB[3]. 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

[2] https://datahelpdesk.worldbank.org/knowledgebase/articles/378832-what-is-the-world-bank-atlas-me accesată ultima dată în ianuarie 2023

[3] https://blogs.worldbank.org/opendata/new-world-bank-country-classifications-income-level-2022-20 accesată în ianuarie 2023

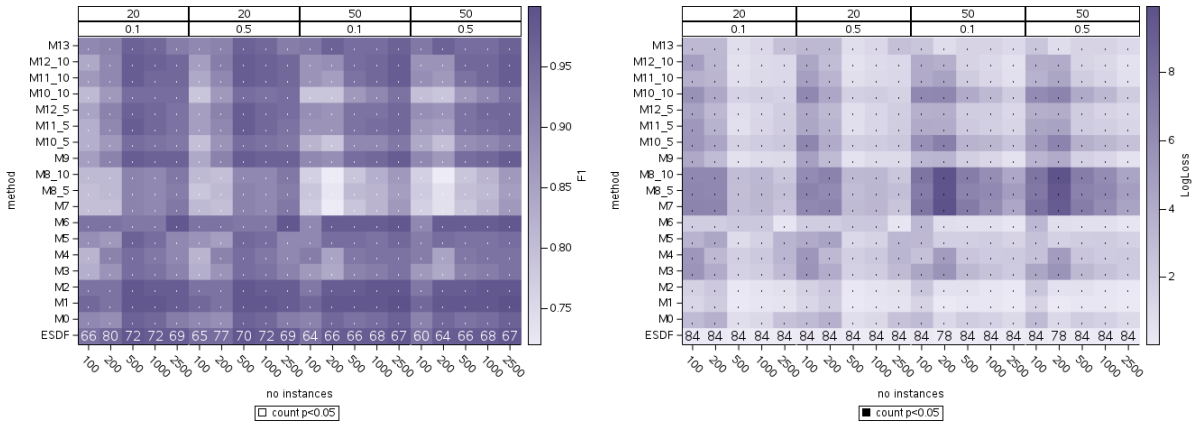

Figura 2: Heatmaps for the F1 and LogLoss indicators for the synthetic data sets. For F1, a higher (darker) value is desirable, while for the LogLoss a smaller (lighter) value is better. The last line also presents the number of times ESDF results were significantly better than the other method based on the p values of the t test comparing results reported for all folds (out of 120). The first line in the heading indicates the number of attributes and the second line the class separator for the data sets.

Tabela 1: 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

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 2. 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.

Tabela 2: 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

Rezultate - selecția atributelor Pentru a ilustra o posibilă interpretare practică a atributelor alese, figura 3 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.

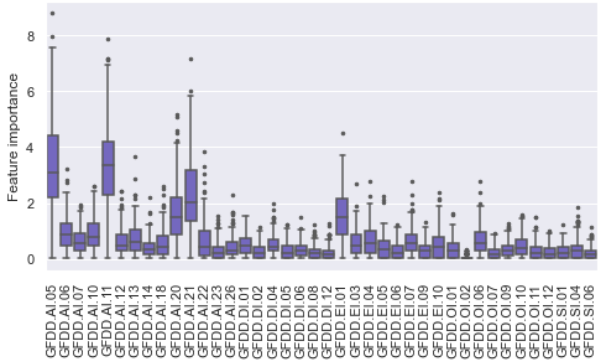

Figura 3: Example of distribution of feature weights for one run on the WBII data set.

3.1.4 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 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.

3.2 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.

3.2.1 Game-theoretic Decision Forests - GT-DF

Schița metodei GT-DF este prezentată în algoritmul 3. 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 3.1.2.

GT-DF are două etape principale. În primul pas (Algorithm 3, linia 4), 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:

$$\omega_j \leftarrow \omega_j + \phi_k(j), j = 1, \ldots, d. \tag{5}$$

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

$$P_j = \frac{\omega_j}{\omega_1 + \omega_2 + \ldots + \omega_d}. \tag{6}$$

Algorithm 3 GT-DF: Game-theoretic decision forest

1: **input**: training set $\mathcal{X}, \mathcal{Y}$, K - number of estimators (trees), p - proportion of attributes used for a tree, μ - maximum tree depth;
2: **output**: predictions C for a (test) set T; Feature weights ω;
3: Initialize $\omega_j = 1/d$, $j = 1, \ldots, d$;
4: *Step 1:*
5: **for** k in $1 : K$ **do**
6: $D_k \leftarrow$ sample of size N with replacement from $\mathcal{X}$;
7: $\overline{D}_k \leftarrow$ sample proportion p of attributes from D_k using probabilities P_j in eq. (6);
8: $DT_k \leftarrow$ Decision tree based on $\overline{D}_k, p, \mu$;
9: Update ω: $\omega_j \leftarrow \omega_j + \phi_k(j), j = 1, \ldots, d.$;
10: **end for**
11: *Step 2:*
12: **for each** $x_t \in T$ **do**
13: $RF(x_t) = \cup_{k=1}^{K} DT_k(x_t)$;
14: Fit LR on $RF(x_t)$;
15: Assign c_t to x_t - probability that x_t has class 1, based on LR;
16: **end for**
17: **return** $C = (c_1, c_2, \ldots, c(x_{|T|^1}))$ and ω - feature weights.

