



UNIVERSITATEA NAȚIONALĂ  
DE ȘTIINȚĂ ȘI TEHNOLOGIE  
POLITEHNICA BUCUREȘTI



Școala Doctorală de Electronică, Telecomunicații  
și Tehnologia Informației

Decizie nr. 127 din 09-11-2023

# REZUMAT TEZĂ DE DOCTORAT

**Ing. Mihai-Corneliu CRISTESCU**

---

ÎMBUNĂTĂȚIRI ALE METRICILOR DE PERFORMANȚĂ ÎN  
VERIFICAREA FUNCȚIONALĂ A CIRCUITELOR UTILIZÂND  
MACHINE LEARNING

PERFORMANCE METRIC IMPROVEMENTS IN CIRCUIT  
FUNCTIONAL VERIFICATION USING MACHINE LEARNING

---

## COMISIA DE DOCTORAT

<b>Prof. Dr. Ing. Gheorghe BREZEANU</b> Universitatea Națională de Știință și Tehnologie Politehnica din București	Președinte
<b>Prof. Dr. Ing. Gheorghe M. ȘTEFAN</b> Universitatea Națională de Știință și Tehnologie Politehnica din București	Conducător de doctorat
<b>Prof. Dr. Ing. Oana AMĂRICĂI- BONCALO</b> Universitatea Politehnica din Timișoara	Referent
<b>Prof. Dr. Ing. Petre OGRUȚAN</b> Universitatea Transilvania din Brașov	Referent
<b>Conf. Dr. Ing. Radu HOBINCU</b> Universitatea Națională de Știință și Tehnologie Politehnica din București	Referent

**BUCUREȘTI 2024**

---



# Cuprins

Cuprins.....	iii
1 Introducere.....	1
1.1 Descrierea problemei: Eforturi ridicate în verificarea pre-siliciu a circuitelor integrate.....	1
1.2 Motivarea alegerii temei.....	1
1.3 Scopul tezei de doctorat .....	2
1.4 Conținutul tezei de doctorat .....	2
2 Analiza literaturii.....	3
2.1 Comparații între strategiile de verificare inteligentă.....	3
3 Fundamente teoretice.....	5
3.1 Învățarea supravegheată pentru <i>metodologia de reducere a redundanței stimulilor</i> (SL4SRRM).....	5
3.1.1 Faza de antrenare în cadrul metodologiei SL4SRRM .....	5
3.1.2 Faza de inferență în cadrul metodologiei SL4SRRM.....	6
4 Instrument bazat pe SL4SRRM pentru <i>generarea de stimuli influențată de acoperire</i> (CASTiG) .....	7
4.1 Modelarea blocului de verificat.....	7
4.2 Tehnologii de bază .....	7
4.3 Punctele forte ale instrumentului software.....	8
4.4 Arhitectura rețelei neuronale artificiale.....	8
5 Aplicații și rezultate experimentale .....	9
5.1 Modelul de analiză comparativă .....	9
5.2 Coverpoint liniar cu distribuție uniformă în $N$ intervale egale.....	10
5.2.1 Modelul de coverpoint .....	10
5.2.2 Rezultatele procesului de învățare automată.....	10
5.2.3 Rezultatele analizei comparative .....	11
5.3 Coverpoint cu distribuție neliniară de tipul „putere a lui doi” .....	14
5.3.1 Modelul de coverpoint .....	14
5.3.2 Rezultatele procesului de învățare automată.....	15
5.3.3 Rezultatele analizei comparative .....	16
6 Concluzii.....	19
6.1 Rezultate obținute.....	19
6.2 Contribuții originale .....	19
6.3 Lista lucrărilor originale.....	21

6.4	Perspectivă de dezvoltare ulterioară.....	22
6.4.1	Integrarea nucleului de învățare automată al instrumentului CASTiG în arhitectura ECTB.....	22
6.4.2	Alte direcții de cercetare .....	23
	Bibliografie .....	24

# Capitolul 1

## Introducere

### 1.1 Descrierea problemei: Eforturi ridicate în verificarea pre-siliciu a circuitelor integrate

În dezvoltarea *circuitelor integrate* (CI), procesul de *verificare pre-siliciu bazat pe simulare* este popular datorită flexibilității și caracteristicilor sale care ajută inginerii să atingă obiectivele de acoperire a scenariilor. Dorința este de a atinge toate aceste obiective cu un cost minim de timp, dar această metodă de verificare, în multe circumstanțe, se dovedește a fi solicitantă și anostă. Acest lucru este cauzat în mare parte de scenariile redundante generate de mecanismele de aleatorizare ale simulatoarelor funcționale de astăzi.

### 1.2 Motivarea alegerii temei

Provocările majore de verificare sunt cauzate de proceduri iterative sau manuale care necesită contribuția inginerilor de verificare. Folosind cele mai populare strategii și metodologii de verificare de astăzi, s-au observat astfel de blocaje:

1. *Problema 1: Extracțiile semantice manuale din documentul de specificație sunt predispuse la inserări involuntare de erori.*
2. *Problema 2: Acoperirea celor câteva procente rămase în metrica de acoperire funcțională poate necesita costuri de timp și resurse inaccesibile.*
3. *Problema 3: Depanarea manuală a simulării pentru identificarea cauzei principale a defecțiunii poate necesita costuri inacceptabile de timp și resurse.*
4. *Problema 4: Încheierea procesului de dovedire a unei funcționalități complexe în verificarea formală poate necesita costuri de timp nepermise.*

Deoarece tehnicile de *data mining* (DM) permit optimizări valoroase ale proceselor [Don22], principalul meu interes de cercetare este valorificarea sinergiilor ascunse între verificarea funcțională și *inteligenta artificială* (IA). Aceste *puncte de sinergie* sunt cunoscute sub numele de metode de *verificare inteligentă* (VI).

## 1.3 Scopul tezei de doctorat

Obiectivul acestei teze de doctorat este de a cerceta și identifica soluții potențiale pentru a atenua efectele problemelor prezentate în subcapitolul 1.2. După ce am cercetat diferite rapoarte de specialitate privind provocările de fabricație a CI [LZC+14], investigația mea a indicat că cel mai costisitor blocaj din industria de dezvoltare a CI este cel evidențiat de *Problema 2*. Practic, costul de timp ar trebui redus folosind o strategie *de ultimă generație* pentru optimizarea proceselor. Prin urmare, soluțiile propuse în această teză de cercetare constau într-o *metodologie* și un *instrument* care filtrează cu succes pachetele redundante de stimuli.

## 1.4 Conținutul tezei de doctorat

Capitolul 2 prezintă o analiză a literaturii de specialitate, cu un accent deosebit pe *sinergiile de ultimă generație* și pe *abordările inteligente* pentru reducerea atât a alocării resurselor, cât și a costurilor de timp în verificarea funcțională a CI. În mod specific, sunt analizate motoarele asistate de IA care optimizează procesele cheie. În ultimele paragrafe, punctele forte și punctele slabe corespunzătoare sunt comparate pentru a identifica cea mai promițătoare strategie VI.

Capitolul 3 prezintă fundamentele teoretice ale tehnicilor de *machine learning* (ML) identificate în capitolul 2. Se prezintă principiul optimizării inteligente pentru reducerea  *timpului de acoperire completă a scenariilor prin reducerea redundanței stimulilor* (SRR). De asemenea, introduc principiile noii *metodologii de învățare automată supravegheată pentru reducerea redundanței stimulilor (SL4SRRM)* pe care am proiectat-o și propus-o ca obiectiv principal de implementare în această cercetare.

Capitolul 4 introduce detaliile de implementare a unui instrument software original numit *generarea stimulilor influențată de acoperirea scenariilor (CASTiG)* care utilizează noua metodologie **SL4SRRM** descrisă în capitolul 3. Primul subcapitol indică *blocul de verificat* (DUT) folosit în acest proiect de cercetare. Cerințele funcționale cheie ale DUT sunt selectate pentru planificarea procesului de verificare utilizând noul instrument. Următoarele subcapitole prezintă tehnologiile alese, alte abordări de punere în practică, precum și punctele forte ale instrumentului asistat de IA. Capitolul se încheie prezentând structura *rețelei neuronale artificiale* (ANN) care constituie *nucleul de învățare automată* propus bazat pe metodologia **SL4SRRM**.

Capitolul 5 listează rezultatele experimentale obținute în urma implementării unor studii de caz pe noua platformă **CASTiG**. Diferite modele de *coverpoint* cu distribuții neliniare sunt analizate pentru a măsura *economiile de timp* pentru închiderea verificării funcționale. Instrumentul este *analizat prin compararea* rezultatelor de performanță cu standardul actual, *verificarea ghidată de acoperirea scenariilor* (CDV).

În cele din urmă, dar important, capitolul 6 listează concluziile și evidențiază contribuțiile originale de cercetare prezentate în această teză. Mai mult, perspectivele viitoare și direcțiile de cercetare ulterioare sunt subliniate în ultimele rânduri.

# Capitolul 2

## Analiza literaturii

### 2.1 Comparații între strategiile de verificare inteligentă

În urma unei analize a literaturii de specialitate privind patru strategii VI diferite [Cri21], am enumerat principalele avantaje, notate *Av.*, și dezavantaje, notate *Dez.*, în tabelul 2.1. Comparația se face pe baza a patru metrici de performanță: *efortul de implementare*, *viteza de procesare*, *acuratețea* și *constrângerile de utilizare*.

