

RAPORTUL ȘTIINȚIFIC ȘI TEHNIC

Proiecte de cercetare pentru stimularea tinerelor echipe
independente (TE)

PN-III-P1-1.1-TE-2021-0231

**Instrument Software pentru Extracția Unitară a Parametrilor
și Caracterizarea Diodelor Schottky, bazate pe Modele
Implicite Diferențiabile Emergente**

Acronim: **SBD-SPECS**

(<https://www.dcae.pub.ro/proiecte/17/sbd-specs/>)

Proiect nr. **TE17 / 2022**

Etapa I

Decembrie, 2022

Coordonator Proiect – Universitatea POLITEHNICA București (UPB)

1. Obiectivele generale ale proiectului sunt:

- Validarea și completarea ecuațiilor modelului p-diode pentru a permite caracterizarea diodelor Schottky pe domenii de temperaturi de lucru cât mai largi (de la nivele criogenice până la 675K).
- Implementarea unor algoritmi de extracție a parametrilor, bazați pe tehnici implicite diferențiabile, cu modelul p-diode subiacent. Dezvoltarea unui produs software pentru caracterizare unitară a diodelor Schottky.
- Fabricarea de diode Schottky pe SiC cu diverse arii active și contacte metal-semiconductor.
- Dezvoltarea unei baze de date cu măsurători de caracteristici directe ale diodelor Schottky obținute.

Se urmărește dezvoltarea de soluții software robuste pentru analiza și caracterizarea diodelor Schottky, bazate pe algoritmi de optimizare de ultimă generație. Pentru a asigura relevanța fizică a parametrilor extrași, instrumentul software va avea la bază un model de neomogenități (p-diode), recent dezvoltat în instituția gazdă, cu implicarea și contribuții esențiale din partea membrilor SBD-SPECS. Produsul informatic va fi testat și calibrat cu măsurători pe diode Schottky pe SiC ce sunt capabile să funcționeze pe game largi de temperaturi.

Etapă 1. Măsurători electrice preliminare, analiza tehnicilor optime de parametrizare și proiectarea fluxului tehnologic.

Etapă I, derulată în 2022, este focalizată pe identificarea tehnicilor adecvate de extracție și optimizare a parametrilor diodelor Schottky. Se urmărește, de asemenea, efectuarea de investigații experimentale asupra diodelor Schottky pe SiC. Se va crea astfel o bază de măsurători asupra caracteristicilor directe, ce va servi la validarea soluției software de parametrizare. Pentru a dezvolta baza materială de dispozitive experimentale a UPB, etapa I a vizat și elaborarea unui proces tehnologic de fabricare a unor loturi noi de diode Schottky pe SiC. Toate activitățile din planul de realizare au fost îndeplinite, după cum reiese din Raportul științific dezvoltat în continuare (**1 Raport de cercetare**).

1.1 Măsurători pe Diodele Schottky (SBD) pe SiC pe game largi de temperaturi pentru caracterizare preliminară și popularea bazei de date.

În cadrul proiectelor desfășurate anterior, UPB, prin **Laboratorul de Dispozitive și Circuite Electronice – Studii Avansate (DCE-SA <https://eeri.eu/ERIF-2000-000M-2131>)** în parteneriat cu Institutul Național de Cercetare și Dezvoltare în Microelectronică – IMT București, a dezvoltat dispozitive pe carbură de siliciu (SiC) și tehnologia de încapsulare adecvată pentru funcționarea la temperaturi înalte, în medii ostile. Printre acestea, se numără și diode Schottky pe SiC, cu arii active și arhitecturi diferite. Astfel, baza materială disponibilă pentru realizarea Activității 1.1 asigură dispozitive cu contacte Ni/4H-SiC și Ti/4H-SiC.

Diodele Schottky masurate electric din cel mai recent lot au fost fabricate pe substrat de SiC de tip n (puternic dopat – 10^{18} cm^{-3}) cu strat epitaxial (slab dopat – $2 \cdot 10^{16} \text{ cm}^{-3}$). Diodele au contacte cu o configurație circulară. Au fost utilizate 2 metale diferite pentru realizarea contactului Schottky, și anume: Ti(100nm), respectiv Ni(100nm). Tratamentele termice post-metalizare s-au efectuat în atmosferă inertă de Ar, la temperaturi diferite. Pentru contactele de Ti, tratamentul a fost la 600°C, 3 min., iar pentru cele cu Ni la 750°C, 3 min. Pentru obținerea contactului ohmic s-au depus succesiv, pe spatele plachetelor, două straturi metalice: Ti (30 nm)/Ni (100 nm). Tratatamentul termic post-metalizare pentru contactul ohmic a fost realizat la o temperatură de 1050°C, timp de 3 minute, tot în atmosferă de Ar. Pentru realizarea pad-urilor de contact și finalizarea contactului ohmic, s-a depus un ansamblu metalic, format din Ti (70nm)/ Ni(175nm)/Au(320nm), pe ambele fețe ale plachetelor.

Structurile de test au fost încapsulate în capsule de tip TO39, utilizând tehnologia firului de aur pentru contactarea anodului. Catodul a fost conectat la ambaza capsulei utilizându-se o pastă pe bază de Ag. Structurile de test au fost caracterizate electric pe o gamă largă de temperaturi, în funcție de capacitățile contactului. Astfel, diodele cu Ni, având bariera Schottky ridicată, au putut fi caracterizate în domeniul 15 – 675K, iar cele cu Ti, de barieră redusă, au fost evaluate până la 500K. Măsurătorile electrice au fost efectuate utilizând o unitate de măsură Keysight B2902A și un criostat cu heliu și circuit închis de tip Janis (CCS-450), capabil să ofere o gamă largă de temperaturi, cuprinsă între 15 și 500K (fig. 1.1). Măsurătorile la temperaturi înalte (300 – 700K) s-au efectuat cu sistemul descris în [1], [1] și figura 1.2, ce cuprinde un sistem Keithley 4200 SCS, conectat la un cuptor cromatograf de tip Varian.



Fig. 1.1 Ansamblu de caracterizare la temperaturi joase pentru diodele Schottky pe SiC.

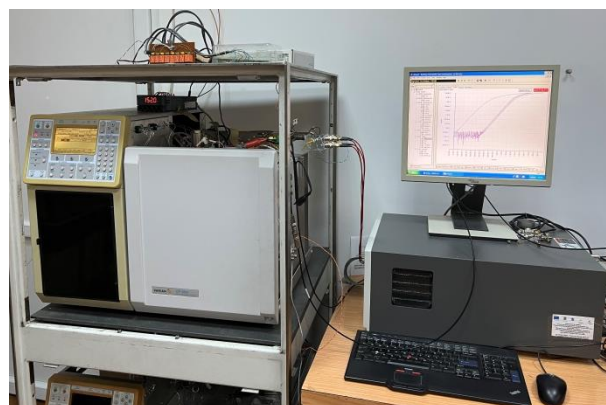


Fig. 1.2 Ansamblu de caracterizare la temperaturi înalte pentru diodele Schottky pe SiC [3].

În figura 1.3 sunt ilustrate exemple de caracteristici directe, măsurate cu cele două sisteme, pe diode cu contact de Ti, respectiv Ni.

