UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

O analiză extinsă a performanței ingineriei datelor și a proiectării modelelor în invațarea profundă

Rezumatul tezei de doctorat

Student-doctorand: Vlad-Ioan Tomescu
Conducător științific: Prof. dr. Czibula Gabriela

2022

Cuvinte cheie: Învățare profundă, învățare automată, ingineria caracteristicilor, rețele convoluționale, hărți de adâncime, predicția defectelor software

Cuprins

Cuprinsul tezei de doctorat

Lista publicațiilor

Clasamentul publicațiilor a fost realizat conform standardelor CNATDCU (Consiliul Naţional de Atestare a Titlurilor, Diplomelor si Certificatelor Universitare) aplicabile pentru studenții doctoranzi înscriși după 1 octombrie 2018. Toate clasamentele sunt listate conform clasificării jurnalelor [1] și a conferințelor [2] în Informatică.

Publicații indexate în Web of Science - Science Citation Index Expanded

[MVIC22] Diana-Lucia Miholca, **Vlad-Ioan Tomescu**, Gabriela Czibula. *An in-depth analysis of the software features' impact on the performance of deep learning-based software defect predictors*. IEEE Access, Volume 10, 64801 — 64818, 2022 (**2021 IF=3.476**, Journal IF Quartile Q2)

Rank B, 4 points.

Publicații indexate în Web of Science, Conference Proceedings Citation Index

[TCNta21] **Tomescu, Vlad-Ioan** and Czibula, Gabriela and Niţică, Ştefan. A study on using deep autoencoders for imbalanced binary classification *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference (KES 2021)*, Poland, Volume 192, 2021, Pages 119-128 (**indexed Web of Science**)

Rank B - CORE2021, 4 points.

[MCT20b] Diana-Lucia Miholca, Gabriela Czibula, **Vlad Tomescu**. COMET: A conceptual coupling based metrics suite for software defect prediction *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24nd International Conference (KES 2020)*, Budapest, Hungary, Procedia Computer Science, Volume 176, 2020, Pages 31–40. (**indexed Web of Science**)

Rank B - CORE2020, 4 points.

[CTC21] George Ciubotariu, **Vlad-Ioan Tomescu**, Gabriela Czibula. *Enhancing the performance of image classification through features automatically learned from depth-maps*, 13th International Conference on Computer Vision Systems (ICVS), September 22-24, 2021, LNCS 12899, Pages 68–81 (**indexed Scopus**).

[1] https://uefiscdi.ro/premierea-rezultatelor-cercetarii-articole
[2] http://portal.core.edu.au/conf-ranks/

Rank C - CORE2021, 2 points.

[NtaCT20] Ştefan Niţică, Gabriela Czibula, **Vlad-Ioan Tomescu**. *A comparative study on using unsupervised learning based data analysis techniques for breast cancer detection*. IEEE 14th International Symposium on Applied Computational Intelligence and Informatics, SACI 2020, Timisoara, Romania, 2020, Pages 99–104. (**indexed Web of Science**)

Rank D - CORE 2020, 1 point.

[Tom20] **Vlad-Ioan Tomescu**. *$FoRConvD$: An approch for food recognition on mobile devices using convolutional neural networks and depth maps*. IEEE 14th International Symposium on Applied Computational Intelligence and Informatics, SACI 2020, Timisoara, Romania, 2020, Pages 129–134. (**indexed Web of Science**)

Rank D - CORE 2020, 1 point.

Scorul publicațiilor: 16 points.

Introducere

Domeniul principal de cercetare al tezei de doctorat este *învățarea profundă*. Teza de doctorat se intitulează „O cercetare extinsă asupra performanței analizei datelor și proiectării modelelor în învățarea profundă" și își propune să dezvolte noi modalități de îmbunătățire a performanței clasificării, atât prin utilizarea analizei inteligente a datelor, cât și prin proiectarea de modele eficiente.

Învățarea profundă este un tip de învățare automată care este renumit pentru arhitecturile de model și metodele de antrenament complexe, precum și pentru utilizarea multor date pentru antrenare și evealuare. Modelele de învățare profundă pot, în teorie, să mimeze orice funcție matematică și, cu antrenarea potrivită, ar putea prezice corect clasa instanțelor din date. Prin urmare, din acest lucru rezultă că, orice fel de analiză a datelor și inginerie a caracteristicilor, utilizate în mod tradițional pentru învățarea automată clasică, devin depășite. Cu toate acestea, este un fapt cunoscut că nu toate modelele de învățare profundă funcționează la fel de bine și chiar dacă majoritatea dintre ele ar putea imita orice funcție, de multe ori antrenamentul modelelor nu ajunge la funția respectivă, ci mai degrabă converge către o funcție corespunzătoare unui minim local.

Acest lucru ridică întrebarea dacă analiza datelor este încă relevantă în sarcinile de clasificare, chiar dacă este folostiă la un model de învățare profundă puternic și complex. Se sugerează posibilitatea ca ingineria inteligentă a caracteristicilor să netezească planul funcției de pierdere al modelului, făcând mai ușor pentru optimizatorul său să găsească minimul global sau chiar unul mai bun local. De asemenea, trebuie luat în considerare și faptul că nu toate datele sunt la fel de relevante, rezultând în unele seturi de date să fie mai dificile decât altele, deoarece conțin multe date irelevante. Din cauză că, de multe ori seturile de date sunt generate fie automat din toate datele disponibile, fie de către un profesionist dintr-un domeniu de aplicare, fără experiență în învățarea automată, cercetarea în analiza datelor ar putea duce în cele din urmă la o extracție mai bună a lor.

Următoarul aspect luat în considerare este legat de modelele de învățare profundă propriu-zise. Deoarece datele pot proveni din diferite domenii de aplicare, acest lucru ar putea indica fie ca unele modele să fie mai bune la clasificare pe un anumit set de date, sau să existe modele superioare în general. Prin urmare, este vital să se înțeleagă raționamentul din spatele alegerii unui model pentru o anumită sarcină. Factorii importanți în ceea ce privește modelele de învățare profundă includ arhitectura și metodele de antrenament ale acestuia. Cercetarea de față investighează cât de relevanți sunt fiecare din acești factori, iar totalitatea acestor factori se va numi *designul modelelor*.
Cele două direcții de cercetare complementare dictează principalele obiective de cercetare:

OC1 Studiul relevanței analizei datelor și a ingineriei caracteristicilor în învățarea profundă.

OC2 Determinarea criteriilor de alegere a unui model de învățare profundă pentru o anumită sarcină de clasificare.

Deoarece obiectivele de cercetare sunt foarte generale, răspunsul la ele ar necesita o analiză extinsă în diferite domenii de aplicare.

Domenii de aplicare

S-au ales 3 domenii de aplicare: vederea artificială, predicția defectelor software și detecția cancerului mamar. Raționamentul din spate este dublu. În primul rând, în timp ce domeniile de aplicare sunt foarte diferite și arată că cercetarea este extinsă, natura datelor în sine este relativ similară, ceea ce duce la aplicarea unor modele pe mai multe domenii, rezultând în generalitatea lor. Al doilea raționament se datorează importanței domeniilor și consecințelor semnificative care rezultă din predicții corecte.

Vederea artificială își propune să acopere o gamă largă de sarcini vizuale pentru a automatiza procesele de luare a deciziilor în domenii precum *conducerea autonomă*, *automatizarea proceselor robotizate*, *controlul calității* sau crearea de medii virtuale pentru Realitatea Virtuală/Realitatea Augmentată. Recent, toate sarcinile de vedere artificială sunt efectuate folosind modele de *învățare profundă*, care constau din mai multe tipuri de straturi pentru procesarea imaginilor cu diferite rezoluții. Cea mai populară arhitectură în procesarea imaginilor este cea de tip *Rețea convoluțională profundă*, care are ca concept de bază *convoluția*. Scopul unor astfel de rețele este de a codifica informațiile spațiale ale imaginilor cu ajutorul convoluțiilor, în timp ce scad rezoluția imaginilor pentru a înțelege mai mult contextul scenei. Astfel de convoluții sunt intens cercetate pentru a maximiza cantitatea de informații extrase și a minimiza costul antrenării rețelei.

Predicția defectelor software (PDS) reprezintă o activitate esențială în timpul dezvoltării software, deoarece contribuie la îmbunătățirea continuă a *calității software-ului*. Prin detectarea modulelor predispuse la defecte în noile versiuni ale unui sistem software, predicția defectelor software contribuie la îmbunătățirea întreținerii și evoluției software. PDS constă în identificarea componentelor software defecte, fiind considerată o activitate esențială în timpul dezvoltării software. Dezvoltarea de sisteme software de înaltă calitate este costisitoare și, prin urmare, PDS este utilizată pentru creșterea eficienței asigurării și testării calității. Prin detectarea modulelor predispuse la defecte în noile versiuni ale unui sistem software, PDS ajută la alocarea efortului pentru a testa acele module mai amănunțit.