Privind *efortul de implementare*, strategia *scenariilor create cu feedback de la modelul de acoperire* are cele mai multe puncte forte. Tehnicile DM nu necesită *cunoștințe specifice*, iar *programarea logică inductivă* (ILP) generează date în formate lizibile. Totuși, acestea din urmă necesită o cantitate mare de *cunoștințe specifice* (CS), iar acest punct slab afectează și strategia *analizei inteligente a cauzelor*. Din pricina aparatului matematic complex, cu greu se poate automatiza folosind a patra strategie. Similar, pentru prima strategie, este nevoie de un efort mare din pricina adnotărilor.

*Viteza de procesare* a *algoritmilor genetici* (GA) este mare întrucât se paralelizează procesele, în timp ce *mașinile cu support vectorial* (SVM) au viteze mult mai mici. *Metodele formale inteligente* reduc timpul de execuție, dar acest avantaj este anulat atunci când se caută valori pentru *hiperparametrii*. A treia strategie VI reduce timpul de depanare, dar avantajul se pierde atunci când se efectuează pași de ordonare a datelor. Strategia de *extragere inteligentă a cerințelor* automatizează implementarea constrângerilor, dar generarea specificației intermediare anulează acest câștig.

Pentru *acuratețea modelului*, GA sunt preferați, deoarece au putut atinge soluții globale. În schimb, *rețelele Bayesiene* (BN) au acurateți scăzute, în timp ce DM oferă rezultate inexacte atunci când se prelucrează exemple de date noi.

Prima strategie VI îmbunătățește acuratețea, reduce erorile, dar aceste lucruri sunt contrabalansate de timpul petrecut cu etichetarea manuală. Mai mult, strategia *analizei inteligente a cauzelor* poate beneficia de modele SVM precise numai dacă este posibilă ordonarea *commit-urilor* din *sistemele de versionare* a fișierelor.

Metrica privind *constrângerile de utilizare* indică strategiile scalabile. Pentru cea de-a doua strategie VI, metodele ILP se dovedesc a fi o soluție de uz general, deoarece mapează dependențe complexe. Mai mult, soluțiile SVM funcționează bine

**Tabelul 2.1** Compararea diferitelor strategii VI

Indicador de performanță		Efort de implem.	Viteza de procesare	Acuratețea procesului	Constrângeri de utilizare
Strategie VI					
<b>Extragere inteligentă a cerințelor</b> (capitolul 2.2)	<i>Av.</i>	-	Generarea automată a constrângerilor	Extracție semantică automată	Evitare erori umane neintenționate
	<i>Dez.</i>	Etichetarea manuală a propozițiilor	Generarea specificațiilor intermediare	Etichetare folosind CS	Standard de documentare IEEE
<b>Creare scenarii cu feedback de la modelul de acoperire</b> (capitolul 2.3)	<i>Av.</i>	DM: CS reduce ILP: Lizibil pentru om	GA: Paralelism	GA: Optim global	SVM: Dependențe de antrenare scăzute  ILP: Mapări complexe
	<i>Dez.</i>	ILP: multe CS	SVM: Reduceri reduce de timp	BN: Asignări inițiale DM: Exemple noi	GA: coverpoint-uri complexe  Nefolosită pt. acoperirea funcțională
<b>Analiză inteligentă a cauzelor</b> (capitolul 2.4)	<i>Av.</i>	-	Spațiu redus pt. analizat bug-uri	SVM	-
	<i>Dez.</i>	Multe CS	Ordonarea datelor Regresii de teste (analiza rutelor)	Clasarea commit-urilor	Nepotrivit pt. proiecte mici (commit-uri reduce)  Commit-uri de referință
<b>Metode formale inteligente</b> (capitolul 2.5)	<i>Av.</i>	Automatizare consistentă	Comutare algoritmi	N/A	-
	<i>Dez.</i>	Matematică complexă	Atribuirii hiperparametrii	N/A	Nepotrivit pt. proiecte mari

atunci când se inferează modele de mapare noi. Din păcate, atunci când se implementează GA pe sarcini complexe, cercetările au indicat performanță limitată.

Strategia de *analiză inteligentă a cauzelor* are aplicabilitate mică, deoarece *sistemul de versionare* trebuie populat cu sute de *commit-uri*. În plus, *metodele formale* sunt inutilizabile în cadrul proiectelor mari cu funcționalități complexe.

În ceea ce privește *extragerea inteligentă a cerințelor*, avantajul este că zecile de tipuri de erori umane neintenționate pot fi minimizezate sau evitate total, dar acest lucru este diminuat, deoarece un număr mic de organizații își scriu documentele de specificații conform *standardului IEEE 830-1998* [Iee98].



# Capitolul 3

## Fundamente teoretice

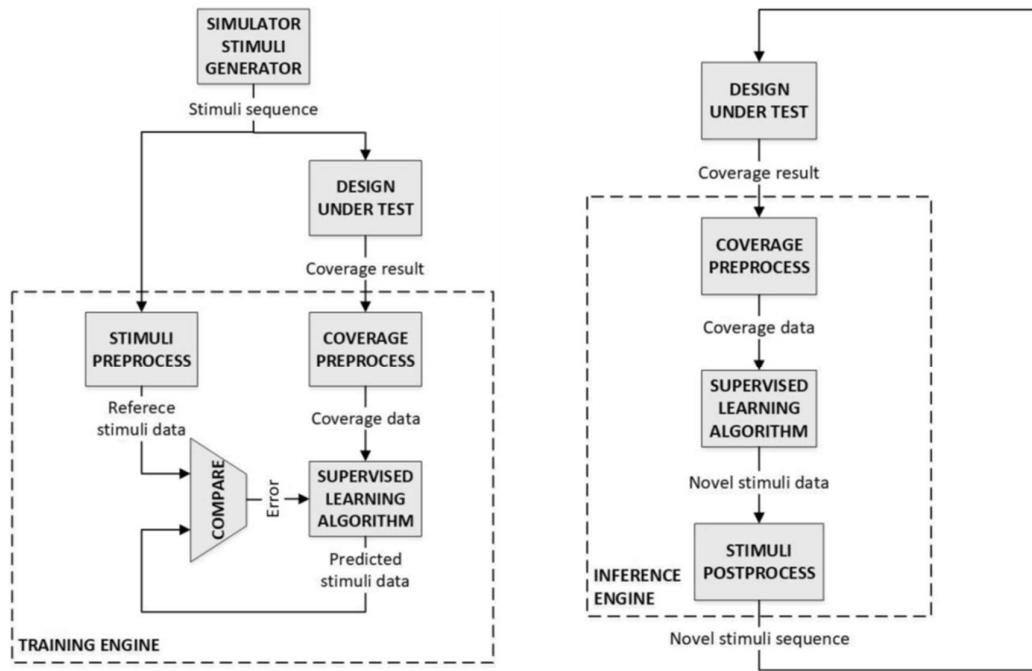
### 3.1 Învățarea supravegheată pentru *metodologia de reducere a redundanței stimulilor (SL4SRRM)*

În capitolul anterior, strategia VI care a indicat cea mai interesantă oportunitate pentru reducerea costurilor verificării CI este generarea *automată de teste noi folosind feedback-ul de acoperire a scenariilor*. Dorința este de a minimiza timpul de simulare prin generarea unor scenarii noi și eliminarea *secvențelor de testare redundante* care au fost utilizate anterior în timpul aceleiași regresii a teste. Acest lucru poate fi realizat prin reducerea *ratei de redundanță a stimulilor*. Din ceea ce cunoaștem, literatura de specialitate indică două sau trei metode [XZB+16] pentru reducerea numărului de *secvențe de testare*, dar nu există terminologie sau cercetări asupra reducerii directe a *ratei de redundanță a stimulilor*. Prin urmare, am introdus terminologia SRR în [CB21].

După ce am analizat stadiul actual al literaturii și am listat observațiile făcute în paragrafele anterioare, am continuat cu proiectarea unei metodologii care poate implementa eficient principiul SRR folosind diverse modele *ML supervizate*. Introduc *modelul de lucru și recomandările de aplicare* pentru noua metodologie **SL4SRRM** în [CB21]. Din punctul meu de vedere, scopul este de a utiliza un algoritm inteligent, preferabil o ANN care poate modela *funcțiile de transfer* ale DUT între *pachetele de stimuli* de intrare și *coverpoint-uri*. În acest fel, unele software alimentată de **SL4SRRM** modelează și utilizează *inversa funcției de transfer* a DUT pe *bucla de feedback* a *modelului de acoperire a scenariilor*.

#### 3.1.1 Faza de antrenare în cadrul metodologiei SL4SRRM

Pentru metodologia **SL4SRRM** descrisă în articolul [CB21], succesiunea pașilor pentru antrenarea modelului este prezentată în figura 3.6 (a). În această fază, DUT este rulat în diferite scenarii generate de un *simulator digital* obișnuit, ca într-o configurație tipică de regresie [Piz04]. Un exemplu de antrenare este creat prin asocierea unei *secvențe de stimuli* de intrare cu *rezultatul de acoperire a scenariilor*. În acest fel, algoritmul ML primește exemplele de antrenare după ce trec prin etape de preprocesare de bază înainte de a fi introduse în nucleul ML [KKP06].