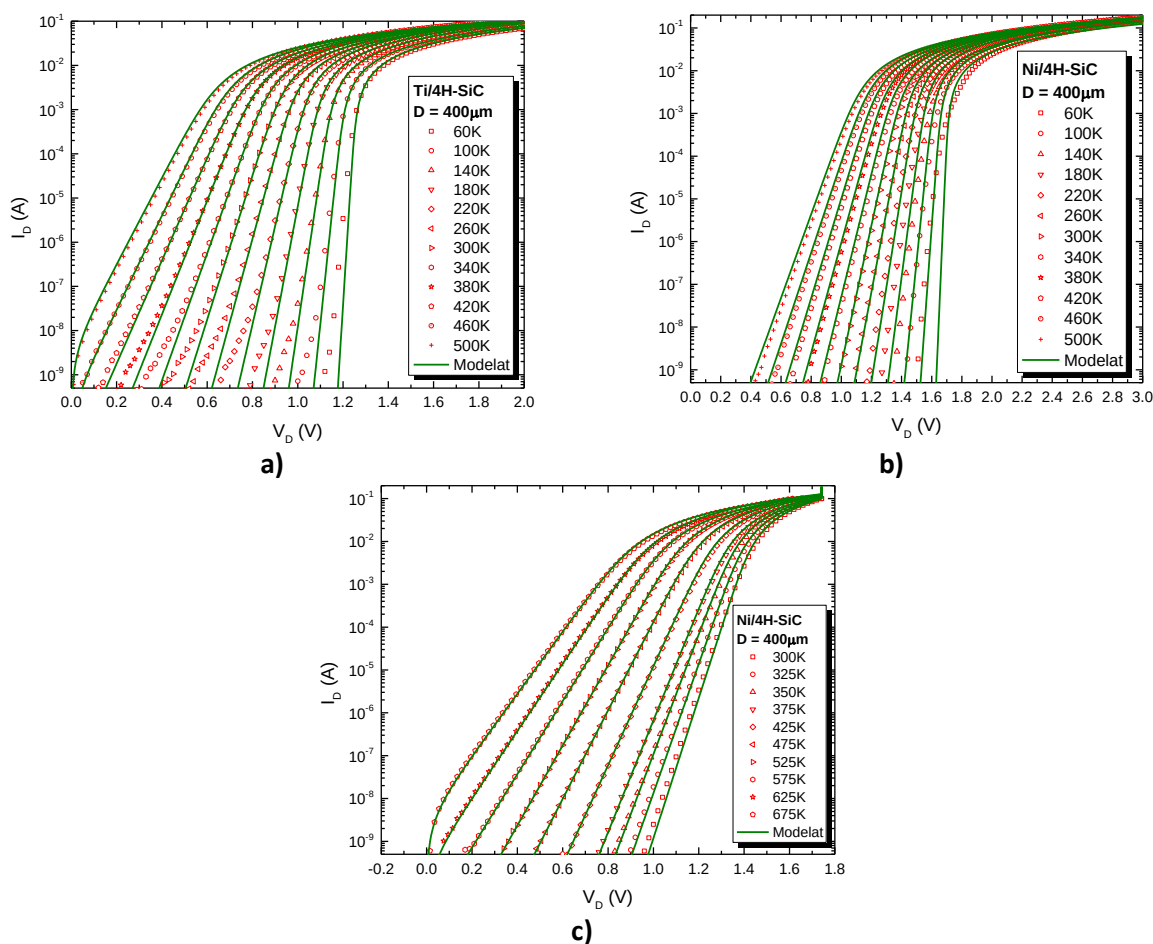


Fig. 1.3 Caracteristici directe pentru diode cu contacte de a) Ti/4H-SiC și b), c) Ni/4H-SiC în diferite game de temperaturi. Date măsurate(puncte) și modelate(linii).

Se poate observa că dispozitivele prezintă o dependență exponențială a curentului de tensiunea aplicată pe cel puțin 5 ordine de mărime. Acest tip de variație este caracteristic

mecanismului de conducție denumit *emisie termionică*. Utilizarea diodelor Schottky în aplicații practice este puternic condiționată de influența dominantă a acestui mecanism. Deși, așa cum s-a precizat anterior, diodele au putut fi caracterizate chiar de la 15K, comportamentul lor electric deviază considerabil de la cel dorit în gama 15 – 60K, după cum se poate observa în figura 1.4.

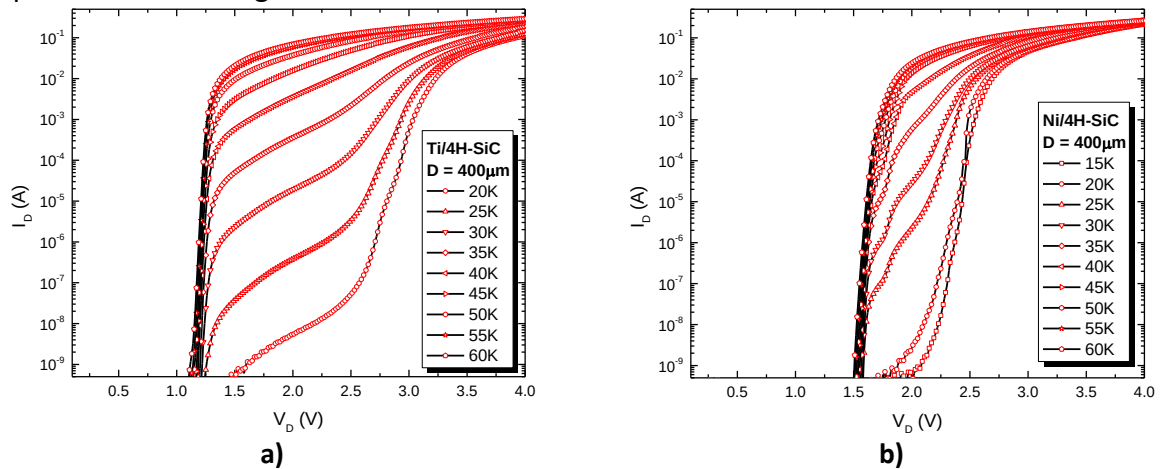


Fig. 1.4 Caracteristici directe, măsurate pe diode cu contacte de a) Ti/4H-SiC și b), Ni/4H-SiC la nivele criogenice de temperatură.

Principalii parametri de interes pentru o diodă Schottky sunt înălțimea barierei (Φ_{Bn}), factorul de idealitate (n) și rezistența serie (R_s). În cazul unui dispozitiv ideal, Φ_{Bn} și n au valori riguros constante pe întregul domeniu de temperaturi investigat [4]. Determinarea cu acuratețe a acestor parametri este mult îngreunată atunci când dispozitivul Schottky analizat prezintă neomogenități pe suprafața contactului [5]-[9]. Acest fenomen este predominant mai ales când sunt investigate diode fabricate pe semiconductori emergenți, precum SiC, deoarece tehnologia nu este suficient de matură pentru a asigura procese de fabricație stabile, cu rezultate reproductibile. Eforturile de cercetare în domeniul diodelor Schottky cu neomogenități de contact au culminat cu apariția conceptului conducției paralele [9], ce constituie o abordare intuitivă în explicarea comportamentului electric pentru astfel de dispozitive. Conform acestuia, un contact Schottky neomogen este format din zone multiple, de arii diferite, fiecare cu propria barieră [4], [6], [11], ce sunt încorporate în suprafața dorită prin proiectare. Fiecare dintre aceste zone „parazite” contribuie variabil la curentul prin dispozitiv în funcție de parametrii specifici, temperatură și tensiunea aplicată diodei. La nivele mari de polarizare și temperaturi înalte, chiar și dispozitivele cu neuniformități pronunțate tind să aibă un comportament cvasi-ideal [4].

Astfel, diodele măsurate în cadrul acestei activități au fost parametrizate inițial folosindu-se doar datele experimentale de la temperaturi înalte (300K-500K pentru diodele cu Ti și 300K-675K pentru cele cu Ni). Rezultatele extracției de parametri efectuate pe datele din fig. 1.3 sunt prezentate în tabelul 1.1. Parametrul p_{eff} este utilizat pentru evaluarea cantitativă a gradului de neuniformitate a contactului [4], [12].

Tabelul 1.1. Parametrii determinați pentru diodele cu Ti și Ni din măsurători peste 300K.