După cum afirmă Organizația Mondială a Sănătății, *cancerul mamar* este cea mai frecventă formă de cancer în rândul femeilor, fiind responsabil pentru 15% din toate decesele cauzate de cancer la femei. Scopul principal al unui model de învățare automată este de a ajuta medicii, prin detectarea tiparelor ascunse ale bolii în datele pe care a fost antrenat. Această asistență ar putea face munca experților mai eficientă. Provocările majore în detectarea cancerului mamar sunt de a maximiza rata instanțelor adevărat pozitive (adică de a maximiza sensibilitatea clasificatorului), dar, între timp, de a minimiza detecția de falsuri pozitive (sau, echivalent, de a maximiza specificitatea clasificatorului). Un fals pozitiv reprezintă o anomalie descoperită de clasificator, dar care nu este cancer. Minimizarea ratei de falsuri pozitive poate duce la reducerea traumelor pacienților (panică, intervenții sau tratamente inutile), precum și la scăderea costului tratamentelor (de exemplu, mamografii sau teste suplimentare inutile). În zilele noastre există un interes din ce în ce mai mare pentru aplicarea tehnicilor convenționale de *învățare automată* și, mai recent, de *învățare profundă* în domeniul detectării cancerului de sân, ajutând experții medicali în depistarea timpurie a bolii.

Contribuții originale

Contribuțiile originale pot fi organizate în 2 categorii. Prima este legată de date, tot ceea ce se face pe partea de date care duce la o creștere a performanței. A doua este legat de model, comparații între diferite design-uri de modele pe un anumit set de date. Contribuțiile sunt împărțite între cele 3 domenii principale de aplicare:

1. *Vederea artificială* este un domeniu care preia imagini și folosește modele de învățare profundă, în special rețele convoluționale, pentru a îndeplini o varietate de sarcini, inclusiv clasificarea, obiectivul actual de studiu. În lucrările originale [Tom20] și [CTC21], se prezintă 2 experimente pe imagini, inclusiv folosind date de adâncime. Primul include construirea unui sistem care ar determina în cele din urmă masa unui produs alimentar, determinând separat clasa și volumul. Al doilea experiment analizează îmbunătățirea performanței la clasificare, prin adăugarea datelor de adâncime. Prin urmare, contribuțiile originale sunt:

   (a) Tipul de date pe care se concentrează cercetarea actuală de vedere artificială sunt datele de adâncime. Datele de adâncime pot fi extrase împreună cu imaginea color originală, folosind diverse metode, de la senzori stereo la camere infraroșu. S-au ales ca obiect de studiu datele de adâncime în special pentru versatilitatea lor, analizându-se diferitele utilizări ale hărților de adâncime, atât ca și caracteristici suplimentare care pot îmbunătăți performanța într-o sarcină de clasificare, cât și ca mijloace autonome de reconstrucție a volumului. Rezultatele arată că simpla adăugare a datelor de adâncime la o imagine și trecerea acesteia printr-un clasificator ar putea îmbunătăți considerabil performanța, în timp ce extragerea și combinarea ulterioară a caracteristicilor ar putea îmbunătăți-o și mai mult. Chiar și atunci când nu sunt utilizate pentru învătare automată, datele de adâncime pot fi folosite pentru o sarcină complementară clasificării, reconstrucția și estimarea volumului cu rezultate apropiate de valorile reale.

   (b) Pe partea de model, s-au folosit doi clasificatori cu arhitecturi diferite. Unul este un clasificator de învățare profundă de ultimă generație, potrivit pentru seturi mari de date cu multe clase, în timp ce al doilea aparține învățării automate clasice. Primul clasificator a fost ales pentru a maximiza performanța pe o sarcină dificilă, în timp ce al doilea a fost suficient pentru clasificarea binară, contribuind totodată la evidențierea creșterii performanței aduse de hărțile de adâncime. Modelul de învățare profundă, deși complex, este conceput pentru a fi suficient de ușor pentru a fi compatibil cu aplicațiile live. Este un model de ultimă generație, atât din punct de vedere al arhitecturii (Efficient [TL19]), cât și al metodei de antrenament (Fastai [H+18]). Al doilea model este un perceptron cu mai multe straturi, un model de învățare automată standard, a cărui implementare disponibilă în scikit-learn [Sci21] servește pentru a sublinia reproductibilitatea experimentelor actuale cu hărți de adâncime.

2. *Predicția defectelor software* este folosită pentru a determina dacă entitățile software sunt defecte sau nu. În lucrarea originală, [MVIC22] se cercetează diverse modele și caracteristici ale datelor, în timp ce în cealaltă lucrare [MCT20a] se introduce un nou set de caracteristici, analizând performanța acestora. Prin urmare, contribuțiile originale pot fi considerate ca fiind:

   (a) S-a efectuat o analiză aprofundată a impactului caracteristicilor software asupra performanței predictorilor de defect software, bazați pe învățarea profundă. Se extinde în continuare

un set de caracteristici pe scară largă propus în literatura de specialitate pentru detectarea predispoziției la defecte, prin adăugarea de caracteristici software conceptuale, care captează semantica codului sursă, inclusiv comentariile. Caracteristicile conceptuale sunt proiectate automat folosind Doc2Vec și LSI, modele de predicție bazate pe rețele neuronale artificiale. Se face chiar un pas mai departe în ingineria caracteristicilor, prin proiectarea propriului set de metrici numit COMET. Acest set este conceput pentru a fi relevant în a ajuta la identificarea instanțelor defecte și este comparat cu setul de metrici standard utilizat în literatură numit Promise [SM15], având ca rezultat o îmbunătățire a performanței.

(b) În ceea ce privește modelul, o evaluare amplă a sistemului software Calcite evidențiază o îmbunătățire semnificativă statistic, obținută prin aplicarea unor clasificatoare bazate pe învățare profundă pentru detectarea defectelor software, atunci când se utilizează caracteristici conceptuale extrase din codul sursă, pentru caracterizarea entităților software. Mai mult, o serie de experimente sunt concepute, pentru a evalua performanța noilor metrice COMET introduse, pe 7 seturi de date. Acestea includ o analiză bazată pe corelație, modele de învățare nesupravegheată și clasificare binară din învățarea supravegheată.

3. *Detecția cancerului mamar*, datorită naturii datelor sale, este o sarcină de clasificare dezechilibrată. Partea de analiză a datelor include reducerea dimensionalității folosind învățarea nesupravegheată, în timp ce partea de proiectare de modele introduce un nou clasificator binar de tip autoencoder.

   (a) Deoarece seturile de date de cancer mamar sunt foarte dezechilibrate, partea de analiză a datelor se concentrează asupra modului în care acest lucru ar putea afecta performanța. Prin urmare, se propun trei modele de *învățare nesupravegheată*, pentru a găsi informații în date. Modele de tip *T-Distributed Stochastic Neighbor Embedding* (t-SNE), *autoencoder* și *self-organizing maps* (SOM) au fost folosite ca tehnici neliniare de reducere a dimensionalității și pentru o clasificare ulterioară nesupravegheată (adică grupare). Experimentele au fost efectuate pe trei seturi de date de cancer mamar (două disponibile public din depozitul UCI [WSM] și unul reprezentând datele SERS utilizate anterior în literatura de specialitate [CMM+15]).

   (b) Ca parte a designului de modelele, se introduce un clasificator binar bazat pe modele de tip autoencoder (AE). Deoarece AE-urile sunt antrenate să recunoască și să reproducă un anumit set de instanțe, acestea pot fi utilizate pe toate cele dintr-o anumită clasă. Prin urmare, prima versiune a modelului ar conține un AE capabil să recunoască o clasă, dar nu și cealaltă, rezultând un clasificator binar. Cu toate acestea, este posibil să se antreneze două AE, câte unul pe fiecare clasă, și la momentul inferenței, ambele ar da o valoare a funcției de pierdere, bazată pe cât de bine fiecare a recunoscut instanța. Decizia ar fi atunci să se considere clasa AE-ului cu pierderea mai mică. Este posibil să se meargă și mai departe și să se considere perechile de pierderi de la cele două AE ca puncte 2D într-un plan, iar granița de decizie să fie determinată de un alt clasificator, în acest caz de tip *Support Vector Machine (SVM)* [PS20a]. Apoi, sistemul format din două autoencodere ar acționa ca un extractor de caracteristici din date, aparținând efectiv primului tip de contribuții originale, legat de date.