(a) The learning phase within SL4SRRM

(b) The inference phase within SL4SRRM

**Figura 3.6** Fazele de învățare automată pentru metodologia **SL4SRRM** [CB21]

În funcție de implementare, algoritmul poate să învețe *funcția de eșantionare inversă* a *modelului de acoperire* al DUT. Acest lucru se face prin selectarea *datelor de acoperire* ca și *attribute de intrare*, în timp ce *datele stimulilor* sunt utilizate ca etichete de ieșire. Algoritmul prezice date noi despre stimuli care sunt comparate *stimulii de referință*. Pentru fiecare *lot de antrenare* [Hea15], *nucleul neural* calculează eroarea de predicție care poate ajusta eficient *ponderile sinaptice* ale modelului în următorul *lot de antrenament* [KKP06]. Mai mult, setul de antrenare poate fi utilizat de-a lungul a sute de *epoci* și, cu fiecare nouă *epocă* de antrenare, instrumentul bazat pe **SL4SRRM** îmbogățește setul de antrenament cu cazuri nemaivăzute anterior. Prin urmare, în această fază, algoritmul ML acționează ca un *nucleu de antrenare*.

### 3.1.2 Faza de inferență în cadrul metodologiei SL4SRRM

Pașii pe care îi propun pentru metodologia **SL4SRRM** în timpul fazei de inferență sunt prezentați în figura 3.6 (b). De-a lungul acestei faze, același DUT nu mai este expus scenariilor generate de *simulatorul digital*. În schimb, modelul ML antrenat trimite DUT-ului *secvența de stimuli* noi generată în timpul iterației anterioare de inferență. Practic, algoritmul primește *datele curente de acoperire* și pe baza experienței sale de antrenare, nucleul neural poate prezice *date de stimuli* nevăzute anterior prin modelarea funcției de eșantionare inversă a DUT. Secvențele de stimuli noi au o probabilitate mare de a îmbunătăți rezultatele *acoperirii de scenarii* cu fiecare nouă iterație de inferență.

Înainte ca DUT să primească *pachetul de stimuli*, un bloc dedicat post-procesează datele și assemblează o nouă *secvență de stimuli* [BF00]. În consecință, în această fază, algoritmul ML acționează ca un *nucleu de inferare*.

# Capitolul 4

## Instrument bazat pe SL4SRRM pentru generarea de stimuli influențată de acoperire (CAStiG)

Pentru explorarea posibilităților de *reducere a redundanței stimulilor*, am implementat unui instrument *flexibil* asistat de ML care poate modela cazuri de utilizare a verificării funcționale. În capitolul 4, descriu tehnicile și configurația utilizate pentru rularea ineditului instrument **CAStiG** bazat pe metodologia **SL4SRRM**.

### 4.1 Modelarea blocului de verificat

Pentru evaluarea *conceptului de validare* propus pentru SRR, se verifică mecanismul de selecție al unui *demultiplexor*. Utilizatorul rulează scenarii de testare care generează  $S = 1000$  pachete de stimuli care ținesc diverse valori ale adresei pe 32 de biți.

După rularea acestui subset de regresie, *baza de date de acoperire* și *baza de date a stimulilor* sunt extrase din datele de ieșire ale simulării. Se obține setul de antrenament, iar rezultatele inițiale indică o rată de *acoperire funcțională* de 85,8% pentru modelul de *coverpoint* care distribuie uniform valorile adresei în  $N$  intervale.

Scopul *nuclului de învățare automată* este de a obține o *rată a acoperirii funcționale* de 100% pe valorile adreselor trimise pe intrarea DUT-ului.

### 4.2 Tehnologiile de bază

Folosind tehnologii de programare *orientate pe obiecte*, *elementele de acoperire* și *secvențele de stimuli* de intrare sunt modelate eficient folosind principiile *moștenire* și *polimorfism*.

Când inginerul de verificare pornește o regresie de testare, *simulatoarele digitale* pot stoca *rezultatele de acoperire* într-o bază de date care respectă formatul UCIS [UCI12]. Acest standard permite utilizatorilor să îmbine diferite *date de acoperire* într-o bază de date unificată. Prin urmare, setul de antrenare ML este extras din baza de date cu ușurință folosind utilitare dedicate, cum ar fi *CoverageLens* [Sta17].

## 4.3 Punctele forte ale instrumentului software

Pentru dezvoltarea nucleelor neuronale complexe, comunitatea de cercetare a creat interfețe dedicate care sunt *ușor de utilizat*, oferă modele pre-antrenate și permit implementarea rapidă a modelelor. O astfel de bibliotecă puternică, preferată pentru modelarea *rețelelor neuronale adânci*, este interfața *Keras* a bibliotecii *open-source TensorFlow* [Goo15]. Prin utilizarea acestei platforme de dezvoltare, instrumentul **CASTiG** [CB21] evită implementarea punctelor slabe asociate strategiei de *învățare „in direct”* și valorifică o soluție inteligentă folosind un mediu de programare pur *Python*.

Pentru o integrare perfectă în *nucleul de învățare automată*, *modelul de acoperire* al DUT este proiectat folosind un modul dedicat *Python*, iar un alt avantaj este că modulul *generator aleatoriu Numpy* este utilizat pentru crearea pachetelor inițiale de stimuli. Prin urmare, alte module *Keras* pot procesa cu ușurință setul de date privind stimulii, deoarece este formatat conform standardului *Numpy binary format*.

## 4.4 Arhitectura rețelei neuronale artificiale

Conform metodologiei **SL4SRRM** descrisă în subcapitolul 3.1, ANN ar trebui să facă parte din *bucla de feedback a modelului de acoperire* și să învețe *inversa funcției de eșantionare a modelului de acoperire*.

După cum s-a subliniat în subcapitolul 3.1, în timpul ambelor faze de învățare automată, datele de acoperire sunt utilizate ca intrare pentru ANN, în timp ce datele stimulilor sunt colectate de la *neuronul stratului de ieșire*. În ceea ce privește modelarea scenariilor, *secvențele de stimuli* sunt proiectate să conțină un singur câmp care reprezintă o valoare de adresă pe 32 de biți folosite de DUT-ului.

*Perceptronii multistrat* sunt utilizați pentru evaluarea instrumentului **CASTiG**. Deoarece structura ANN depinde de dimensiunea *coverpoint-ului*, structura ANN este ușor diferită de la un caz la altul. Mai exact, stratul de intrare a ANN are un număr de neuroni egal cu numărul de *bin-uri* al *coverpoint-ului*. Astfel, pentru un *coverpoint* cu o dimensiune  $C = 8$  *bin-uri*, structura *perceptronului multistrat* are  $C = 8$  neuroni pe stratul de intrare.

Cea mai stabilă *funcție de activare* s-a dovedit a fi *Rectified Linear Unit (ReLU)* în comparație cu alte opțiuni, cum ar fi funcțiile *SoftMax* sau *Sigmoid*. Mai mult, analiza a arătat că cel mai eficient optimizator de învățare este *estimarea momentului adaptiv (Adam)*, deoarece oferă cea mai bună precizie și cel mai rapid timp de convergență.

Indiferent de *coverpoint-ul țintă*, instrumentul **CASTiG** este configurat pentru a implementa pașii de învățare cu valori fixe ale *hiperparametrilor*. Prin urmare, în timpul fazei de antrenare, instrumentul **CASTiG** utilizează  $E = 400$  *epoci de învățare* care oferă suficiente iterații pentru a ajusta optim *ponderile sinaptice*. Mai mult, fiecare lot de *învățare* este dimensionat la  $B = 10$  exemple de antrenament / lot, ceea ce oferă cel mai bun compromis între timpul de execuție și acuratețea finală a învățării.

# Capitolul 5

## Aplicații și rezultate experimentale

Pentru *validarea conceptului* de bază al metodologiei **SL4SRRM** descrise în subcapitolul 3.1, instrumentul **CAStiG** introdus în capitolul 44 este analizat prin soluționarea celor mai comune *coverpoint-uri* în procesele tipice de verificare ASIC.

Următoarele subcapitole subliniază modul în care două aplicații de *coverpoint* sunt implementate utilizând instrumentul **CAStiG**. De asemenea, aceste aplicații sunt analizate astfel încât evaluarea comparativă a instrumentului propus indică o reducere semnificativă a numărului total de *pachete de stimuli* aplicate intrărilor DUT-ului.

### 5.1 Modelul de analiză comparativă

Pentru a indica mai bine efectul SRR în cadrul metodologiei **SL4SRRM** și modul în care timpul de completare a acoperirii este redus semnificativ atunci când se folosește instrumentul **CAStiG**, introduc un *model de analiză comparativă* [Cri23] care monitorizează numărul de *pachete de stimuli* care sunt primite către DUT.

*Modelul de analiză comparativă* nu utilizează vreo componentă *terță* și este proiectat ca o caracteristică a instrumentului **CAStiG** introdus în capitolul 4. Pentru abordarea tipică CDV, *modelul de analiză comparativă* utilizează un *generator de numere aleatorii pure* cu o *distribuție uniformă a probabilităților*. De asemenea, procesul de *analiză comparativă* efectuează o explorare a designului prin implementarea și colectarea datelor pentru câteva tipuri și dimensiuni de *coverpoint-uri*. În plus, analiza este extinsă prin compararea cazurilor cu *rate inițiale de acoperire* de 33%, 50% sau 75%.