Contactul Schottky	Ti/4H-SiC	Ni/4H-SiC
Φ_{Bn} [V]	1,285	1,72
n	1,03	1,03
P_{eff}	0,39	1,18
R_s [Ω]	8-12	2,5-6

Datele din tabelul 1.1 au fost utilizate în ecuația

$$I_D \cong A_n A_S T^2 \exp\left(\frac{-\Phi_{Bn}}{V_{th}} - p_{eff}\right) \left[\exp\left(\frac{V_D - R_S I_F}{n V_{th}}\right) - 1 \right] \quad (1.1)$$

pentru a modela caracteristicile experimentale din fig. 1.3. A_n reprezintă constanta lui Richardson ($146 \text{ A/K}^2\text{cm}^2$ pentru 4H-SiC), A_S este aria nominală a contactului Schottky iar V_{th} este tensiunea termică. Concordanța calcule-măsurători este menținută doar la temperaturi peste 200K. De asemenea, sunt relevate discordanțe și la nivele scăzute de polarizare. Acest fapt a fost observat pentru majoritatea dispozitivelor investigate, indiferent de metal (Ti/Ni). Modelarea acestui tip de comportament neuniform necesită utilizarea unui model mai complex, precum cel descris în [2], [4], bazat pe ecuația

$$I_D = \sum_{i=1}^m I_{D,i} = \sum_{i=1}^m I_{S,i} \left[\exp\left(\frac{V_D - R_{S,i} I_{D,i}}{n V_{th}}\right) - 1 \right], \quad (1.2)$$

unde $I_{S,i}$ reprezintă curentul de saturație:

$$I_{S,i} = A_n A_S T^2 \exp\left(\frac{-\Phi_{Bn,i}}{V_{th}} - p_{eff,i}\right). \quad (1.3)$$

Modelul utilizat (*p-diode*, dezvoltat în UPB [4]), presupune curgerea curentului total (I_F) prin m diode Schottky ideale, conectate în paralel. Fiecare *diodă-paralel* are asociată o barieră Schottky ($\Phi_{Bn,i}$) și o rezistență serie ($R_{S,i}$) specifice. Termenii $p_{eff,i}$ reprezintă parametri de neuniformitate asociați fiecărei diode-paralel. Se consideră o singură valoare (de maxim 1,03, datorată efectelor precum forță-imagie [11]-[14]), cvasi-constantă, asociată întregii diode, pentru factorul de idealitate (n). Extracția de parametri bazată pe ecuația 1.2, transcendentă, este laborioasă și consumatoare de timp dacă se efectuează prin metode iterative. Scopul proiectului **SBD-SPECS** este de a dezvolta o soluție ce scade dramatic timpul de extracție cu acest model, utilizându-se modele implicite diferențiale.

1.2 Proiectarea fluxului tehnologic de fabricație pentru loturile de SBD.

Validarea soluției software ce urmează să fie implementate presupune caracterizarea de diode Schottky cu metale variate, pentru a se asigura versatilitate și reproductibilitate. Astfel, este necesară obținerea loturilor suplimentare de diode. Proiectarea fluxului tehnologic de fabricație pentru aceste noi loturi este descris succint în continuare.

1. Se vor utiliza plachete de SiC de tip n , politipul 4H-SiC, puternic dopat (cu o concentrație mai mare de 10^{18} cm^{-3}) și care să aibă un strat epitaxial slab dopat (sub 10^{16} cm^{-3}) necesar formării contactului Schottky. În acest strat slab dopat se va forma regiunea de sarcină spațială a diodei Schottky.
2. Vor fi utilizate tehnici de curățiri chimice speciale pentru pregătirea suprafețelor plachetelor de SiC de exemplu folosind soluția 5xDIW (apa deionizată) / 1x 29% NH_3 (amoniac) / 1x30% H_2O_2 (apa oxigenată): 75°C (10min.) + 1xHF (acid fluorhidric)/200xDIW: 25°C (15 sec.) + 6xDIW / 1x37 %HCl (acid clorhidric) / 1x30% H_2O_2 : 25°C (10min.).
3. Ariile diodelor Schottky vor fi definite într-un oxid de câmp, cu o grosime suficient de mare astfel încât să ofere o izolare din punct de vedere electric între diodele definite; o grosime a oxidului cuprinsă între 0,5 - 1 μm reprezintă o soluție optimă pentru aceasta etapă, ținând cont de faptul că diodele se vor caracteriza, în

special, în polarizare directă, la tensiuni mici ($<5V$). Astfel, nu există riscul unei străpungeri a oxidului sau apariția unor curenți de pierderi mari. Vor fi definite configurații circulare, cu arii diferite, având diametre cuprinse între 200 - 400 μm , utilizând tehnici de fotolitografie. Pentru a obține o terminație cu un profil de rampă, se vor depune două straturi de oxid. Primul va fi supus unui tratament termic la o temperatură de minim 950°C pentru a obține un strat mai dens, iar cel de-al doilea se va depune standard, la o temperatură de 750°C. În timpul corodării, din cauza densității diferite a celor două straturi de oxid, se va obține un profil de rampă care are rolul de a asigura o densitate mai uniformă a curentului prin dispozitiv și o tensiune de străpungeră mai mare.

4. Primul proces de fotogravură are rolul de a deschide ferestre în oxidul de câmp și de a defini ariile active ale diodelor Schottky pe SiC. Un proces de fotogravură constă în următorii pași: depunere fotorezist și tratament termic, alinierea măștii fotolitografice, expunere la UV a fotorezistului, dezvoltare fotorezist, corodarea oxidului prin masca fotolitografică. Corodarea va avea loc fără protecția spatelui plachetei pentru a elimina și oxidul care a fost crescut acolo în timpul depunerii/tratamentului termic efectuat. Urmează îndepărtarea fotorezistului în acetonă.
5. Pentru obținerea unei rezistențe serie mici, calitatea înaltă a contactului ohmic este esențială. Acesta va fi realizat pe spatele plachetelor de SiC prin depunerea unui metal sau a unor straturi metalice succesive (de ex.: doar Ni sau un ansamblu metalic format din Ti/Ni). După depunerea metalului pe spatele plachetelor de SiC urmează un tratament termic post-metalizare, în atmosferă inertă și la o temperatură ridicată ($T > 1000^{\circ}C$). Aceste etape tehnologice trebuie să ofere o rezistivitate specifică de contact scăzută;
6. Contactele Schottky vor fi definite prin tehnici de fotolitografie, care vor avea atât arii, cât și metale diferite (Ti, Ni, Cr, Pt). După depunerea acestora, se va optimiza tratamentul termic post-metalizare pentru fiecare metal în parte, deoarece formarea siliciurilor metalice are loc la temperaturi diferite. Gradul de neomogenitate al unei diode Schottky este cel mai puternic influențat de acest pas.
7. Pentru realizarea pad-urilor de contact și finalizarea contactului ohmic se vor depune succesiv, pe ambele fețe ale plachetelor, compoziții metalice formate din Ti/Ni/Au. Stratul de Ti oferă o aderență bună la siliciura metalică rezultată în urma tratamentului termic post-metalizare. Nichelul reprezintă un strat de barieră, ce oprește inter-difuzia celorlalte metale la temperaturi înalte, iar aurul are un grad foarte scăzut de oxidare și oferă un contact electric cu rezistivitate foarte scăzută, ținând cont și de procesul de încapsulare (capsula TO39 are terminale aurite).

1.3 Analiza și selecția tehnicilor optime implicate diferențiabile pentru parametrizarea diodelor Schottky pe SiC.

Studiind ecuațiile 1.2 și 1.3, se disting trei tipuri de termeni:

- Constante, cu valoare bine stabilită (A_n, A_s, V_{th})

- Date obținute în urma măsurătorilor (V_D, I_D, T)
- Parametri ce trebuie estimați cu precizie ($\Phi_{Bn}, p_{eff}, n, R_S$)

În mod ideal, se dorește determinarea unei soluții analitice pentru ecuația dată, deoarece aceasta ar avea avantajele de a fi exactă și rapid de calculat. Însă, cum modelul ales nu este tratat analitic, trebuie recurs la abordarea alternativă de utilizare a optimizării numerice [15] pentru a căuta iterativ soluția optimă. În acest scop, se formulează problema rezolvării ecuației ca o sarcină de optimizare și se utilizează informațiile de gradient pentru a ghida căutarea.

Pentru cazul nostru concret, definim următoarea problemă de optimizare cu constrângeri:

Minimizează θ în $\sum_k (\log I_{D_k} - \log \sum_{i=1}^m I_{D,i_k})^2$ dacă $I_{D,i_k} = \psi(I_{D,i_k}, V_k, T_k, \theta_i)$, unde

- Ψ este ecuația 1.2;
- θ reprezintă toți parametrii care trebuie extrași;
- k este indexul măsurătorii.