Capitolul 1

Fundamente teoretice

În acest capitol se dă un context problemelor abordate din teză. Majoritatea contribuțiilor originale sunt abordări bazate pe învățare supravegheată și nesupravegheată, unele modele fiind aplicate în mai multe domenii. Prin urmare, se oferă o privire de ansamblu asupra modelelor de învățare automată în secțiunea 1.1. Acest capitol al tezei prezintă, de asemenea, descrierea și importanța problemelor abordate, precum și o revizuire a literaturii de specialitate care descrie lucrări similare conexe.

1.1 Modele de învățare automată utilizate

În această secțiune se prezintă modelele utilizate în studiul prezent. Principalele roluri pe care trebuie să le îndeplinească sunt: analiza calitativă, extragerea caracteristicilor și clasificarea datelor. O analiză calitativă de succes ar trebui să se potrivească cu rezultatele numerice ale clasificării, în timp ce extragerea caracteristicilor și introducerea lor într-un clasificator ar trebui să-i îmbunătățească performanța. În timp ce unele modele excelează doar la un singur rol, există unele care pot fi folosite în mai multe situații, cu succes considerabil. Mai multe detalii despre aplicațiile lor pot fi găsite în capitolele ulterioare.

1.1.1 Învățare nesupravegheată

t-SNE [vdMH08] este o tehnică neliniară de reducere a dimensionalității, care urmărește maximizarea asemănării dintre distribuția probabilităților în spațiul original și distribuția probabilităților în spațiul cu puține dimensiuni. Distanțele dintre două puncte de date sunt convertite în probabilități condiționate și divergența Kullback-Leibler este utilizată pentru a exprima similaritatea a două distribuții de probabilitate. Printr-un hiperparametru numit *perplexity*, t-SNE păstrează structura locală a datelor, menținând în același timp structura globală.

Self-organising maps (SOM) sunt modele de învățare nesupravegheate utilizate pe scară largă în literatura de învățare automată pentru vizualizarea datelor și reducerea dimensionalității neliniare, precum și pentru algoritmul de grupare. Modelul SOM [Koh13] este conectat la paradigma învățării competitive. Așa-numita *hartă* este de obicei o rețea 2D de neuroni obținută printr-o mapare neliniară a instanțelor de intrare de mari dimensiuni într-un spațiu 2D. O caracteristică principală a mapării SOM este aceea de a păstra proprietățile topologice ale spațiului de intrare, astfel încât cazurile care sunt similare în spațiul de intrare de dimensiuni mari sunt mapate pe neuroni care sunt aproape unul de celălalt [KOS09]. Datorită acestei proprietăți, un SOM antrenat oferă grupuri de instanțe de intrare similare.

1.1.2 Autoencoder

Un *autoencoder* (AE) [GBC16] este o rețea neuronală de tip *feed forward*, autosupravegheată, care își propune să învețe funcția de identitate, mai precis să recreeze datele de intrare. Există două componente principale ale unui AE: (1) *codificator*, care mapează spațiul de intrare n-dimensional într-un spațiu ascuns m-dimensional; și (2) un *decodificator*, care învață să reconstruiască spațiul de intrare original din spațiul ascuns. Dacă dimensionalitatea spațiului ascuns este mai mică decât dimensionalitatea spațiului de intrare, codificatorul poate fi utilizat nesupravegheat pentru reducerea dimensionalității, adică starea ascunsă exprimă o reprezentare cu puține dimensiuni a datelor de intrare.

1.1.3 Rețele neuronale convoluționale

Rețelele neuronale convoluționale sunt un tip specializat de rețele neuronale care se numește așa datorită unui strat numit strat convoluțional. Aplicația lor principală este pentru datele cu imagini bidimensionale și tridimensionale, deși au fost utilizate cu succes și pe diferite tipuri de date, cum ar fi tabelare. Rețelele convoluționale sunt lanțuri lungi de straturi convoluționale, care sunt, de obicei, însoțite de funcții de activare, pentru a introduce neliniaritate în sistem, și straturi de normalizare pentru un antrenament mai fluid. Ele sunt antrenate propagând înapoi gradienții. Pentru a rezolva problema dispariției gradientului, arhitecturile moderne conține conexiuni de ocolire și pentru a evita învățarea pe de rost a setului de antrenare, sunt folosite conexiuni de tip *drop*. Pentru sarcinile de clasificarea, rețelele se termină cu un strat complet conectat și o funcție de activare *softmax*.

1.1.4 Support Vector Machines

Support vector machines (SVM) sunt metode de învățare supravegheată utilizate atât pentru clasificare, cât și pentru regresie [PS20b]. Având în vedere un set de vectori cu valori reale de dimensiuni mari (puncte de date în $\mathbb{R}^d$), un SVM construiește un hiperplan sau un set de hiperplanuri de separare optimă într-un spațiu de dimensiuni mari, o bună separare fiind realizată de către hiperplanul cu *marja funcțională* maximă (adică având cea mai mare distanță până la cele mai apropiate puncte de date de antrenament din orice clasă). Datorită marjei mari, SVM-urile sunt cunoscute ca clasificatori de marjă mare. Intuiția este că, prin maximizarea marjei clasificatorului, riscul unor decizii de clasificare incerte (adică puncte de date din apropierea graniței de decizie) va fi minimizat.

Problema de optimizare a SVM poate fi formulată ca găsirea unui hiperplan de separare optim (adică lăsând cea mai mare fracție posibilă de puncte din aceeași clasă pe aceeași parte a hiperplanului și maximizând distanța fiecărei clase de hiperplan) și minimizând probabilitatea de clasificare greșită a instanțelor de antrenament și a celor nevăzute de testare. SVM-urile sunt cunoscute pentru implementarea controlului automat al complexității pentru a evita supraadaptarea și, datorită marjelor mari, sunt clasificatoare simple, chiar dacă au o mulțime de hiperparametri.

1.1.5 Perceptronul cu mai multe straturi

Perceptronul cu mai multe straturi [AC20] este un tip de rețea neuronală artificială, care este format din mai multe straturi complet conectate. Conține un strat de intrare, un strat de ieșire și cel puțin un strat ascuns, fiecare cu noduri care sunt toate conectate între ele. Rețeaua conține ponderi la fiecare nod, care pot fi antrenate utilizând propagarea inversă a gradientului, folosind regula lanțului,i și apoi pot face predicții la momentul inferenței. Antrenamentul se face folosind un optimizator precum *Stochastic Gradient Descent* (SGD) sau Adam, până când se ajunge la convergență. Funcțiile de activare sunt de obicei introduse pentru a adăuga neliniaritate în sistem, care, în teorie, ar putea apoi

simula orice funcție. Alegerea mărimii adecvate, în funcție de dificultatea setului de date, ajută la evitarea situațiilor nedorite, cum ar fi supraadaptarea și subadaptarea. Perceptronul poate fi utilizat pe mai multe tipuri de date.

Capitolul 2

Modele de învățare profundă în domeniul vederii artificiale

Cele două direcții principale de cercetare ale tezei sunt: (1) analiza modului în care natura datelor influențează performanța în clasificare și (2) proiectarea de modelele optime pentru sarcini specifice pe un anumit set de date. Deoarece aceasta este o analiză extinsă și rezultatele ar trebui să generalizeze, se cercetează diverse domenii de aplicare foarte diferite unul de celălalt. Unul dintre aceste domenii este Vederea Artificială, care este prezentat în acest capitol.

Vederea Artificală (VA) își propune să acopere o gamă largă de sarcini vizuale pentru a automatiza procesele de luare a deciziilor în domenii precum *conducerea autonomă*, *procese robotizate*, *controlul calității produselor* sau crearea de medii virtuale pentru Realitatea Virtuală/Realitatea Augmentată. Recent, toate sarcinile VA sunt efectuate folosind modele *Învățare Profundă* care constau din mai multe tipuri de straturi pentru procesarea imaginilor de intrare la rezoluții diferite. Cea mai populară arhitectură în procesarea imaginilor este cea a *Rețele Convoluționale Adânci*, care are ca și concept de bază *convoluția*. Scopul unor astfel de rețele este de a codifica informațiile spațiale ale imaginilor cu ajutorul convoluțiilor în timp ce scade rezoluția imaginilor, pentru a înțelege mai mult contextul scenei. Astfel de convoluții sunt intens cercetate pentru a maximiza cantitatea de informație extrasă și a minimiza costul de calcul al antrenamentului rețelelor.

Prin urmare, pe partea de model, se studiază diferite arhitecturi și metode de antrenament ale rețelelor neuronale, în timp ce pe partea de date, cercetărea se concentrează pe utilitatea datelor de adâncime în contextul învățării profunde aplicate la vederea artificială. Contribuțiile originale pot fi rezumate astfel:

1. Au fost utilizate două clasificatoare cu arhitecturi diferite. Unul este un clasificator de învățare profundă de ultimă generație, potrivit pentru seturi mari de date cu multe clase, în timp ce al doilea aparține învățării automate clasice. Primul clasificator a fost ales pentru a maximiza performanța pe o sarcină dificilă, în timp ce al doilea a fost suficient pentru clasificarea binară, contribuind totodată la evidențierea performanței aduse de *hărțile de adâncime*.