O metrică importantă utilizată în procesul de *analiză comparativă* este *numărul de pachete de stimuli care trec prin DUT (NSTD)*. În plus, deoarece instrumentul **CAStiG** adaugă un cost în urma procesării pentru generarea pachetelor de stimuli, este interesant să se măsoare și *numărul de pachete de stimuli generate de ANN (NSGA)*. Astfel, obiectivul principal al procesului de *analiză comparativă* este identificarea configurațiilor de *coverpoint* care minimizează valorile pentru metrica **NSTD**. Deoarece **NSGA** indică volumul de lucru efectuat de către ANN, este important să se reducă și aceste numere, dar cu prioritate mai mică în comparație cu valorile **NSTD**.

## 5.2 Coverpoint linear cu distribuție uniformă în $N$ intervale egale

### 5.2.1 Modelul de coverpoint

Unul dintre cele mai comune *coverpoint-uri* este cel care împarte întreaga gamă de valori ale adreselor distribuite în mod egal în  $N$  intervale. Considerând că magistrala de adresare are o lățime de  $W = 32$  biți, în loc să acopere individual toate valorile adreselor în număr de  $M = 2^W = 2^{32}$ , din punct de vedere funcțional, este suficient să se ia în considerare un *coverpoint* cu  $N$  intervale distribuite pe gama valori întregi  $[0: M-1]$ . Limitele celor  $C = N = 1000$  intervale sunt prezentate în expresia (5.3). Analizând (5.3) și având în vedere că ANN este concepută pentru a reproduce inversul *comporta-*

$$bin_i \leftrightarrow \left[ i \frac{2^{32}}{1000} : (i + 1) \frac{2^{32}}{1000} - 1 \right] , i \in [0, 999] \cap \mathbb{N} \quad (5.3)$$

*mentului de eșantionare* a DUT-ului, funcția modelată de algoritmul de *învățare automată supravegheată* este menționată în ecuația (5.4).

$$predicted_{address} = target_{bin_{index}} \cdot \frac{2^{32}}{1000} \quad (5.4)$$

În cazul în care valorile magistralei sunt generate folosind o distribuție uniformă, o funcție care indică probabilitățile de acoperire a fiecărui interval al *coverpoint-ului* este *funcția de masă de probabilitate* (PMF). Acesta indică o distribuție uniformă a probabilității în toate intervalele și este calculat în ecuația (5.5).

$$P(b) = \frac{M-2}{N \cdot M} = \frac{2^{32}-2}{1000 \cdot 2^{32}} , i \in [0: 999] \cap \mathbb{N} \quad (5.5)$$

### 5.2.2 Rezultatele procesului de învățare automată

Pentru a obține o învățare care rezolvă în mod optim această sarcină, implementez zeci de modele de ANN folosind diferite combinații de *hiperparametri* până când valorile performanței de învățare indică o soluție optimă. *Hiperparametrii* și intervalele lor de valori sunt prezentate în tabelul 5.1. După terminarea etapei de analiză a hiperparame-

**Tabelul 5.1** Valorile hiperparametrilor utilizate în timpul proceselor de explorare a învățării automate

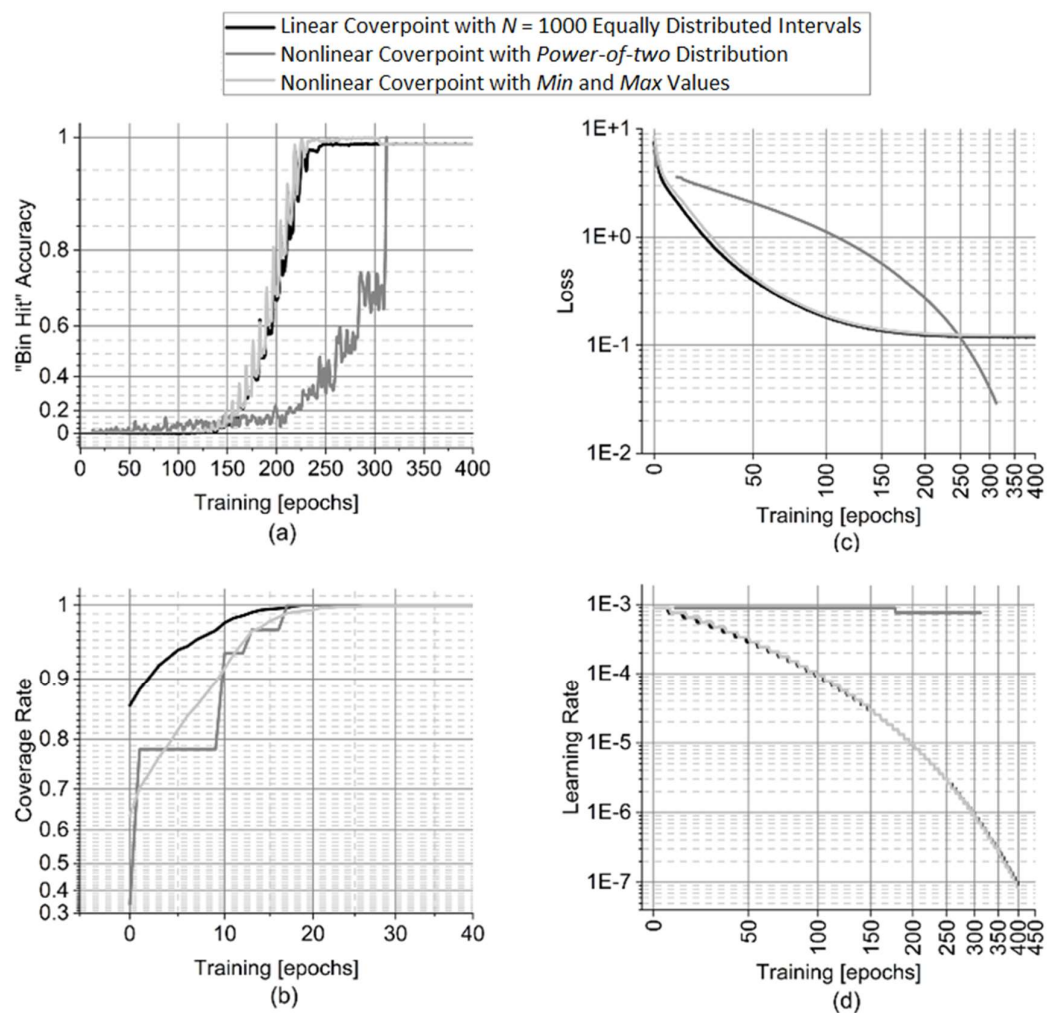
Hiperparametru	Notăție	Interval de valori	
		Min.	Max.
Rata de învățare inițială	<i>initial_lr</i>	1.00E-06	1.00E-03
Pasul de degradare	<i>decay</i>	1.00E+03	1.00E-05
Rata de degradare	<i>decay_lr</i>	0.5	0.99

trilor, rezultatele optime sunt identificate manual și prezentate în figura 5.2.

Se poate observa că *precizia modelului* de a acoperi un nou *bin* crește rapid după efectuarea a 150 de epoci de învățare. După interpretarea acestei soluții optime, observațiile sunt enumerate în tabelul 5.2. Se poate observa că sunt necesare doar 20 de epoci pentru *completarea acoperirii de scenarii*. Cu toate acestea, celelalte metrice de *învățare automată* obțin valori optime cu mult peste pragul de 20 de epoci de învățare.

### 5.2.3 Rezultatele analizei comparative

Pentru acest *coverpoint*, procesul de *analiză comparativă* al instrumentului **CASTiG** acoperă multe configurații cu valori diferite pentru  $N$ , variind de la  $C = 64$  de *bin-uri* până la  $C = 8192$  de *bin-uri*. În comparație cu abordarea clasică a CDV, obiectivul



**Figura 5.2** Configurațiile optime de învățare automată obținute pentru toate cele trei aplicații de coverpoint

(a) Acuratețea învățării pentru atingerea bin-urilor noi

(b) Rata acoperirii (obiectivul de verificare)

(c) Funcția de cost a învățării

(d) Rata de învățare

**Tabelul 5.2** Rezultatele performanței de învățare automată pentru coverpoint-ul liniar cu distribuție uniformă în  $N = 1000$  intervale egale

Metrică	Valoare	Numărul de epoci <sup>a</sup>
Acuratețe	98,51%	240
Rata acoperiri	100%	20
Funcția de cost	0.14	250
Rata de învățare	8.50E-08	400

a. Numărul de epoci de învățare după care se atinge valoarea prezentată

este să demonstrez dacă instrumentul **CASTiG** reușește să reducă numărul total de pachete de stimuli necesare pentru a ajunge la *acoperire completă a scenariilor*.

După ce toate configurațiile de coverpoint-uri sunt implementate și după ce toate valorile de performanță asociate sunt măsurate, rezultatele *analizei comparative* indică o reducere considerabilă a **NSTD** total pentru această aplicație. Cele mai relevante statistici de analiză comparativă sunt menționate în tabelul 5.3.

Pentru a indica îmbunătățirile de performanță, a zecea coloană prezintă reducerea **NSTD** utilizând instrumentul **CASTiG** în comparație cu strategia CDV. Mai exact, cel puțin de două ori și de până la 8,6 ori mai puțini **NSTD** sunt suficienți pentru atingerea aceluiași obiectiv de verificare folosind instrumentul **CASTiG** bazat pe metodologia **SL4SRRM** comparativ cu abordarea clasică CDV.

Pentru o mai bună vizualizare a rezultatelor *analizei comparative*, cele mai importante date prezentate în tabelul 5.3 sunt utilizate pentru crearea graficelor prezentate în figura 5.3. Pentru instrumentul **CASTiG**, graficul (a) indică modul în care valorile **NSTD** totale cresc împreună cu dimensiunea *coverpoint-ului*.