De asemenea, subliniem faptul că funcția de cost minimizată este eroarea pătratică medie și este definită în spațiul logaritm, ceea ce asigură că măsurătorile la curenți mici nu sunt complet ignorate.

Abordarea comună pentru a rezolva o astfel de problemă de optimizare implică calcularea gradientilor și apoi utilizarea acestor informații într-un algoritm de optimizare iterativ, cum ar fi algoritmul gradientului [16] sau metoda LBFGS (limited-memory BFGS, Broyden–Fletcher–Goldfarb–Shanno) [17]. Cu toate acestea, calcularea manuală a gradientilor este o sarcină dificilă și predispusă la erori. Din acest motiv, ne vom baza pe diferențierea automată – o tehnică ce evaluează gradientii numeric, în mod eficient, utilizând teorema derivării funcțiilor compuse (*chain rule*). Aceste tehnici au fost dezvoltate și popularizate în contextul învățării profunde (*deep learning*) pentru optimizarea funcțiilor multi-dimensionale și non-liniare.

În timp ce problemele uzuale de optimizare implică doar funcții, în cazul nostru, avem de-a face cu o optimizare imbricată (cunoscută și sub denumirea de “optimizare binivel” [18] sau “rețele declarative” [19]) cu problema interioară care reprezintă ecuația unui punct fix, de forma $I = \psi(I)$. Pentru aceasta, se va studia posibilitatea implementării tehnicii mai avansate a diferențierii implicite.

1.3.1 Diferențierea automată pentru optimizarea numerică

Calcularea gradientului unei funcții este esențială pentru învățarea automată și tehnicile de optimizare. Pe măsură ce modelele devin mai complexe (și totodată mai flexibile), devine din ce în ce mai dificil să se calculeze derivata în mod manual. Mai mult, această abordare împiedică explorarea în ceea ce privește modelul și topologia acestuia, deoarece orice modificare trebuie să fie reflectată și în calculul derivatei.

Diferențierea automată (*automatic differentiation*) ne oferă o familie de algoritmi care calculează automat gradientii unei expresii date. Fundamentul acestei tehnici este teorema derivării funcțiilor compuse, care permite determinarea gradientului într-un mod compozițional: cunoașterea gradientilor unor expresii primitive permit calculul gradientilor unei funcții arbitrare, compuse din aceste expresii de bază.

Există două variante principale de diferențiere automată – modul direct (*forward mode*) și modul invers (*reverse mode*) – ambele având compromisuri în ceea ce privește cerințele de calcul și memorie. Mai exact, modul direct este eficient atunci când funcția pe care o diferențiem are mai multe ieșiri decât intrări și nu trebuie să stocheze activări intermediare, în timp ce modul invers este eficient atunci când funcția are mai multe intrări decât ieșiri și trebuie să stocheze calcule intermediare. Din punctul de vedere al învățării profunde moderne (unde spațiul de intrare este, în general, de dimensiuni foarte mari, spre exemplu, rețelele au zeci de milioane de parametri, iar ieșirea este un singur număr, valoarea funcției de cost), diferențierea automată de tip mod invers oferă o modalitate de a calcula gradientii eficient. Dimpotrivă, modul direct ar necesita multiple rulări ale algoritmului de diferențiere, ceea ce este probabil prohibitiv din punct de vedere computațional pentru multe dintre aplicațiile curente.

O instanțiere comună a diferențierii automate este algoritmul *backpropagation* [20], care este o variantă a algoritmului de diferențiere automată de tip mod invers pentru cazul când rezultatul funcției este un scalar (un singur număr); acesta este de obicei cazul pentru învățarea profundă, în care o funcție de cost este utilizată pentru a evalua potrivirea combinației curente de parametri.

Implementarea corectă a tehnicii diferențierii automate reprezintă un proces delicat [21], însă eforturile recente de cercetare [22]-[24] s-au focalizat pe dezvoltarea unor baze teoretice mai bune, conform cărora validarea poate fi mai ușor efectuată.

Chiar dacă diferențierea automată este, în general, un algoritm complex, aceasta este ușor de utilizat datorită numeroaselor sale implementări în mai multe limbaje de programare, cum ar fi PyTorch [25] și Jax [26] pentru Python, ReverseDiff și Zygote [27] pentru Julia; Tensorflow [28], care funcționează pentru mai multe limbaje, inclusiv Python, C++, Java, Go, JavaScript, R. Toate aceste biblioteci facilitează crearea de modele complexe și flexibile și sunt un motiv important pentru creșterea rapidă a aplicațiilor de deep learning.

1.3.2 Diferențierea implicită pentru optimizarea binivel

Optimizarea standard și diferențierea automată implică *funcții*, care sunt compuse într-o singură funcție de cost. Cu toate acestea, pentru o anumită clasă de probleme, ne interesează relații mai generale între intrări și ieșiri, precum constrângeri (neliniare) sau probleme imbricate de optimizare. Mai precis, presupunem următoare două tipuri de formulări generale:

Minimizează u în $f(u, w)$ dacă $g(u, w) = 0$

Sau

Minimizează u în $f(u, w^*)$ dacă $w^* \in \arg \min_w g(u, w)$

unde

- f este problema de optimizare exterioară;
- g este constrângerea (neliniară) sau problema interioară de optimizare;
- u sunt parametrii externi;
- w sunt parametrii interni.

Acest cadru apare sub diferite denumiri în literatură, *rețelele declarative* [19] și *optimizare binivel* sau *imbricată* [18] fiind cele mai uzuale. Multe probleme de învățare automată pot fi interpretate în acest cadru, astfel încât dezvoltarea de soluții generale are potențialul de a rezolva multe sarcini de interes. Mai important, problema noastră poate fi și ea formulată în acest mod, unde constrângerea neliniară corespunde ecuației punctului fix.

Există două tipuri de soluții pentru această problemă: diferențierea derulată (*unrolled differentiation*) și diferențierea implicită. Diferențierea derulată rezolvă problema interioară cu un algoritm de optimizare și apoi utilizează diferențierea automată pentru a propaga gradientii prin iterațiile algoritmului de rezolvare a ecuației. Diferențierea implicită se bazează pe teorema funcției implicite [29] și ne oferă o modalitate eficientă de a calcula derivata soluției în raport cu parametrii.

1.3.2.1 Aplicabilitate

În această subsecțiune prezentăm mai multe exemple de sarcini de învățare automată care pot fi formulate ca o problemă de optimizare de tip binivel. Aceste numeroase aplicații arată importanța abordării, oferind indicii privitoare la modul de utilizare a tehnicii de diferențiere implicită pentru problema noastră.

În tabelul de mai jos rezumăm șapte sarcini, listând obiectivele lor interne, parametrii interni, obiectivele externe și parametrii externi.

Sarcina	Obiectiv interior	Parametri interni	Obiectiv exterior	Parametri externi
optimizarea hiper-parameterilor	funcția de cost pe datele de antrenare	ponderile rețelei	funcția de cost pe datele de validare	hiper-parametri
modele profunde cu punct de echilibru	soluție de tip punct fix	ponderile rețelei	funcția de cost	alte ponderi ale rețelei
distilarea datelor	funcția de cost pe datele distilate	ponderile rețelei	funcția de cost pe datele originale	set de date distilate
învățarea modelelor structurate	predicție structurată	ieșirea structurată	funcția de cost	ponderile rețelei
învățarea de dicționar pe baza unei sarcini	învățarea dicționarului	dicționarul și reprezentarea	funcția de cost pentru sarcina finală	ponderile rețelei
funcții de influență	funcția de cost pe datele de antrenare	ponderile rețelei	impactul eșantionaelor	contribuția fiecărui eșantion
meta-învățare	funcția de cost pentru sarcinile	parametri specifici fiecărei sarcini	funcția de cost pe datele de testare	meta-parametri

Notație. În cele ce urmează (dacă nu este specificat altfel), vom presupune un scenariu tipic de învățare automată. Se consideră un set de perechi de date ce constituie eșantioane intrare-ieșire $D = \{(x_i, y_i)\}$ (unde x_i reprezintă intrările iar y_i ieșirile asociate). Setul de date are, de obicei, trei părți, corespunzătoare antrenării (*training*, tr), validării (*validation*, va) și testării (*testing*, te). Scopul este de a învăța o funcție f_θ (parametrizată prin ponderile θ), care asociază intrarea x la ieșirea y astfel încât o funcție de cost L să fie minimizată. Funcția f_θ este, de obicei, implementată ca o rețea neuronală (compoziție alternativă de neliniarități și liniarități) și este calibrată pe partea de date de antrenare (tr). Partea de date de validare este utilizată pentru a evalua corectitudinea parametrilor și pentru a căuta hiper-parametri optimi, care pot controla arhitectura (*de ex.*, numărul de straturi, tipuri de straturi) sau optimizarea (*de ex.*, rata de învățare).