2. Analizăm diferitele utilizări ale *hărților de adâncime*, atât ca și caracteristici suplimentare care pot îmbunătăți performanța într-o sarcină de clasificare, cât și ca mijloace independente de reconstrucție a volumului. Rezultatele arată că simpla adăugare a datelor de adâncime la o imagine și trecerea acesteia printr-un clasificator ar putea îmbunătăți considerabil performanța, în timp ce extragerea și combinarea ulterioară a caracteristicilor ar putea îmbunătăți-o și mai mult. Chiar și atunci când nu sunt utilizate pentru învățare automată, datele de adâncime pot fi utilizate pentru reconstrucția și estimarea volumului, cu rezultate apropiate de valorile reale,

justificând astfel versatilitatea lor.

Restul capitolului este organizat după cum urmează:

Secțiunea 2.1 în ansamblu se bazează pe lucrarea originală [Tom20] și introduce un sistem care ar folosi datele vizuale pentru a determina în cele din urmă masa unui aliment. Sistemul trebuie să fie rapid, dar performant, pentru a fi aplicat de pe un dispozitiv mobil. Secțiunea **??** reprezintă o altă lucrare originală [CTC21], care cercetează utilitatea datelor de adâncime în îmbunătățirea clasificării imaginilor de interior-exterior.

2.1 $FoRConvD$: O abordare pentru recunoașterea alimentelor de pe dispozitive mobile folosind rețele neuronale convoluționale și hărți de adâncime

Alimentația corectă este fundamentală pentru un stil de viață sănătos, mai ales în lumea modernă, unde multe boli ar putea fi evitate cu o alimentatie corecta. S-a introdus [Tom20] o abordare numită $FoRConvD$, care constă în (*Recunoașterea alimentelor folosind rețele neuronale convoluționale și hărți de adâncime*) pentru detectarea tipurilor de alimente și a masei acestora de pe dispozitivele mobile, folosind doar camera telefonului. Abordarea aceasta constă în două componente principale: *detectarea tipului de alimente* și *estimarea volumului*. Detectarea tipului de hrană se face cu EfficientNet, un model de *rețea neuronală convoluțională* de ultimă generație, potrivit pentru telefoanele mobile. A fost aplicat pe un set de date de peste 80000 de imagini și 382 de clase. Din ce se știe, aceasta este prima dată când EfficientNet a fost utilizat pe un set mare de date de alimente cu multe clase.

Metoda folosită pentru estimarea volumului se numește *fuziunea hărților de adâncime* și implică realizarea de imagini diferite din diverse unghiuri, împreună cu hărțile lor de adâncime, și calcularea volumului unui model 3D al obiectului.

Fiabilitatea și acuratețea metodei propuse pentru estimarea volumului alimentelor este dovedită științific prin rezultatele experimentale, rezultând o ușoară supraestimare a volumului de 0% -10%, în funcție de forma obiectului. Este împărțită în patru sarcini: *reconstrucția modelului*, *fuziunea norului de puncte*, *din nor de puncte la grila cu voxeli* și *calculul volumului*. Metoda introdusă în această lucrare convertește hărțile de adâncime direct într-un nor de puncte și utilizează o tehnică de îmbinare a norilor de puncte numită *Punctul cel mai apropiat, iterativ* (ICP) [CM92] [BM92] pentru a obține norul fuzionat. Norului de puncte i se generează apoi un volum prin conversia acestuia într-o grilă de voxeli, iar acel volum este apoi determinat folosind însumarea unităților finite de suprafață și înmulțirea acestora cu înălțimea lor respectivă.

O provocare în abordarea noastră a fost să facem ICP mai robust împotriva erorilor de translație și rotație în situația dată: un obiect pe o suprafață plană. Acest lucru a fost realizat prin împărțirea norilor de puncte între obiect și plan și efectuând mai multe iterații ICP într-un anumit mod. De asemenea, pentru a determina cât mai bine volumul, modelul a suferit diverse transformări inclusiv o eliminare a zgomotului. Un alt factor care trebuia optimizat este dimensiunea unității de volum. Dacă este prea mic, norul de puncte nu este suficient de dens, lăsând spații goale, în timp ce dacă este prea mare, colțurile nedorite ale unităților de volum cubice adaugă volum redundant.

Fiabilitatea și acuratețea metodei propuse pentru estimarea volumului alimentelor este dovedită empiric de rezultatele experimentale, rezultând o ușoară supraestimare a volumului de 0% - 10%, în funcție de forma obiectului.

2.2 Îmbunătățirea performanței clasificării imaginilor prin caracteristici învățate automat din hărți de adâncime

În lucrarea originală [CTC21] s-a abordat o problemă de vedere artificială, aceea a clasificării imaginilor de interior și exterior, folosind modele de *învățare automată* și *învățare profundă*. Pentru a face acest lucru, s-a efectuat o analiză bazată pe învățarea nesupravegheată cu scopul de a determina relevanța hărților de adâncime în contextul clasificării. Pentru teste suplimentare pentru a decide asupra granularității mijloacelor de extracție a informațiilor, caracteristicile au fost agregate din subimagini de diferite dimensiuni, din setul de date DIODE, comparând mai multe rezoluții. Patru seturi de caracteristici au fost propuse și analizate comparativ în contextul clasificării imaginilor de interior-exterior. Pentru a confirma empiric avantajul utilizării caracteristicilor învățate automat din hărțile de adâncime, caracteristicile au fost introduse într-un model de clasificare supravegheat. Performanța clasificării folosind seturile de caracteristici propuse este apoi comparată cu rezultatele lucrărilor similare, existente în literatura de specialitate, evidențiind avantajul clar al utilizării caracteristicilor care folosesc informații de adâncime. Îmbunătățirile generale ale acurateței constau în 18,8% pentru abordarea de învățare automată și 1,1% pentru cea de învățare profundă atunci când la date au fost adăugate informații de adâncime.

Capitolul 3

Modele de învățare profundă pentru detectarea defectelor software

Predicția defectelor software reprezintă o activitate esențială în timpul dezvoltării software, deoarece contribuie la îmbunătățirea continuă a calității software-ului. Prin detectarea modulelor predispuse la defecte în noile versiuni ale unui sistem software, predicția defectelor software contribuie la îmbunătățirea întreținerii și evoluției software.

Predicția defectelor software (PDS) constă în detectarea defectelor software, fiind considerată o activitate esențială în timpul dezvoltării software. Reprezintă activitatea de identificare a modulelor software, care sunt defecte, în mai multe versiuni ale aceluiași sistem software [hCMZ11]. PDS este considerat de mare importanță în ingineria software, deoarece contribuie la îmbunătățirea continuă a *calității software*. Dezvoltarea de sisteme software de înaltă calitate este costisitoare și, prin urmare, PDS este utilizat pentru creșterea eficienței asigurării și testării calității [CMC14]. Prin detectarea modulelor predispuse la defecte în mai multe versiuni ale aceluiași sistem software, PDS ajută la alocarea efortului astfel încât să se testeze mai amănunțit acele module [hCMZ11].

PDS ajută la urmărirea evoluției proiectului, susține managementul proceselor [CZ01], prezice fiabilitatea software-ului [Zhe09], ghidează testarea și revizuirea codului [hCMZ11]. Toate aceste activități permit reducerea semnificativă a costurilor implicate în dezvoltarea și întreținerea produselor software [HM18]. În plus, în special în cazul sistemelor critice pentru siguranță, PDS ajută la detectarea anomaliilor software cu posibile efecte negative asupra vieților umane.

Pe măsură ce complexitatea sistemelor software crește, numărul de defecte software generate în timpul dezvoltării software va crește, de asemenea, semnificativ. Această complexitate tot mai mare a proiectelor software duce la necesitatea unei atenții sporite la analiza și testărea acestora. Numeroase cercetări din literatura PDS se bazează pe extragerea informațiilor istorice din codul sursă în timpul procesului de dezvoltare a software-ului și apoi construirea unui model de clasificare (statistic, bazat pe învățarea automată sau altele) pentru a prezice defectele software [ZZYW20].

Există două tipuri principale de predictori de defect software în literatura de specialitate PDS: predictori de defect în cadrul proiectului și inter-proiect. Pentru predicția defectelor în cadrul proiectului, unele date cu defecte dintr-un proiect software sunt folosite ca set de antrenament, pentru a construi modelul de predicție, în timp ce datele rămase sunt folosite pentru a testa performanța modelului [ZZYW20]. Pe de altă parte, predictorul de defecte inter-proiect permite prezicerea defectelor într-un sistem software țintă pe baza datelor istorice de la alte sisteme. Prin urmare, acestea sunt mai generale și permit prezicerea defectelor în proiecte cu date istorice limitate. Modelele PDS inter-proiect sunt create pe baza datelor extrase dintr-un set de sisteme software, dar aplicate și testate pe diferite sisteme software.