Deoarece configurațiile cu o *rată inițială de acoperire* de 33% generează cele mai bune rezultate **NSTD**, o comparație între CDV și **CASTiG** este prezentată în figura 5.4. Se poate observa că valorile **NSTD** obținute cu instrumentul **CASTiG** sunt semnificativ mai mici decât valorile **NSTD** corespunzătoare obținute cu abordarea clasică CDV. În plus, diferența crește împreună cu dimensiunea *coverpoint-ului*.

Graficul (c) din figura 5.3 reprezintă rezultatele din cea de a zecea coloană a tabelului 5.3 și se poate observa că situațiile cu o dimensiune mai mare a *coverpoint-ului* oferă îmbunătățiri mai accentuate ale **NSTD**. În plus, variațiile de performanță între *rate inițiale de acoperire* cresc împreună cu dimensiunea *coverpoint-ului*.

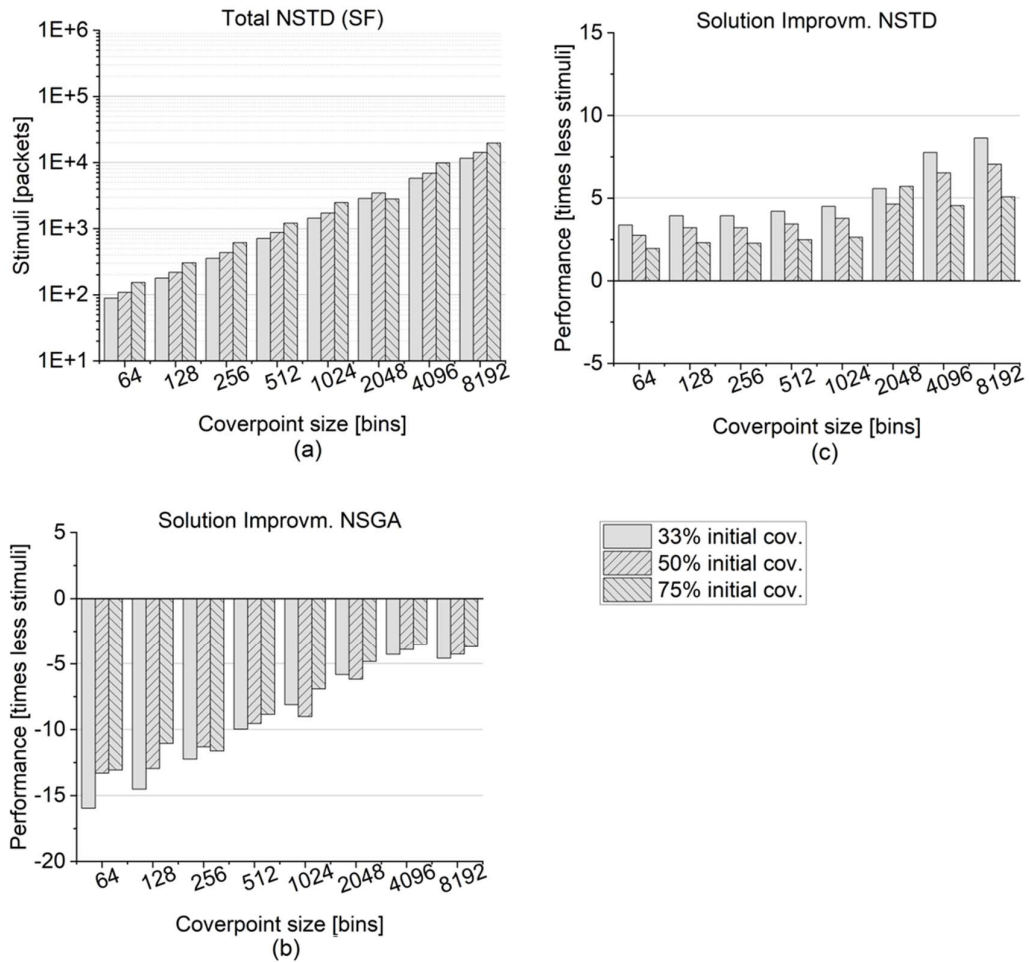
A unsprezecea coloană din tabelul 5.3 este reprezentată în graficul (b) din figura 5.3. Deoarece **NSGA** corespunzător instrumentului **CASTiG** este mai mare în comparație cu **NSGA** corespunzător CDV, graficul indică doar rezultate negative de performanță. La fel ca raportul **NSTD**, raportul **NSGA** se îmbunătățește împreună cu dimensiunea *coverpoint-ului*. Practic, costurile aferente **NSGA** scad pe măsură ce dimensiunea *coverpoint-ului* crește. Cu toate acestea, configurațiile care au *rate inițiale de acoperire* mai mari au rapoarte **NSGA** mai bune.

În consecință, cel mai bun raport **NSTD** este obținut atunci când *rata inițială de acoperire* este de 33%, deoarece instrumentul **CASTiG** este implicată mai mult în



**Tabelul 5.3** Rezultatele analizei comparative pentru coverpoint-ul liniar cu distribuție uniformă în  $N$  intervale egale

Rata inițială de acoperire [%]	Numărul de <i>bin-uri</i> din coverpoint ( $N$ )	NSTD (pentru CDV)	NSTD (Faza de antrenare pentru CAStiG)	NSTD (Ambele faze pentru CAStiG)	NSTD total (pentru CAStiG)	NSGA (Faza de antrenare pentru CAStiG)	NSGA (Faza de inferență pentru CAStiG)	NSGA total (pentru CAStiG)	Îmbun. CAStiG NSTD [ori mai puțin stimuli]	Îmbun. CAStiG NSGA [ori mai puțin stimuli]
33%	64	300	25	64	89	2099	2667	4791	<b>3.37</b>	<b>-15.97</b>
	128	700	50	128	178	4610	5504	10164	<b>3.93</b>	<b>-14.52</b>
	256	1400	100	256	356	7746	9301	17147	<b>3.93</b>	<b>-12.25</b>
	512	3000	200	512	712	12990	16718	29908	<b>4.21</b>	<b>-9.97</b>
	1024	6500	420	1024	1444	23728	28535	52683	<b>4.50</b>	<b>-8.11</b>
	2048	16000	820	2048	2868	41760	50517	93097	<b>5.58</b>	<b>-5.82</b>
	4096	45000	1700	4096	5796	87772	103219	192691	<b>7.76</b>	<b>-4.28</b>
8192	100000	3400	8192	11592	213066	241664	458130	<b>8.63</b>	<b>-4.58</b>	
50%	64	300	45	64	109	1789	2159	3993	<b>2.75</b>	<b>-13.31</b>
	128	700	90	128	218	4214	4762	9066	<b>3.21</b>	<b>-12.95</b>
	256	1400	180	256	436	7268	8380	15828	<b>3.21</b>	<b>-11.31</b>
	512	3000	360	512	872	13150	15104	28614	<b>3.44</b>	<b>-9.54</b>
	1024	6500	700	1024	1724	27241	30720	58661	<b>3.77</b>	<b>-9.02</b>
	2048	16000	1400	2048	3448	45555	51883	98838	<b>4.64</b>	<b>-6.18</b>
	4096	45000	2800	4096	6896	81080	91477	175357	<b>6.53</b>	<b>-3.90</b>
8192	100000	6000	8192	14192	200253	219546	425799	<b>7.05</b>	<b>-4.26</b>	
75%	64	300	90	64	154	1823	2005	3918	<b>1.95</b>	<b>-13.06</b>
	128	700	177	128	305	3638	3917	7732	<b>2.30</b>	<b>-11.05</b>
	256	1400	360	256	616	7680	8226	16266	<b>2.27</b>	<b>-11.62</b>
	512	3000	700	512	1212	12540	13312	26552	<b>2.48</b>	<b>-8.85</b>
	1024	6500	1450	1024	2474	20926	22528	44904	<b>2.63</b>	<b>-6.91</b>
	2048	16000	2900	2048	2800	35853	38571	77324	<b>5.71</b>	<b>-4.83</b>
	4096	45000	5800	4096	9896	73436	78507	157743	<b>4.55</b>	<b>-3.51</b>
8192	100000	11500	8192	19692	174324	183501	369325	<b>5.08</b>	<b>-3.69</b>	



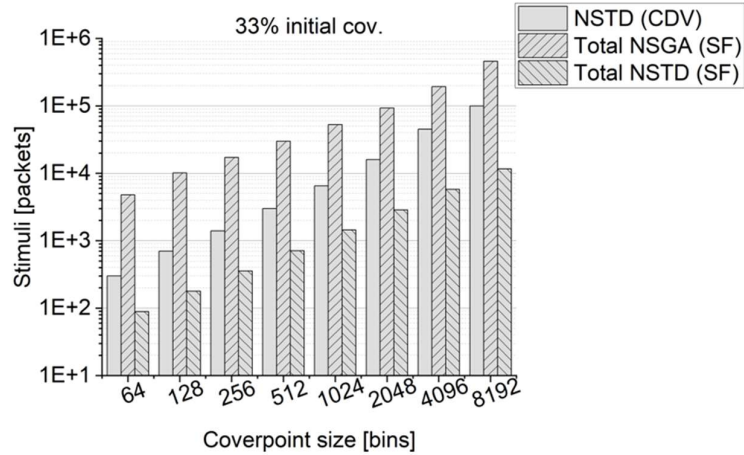
**Figura 5.3** Rezultatele analizei comparative obținute pentru mai multe configurații ale coverpoint-ului liniar cu distribuție uniformă în  $N$  intervale egale  
 (a) Valorile NSTD totale obținute utilizând instrumentul CASTiG  
 (b) Performanța NSGA a instrumentului CASTiG  
 (c) Performanța NSTD a instrumentului CASTiG în comparație cu abordarea CDV

generarea de noi scenarii care *reduc redundanța stimulilor*. Analiza scoate la lumină un compromis între cele două metrici de performanță și se poate observa că atunci când raportul NSTD este cel mai bun, raportul NSGA este cel mai puțin bun și *vice versa*.