Optimizarea hiper-parametrilor

Unul dintre cele mai populare exemple de optimizare binivel în învățarea automată este sarcina de optimizare a hiper-parametrilor. De obicei, antrenarea unei rețele neuronale constă în optimizarea parametrilor care controlează direct predicția (aceștia sunt cunoscuți și sub numele de ponderi). Cu toate acestea, există o suită de parametri suplimentari care controlează alte aspecte ale rețelei, cum ar fi rata de învățare a optimizatorului, adâncimea rețelei, numărul de canale pentru un strat, preprocesarea aplicată intrărilor. Acești parametri sunt numiți *hiper-parametri* și, deși uneori sunt considerați secundari, ei sunt cruciali pentru atingerea unei performanțe cât mai bune [30]. Hiper-parametrii sunt, de obicei, optimizați separat, folosind abordări tip “black-box” (cum ar fi căutarea exhaustivă sau căutarea aleatorie [31]). Această abordare folosește informații de gradient doar pentru a găsi ponderile rețelei, dar hiper-parametrii sunt determinați prin încercări multiple, fără a utiliza informații de gradient. Procesul se potrivește foarte natural ca problemă de optimizare binivel și poate fi astfel rezolvată mai eficient. Din punct de vedere matematic, avem două obiective: cel exterior optimizează performanța pe setul de validare prin potrivirea hiper-parametrilor, în timp ce cel interior optimizează performanța pe partea de test, prin potrivirea ponderilor rețelei. Ca în cazul optimizării binivel, parametrii interni (ponderile rețelei) depind implicit de parametrii externi (hiper-parametrii). Formularea devine:

Minimizează λ în $\sum_{x,y \sim D_{va}} L(f_{\theta^*, \lambda}(x), y)$ dacă $\theta^* \in \arg \min_w \sum_{x,y \sim D_{tr}} L(f_{\theta^*, \lambda}(x), y)$

Deși această formulare a fost aplicată inițial pentru modele de complexitate redusă [32], lucrări mai recente au scalat-o la milioane de hiper-parametri [33].

Modele profunde cu punct de echilibru

Modelele profunde cu punct de echilibru (*deep equilibrium models* [34]) rezultă din ducerea la limită a două caracteristici cheie ale rețelelor neuronale moderne – adâncimea lor și partajarea ponderilor:

1. *Rețele mai adânci.* Conexiunile reziduale [35] și tehnicile de normalizare [36], [37] au permis construirea de rețele din ce în ce mai adânci, care excelează din punct de vedere al expresivității [38] și la modelarea conceptelor relevante din punct de vedere semantic [39].
2. *Partajarea ponderilor între straturi.* Rețelele neuronale convoluționale [40], rețelele neuronale recurente [41] și alte abordări aferente [42]-[44], toate arată beneficiile reutilizării ponderilor în mai multe locații și niveluri din rețea.

Ducând la extrem aceste două observații, obținem o rețea de adâncime infinită cu ponderi comune pe toate straturile: $f_{\theta}(x) = w^T NN_{\theta}(x, NN_{\theta}(x, \dots NN_{\theta}(x, z_0)))$, unde z_0 este un vector de inițializare aleatorie iar w este un strat de predicție final, care proiectează activarea în spațiul de ieșire.

Deoarece iterația repetată converge de obicei către un punct fix, antrenamentul rețelei poate fi exprimat și într-o formulare pe două niveluri:

Minimizează θ, w în $\sum_{x,y \sim D_{tr}} L(w^T z^*, y)$ dacă $z^* = \text{PunctFix}(NN_{\theta}(x, -))$

Observăm că această formulare este similară cu cea pentru problema noastră, în sensul că ambele implică o soluție de tip punct fix. Diferența este că, în acest caz, se antrenează o rețea generică (NN_{θ} este un strat tipic, cum ar fi perceptronul multistrat sau convoluțional), în timp ce parametrii din modelul nostru (*p-diode*) sunt plauzibili din punct de vedere fizic.

Distilarea datelor

Distilarea datelor [45] își propune să creeze un set mic de date sintetice, astfel încât un model antrenat pe aceste date să funcționeze bine pe datele originale (reale). Drept urmare, cu această abordare este posibil să comprimăm seturi de date de zeci de mii de eșantioane în doar câteva exemple. În cazul unei sarcini de clasificare, generăm un exemplu sintetic $\bar{x}_k$ pentru fiecare clasă k ; formularea ca problemă binivel este:

Minimizează $\bar{x}$ în $\sum_{x,y \sim D_{tr}} L(f_{\theta^*}(x), y)$ dacă $\theta^* = \arg \min_{\theta} \sum_k L(f_{\theta}(\bar{x}_k), k)$

Învățarea modelelor structurate

În multe probleme de învățare automată, rezultatul are mai multă structură decât un singur număr (cazul sarcinii de regresie) sau o etichetă (cazul sarcinii de clasificare); de exemplu, în problemele de vedere artificială, putem dori să prezicem poziția membrilor corpului unei persoane (o secvență de coordonate) sau în procesarea limbajului natural dorim să prezicem o propoziție (o secvență de cuvinte). În aceste cazuri, pentru a obține acest tip de ieșire structurată (secvențe) definim, de obicei, o funcție de compatibilitate E_{θ} între intrare și ieșire. Pentru a obține predicția $\hat{y}$ trebuie să rezolvăm o problemă de optimizare (internă) care caută în spațiul de ieșire valoarea care maximizează compatibilitatea:

Minimizează θ în $\sum_{x,y \sim D_{tr}} L(y, \hat{y})$ dacă $\hat{y} = \arg \max_y E_{\theta}(x, y)$

Antrenarea modelelor structurate poate fi încadrată ca o problemă de optimizare pe două niveluri ce poate fi rezolvată cu o metodă similară diferențierii derulate (discutată mai jos) [46].

Învățarea de dicționar pe baza unei sarcini

Învățarea unui dicționar constă în factorizarea matricii datelor X în produsul a două matrici: un dicționar de atomi $A = \{a_1, \dots, a_j, \dots\}$ și reprezentarea lor R (o matrice rară care indică din ce atomi se poate reconstitui un eșantion de date):

$$x_i = \sum_j R_{ij} a_j$$

Învățarea unui dicționar pentru o sarcină [47] optimizează această factorizare matriceală către o sarcină finală (cum ar fi clasificarea peste reprezentările R). Încadrată ca o problemă de optimizare pe două niveluri, formularea este următoarea:

Minimizează θ în $\sum_{x,y \sim D_{tr}} L(y, f_{\theta}(r^*))$ dacă $A^*, R^* = \arg \min_{A,R} \|X - AR\|_F^2 + \lambda \|R\|_*$

Algoritmul propus în [47] funcționează pentru alegeri particulare ale funcției de cost și a normelor, dar diferențierea implicită ne permite o soluție mai generală.