[1] $|\cdot|$ denotes the cardinality of a set.

În al doilea pas (Algorithm 3, linia 11), 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) [12]. În acest sens, GT-DF acționează similar cu metoda celui mai apropiat vecin. O abordare similară poate fi găsită în [24].

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.

3.2.2 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.

3.3 Selecția atributelor pe baza corelație

Selecția atributelor pe baza corelație [10] 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:

$$M(s) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \tag{7}$$

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 [10].

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:

$$m(s, i) = M(s) - M(s_{-i}), \tag{8}$$

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:

$$MM(s) = \sum_{i=1}^{k} m(s, i). \tag{9}$$

$MM(s)$ 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 3.1.3). 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 [12] raportează folosind 10-fold cross-validation 79.09% acurateţe. Folosind metoda greedy bazată pe merit [10] acurateţea creşte la 87.69%. Algoritmul genetic oferă o acurateţe de 89.84%.

Referințe

[1] Aich, S., Younga, K., Hui, K. L., Al-Absi, A. A., and Sain, M. 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)* (2018), pp. 638–642.

[2] Bala, J., Huang, J., Vafaie, H., Dejong, K., and Wechsler, H. Hybrid learning using genetic algorithms and decision trees for pattern classification. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1* (San Francisco, CA, USA, 1995), IJCAI'95, Morgan Kaufmann Publishers Inc., p. 719–724.

[3] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., and Lang, M. Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis 143* (2020), 106839.

[4] Breiman, L. Random Forests. *Machine Learning 45*, 1 (Oct. 2001), 5–32.

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

[6] Cai, J., Luo, J., Wang, S., and Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing 300* (2018), 70–79.

[7] Dua, D., and Graff, C. UCI machine learning repository, 2017.

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

[9] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., USA, 1989.

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

[11] Hansen, L., Lee, E. A., Hestir, K., Williams, L. T., and Farrelly, D. Controlling feature selection in random forests of decision trees using a genetic algorithm: Classification of class i mhc peptides. *Combinatorial Chemistry & High Throughput Screening 12*, 5 (2009), 514–519.

[12] Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference and prediction*, 2 ed. Springer, 2009.

[13] Irsoy, O., Yildiz, O. T., and Alpaydin, E. Soft decision trees. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)* (2012), IEEE, pp. 1819–1822.

[14] Jovanovic, M., Delibasic, B., Vukicevic, M., Suknović, M., and Martic, M. Evolutionary approach for automated component-based decision tree algorithm design. *Intelligent Data Analysis* (2014).

[15] Kretowski, M., and Grześ, M. Evolutionary learning of linear trees with embedded feature selection. In *Artificial Intelligence and Soft Computing – ICAISC 2006* (Berlin, Heidelberg, 2006), L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, Eds., Springer Berlin Heidelberg, pp. 400–409.

[16] Mao, Q., Wang, X., and Zhan, Y. Speech emotion recognition method based on improved decision tree and layered feature selection. *International Journal of Humanoid Robotics* (2010).

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

[18] Nogueira, S., and Brown, G. Measuring the stability of feature selection. In *Machine Learning and Knowledge Discovery in Databases* (Cham, 2016), P. Frasconi, N. Landwehr, G. Manco, and J. Vreeken, Eds., Springer International Publishing, pp. 442–457.

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

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

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

[22] Stein, G., Chen, B., Wu, A. S., and Hua, K. A. Decision tree classifier for network intrusion detection with ga-based feature selection. In *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 2* (New York, NY, USA, 2005), ACM-SE 43, Association for Computing Machinery, p. 136–141.

[23] Suciu, M., and Lung, R. I. A new filter feature selection method based on a game theoretic decision tree. In *Hybrid Intelligent Systems* (Cham, 2023), A. Abraham, T.-P. Hong, K. Kotecha, K. Ma, P. Manghirmalani Mishra, and N. Gandhi, Eds., Springer Nature Switzerland, pp. 556–565.

[24] Suciu, M.-A., and Lung, R. I. A new game theoretic based random forest for binary classification. In *Hybrid Artificial Intelligent Systems* (Cham, 2022), P. e. a. García Bringas, Ed., Springer International Publishing, pp. 123–132.

[25] Vafaie, H., and De Jong, K. Genetic algorithms as a tool for feature selection in machine learning. In *Proceedings Fourth International Conference on Tools with Artificial Intelligence TAI '92* (1992), pp. 200–203.

[26] Wang, S., Tang, J., and Liu, H. Embedded Unsupervised Feature Selection. *Proceedings of the AAAI Conference on Artificial Intelligence 29*, 1 (Feb. 2015).

[27] Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. Top 10 algorithms in data mining. *Knowledge and Information Systems 14*, 1 (Jan. 2008), 1–37.

[28] XUE, B., CERVANTE, L., SHANG, L., BROWNE, W. N., and ZHANG, M. Multi-objective evolutionary algorithms for filter based feature selection in classification. *International Journal on Artificial Intelligence Tools 22*, 04 (2013), 1350024.

[29] Xue, B., Zhang, M., Browne, W. N., and Yao, X. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation 20*, 4 (2016), 606–626.

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

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

Director Proiect,
Suciu Mihai-Alexandru