## 5.3 Coverpoint cu distribuție neliniară de tipul „putere a lui doi”

### 5.3.1 Modelul de coverpoint

Un *coverpoint* care utilizează o grupare neliniară a valorilor adresei este cel având  $C$  intervale cu distribuție de tipul „putere a lui doi” [Bir15]. Mapările dintre fiecare *bin* și intervalul de adrese asociate sunt sugerate în expresia (5.6). Se poate observa că, cu cât



**Figura 5.4** O comparație între CDV și instrumentul *CASiG* când rata inițială de acoperire este de 33% iar configurațiile de coverpoint-uri sunt liniare cu distribuție în  $N$  intervale egale

$$b_i \leftrightarrow [2^i : 2^{i+1} - 1] \quad , i \in [0: N - 1] \cap \mathbb{N} \quad (5.6)$$

este mai mare indicele *bin-ului*, cu atât mai multe valori ale magistralei sunt incluse în acel interval. Caracteristica PMF neliniară pentru acest *coverpoint* este definită în (5.7)

$$P(b) = P(b_i) = \frac{2^i}{2^{N-1}} \quad , i \in [0: N - 1] \cap \mathbb{N} \quad (5.7)$$

În acest caz, probabilitatea crește împreună cu numărul de valori incluse în intervalul respectiv.

### 5.3.2 Rezultatele procesului de învățare automată

Ca și în studiul de caz anterior, se utilizează același set de *hiperparametri* de învățare. În mod similar, *procesul de reglare* utilizează aceleași intervale de valori ale *hiperparametrilor*, așa cum sunt prezentate în tabelul 5.1. Analiza se desfășoară în mai multe iterații, iar figura 5.2 arată că metricile de învățare indică rezultate interesante.

După analizarea soluției optime, observațiile sunt menționate în tabelul 5.4. Mai exact, după 312 *epoci* de învățare, *acuratețea* ajunge la 100%, dar *rata de acoperire* crește semnificativ de la 34,38% la 100% în doar 17 *epoci*. Comparativ cu acel

**Tabelul 5.4** Rezultatele performanței de învățare automată pentru coverpoint-ul neliniar cu distribuție de tipul „puterea a lui doi”

Metrică	Valoare	Numărul de epoci <sup>a</sup>
Acuratețe	100%	312
Rata de acoperire	100%	17
Funcția de cost	0,029	312
Rata de învățare	7.65E-04	312

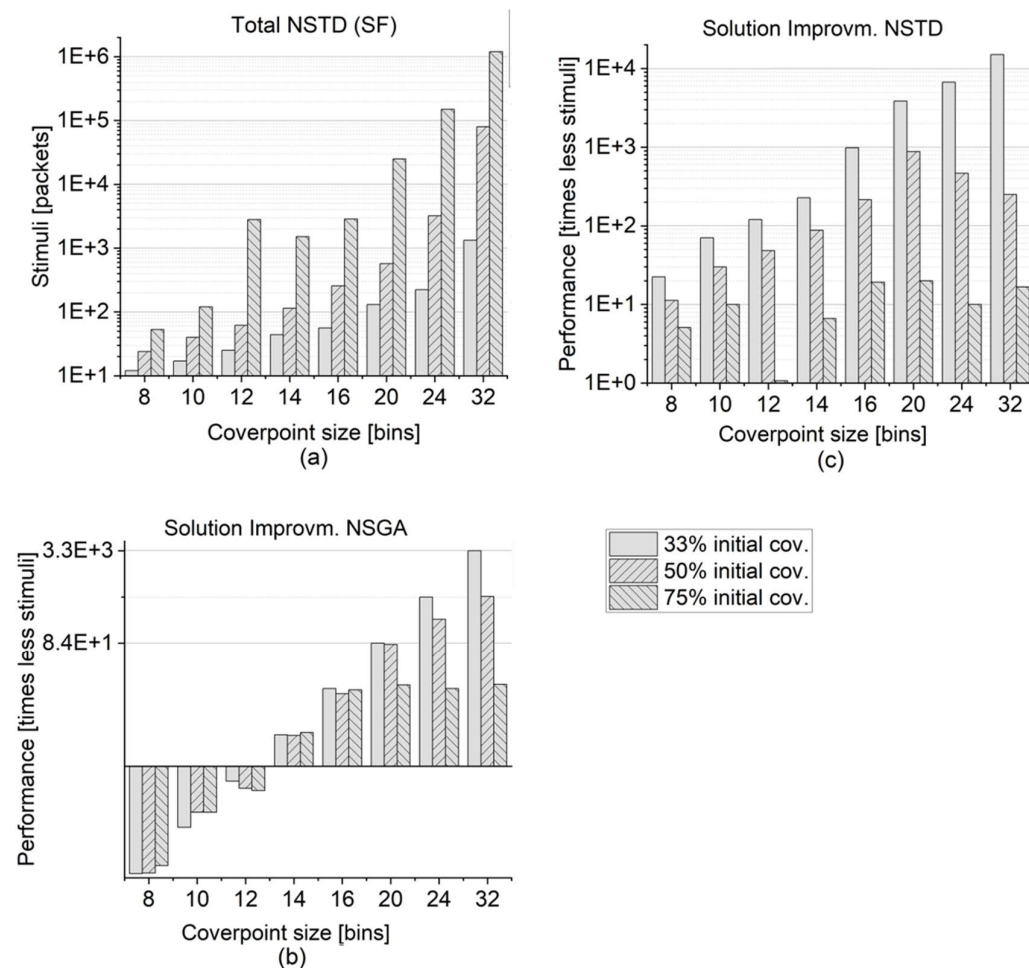
a. Numărul de epoci de învățare după care se atinge valoarea prezentată

coverpoint liniar având  $N = 1000$  intervale distribuite egal, acest model are doar  $C = 32$  bin-uri, ceea ce necesită un număr mai mic de epoci de învățare pentru a ajunge la acoperire completă de scenarii. Mai mult, procesul de învățare automată se încheie la epoca cu numărul 312, deoarece acuratețea ajunge la 100% în acel moment.

### 5.3.3 Rezultatele analizei comparative

Lista completă a rezultatelor analizei comparative este prezentată în tabelul 5.5. Deoarece forma acestui tabel este similară cu cea din tabelul 5.3, semnificația fiecărei coloane rămâne aceeași cu cele descrise deja în subcapitolul 5.2.3.

5.2.3 Din fericire, tabelul 5.5 indică o rată NSTD de cel puțin 5 ori și de până la 15000 mai bună în favoarea instrumentului **CAStiG** bazat pe metodologia **SL4SRRM**, față de tehnica CDV. Figura 5.6 are o reprezentare similară cu figura 5.3 și prezintă



**Figura 5.6** Rezultatele analizei comparative obținute pentru mai multe configurații ale coverpoint-ului neliniar cu distribuție de tipul „putere a lui doi”  
 (a) Valorile NSTD totale obținute utilizând instrumentul **CAStiG**  
 (b) Performanța NSGA a instrumentului **CAStiG**  
 (c) Performanța NSTD a instrumentului **CAStiG** în comparație cu abordarea CDV

**Tabelul 5.5** Rezultatele analizei comparative pentru coverpoint-ul neliniar cu distribuție de tipul „puterea a lui doi”