În lucrarea originală [MVIC22], s-a efectuat o analiză aprofundată a impactului caracteristicilor software asupra performanței predictorilor de defecte software bazați pe învățarea profundă. În continuare, se extinde un set de caracteristici, de scară largă propus în literatura de specialitate pentru detectarea predispoziției la defecte, prin adăugarea de caracteristici software conceptuale care captează semantica codului sursă, inclusiv comentariile. Caracteristicile conceptuale sunt proiectate automat folosind Doc2Vec și LSI, modele de predicție bazate pe rețele neuronale artificiale. O evaluare amplă efectuată asupra sistemului software Calcite evidențiază o îmbunătățire semnificativă, statistic obținută prin aplicarea unor clasificatoare bazate pe învățare profundă pentru detectarea defectelor software, atunci când se utilizează caracteristici conceptuale extrase din codul sursă pentru caracterizarea entităților software.

În cea de-a doua lucrare originală referitoare la predicția defectelor software [MCT20a], se face un pas și mai departe în ingineria caracteristicilor prin proiectarea propriului set de metrici. Ele se bazează pe cuplarea conceptuală, relația semantică dintre entități la nivel de cod sursă. Caracteristicile sunt extrase din entități folosind Doc2Vec și LSI, iar agregare este efectuată asupra acestora, rezultând măsuri specifice (*maximul*, *media* și *deviația standard*) care sunt utilizate sub forma unui nou set de metrici numit COMET. Acest set este conceput pentru a fi relevant in ajutarea la identificarea instanțelor defecte și este comparat cu setul de metrici standard utilizat în literatură, metricile Promise [SM15], într-o serie de experimente: (1) analiza bazată pe corelație, (2)) reprezentarea învățării nesupravegheate și (3) clasificarea bazată pe învățarea supravegheată.

După cum s-a menționat în capitolele anterioare, scopul principal al tezei este de a identifica relevanța ingineriei caracteristicilor, precum și proiectarea modelelor, în contextul învățării automate în general și mai specific al învățării profunde. Secțiunea 3.1 se concentrează mai mult pe partea de model, comparând diferite modele de învățare profundă și învățăre automată clasică, arhitecturile de învățare profundă antrenate folosind metode complexe fiind superioare. Cu toate acestea, sectiunea conține și o parte semnificativă a ingineriei caracteristicilor, care este utilizată pentru a îmbunătăți performanța modelului. În schimb, Secțiunea 3.2 analizează mai mult partea caracteristicilor, prin introducerea unui nou set de caracteristici, care este testat temeinic și comparat cu literatura de specialitate, rezultând o performanță îmbunătățită față de standardul actual.

Rezultatele din acest capitol arată că atât ingineria caracteristicilor, cât și proiectarea modelului sunt importante pentru obținerea unor performanțe ridicate în domeniul predicției defectelor software.

3.1 O analiză aprofundată a impactului caracteristicilor software asupra performanței predictorilor de defect software bazați pe învățarea profundă

În ciuda importanței și aplicabilității sale extinse, PDS rămâne o problemă dificilă, în special în sistemele complexe la scară largă, și o zonă de cercetare foarte activă [HBB+11]. Condițiile pentru ca un modul software să aibă defecte sunt greu de identificat și, prin urmare, problema de predicție a defectelor este dificilă din punct de vedere computațional. Din punct de vedere al învățării supravegheate, prezicerea defectelor este o sarcină dificilă, deoarece datele de antrenament utilizate pentru construirea predictorilor de defect sunt foarte dezechilibrate. Modulele defecte dintr-un sistem software sunt mult depășite numeric de modulele fără erori. Astfel, algoritmii de învățare convenționali sunt adesea părtinitori către clasa nedefectuoasă.

O altă problemă importantă în PDS este legată de caracteristicile utilizate pentru caracterizarea entităților software (o entitate poate fi o componentă, clasă, modul – în funcție de nivelul de detaliu vizat). Deoarece, în general, în *învățarea automată*, abordarea clasică este utilizarea caracteristicilor

proiectate manual, metricile software tradiționale sunt de obicei utilizate în PDS ca și caracteristici care descriu entitățile software date. Analizele literaturii din PDS au arătat că aproximativ 87% [Mal15] dintre studiile de caz au folosit metrici procedurale sau orientate pe obiecte, în timp ce peste 95% [HTG19] dintre studiile pentru predicția defectelor inter-proiect s-au bazat pe metrici software.

O a treia problemă este că, în timp ce defectele pot fi de diferite tipuri (de exemplu, erori numerice, probleme de complexitate, probleme de indicator etc.) și este foarte probabil ca fiecare tip de defect să aibă propriile sale proprietăți, abordările SDP existente iau în considerare toate tipurile de defecte împreună și încearcă să vină cu un model universal de predicție a defectelor.

Cele două direcții de cercetare predominante în literatura SDP sunt: propunerea de caracteristici software relevante pentru discriminarea între entitățile software defecte și fara defect și construirea sau recomandarea modelelor performante de predicție a defectelor.

Când vine vorba de cantități mari de date, modelele de învățare profundă sunt unele dintre cele mai bune pentru a face predicții precise, indiferent de originea acelor date. Atâta timp cât există o corelație între informațiile de intrare și de ieșire, modelele o vor descoperi. Pentru a utiliza învățarea profundă, caracteristicile software de intrare sunt scrise în formă tabelară, o formă de date care a fost cercetată amănunțit și pentru care sunt disponibile multe modele [B+21].

În lucrarea originală [MVIC22], s-au urmat ambele direcții menționate mai sus. Munca a pornit din trei întrebări de cercetare:

ÎC1 Ar putea performanța de predicție a defectelor software fi îmbunătățită prin extinderea caracteristicilor software propuse pentru PDS cu caracteristici conceptuale extrase din codul sursă? Care este setul de caracteristici cel mai potrivit pentru a face distincția între entitățile software defecte și nedefective și în ce măsură este semnificativă îmbunătățirea performanței din punct de vedere statistic?

ÎC2 Ar putea relevanța caracteristicilor conceptuale software să fie susținută empiric atât de analize nesupravegheate, cât și de analize supravegheate efectuate pe un sistem software la scară largă?

ÎC3 Aduce predicția defectelor bazată pe învățarea profundă o îmbunătățire semnificativă statistic în comparație cu clasificatorii tradiționali supravegheați?

Având în vedere aceste întrebări de cercetare, s-a efectuat o analiză aprofundată a impactului caracteristicilor software asupra performanței predictorilor de defecte software. S-a extins colecția mare de caracteristici SDP propuse de Herbond și colab. [HTTL22] cu caracteristici software conceptuale bazate pe Doc2Vec care captează semantica codului sursă (inclusiv comentariile). Un studiu amplu realizat pe diferite versiuni ale setului de date Calcite evidențiază, atât prin analize nesupravegheate, cât și prin analize bazate pe învățare supravegheată, că acele caracteristici conceptuale aduc o îmbunătățire semnificativă dinpunct de vedere statistic asupra performanței PDS. Ca o a doua direcție de cercetare, s-a examinat pe scara largă efectul setului de caracteristici, identificat ca fiind cel mai relevant, asupra performanței diferiților predictori de defecte. Din câte se știe, până acum nu a fost propus în literatură un studiu similar cu acesta.

3.2 COMET: Un nou set de metrici pentru predicția defectelor software folosind cuplarea conceptuală

Zona de cercetare a Predicției Defectelor Software (PDS) este vastă, datorită naturii complexe și dificile a sarcinii sale. Deși există multe direcții de cercetare posibile, care ar putea duce la o mai bună identificare a defectelor în sistemele software, 2 dintre cele principale, așa cum s-a văzut în

secțiunea anterioară, sunt identificarea unor metrici relevante care sunt corelate cu prezența defectelor și crearea de modele de clasificare care ar extrage informații semnificative din acele metrici. În această secțiune, se considerară în principal prima direcție, legată de cercetărea asupra modului în care *cuplarea conceptuală* ar putea ajuta la extragerea unor valori superioare din codul sursă și se bazează pe lucrarea originală [MCT20b].