Funcții de influență

Funcțiile de influență [48] ne oferă o modalitate de a măsura cât de mult un eșantion din setul de date de antrenare afectează performanța modelului învățat. Abordarea naivă ar consta în re-antrenarea modelului într-un cadru de excludere (prin excluderea fiecărui punct de antrenament în mod iterativ) și apoi calcularea funcției de cost pe datele de validare pentru fiecare dintre modelele rezultate; cu cât diferențele funcției de cost sunt mai mari (comparativ cu modelul antrenat pe toate datele), cu atât un punctul respectiv este mai important. Cu toate acestea, această abordare simplă este prohibitiv de lentă, deoarece timpul crește liniar cu numărul de puncte de date din setul de antrenament. O modalitate mai eficientă este de a defini următoarea problemă pe două niveluri:

Minimizează η în $\sum_{x,y \sim D_{va}} L(y, f_{\theta^*}(x^*))$ dacă $\theta^* = \arg \min_{\theta} \sum_{i=1}^{N_{tr}} \eta_i L(y_i, f_{\theta}(x_i))$

unde η_i este ponderea eșantionului i în procesul de antrenare; un η de forma [...1,1,0,1...] ar corespunde antrenării fără un singur punct. În acest caz, gradientul problemei exterioare aproximează abordarea naivă [48].

Meta-învățare

În sarcina de meta-învățare [49] sunt antrenați algoritmi care generalizează peste mai multe sarcini. Fiecare sarcină t este, de obicei, definită de un set mic de date D^t și scopul este ca, date câteva mostre adnotate pentru o sarcină nouă, metoda să fie capabilă să învețe. Ideea principală este de a avea un set de parametri θ partajați de toate sarcinile, care să faciliteze transferul cunoștințelor de la o sarcină la alta; acești parametri sunt cunoscuți și ca meta-parametri și sunt antrenați împreună cu parametrii specifici fiecărei sarcini ϕ_t . Meta-învățarea poate fi formulată ca o problemă de optimizare pe două niveluri, după cum urmează [50]:

Minimizează θ în $\sum_{x,y \sim D_{va}^t} L(g_{\phi_t^*}(x), y)$ dacă

$$\phi_t^* = \arg \min_{\phi_t} \sum_{x,y \sim D_{tr}^t} L(g_{\phi_t}(x), y) + \|\phi_t - \theta\|_2^2$$

1.3.2.2 Diferențierea derulată

Revenim la problema generală de optimizare pe două niveluri și discutăm una dintre abordările propuse: diferențierea derulată (*unrolled differentiation*). Problema pe care încercăm să o rezolvăm are următoarea formă:

Minimizează u în $f(u, w^*)$ dacă $w^* \in \arg \min_w g(u, w)$

Tehnica diferențierii derulate este cea mai directă abordare; constă inițial în căutarea soluției problemei interioare prin aplicarea unui algoritm de optimizare iterativ T pentru K pași:

$$w_0 \xrightarrow{T} w_1 \xrightarrow{T} \dots \xrightarrow{T} w_K$$

Această metodă va produce o valoare aproximativă w_K a valorii optime w^* , iar dacă T este diferențiabil, putem folosi diferențierea automată pentru a propaga înapoi gradientii prin cei K pași de optimizare. Această abordare are un mare avantaj – simplitatea sa – dar suferă și

de câteva dezavantaje. În primul rând, este o metodă inefficientă în ceea ce privește memoria: deoarece modul invers de diferențiere automată stochează activările intermediare, gradul de încărcare al memoriei crește liniar cu numărul de pași de derulare K . În al doilea rând, pentru a aplica această metodă, trebuie să implementăm algoritmul de optimizare interior în cadrul librăriei de diferențiere automată, ceea ce limitează utilizarea opțiunilor de optimizare, care sunt adesea implementate în alte librării sau alte limbaje de programare.

1.3.2.3 Diferențierea implicită

Tehnica de diferențiere derulată calculează în mod explicit gradientii problemei de optimizare interioară. Diferențierea implicită, pe de altă parte, oferă o abordare în care calculul gradientului se face implicit: soluția problemei interioare poate fi privită ca o *funcție* a parametrilor și ca atare putem calcula gradientul funcției de cost relativ la parametrul.

Teoria diferențierii implicite. În continuare se prezintă teoria diferențierii implicite [29], [51].

Fiind dată o funcție $f: R^p \times R^n \rightarrow R^n$, putem interpreta o soluție a funcției (adică a_0 și z_0 astfel încât $f(a_0, z_0) = 0$) ca un mod de a specifica un sistem de n ecuații (neliniare) între n variabile (notate cu z), care este parametrizată de p alte variabile (notate cu a). Teorema funcției implicite afirmă că, dacă reușim să găsim această soluție (a_0, z_0) , în anumite condiții de regularitate (descrise mai jos), putem defini o *funcție* $z^*: R^p \rightarrow R^n$, care asociază variabila a variabilei z , iar acea funcție este *diferențiabilă*; prin urmare, putem calcula derivata soluției în raport cu parametrul p care guvernează sistemul de ecuații.

Prezentăm mai jos declarația formală a teoremei:

Fiind dată o funcție $f: R^p \times R^n \rightarrow R^n$ și două puncte a_0 și z_0 astfel încât

1. cele două puncte reprezintă o soluție a ecuației $f(a_0, z_0) = 0$,
2. Jacobianul lui f în raport cu z este nesingular în punctul soluție: $\det \partial_1 f(a_0, z_0) \neq 0$ atunci există mulțimile deschise $S_{a_0} \subset R^p$ (conținând a_0) și $S_{z_0} \subset R^n$ (conținând z_0) și o funcție continuă unică $z^*: R^p \rightarrow R^n$ astfel încât

1. $z^*(a_0) = z_0$
2. $f(a, z^*(a)) = 0$ pentru orice $a \in S_{a_0}$.
3. funcția z^* este diferențiabilă pentru orice $a \in S_{a_0}$.

Intuitiv, teorema funcției implicite spune că în vecinătatea unei soluții date, putem găsi o funcție z^* (de la parametri la soluție). Deoarece această funcție este, de asemenea, diferențiabilă, putem calcula gradientul său și îl putem folosi în metode standard de optimizare bazate pe gradient.

Calculul gradientului funcției implicite. Deși teorema funcției implicite nu precizează gradientul funcției implicite, acesta nu este dificil de calculat. Pornim de la ecuația:

$$f(a, z^*(a)) = 0, \forall a \in S_{a_0}.$$

Calculând derivata acestei ecuații obținem:

$$\partial_0 f(a, z^*(a)) + \partial_1 f(a, z^*(a)) \circ \partial z^*(a) = 0,$$

$$\text{de unde rezultă } \partial z^*(a) = -[\partial_1 f(a, z^*(a))]^{-1} \partial_0 f(a, z^*(a)).$$

Dacă trebuie să luăm în considerare derivata la soluția (a_0, z_0) , atunci obținem

$$\partial z^*(a_0) = -[\partial_1 f(a_0, z_0)]^{-1} \partial_0 f(a_0, z_0).$$

Observăm două aspecte cruciale care fac diferențierea implicită extrem de eficientă în practică:

1. Derivata (la punctul de soluție) are nevoie doar de informații despre soluție, așa că nu trebuie să știm cum s-a ajuns acolo (spre deosebire de diferențierea derulată). Ca atare, putem folosi soluții de ultimă generație care pot fi implementate în alte limbaje sau paradigme de programare.
2. Nu trebuie să aflăm în mod explicit funcția care leagă parametrii de soluție, deoarece derivata este exprimată numai în termeni de derivate ale funcției date f . Ca atare, putem aplica diferențierea automată pentru a calcula gradientii.

În continuare, prezentăm modul în care teorema funcției implicite ajută la două dintre sarcinile pe care le-am discutat în secțiunile anterioare: A · optimizare cu soluție fixă; B · optimizare pe două niveluri.

A · Optimizare cu soluție fixă. Calcularea soluției fixe a unei funcții Φ poate fi formulată în mod natural folosind diferențierea implicită. Dorim să găsim z astfel încât $\Phi(a, z) = z$, ceea ce poate fi rescris ca $f(a, z) = \Phi(a, z) - z = 0$ și, astfel, ajungem în forma dorită. Gradientul este: $\partial z^*(a_0) = -[\partial_1 f(a_0, z_0) - I_D]^{-1} \partial_0 \phi(a_0, z_0)$.