Rata inițială de acoperire [%]	Numărul de bin-uri din coverpoint (N)	NSTD (pentru CDV)	NSTD (Faza de antrenare pentru CASfiG)	NSTD (Ambele faze pentru CASfiG)	NSTD total (pentru CASfiG)	NSGA (Faza de antrenare pentru CASfiG)	NSGA (Faza de inferență pentru CASfiG)	NSGA total (pentru CASfiG)	Îmbun. CASfiG NSTD [ori mai puțin stimuli]	Îmbun. CASfiG NSGA [ori mai puțin stimuli]
33%	32	2.0E+7	1300	32	1332	2297	2441	6038	<b>15,015</b>	<b>3,312</b>
	24	1.5E+6	200	24	224	3622	3955	7777	<b>6,696</b>	<b>192</b>
	20	5.0E+5	110	20	130	2717	3093	5920	<b>3,846</b>	<b>84</b>
	16	5.5E+4	40	16	56	2552	3067	5659	<b>982</b>	<b>9.72</b>
	14	10000	30	14	44	2112	2538	4680	<b>227</b>	<b>2.14</b>
	12	3000	13	12	25	1853	1188	3054	<b>120</b>	<b>-1.02</b>
	10	1200	8	9	17	1678	2661	4347	<b>70</b>	<b>-3.62</b>
	8	270	5	7	12	1170	1879	3054	<b>22</b>	<b>-11.31</b>
50%	32	2.0E+7	80000	32	80032	2917	3066	85983	<b>249</b>	<b>232</b>
	24	1.5E+6	3200	24	3224	3594	3874	10668	<b>465</b>	<b>140</b>
	20	5.0E+5	550	20	570	2620	2888	6058	<b>877</b>	<b>82</b>
	16	5.5E+4	240	16	256	2774	3172	6186	<b>214</b>	<b>8.89</b>
	14	10000	100	14	114	2104	2554	4758	<b>87</b>	<b>2.10</b>
	12	3000	50	12	62	1791	2430	4271	<b>48</b>	<b>-1.42</b>
	10	1200	30	10	40	1429	1867	3326	<b>30</b>	<b>-2.77</b>
	8	270	17	7	24	1257	1749	3023	<b>11.25</b>	<b>-11.20</b>
75%	32	2.0E+7	1.2E+6	32	1.2E+6	3679	3865	1.2E+6	<b>16.67</b>	<b>16.56</b>
	24	1.5E+6	1.5E+5	24	1.5E+5	1940	2032	1.5E+5	<b>10.00</b>	<b>9.74</b>
	20	5.0E+5	2.5E+4	20	2.5E+4	3460	3803	3.2E+4	<b>19.98</b>	<b>15.50</b>
	16	5.5E+4	2850	16	2866	1406	1522	5778	<b>19.19</b>	<b>9.52</b>
	14	10000	1500	14	1514	1197	1301	3998	<b>6.61</b>	<b>2.50</b>
	12	3000	300	12	2800	2015	2324	4639	<b>1.07</b>	<b>-1.55</b>
	10	1200	110	10	120	1501	1711	3322	<b>10.00</b>	<b>-2.77</b>
	8	270	45	8	53	1200	1449	2694	<b>5.09</b>	<b>-9.98</b>

rezultatele *analizei comparative* listate în tabelul 5.5.

În graficul (a), valorile totale ale **NSTD** cresc împreună cu dimensiunea *coverpoint-ului*, iar acest comportament este similar cu cel observat în aplicația anterioară. O altă asemanare este că valorile **NSTD** scad împreună cu *rata inițială de acoperire*. Cu toate acestea, diferențele dintre valorile **NSTD** pentru *diferite rate inițiale de acoperire* sunt semnificativ mai mari pentru acest tip de *coverpoint*.

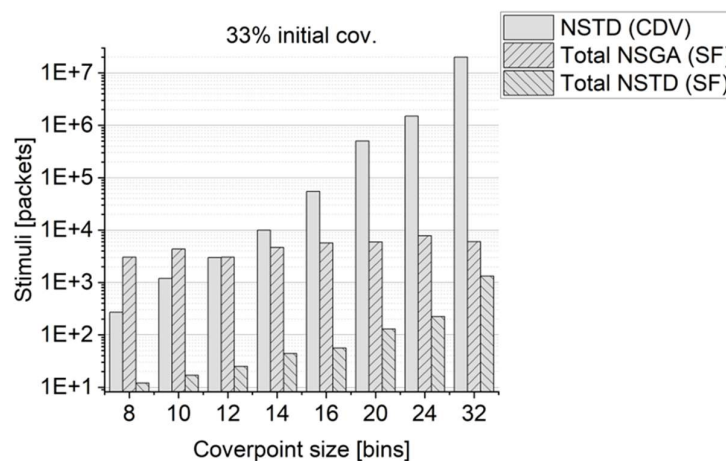
În mod similar, configurațiile cu o *rată inițială de acoperire* de 33% generează cele mai bune rezultate **NSTD**. Prin urmare, figura 5.7 prezintă o comparație între abordarea CDV și soluția instrumentului **CAStiG**. Ca în cazul distribuției liniare, diferențele **NSTD** cresc împreună cu dimensiunea *coverpoint-ului*.

În figura 5.6, graficul (c), îmbunătățirile **NSTD** cresc împreună cu dimensiunea *coverpoint-ului*, ca în cazul distribuției liniare. Cu toate acestea, variațiile de performanță între diferitele *rate inițiale de acoperire* sunt mult mai mari pentru această distribuție neliniară.

Graficul (b) indică performanța **NSGA**, iar rezultatele pozitive sunt obținute începând cu o dimensiune a *coverpoint-ului* de  $C = 14$  intervale. Pentru situațiile în care numărul de intervale este mai mic, ANN manifestă un proces costisitor în timpul epocilor de învățare, ceea ce le face mai puțin eficiente decât abordările CDV. Mai mult, penalitățile asupra **NSGA** scad, pe măsură ce dimensiunea *coverpoint-ului* crește.

Similar cu rezultatele obținute pentru *coverpoint-ul cu distribuție liniară în N intervale egale*, cel mai bun raport **NSTD** se obține atunci când *rata inițială de acoperire* este de 33%. Spre deosebire de cazul de utilizare anterior, cel mai bun raport **NSGA** este capturat atunci când *rata inițială de acoperire* este de 33%.

Spre deosebire de aplicația cu distribuție liniară, pentru care modelul de *analiză comparativă* a generat rezultate negative, obiectivul de *completare a acoperirii* pentru acest tip de *coverpoint* cu distribuție „putere a lui doi” a fost atins mai convenabil folosind instrumentul **CAStiG**.



**Figura 5.7** O comparație între CDV și instrumentul **CAStiG** când *rata inițială de acoperire* este de 33% iar configurațiile de *coverpoint-uri* sunt neliniare cu distribuție de tipul „puterea a lui doi”

# Capitolul 6

## Concluzii

### 6.1 Rezultate obținute

În capitolul 2, după prezentarea stadiului actual al literaturii privind strategiile VI, am comparat principalele abordări din tabelul 2.1 și am subliniat potențialul *automatizării generării de teste noi folosind feedback-ul de acoperire a scenariilor*. Aceasta este piatra de temelie pentru strategia mea de implementare a instrumentului **CASiG** bazat pe inedita metodologie **SL4SRRM**.

În capitolul 3, am evidențiat fundamentele teoretice pentru arhitectura noii metodologii **SL4SRRM**. Practic, am proiectat modelele de arhitecturale pentru fazele de antrenare și inferență.

În capitolul 4 descriu detaliile de implementare pentru noul instrument **CASiG** bazat pe metologia **SL4SRRM**. Acesta este un program software bazat pe interfața *Keras* [Goo15]. Am identificat cea mai eficientă arhitectură ANN și am descoperit că modelul de învățare automată *Adam* oferă cea mai bună acuratețe a învățării, dar și o convergență mai rapidă.

În capitolul 5, am selectat două dintre cele mai întâlnite distribuții de *coverpoint* și am efectuat un proces de *analiză comparativă* care indică reduceri semnificative ale numărului total de cicluri de simulare. Introduc metricile de performanță **NSTD** și **NSGA**, iar cele mai interesante rezultate indică o reducere a scorului **NSTD** de peste 15000 de ori, în timp ce rata de îmbunătățire a **NSGA** este mai mare de 3300.

### 6.2 Contribuții originale

Pe măsură ce industria identifică nevoia tot mai mare de noi metode și mai eficiente pentru a reduce costurile aferente încheierii verificării funcționale a CI, cercetarea mea a determinat următoarele contribuții originale:

1. În articolul de jurnal [Cri21], trec în revistă stadiul actual al automatizării verificării funcționale și contribui prin publicarea unui tabel comparativ, mai

precis, Tabelul 2.1. Descriu punctele forte și punctele slabe ale soluțiilor VI care prezintă perspective promițătoare. Mai mult, această contribuție este mult apreciată de către comunitatea academică, deoarece articolul meu din jurnalul [Cri21] a fost citat de cel puțin 21 de ori în perioada ianuarie 2021 - iunie 2023.

2. În lucrările mele de conferință [CB21, CC21], abordez strategia de *generarea de teste noi folosind feedback de la modelul de acoperire* prin implementarea unei ANN ca algoritm de *învățare automată*. După cum am subliniat în analiza de literatură [Cri21], nu au existat încercări anterioare de a realiza această automatizare folosind ANN. Astfel, am profitat de biblioteca *open-source TensorFlow* [Goo15] și am implementat instrumentul **CAStiG** descris în capitolul 4. Astfel,
  - a. Cu articol [CB21] contribuî prin introducerea noii metodologii **SL4SRRM** bazată pe ANN și demonstrez că *funcția de eșantionare* a unui *coverpoint cu distribuție liniară având  $N = 1000$  intervale egale* poate fi modelată cu succes folosind un *perceptron multistrat*. Unealta **CAStiG** bazată pe metodologia **SL4SRRM** atinge cu succes obiectivul de *acoperire completă a scenariilor* cu un efort de implementare redus. De asemenea, se indica flexibilitatea uneltei software în implementarea modelelor pentru orice tip de *coverpoint* definit de utilizator. Practic, procesul de *învățare automată* poate fi ușor paralelizat folosind o ANN pentru fiecare *coverpoint* al *modelului de acoperire a scenariilor*.
  - b. În articolul [CC21], extind analiza instrumentului **CAStiG** începută în [CB21] și contribuî demonstrând că *funcțiile de eșantionare neliniare* pentru *coverpoint-urile* cu distribuții de tipul „putere a lui doi” sau cele de tipul „*min și max*” sunt modelate cu succes folosind structuri ANN specifice. Soluțiile din această lucrare atenuază problemele identificate în literatură privind tehnicile de optimizare a proceselor asistate de ML. Rezultatele de performanță obținute indică perspective promițătoare pentru procesele de verificare funcțională asistate de instrumentele EDA bazate pe ANN.
3. În articolul de jurnal [Cri23], fac o explorare amplă pe diverse configurații de *coverpoint-uri* liniare și neliniare, introduc noile metrici de performanță, anume ratele de îmbunătățire **NSTD** și **NSGA** și contribuî la listarea rezultatelor de *analiză comparativă* care confirmă funcționalitatea **SL4SRRM** folosind ANN.
4. În articolul de conferință [VDC23], împreună cu ceilalți coautori, integrăm nucleul de *învățare automată* bazat pe **SL4SRRM** al instrumentului **CAStiG** [CB21] în arhitectura AMIQ ECTB [VD23]. Astfel, contribuî prin explorarea și listarea mai multor rezultate experimentale cu privire la modul în care nucleul **SL4SRRM** funcționează pentru un model de DUT real având funcționalități mai complexe.