Cuplarea conceptuală se referă la relațiile dintre entitățile software la nivel semantic (clase, metode) iar relevanța acesteia în contextul PDS a fost descrisă în literatura [WLT16,MC19]. Prin urmare, un nou set de metrici bazate pe cuplarea conceptuală, numit COMET, este introdus în acest capitol. Pentru a valida relevanța parametrilor COMET pentru PDS, au fost efectuate mai multe experimente, inclusiv o analiză bazată pe corelație, modelare de învățare nesupravegheată și clasificare binară supravegheată. S-a făcut o comparație directă cu valorile standard Promise (disponibile aici [**?**]), care sunt utilizate pe scară largă în literatură. Rezultatele arată o performanță mai bună a metricilor COMET față de cele Promise, marcându-le ca fiind mai relevante în contextul PDS.

Direcțiile noastre de cercetare ar putea fi rezumate în 2 întrebări distincte, fiecare cu sub-secțiunea aferentă, care se concentrează pe răspunsul aceasteia. Aceste întrebări sunt:

ÎC1 Cum poate fi folosită cuplarea conceptuală pentru a genera o suită de metrici, care ar putea fi capabilă să prezică dacă sistemele software conțin erori? În acest context, un nou set de 36 de metrici, denumit COMET, este definit.

ÎC2 Care este performanța acestui set de metrice COMET și cum se comportă relativ la altele din literatură? Pentru a determina acest lucru, sunt efectuate mai multe experimente, inclusiv o analiză bazată pe corelație, precum și abordări de învățare supravegheată și nesupravegheată.

Capitolul 4

Modele de învățare profundă pentru detectarea cancerului mamar

Al treilea domeniu de aplicare este medicină, mai exact detectarea cancerului mamar, unde sarcina concretă este detectarea prezențelor de tumori maligne la pacienți. Natura datelor care conțin informațiile pacienților, împreună cu apariția rară a cancerului la populație în general, rezultă într-o problemă de clasificare nebalansată. Ca domeniu de aplicare, s-a considerat detectarea cancerului mamar, deoarece este o problemă de clasificare de mare importanță in domeniul medical. Conform Organizației Mondiale de Sănătate, cancerul mamar este forma de cancer responsabilă pentru cele mai multe decese la femei.

În cadrul *învățării supervizate*, *clasificarea nebalansată* reprezintă o provocare, deoarece o distribuție inegală de clase in setul de antrenare de obicei duce a o performanță slabă a prezicerii pe clasa minoritară. Totuși, de obicei clasa minoritară este cea mai relevantă, dar din cauza nebalansării setului de antrenare, erorile de clasificare pentru clasa minoritară sunt mai mari, clasificatoarele fiind părtinitoare în a prezice clasa majoritară.

În contextul mai larg al tezei, a cărei scop este atât de a cerceta cum natura datelor afectează performanța la clasificare, cât și de a construi modele performante, se poate observa că acest capitol se axează mai mult pe designul modelelor.

Contribuția originală in acest capitol este in două rânduri. În primul rând, se folosesc modele de Învățare Automată in rolul lor tradițional, cu performanță ridicată, de exemplu t-SNE pentru analiza calitativă. În al doilea rând, se aplică cu succes unele modele pentru sarcini diferite. Modelele de tip autoencoder sunt folosite în principal pentru extragerea caracteristicilor, dar s-a reușit folosirea lor pentru clasficare binară, cu performanță comparabilă și uneori mai bună decât a altor clasificatoare.

Legat de partea cu analiza datelor, se compară diverse seturi de date și se observă că performanța depinde mult de natura setului de date, unele fiind mai dificile decât altele.

Acest capitol introduce contribuțiile originale in domeniul detectării cancerului mamar. Secțiunea 4.1 introduce trei modele de învățare nesupervizat (modele de tip *t-Distributed Stochastic Neighbor Embedding* sau t-SNE, *autoencoder* și *self-organizing maps* sau SOM) analizate în lucrarea originală proprie [NtaCT20] cu scopul de a detecta nesupervizat instanțe din clasele *benign* and *malign*. Secțiunea 4.2 investighează abilitatea modelelor de tip *autoencoder* de a învăța caracteristici ale instanțelor din clasele *benign* and *malign* și introduce trei clasficatoare pe baza modelelor de tip *autoencoder* pentru detectarea cancerului mamar. Metodologia din secțiunea 4.2 a fost introdusă in lucrarea originală prorpie [TCNta21].

4.1 Un studiu comparativ privind utilizarea tehnicilor de analiză a datelor, bazate pe învățarea nesupervizată, pentru detectarea cancerului mamar

Conform Orgainzatiei mondiale a sănătății, cancerul mamar este cea mai frecventă formă de cancer la femei, fiind responsabil de 15% din numarul de decese din această categorie. Multă cercetare s-a efectuat legat de folosirea diverselor modele de Învățare Automată pentru prezicerea cancerului mamar, de la clasificatori convenționali la tehnici de Învățare Aprofundată. Trei modele de Învățare nesupervizată, de tip (*t-Distributed Stochastic Neighbor Embedding*, *autoencoders* și *self-organizing maps*), au fost introduse și analizate comparativ în lucrarea originală [NtaCT20], cu scopul de a detecta nesupervizat instanțe din clasele bening și malign. Experimente pe seturi de date, foliste anterior in literatura de specialitate, arată o performanță bună a modelelor propuse. Cea mai bună performanță a fost obținută folosind modele de tip *autoencoder*, care pentru metrica de evaluare *aria de sub curba ROC* au obținut valor mai mari de 0.935.

Contribuțiile din această secțiune sunt de două feluri [NtaCT20]. În primul rând, se propun trei modele de Învățare nesupervizată pentru detectarea de cancer mamar. Modele de tip *t-Distributed Stochastic Neighbor Embedding* (t-SNE), *autoencoder* AE și *self-organizing map* (SOM) au fost folosite ca tehnici de reducere nonlineară a dimensionalității și pentru alte metode de clasificare nonlineară, gen *clustering*. Diferite experimente au fost făcute pe trei seturi de date de cancer mamar (două disponibile public din repertoriul UCI [WSM] și unul reprezentând date SERS folosit anterior în literatură [CMM+15]). Un al doilea obiectiv al studiului a fost de a examina comparativ performanța modelelor propuse: t-SNE, AE și SOM au fost aplicate pentru maparea spațiului original al vetorilor instanțelor într-un spațiu bidimensional. Rezultatele experimentale obținute evidențiază o corelație bună între clusterele obținute nesupervizat în date (după aplicarea modelelor nesupervizate menționate anterior) și partițiile de adevăr de bază (adică seturile cunoscute de instanțe *benigne* și *maligne*). O performanță puțin mai bună a fost obținută folosind modele de tip AE, care au furnizat o valoare a metricii *Aria sub curba ROC* mai mare de 0,935 pentru toate seturile de date. În timp ce modele nesupervizte similare au fost raportate în literatura de specialitate pentru analiza datelor din depozitul UCI [WSM], din ce se știe, niciunul dintre ele nu a fost aplicat pentru analiza spectrelor SER dobândite pe serul sanguin pentru a detecta cancerul mamar.

4.2 Un studiu privind utilizarea modelelor profunde de tip *autoencoder* pentru clasificarea cancerului de sân

În această secțiune se investighează utilizarea modelelor profunde de tip *autoencoder* (AE) pentru îmbunătățirea performanței predictive la problemele de clasificare binară dezechilibrate. AE-urile sunt de obicei utilizate în literatură într-un context de învățare nesupervizat pentru a extrage caracteristici relevante (de exemplu, din imaginile histopatologice) care sunt furnizate în continuare clasificatorilor pentru a detecta anomalii [FZM18, XXHW14, XXL+15]. Dintr-o perspectivă de învățare supervizată, AE-urile sunt utilizate în principal ca detectoare de anomalii [CCT21, RDBV20] și mai rar ca clasificatori binari, ca în situația de față.

În lucrarea originală [TCNta21], au fost propuși și comparați trei clasificatori binari bazați modele de tip AE, pentru a determina dacă o instanță este malignă sau benignă. Primul clasificator propus, C_{1AE}, se bazează pe utilizarea AE-urilor ca clasificatori de o singură clasă (un AE este antrenat pe setul de instanțe *benign* și în etapa de clasificare instanțele *maligne* sunt detectate ca anomalii în raport cu clasa învățată de AE). Al doilea clasificator C_{2AE} se bazează pe ideea antrenării a două AE, câte un

AE pentru fiecare clasă (*benign* și *maligne*) și apoi o nouă instanță va fi atribuită clasei reprezentate de AE-ul „cel mai apropiat" (adică AE-ul care oferă valoarea minimă a pierderii pentru instanță). Al treilea clasificator, $C_{2AE-SVM}$, îmbunătățește C_{2AE} cu un model de tip *Support Vector Machine* utilizat pentru a decide granița de decizie dintre clase. Experimentele vor fi efectuate pe trei seturi de date de cancer (două disponibile public din depozitul UCI [WSM] și unul reprezentând datele SERS utilizate anterior în literatură [CMM+15]) cu scopul de a analiza comparativ performanța modelelor. Din câte se cunosc, modelele introduse în această secțiune sunt noi în literatura de detecție de cancer mamar bazată pe Învătare Automată.