B · Optimizare pe două niveluri. În cazul optimizării pe două niveluri, problema de optimizare interioară are forma $\arg \min_z f(a, z)$.

Optimul este atins pentru $\partial_1 f(a, z) = 0$.

Astfel, problema ia forma unei diferențieri implicite, ecuațiile neliniare fiind date de gradient. Gradientul funcției implicite z^* este:

$$\partial z^*(a_0) = -[\partial_1 \partial_1 f(a_0, z_0)]^{-1} \partial_0 \partial_1 \phi(a_0, z_0).$$

De observat că acest gradient implică matricea Hessiană (adică derivata de ordinul doi).

1.3.2.4 Utilitare pentru diferențierea implicită

Similar cu modul în care diferențierea automată face mecanic procesul de calcul al gradientului, recent s-au dezvoltat utilitare pentru a automatiza diferențierea implicită. Aceste metode simplifică definirea modelelor care conțin funcții implicite, utilitarele numerice având grijă de implementarea corectă a calculului derivatei. Astfel au apărut librării noi în această direcție, cum ar fi, în special cele din ecosistemul Python: JaxOpt pentru Jax [52], [TorchOpt](#) și [HyperTorch](#) pentru Pytorch.

1.3.3 Soluție alternativă

În timp ce diferențierea implicită oferă cea mai directă abordare, o altă modalitate de rezolvare a acestei probleme este integrarea constrângerilor de egalitate în funcția obiectiv, sub forma unui termen de penalizare. Apoi, se optimizează atât parametrii θ cât și curenții $I_{D,i}$, corespunzători fiecăreia dintre cele m diode individuale, dispuse în paralel:

Minimizează $\theta, I_{D,i}$ astfel încât

$$\sum_k (I_{D_k} - \sum_{i=1}^m I_{D,i_k}) + \beta \sum_k \sum_{i=1}^m (I_{D,i_k} - \psi(I_{D,i_k}, V_{D_k}, T_k, \theta_i))^2$$

unde β este un termen de penalizare. Cu cât este mai mare, cu atât se pune mai mult accent pe rezolvarea precisă a ecuației cu punct fix. De obicei, rezolvăm succesiv această problemă prin creșterea lui β în factori de 2. Se va investiga și posibilitatea implementării acestei metode, însă este de așteptat să fie dificil de optimizat cu cât β este mai mare ($\beta \rightarrow \infty$ pentru o determinare exactă a parametrilor), deoarece algoritmul va trebui să găsească puncte fixe din ce în ce mai precise.

1.4 Diseminarea rezultatelor

În etapa I, s-au publicat 2 lucrări la International Semiconductor Conference (CAS 2022) și s-a prezentat o lucrare sub formă de poster la International Conference on Silicon Carbide and Related Materials (ICSCRM 2022) (**2 participări la conferințe**):

1. R. Pascu, G. Pristavu, M. Stoian, C. Romanitan, M. Kusko, F. Draghici, D.-T. Oneata, G. Brezeanu, "The effect of forming gas annealing on Al/Ti/n-Si contacts," 2022 *International Semiconductor Conference (CAS)*, 2022, pp. 127-130, doi: 10.1109/CAS56377.2022.9934472.
2. V. Moise, F. Draghici, G. Pristavu, R. Pascu, D. -T. Oneata, G. Brezeanu, "Intelligent Temperature Sensor with SiC Schottky Diode," 2022 *International Semiconductor Conference (CAS)*, 2022, pp. 123-126, doi: 10.1109/CAS56377.2022.9934369.
3. V. Moise, F. Draghici, G. Pristavu, R. Pascu, F. Mitu, G. Brezeanu, "Wide Range Temperature Sensor with SiC Schottky Diode – Error Source Analysis," prezentare poster la The 19th edition of the International Conference on Silicon Carbide and Related Materials (ICSCRM), 11-16 Sept., 2022.

Sumar al progresului:

Toate activitățile prevăzute la această etapă sunt îndeplinite, având următoarele rezultate:

1. **Raport de cercetare** (prezentul material).
2. **Două participări la conferințe** (CAS 2022, ICSCRM 2022) în urma cărora au fost publicate în proceedings două lucrări, iar una a fost prezentată ca poster (conform secțiunii 1.4)
3. **O pagină Web** (<https://www.dcae.pub.ro/proiecte/17/sbd-specs/>)

Rezumat executiv al activităților realizate în perioada de implementare

Titlul proiectului: Instrument Software pentru Extracția Unitară a Parametrilor și Caracterizarea Diodelor Schottky, bazate pe Modele Implicite Diferențiabile Emergente

Obiectivele generale ale proiectului

- Validarea și completarea ecuațiilor modelului p-diode pentru a permite caracterizarea diodelor Schottky pe domenii de temperaturi de lucru cât mai largi (de la nivele criogenice până la 675K).
- Implementarea unor algoritmi de extracție a parametrilor, bazați pe tehnici implicite diferențiabile, cu modelul p-diode subiacent. Dezvoltarea unui produs software pentru caracterizare unitară a diodelor Schottky.
- Fabricarea de diode Schottky pe SiC cu diverse arii active și contacte metal-semiconductor (Ni, Pt, Cr, Ti / 4H-SiC).
- Dezvoltarea unei baze de date cu măsurători de caracteristici directe ale diodelor Schottky obținute.

Etapa 1. Măsurători electrice preliminare, analiza tehnicilor optime de parametrizare și proiectarea fluxului tehnologic.

Etapa I, derulată în 2022, a urmărit formarea unei baze cuprinzătoare de măsurători pe diode Schottky pe SiC și parametrizarea inițială a acestora pentru a identifica particularitățile și proprietățile comune. Măsurătorile au fost efectuate pe game extinse de temperaturi, mult peste limita semiconductorilor convenționali, pentru a se putea obține o parametrizare cât mai relevantă.

S-a elaborat un proces tehnologic de fabricație pentru noi loturi de dispozitive, în vederea extinderii familiilor de diode pe care soluția software de caracterizare le poate analiza.

În principal, activitatea de cercetare s-a focalizat pe identificarea tehnicilor adecvate de extracție și optimizare a parametrilor diodelor Schottky, mai ales diferențierea implicită pentru optimizarea binivel, prin evaluarea diferitelor problematizări și formulări pentru problemele exterioară și interioară.

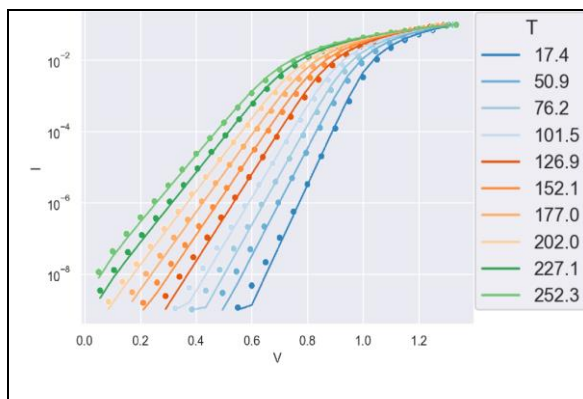


Fig. 1 Caracteristici directe și modelare teoretică pentru o diodă Schottky cu contact de Ti/4H-SiC la diverse temperaturi.

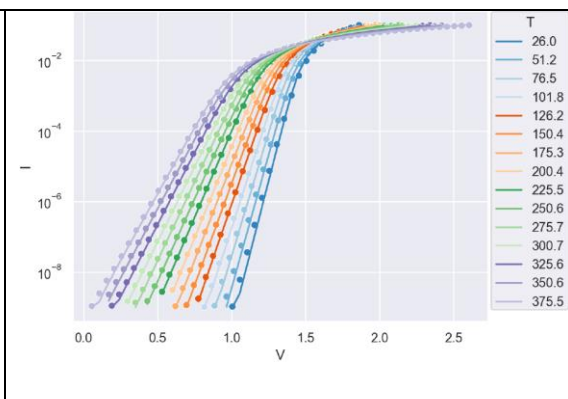


Fig. 2 Caracteristici directe și modelare teoretică pentru o diodă Schottky cu contact de Ni/4H-SiC la diverse temperaturi.