Pentru a rezuma, studiul realizat în această secțiune este organizat în jurul a trei întrebări de cercetare:

ÎC1 Sunt modelele de tip AE capabile să codifice relații ascunse între instanțe (pacienți) aparținând aceleiași clase (*benign* sau *malign*)? Cum să se utilizeze un AE pentru codificarea tiparelor ascunse în instanțe *benigne* și pentru detectarea de către respectivul model antrenat a instanțelor *maligne* ca fiind anomalii?

ÎC2 Cum să se utilizeze două modele de tip AE (adică un AE pentru fiecare dintre clasele *benign* și *malign*) pentru a clasifica supervizat instanțe, pe baza relațiilor codificate dintre instanțele aparținând aceleiași clase?

ÎC3 Cât de performante sunt abordările introduse pentru a răspunde la ÎC1 și ÎC2 și cum se compară abordările cu lucrări similare existente pentru detectarea cancerului mamar?

Se remarcă faptul că clasificatorul binar C_{2AE} introdus în această secțiune poate fi ușor extins la un clasificator cu mai multe clase, C_{nAE} (de exemplu, pentru detectarea stadiului cancerului sau alte probleme de clasificare cu mai multe clase). Presupunând că sunt date mai multe clase, clasificatorul ar avea un AE pentru fiecare clasă cu scopul de a recunoaște instanțele din respectiva clasă. Ulterior, în etapa de clasificare, unei noi instanțe ii va fi atribuită o clasă, al cărei AE oferă cea mai bună reconstrucție pentru instanța respectivă (adică minimizează valoarea pierderii pentru instanță).

Concluzii

Cele două direcții principale de cercetare din teza de doctorat sunt: (1) analiza datelor pentru a determina importanța lor in clasificare si (2) designul de modele eficiente pentru performanță ridicată. Prima direcție poate fi descrisă ca *analiza datelor si ingineria caracterisiticilor*, in timp ce a doua este intitulată *designul modelelor*. Domeniile de aplicare sunt *Vizualizare Computerizată*, *Predicția defectelor software* și *Detecție de cancer mamar*.

In domeniul *Vizualizare Computerizată*, atenția a fost asupra datelor *de adâncime*. Din moment ce se pot extrage impreuna cu imaginea color, dar nu se intâmplă așa in mod automat, demonstrarea importanței datelor de adâncime ar putea duce la o schimbare a modului general de colectare a datelor video. Din cercetarea prezentă in teză reiese ca datele de adâncime sunt versatile. În primul rând, pot fi folosite pentru a îmbunătăți performanța la clasificare, astfel atingând primul obiectiv de cercetare. Adăugând pur si simplu date de adâncime la caracteristicile color a arătat o creștere a acurateții de la 0.69% la 0.88%, iar folosind modele performante de Învățare Profundă, mai exact segmentare semantică, poate să crească performanța si mai mult.

În al doilea rând, chiar daca datele de adâncime nu sunt folosite in mod direct pentru clasificare, pot fi folosite pentru o sarcină complementară. Pentru a calcula masa unui aliment, trebuie cunoscute volumul si densitatea. Pentru estimarea volumului s-au folosit hărți de adâncime, iar densitatea este standard pentru alimente, atâta timp cât clasa lor poate fi determinată din clasificare. Cu metoda de calcul a volumului folosită, rezultatele au avut o marjă de eroare de 10% fața de valorile actuale, un rezultat foarte bun pentru sarcina de față. Astfel, versatilitatea datelor de adâncime sugerează extragerea lor in general alături de datele video color in domeniul Vizualizării Computerizate.

Pe partea de model, știind faptul că in Vizualizarea Computerizată se folosesc rețele convolutionale pentru clasificare, cercetarea de față arată faptul că un model, care utilizează arhitectura EfficientNet antrenată cu metoda *fit one cycle* din librăria Fastai, ar fi atât rapid cât si performant, fiind astfel potrivit pentru aplicații mobile. Câteodată totuși, scopul nu este de a se obține modele cu performanță de ultima oră, ci mai degrabă un model care poate evidenția bine contribuțiile originale: un set nou de caracteristici. Un astfel de model ar trebui să fie disponibil și ușor de implementat, pentru a face experimentele reproductibile. Modelul ales pentru această sarcină este Perceptronul Cu Mai Multe Straturi, un model standard de Învățare Automată implementat in librăria scikit-learn, care obține resultate bune dacă este folosit impreună cu caracteristici bine alese.

Datele folosite petru detecția defectelor software au fost generate pe baza *cuplării conceptuale*, care reprezintă relația dintre entitățille software la nivelul codului sursă. Există mai multe modalități de extragere a informației din același cod, dar nu sunt toate la fel de relevante. Pe sursele de date de tip calcite, rezultate experimentelor au arătat o performanță mai bună a unui set care conținea 60 de caracteristici de tip Doc2Vec și LSI, decât a unuia care conținea peste 4000 de caracteristici. Un număr mai scăzut de caracteristici inseamnă si o inferență mai rapidă, care este relevantă pentru aplicațiile care rulează in timp real. Astfel, se dovedește ca ingineria caracteristicilor este extrem de relevantă, chiar și atunci când modelul folosit este unul performant de Învățare Profundă.

Cercetarea ajunge inclusiv la nivelul următor, prin designul unui set de caracateristici propriu nu-

mit COMET, bazat pe cuplare conceptuală si care folosește Doc2Vec și LSI pentru extragere. Setul nou introdus a fost analizat în amănunțit și comparat cu setul Promise, standard in literatura de specialitate, in diverse teste: analiză pe bază de corelare, reprezentare folosing învățare nesupervizată și clasificare binara folosind învățare supervizată. Metricile COMET au avut performanță mai bună ca cele Promise, dovedindu-și astfel relevanța in contextul detectării de defecte software.

Mai multe modele au fost folosite in experimentele de detecție a defectelor software, atât supervizate cât și nesupervizate, dar unul a cărui performanță iese in evidență este un clasificator de învățare profundă cu o arhitectură pe baza Rețelelor Neuronale Artificiale, antrenat folosind metoda *fit one cycle* din libraria Fastai. Se poate observa ca modelele antrenate cu libraria Fastai au performanță mai bună ca alte modele, chiar și față de cele cu aceeași arhitectură implementată diferit. De asemenea, se pare că metoda *fit one cycle* poate fi aplicată indiferent de arhitectură și tipul de date, de la date vizuale până la date vectorial tabelare, rezultând în performanță ridicată.

Atât in cazul prezicerii defectelor software cât și in cel al detecției cancerului mamar, au fost intâlnite seturi de date nebalansate, din cauza naturii datelor generate din respectivele domenii de aplicare. Pentru a cuantiza exact cât de dificile devin acele sarcini in asemenea condiții, s-a introdus o măsură numită dificultate, care are legatură cu numărul de vecini al instanțelor care sunt din aceeași clasă, in special pentru clasa pozitivă. Pentru experimentele de predicție a defectelor software, se face de asemenea o comparație cu rezultatele obținute cu un clasificator care face preziceri la întâmplare, care ar prezice clasa corectă cu șanse de 50% în cazul unui set de date balansat. Totuși, pentru clasa pozitivă, dată de metrica numită precizie, această șansă poate coborî chiar și pană la 3.3% în unele cazuri, făcând sarcina modelului mult mai dificilă.

Modelele folosite pentru detecția defectelor software sunt în general aceleași cu cele din alte domenii de aplicare, de exemplu modelul t-SNE care folosește învățare nesupervizată. Totuși, se introduce un clasificator binar nou, care este pe baza modelelor de tip *autoencoder*. Modelele de tip *autoencoder* sunt folosite pentru a reconstrui datele care intră in model și sunt antrenate pe o singură clasă. Prima metodă de clasificare propusă constă dintr-un model de tip *autoencoder* antrenat pe clasa majoritară și pentru care sunt folosite ambele clase la testare. Instanțele din clasa pe care a fost antrenat modelul vor avea pierderi mici, pe când cele din cealaltă clasă vor avea pierderi mai mari, rezultând într-o diferență intre ele.

Al doilea clasificator este format din din două modele de tip *autoencoder*, fiecare antrenate pe o singură clasă și facând preziceri pe ambele clase, decizia constând in alegerea pierderii celei mai mici. De asemenea, pierderile pot fi considerate ca puncte 2D intr-un plan, granița de decizie fiind dictată de un alt clasificator, rezultând in modelele de tip *autoencoder* sa devină practic extractoare de caracteristici. *Designul modelelor* ar duce la *ingineria caracteristicilor*, ambele direcții de cercetare ale tezei combinate.

Concluzia că atât *analiza datelor* cât și *designul modelelor* sunt importante poate fi trasă din analiza amănunțită a diverselor domenii de aplicare.

Legat de primul obiectiv de cercetare, OC1, bazat pe *designul modelelor*, principalele descoperiri sunt: (1) Date de adâncime sunt folositoare și versatile și ar trebui extrase impreună cu alte date vizuale și (2) Ingineria caracteristicilor este încă foarte relevantă chiar si in contextul învățării profunde.

Urmărind al doilea obiectiv de cercetare, OC2, se pot trage următoarele concluzii: (1) Metoda de antrenare din librăria fastai [H+18] este eficientă în mod constant pentru diferite tipuri de date și (2) Modelele de tip autoencoder pot fi folosite cu succes atât pentru extragerea caracteristicilor, cât si pentru clasficare binară.

Bibliografie

[AC20] S. Abirami and P. Chitra. Energy-efficient edge based real-time healthcare support system. *Advances in Computers*, 117(1):339–368, 2020.

[B+21] Vadim Borisov et al. Deep neural networks and tabular data: A survey. *CoRR*, abs/2110.01889, 2021.

[BM92] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[CCT21] Gabriela Czibula, Carmina Codre, and Mihai Teletin. Anomalp: An approach for detecting anomalous protein conformations using deep autoencoders. *Expert Systems with Applications*, 166:114070, 2021.

[CM92] Yang Chen and Gérard G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.

[CMC14] Gabriela Czibula, Zsuzsanna Marian, and Istvan Gergely Czibula. Software defect prediction using relational association rule mining. *Information Sciences*, 264:260–278, 2014. Serious Games.

[CMM+15] Silvia Cervo, Elena Mansutti, Greta Mistro, Riccardo Spizzo, Alfonso Colombatti, Agostino Steffan, Valter Sergo, and Alois Bonifacio. Sers analysis of serum for detection of early and locally advanced breast cancer. *Analytical and Bioanalytical Chemistry*, 407:7503–7509, 2015.

[CTC21] George Ciubotariu, Vlad-Ioan Tomescu, and Gabriela Czibula. Enhancing the performance of image classification through features automatically learned from depth-maps. In Markus Vincze, Timothy Patten, Henrik I Christensen, Lazaros Nalpantidis, and Ming Liu, editors, *13th International Conference on Computer Vision Systems, ICVS 2021, Virtual Event, September 22-24, 2021, Proceedings*, volume 12899 of *Lecture Notes in Computer Science*, pages 68–81. Springer, 2021.

[CZ01] B. Clark and D. Zubrow. How good is a software: a review on defect prediction techniques. In *Software Engineering Symposium*, pages 1–35, Carneige Mellon University, 2001.

[FZM18] Yangqin Feng, Lei Zhang, and Juan Mo. Deep manifold preserving autoencoder for classifying breast cancer histopathological images. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(1):91–101, 2018.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[H+18] Jeremy Howard et al. fastai. https://github.com/fastai/fastai, 2018.

[HBB+11] Tracy Hall, Sarah Beecham, David Bowes, David Gray, and Steve Counsell. A systematic review of fault prediction performance in software engineering. *IEEE Transactions on Software Engineering*, 38(6):1276–1304, 2011.

[hCMZ11] Rui hua Chang, Xiaodong Mu, and Li Zhang. Software defect prediction using non-negative matrix factorization. *JSW*, 6(11):2114–2120, 2011.

[HM18] Jaroslaw Hryszko and Lech Madeyski. Cost effectiveness of software defect prediction in an industrial project. *Foundations of Computing and Decision Sciences*, 43(1):7 – 35, 2018.

[HTG19] Seyedrebvar Hosseini, Burak Turhan, and Dimuthu Gunarathna. A systematic literature review and meta-analysis on cross project defect prediction. *IEEE Transactions on Software Engineering*, 45(2):111–147, 2019.

[HTTL22] Steffen Herbold, Alexander Trautsch, Fabian Trautsch, and Benjamin Ledel. Problems with szz and features: An empirical study of the state of practice of defect prediction data collection. *Empirical Software Engineering*, 27(2), Jan 2022.

[Koh13] Teuvo Kohonen. Essentials of the self-organizing map. *Neural Networks*, 37:52–65, 2013.

[KOS09] Andreas Köhler, Matthias Ohrnberger, and Frank Scherbaum. Unsupervised feature selection and general pattern discovery using self-organizing maps for gaining insights into the nature of seismic wavefields. *Computers & Geosciences*, 35(9):1757 – 1767, 2009.

[Mal15] Ruchika Malhotra. A systematic review of machine learning techniques for software fault prediction. *Applied Soft Computing*, 27:504 – 518, 2015.

[MC19] Diana-Lucia Miholca and Gabriela Czibula. Software defect prediction using a hybrid model based on semantic features learned from the source code. In Christos Douligeris, Dimitris Karagiannis, and Dimitris Apostolou, editors, *Knowledge Science, Engineering and Management -LNCS, volume 11775*, pages 262–274, Cham, 2019. Springer International Publishing.

[MCT20a] Diana-Lucia Miholca, Gabriela Czibula, and Vlad Tomescu. COMET: A conceptual coupling based metrics suite for software defect prediction. *Procedia Computer Science*, 176:31–40, 2020. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

[MCT20b] D.L Miholca, Gabriela Czibula, and Vlad Tomescu. Comet: A conceptual coupling based metrics suite software defect prediction. In *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24nd International Conference (KES 2020)*, volume Procedia Computer Science, Vol. 176, pages 31–40, 2020.

[MVIC22] Diana-Lucia Miholca, Tomescu Vlad-Ioan, and Gabriela Czibula. An in-depth analysis of the software features' impact on the performance of deep learning-based software defect predictors. *IEEE Access*, page submitted, 2022.

[NtaCT20] ¸Stefan Ni¸tică, Gabriela Czibula, and Vlad-Ioan Tomescu. A comparative study on using unsupervised learning based data analysis techniques for breast cancer detection. In *2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 99–104, 2020.

[PS20a] Derek A. Pisner and David M. Schnyer. Chapter 6 - support vector machine. In Andrea Mechelli and Sandra Vieira, editors, *Machine Learning*, pages 101–121. Academic Press, 2020.

[PS20b] Derek A. Pisner and David M. Schnyer. Chapter 6 - support vector machine. In Andrea Mechelli and Sandra Vieira, editors, *Machine Learning*, pages 101–121. Academic Press, 2020.

[RDBV20] Stefania Russo, Andy Disch, Frank Blumensaat, and Kris Villez. Anomaly detection using deep autoencoders for in-situ wastewater systems monitoring data, 2020.

[Sci21] Scikit-learn. Machine learning in Python, 2021. http://scikit-learn.org/stable/.

[SM15] Shirabad Sayyad and Tim Menzies. The PROMISE Repository of Software Engineering Databases, University of Ottowa, Kanada. School of Information Technology and Engineering, University of Ottawa, Canada, 2015.

[TCNta21] Vlad-Ioan Tomescu, Gabriela Czibula, and ¸Stefan Ni¸tică. A study on using deep autoencoders for imbalanced binary classification. In *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference (KES 2021)*, volume Procedia Computer Science, Vol., page to be published, 2021.

[TL19] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[Tom20] Vlad-Ioan Tomescu. $forconvd$: An approch for food recognition on mobile devices using convolutional neural networks and depth maps. In *2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 129–134, 2020.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[WLT16] Song Wang, Taiyue Liu, and Lin Tan. Automatically learning semantic features for defect prediction. In *Proc. of the 38th Int. Conf. on Softw. Engineering*, ICSE '16, pages 297–308, New York, NY, USA, 2016. ACM.

[WSM] William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian. Breast cancer wisconsin (diagnostic) data set.

[XXHW14] Jun Xu, Lei Xiang, Renlong Hang, and Jianzhong Wu. Stacked sparse autoencoder (ssae) based framework for nuclei patch classification on breast cancer histopathology. In *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*, pages 999–1002. IEEE, 2014.

[XXL+15] Jun Xu, Lei Xiang, Qingshan Liu, Hannah Gilmore, Jianzhong Wu, Jinghai Tang, and Anant Madabhushi. Stacked sparse autoencoder (ssae) for nuclei detection on breast

cancer histopathology images. *IEEE transactions on medical imaging*, 35(1):119–130, 2015.

[Zhe09] Jun Zheng. Predicting software reliability with neural network ensembles. *Expert Systems with Applications*, 36(2):2116–2122, 2009.

[ZZYW20] Kun Zhu, Nana Zhang, Shi Ying, and Xu Wang. Within-project and cross-project software defect prediction based on improved transfer naive bayes algorithm. *Computers, Materials & Continua*, 63(2):891–910, 2020.