Bibliografie

- [1] F. Draghici et al., "400 °C sensor based on Ni/4H-SiC schottky diode for reliable temperature monitoring in industrial environments," *Sensors (Switzerland)*, vol. 19, no. 10, p. 2384, 2019.
- [2] R. Pascu et al., "60–700 K CTAT and PTAT Temperature Sensors with 4H-SiC Schottky Diodes," *Sensors*, vol. 21, p. 942, 2021.
- [3] Brezeanu, G.; Badila, M.; Draghici, F.; Pascu, R.; Pristavu, G.; Craciunoiu, F.; Rusu, I.; „High temperature Sensors Based on Silicon Carbide (SiC) Devices”, in *Proc. of the 38th International Semiconductor Conference*, 12-14 Oct. 2015, Sinaia, Romania, pp.3-10.
- [4] G. Brezeanu, G. Pristavu, F. Draghici, R. Pascu, F. Della Corte, and S. Rascuna, "Enhanced Non-Uniformity Modeling of 4H-SiC Schottky Diode Characteristics Over Wide High Temperature and Forward Bias Ranges," *IEEE J. Electron Devices Soc.*, vol. 8, no. September, pp. 1339–1344, 2020.
- [5] F. Roccaforte, F. La Via, V. Raineri, R. Pierobon, E. Zanoni, „Richardson’s constant in inhomogeneous silicon carbide Schottky contacts”, *Journal of Applied Physics*, vol. 93, pp. 9137-9144 2003.
- [6] J.H. Werner, H.H. Güttler, „Barrier inhomogeneities at Schottky contacts”, *Journal of Applied Physics*, vol. 69, pp. 1522-1533, 1991.
- [7] Y. P. Song, R. L. Van Meirhaeghe, W. H. Laflère, and F. Cardon, "On the difference in apparent barrier height as obtained from capacitance-voltage and current-voltage-temperature measurements on Al/p-InP Schottky barriers," *Solid State Electron.*, vol. 29, no. 6, pp. 633–638, 1986.
- [8] W. Jin et al., "Barrier height inhomogeneity in mixed-dimensional graphene/Single CdSe nanobelt schottky junctions," *IEEE Electron Device Lett.*, vol. 40, no. 1, pp. 119–122, 2019.
- [9] K. Zeghdar, L. Dehimi, F. Pezzimenti, M. L. Megherbi, and F. G. Della Corte, "Analysis of the Electrical Characteristics of Mo/4H-SiC Schottky Barrier Diodes for Temperature-Sensing Applications," *J. Electron. Mater.*, vol. 49, no. 2, pp. 1322–1329, 2020.
- [10] Ohdomari and K. N. Tu, "Parallel silicide contacts," *J. Appl. Phys.*, vol. 51, no. 7, pp. 3735–3739, 1980.
- [11] R.T. Tung, „Recent advances in Schottky barrier concepts”, *Materials Science and Engineering: R: Reports*, vol. 35, no. 1–3, pp. 1-138, 2001.
- [12] G. Brezeanu, G. Pristavu, F. Draghici, M. Badila, and R. Pascu, "Characterization technique for inhomogeneous 4H-SiC Schottky contacts: A practical model for high temperature behavior," *J. Appl. Phys.*, vol. 122, no. 8, p. 084501, 2017.
- [13] P. M. Gammon et al., „Modelling the inhomogeneous SiC Schottky interface”, *J. Appl. Phys.* 114, pp. 223704, 2013.
- [14] R. F. Schmitsdorf, T. U. Kampen, and W. Monch, "Explanation of the linear correlation between barrier heights and ideality factors of real metal-semiconductor contacts by laterally nonuniform Schottky barriers," *J. Vac. Sci. Technol. B Microelectron. Nanom. Struct.*, vol. 15, no. 4, p. 1221, 1997.
- [15] Nocedal, Jorge, and Stephen Wright. Numerical optimization. *Springer Science & Business Media*, 2006.
- [16] Cauchy, Augustin. "Méthode générale pour la résolution des systemes d’équations simultanées." *Comp. Rend. Sci. Paris* 25.1847 (1847): 536-538.
- [17] Broyden, Charles George. "The convergence of a class of double-rank minimization algorithms 1. general considerations." *IMA Journal of Applied Mathematics* 6.1 (1970): 76-90.
- [18] Sinha, Ankur, Pekka Malo, and Kalyanmoy Deb. "A review on bilevel optimization: From classical to evolutionary approaches and applications." *IEEE Transactions on Evolutionary Computation* 22.2 (2017): 276-295.
- [19] Gould, Stephen, Richard Hartley, and Dylan Campbell. "Deep declarative networks: A new hope." *arXiv preprint arXiv:1909.04866* (2019).
- [20] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." *Nature* 323.6088 (1986): 533-536.
- [21] Manzyuk, Oleksandr, et al. "Perturbation confusion in forward automatic differentiation of higher-order functions." *Journal of Functional Programming* 29 (2019).

- [22] Elliott, Conal. "The simple essence of automatic differentiation." ICFP (2018): 1-29.
- [23] Elsmann, Martin, et al. "Combinatory adjoints and differentiation." MSFP (2022).
- [24] Krawiec, Faustyna, et al. "Provably correct, asymptotically efficient, higher-order reverse-mode automatic differentiation." POPL (2022): 1-30.
- [25] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." arXiv preprint arXiv:1912.01703 (2019).
- [26] Frostig, Roy, Matthew James Johnson, and Chris Leary. "Compiling machine learning programs via high-level tracing." Systems for Machine Learning 4.9 (2018).
- [27] Innes, Michael. "Don't unroll adjoint: Differentiating ssa-form programs." arXiv preprint arXiv:1810.07951 (2018).
- [28] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.
- [29] Spivak, Michael. Calculus on manifolds. CRC press, 1965.
- [30] Melis, Gábor, Chris Dyer, and Phil Blunsom. "On the state of the art of evaluation in neural language models." arXiv preprint arXiv:1707.05589 (2017).
- [31] Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of machine learning research 13.2 (2012).
- [32] Bengio, Yoshua. "Gradient-based optimization of hyperparameters." Neural computation 12.8 (2000): 1889-1900.
- [33] Lorraine, Jonathan, Paul Vicol, and David Duvenaud. "Optimizing millions of hyperparameters by implicit differentiation." AISTATS. 2020.
- [34] Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "Deep equilibrium models." NeurIPS. 2019.
- [35] He, Kaiming, et al. "Deep residual learning for image recognition." CVPR. 2016.
- [36] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." ICML. 2015.
- [37] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450 (2016).
- [38] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
- [39] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In International Conference on Machine Learning (ICML), 2009.
- [40] Fukushima, Kunihiro, and Sei Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." Competition and cooperation in neural nets. Springer, Berlin, Heidelberg, 1982. 267-285.
- [41] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
- [42] Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018).
- [43] Dabre, Raj and Atsushi Fujita. Recurrent stacking of layers for compact neural machine translation models. In AAAI, 2019.
- [44] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In ICLR, 2019.
- [45] Wang, Tongzhou, et al. "Dataset distillation." arXiv preprint arXiv:1811.10959 (2018).
- [46] Domke, Justin. "Generic methods for optimization-based modeling." Artificial Intelligence and Statistics. PMLR, 2012.
- [47] Mairal, Julien, Francis Bach, and Jean Ponce. "Task-driven dictionary learning." IEEE TPAMI 34.4 (2011): 791-804.
- [48] Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." ICML. 2017.
- [49] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." ICML. 2017.
- [50] Rajeswaran, Aravind, et al. "Meta-learning with implicit gradients." NeurIPS (2019).
- [51] Kolter, Zico et al. "Deep Implicit Layers - Neural ODEs, Deep Equilibrium Models, and Beyond."

NeurIPS, 2020. URL: <http://implicit-layers-tutorial.org/> (accessed on 2022-11-24).
[52] Blondel, Mathieu, et al. "Efficient and modular implicit differentiation." arXiv preprint arXiv:2105.15183 (2021).

Director Proiect,
S.L. Dr. Ing. Gheorghe PRISTAVU