## 6.3 Lista lucrărilor originale

În timpul acestui program de cercetare doctorală, am publicat cinci lucrări originale care abordează subiectul de cercetare ales și inițiativele mele descrise în această teză.

Am publicat două articole în conferințe internaționale indexate ISI, mai exact [CB21, CC21]. În plus, articolul [VDC23] a fost acceptat de către o conferință internațională neindexată ISI numită *Design and Verification Conference Europe 2023* (DVCON). Articolul va fi prezentat pe parcursul lunii noiembrie 2023. *DVCON Europe* este o conferință anuală care vizează subiectele de proiectare și verificare a CI și este sponsorizată de *Accellera System Initiative* [Acc00]. De asemenea, am publicat și două lucrări în reviste internaționale indexate ISI, care sunt [Cri21, Cri23].

În plus față de publicațiile menționate mai sus, am depus patru rapoarte ca parte a programului de cercetare științifică, și anume [Rep 1, Rep2, Rep3, Rep4]. În mod specific, raportul la care se face referire în [Rep1] este echivalat folosind manuscrisul înainte de publicarea articolului [Cri21]. În mod similar, [Rep2] este echivalat folosind lucrarea din [CB21]. [Rep3] este echivalat folosind lucrarea din [CC21], în timp ce [Rep4] este echivalat folosind articolul din [Cri23].

Lista completă a lucrărilor care vizează subiectul tezei de doctorat depuse în timpul programului meu de cercetare științifică este prezentată mai jos.

[CB21] M.C. Cristescu and C. Bob, *Flexible Framework for Stimuli Redundancy Reduction in Functional Verification Using Artificial Neural Networks*, IEEE, 2021 International Symposium on Signals, Circuits and Systems (ISSCS), Jul. 2021, pp. 1-4, doi:10.1109/ISSCS52333.2021.9497443

[CC21] M.C. Cristescu and D. Ciupitu, *Stimuli Redundancy Reduction for Nonlinear Functional Verification Coverage Models Using Artificial Neural Networks*, IEEE, 2021 International Semiconductor Conference (CAS), Oct. 2021, pp. 217-220, DOI:10.1109/CAS52836.2021.9604141

[VDC23] A. Vintilă, S. Dudă, and M.C. Cristescu, *An analysis on the impact of AI on digital IC verification coverage closure*, Design and Verification Conference (DVCon) Europe 2023, contribuție acceptată pe 23 iunie 2023, va fi prezentată la conferință pe 14 noiembrie 2023.

[Cri21] M.C. Cristescu, *Machine Learning Techniques for Improving the Performance Metrics of Functional Verification*, Romanian Journal of Information Science and Technology (ROMJIST), vol. 24, no. 1, Apr. 2021, pp. 99-116, ISSN:1453-8245

[Cri23] M.C. Cristescu, *Benchmarking a Smart Framework for Reducing the Coverage Closure Time in ASIC Functional Verification*, University POLITEHNICA of

Bucharest, Scientific Bulletin Series C - Electrical Engineering and Computer Science, 2023, ISSN: 2286-3540 (în proces de recenzare)

[Rep1] M.C. Cristescu, “*Tehnici de Machine Learning pentru îmbunătățirea metricilor de performanță în verificarea funcțională – O analiză a literaturii*”, Scientific Research Report no. 1, University POLITEHNICA of Bucharest, June 2020, echivalat folosind manustrisul articolului [Cri21].

[Rep2] M.C. Cristescu, *Flexible Framework for Stimuli Redundancy Reduction in Functional Verification Using Artificial Neural Networks*, Scientific Research Report no. 2, University POLITEHNICA of Bucharest, June 2021 echivalat folosind manustrisul articolului [CB21].

[Rep3] M.C. Cristescu, *Stimuli Redundancy Reduction for Nonlinear Functional Verification Coverage Models Using Artificial Neural Networks*, Scientific Research Report no. 3, University POLITEHNICA of Bucharest, June 2021, echivalat folosind manustrisul articolului [CC21].

[Rep4] M.C. Cristescu, *Benchmarking a Smart Framework for Reducing Coverage Closure Time in Functional Verification*, Scientific Research Report no. 4, University POLITEHNICA of Bucharest, June 2022, echivalat folosind manustrisul articolului [Cri23].

## 6.4 Perspective de dezvoltare ulterioară

### 6.4.1 Integrarea nucleului de învățare automată al instrumentului CASTiG în arhitectura ECTB

O primă dezvoltare ulterioară constă în implementarea aplicațiilor descrise în subcapitolele 5.2 5.25.35.3. Mai exact, am identificat o oportunitate interesantă de cercetare adusă de articolul menționat mai sus și am decis să integrez *nucleul de învățare automată* bazat pe ANN al instrumentului CASTiG în arhitectura AMIQ ECTB. Prin urmare, selectarea noilor definiții ale constrângerilor poate fi automatizată folosind modelul ANN oferit de instrumentul CASTiG.

După încheierea implementării subliniate în paragraful menționat mai sus, împreună cu ceilalți coautori, am descris acest proces în articolul [VDC23] care a fost acceptat de conferința *DVCon Europe 2023* și va fi prezentat oficial pe 14 noiembrie 2023. Cu toate acestea, experimentele pentru generarea unor rezultate de performanță și mai noi sunt în curs de dezvoltare și intenționăm să oferim rezultate preliminare atunci când lucrarea [VDC23] va fi prezentată oficial.

## 6.4.2 Alte direcții de cercetare

O dezvoltare viitoare interesantă este îmbunătățirea modelului ML al instrumentului **CAStiG** pentru *coverpoint-ul neliniar având bin-uri de tipul „min și max”*, astfel încât obținerea *acoperirii complete a scenariilor* să poată fi atinsă cu un efort de implementare rezonabil.

Analiza ulterioară a evidențiat noi perspective de extindere a capacităților instrumentului **CAStiG** prin investigarea unor arhitecturi ANN mai noi care se pot potrivi unor configurații de *coverpoint-uri* cu distribuții mai complexe.

O altă oportunitate de cercetare viitoare este de a utiliza instrumentul **CAStiG** alimentat de metodologia **SL4SRM** pentru optimizarea aplicațiilor de *acoperire completă a scenariilor* disponibile în fazele de verificare a proiectelor reale.

O altă direcție de investigare este combinarea punctelor forte ale algoritmilor GA și ILP într-un cadru hibrid. GA este potrivit pentru îmbunătățirea calității setului de antrenare, în timp ce ILP ar putea fi implicat în realizarea modelelor complexe de *coverpoint-uri*.

# Bibliografie

- [Don22] Z. Dong, *Research of Big Data Information Mining and Analysis: Technology Based on Hadoop Technology*, IEEE, 2022 International Conference on Big Data, Information and Computer Network (BDICN), Jan 2022, pp. 173-176, DOI: 10.1109/BDICN55575.2022.00041
- [LZC+14] Y.Q. Lv, Q. Zhou, Y.C. Cai, and G. Qu, Trusted Integrated Circuits: The Problems and Challenges, *Journal of Computer Science and Technology*, vol. 29, no. 5, Sep. 2014, pp. 918-928, DOI:10.1007/s11390-014-1479-9
- [Cri21] M.C. Cristescu, *Machine Learning Techniques for Improving the Performance Metrics of Functional Verification*, *Romanian Journal of Information Science and Technology (ROMJIST)*, vol. 24, no. 1, Apr. 2021, pp. 99-116, ISSN:1453-8245
- [Iee98] Standard 830-1998 - *IEEE Recommended Practice for Software Requirements Specifications*, IEEE, Oct. 1998, DOI:10.1109/IEEESTD.1998.88286
- [XZB+16] B. Xue, M. Zhang, W.N. Browne, and X. Yao, *A Survey on Evolutionary Computation Approaches to Feature Selection*, *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, Aug. 2016, pp. 606-626, DOI:10.1109/TEVC.2015.2504420
- [CB21] M.C. Cristescu and C. Bob, *Flexible Framework for Stimuli Redundancy Reduction in Functional Verification Using Artificial Neural Networks*, IEEE, 2021 International Symposium on Signals, Circuits and Systems (ISSCS), Jul. 2021, pp. 1-4, doi:10.1109/ISSCS52333.2021.9497443
- [Piz04] A. Piziali, Information Technology: Transmission, Processing and Storage. In: *Functional Verification Coverage Measurement and Analysis*, vol. 1, Springer New York, Jun. 2004, pp. 28, ISBN: 978-1-4020-8025-8
- [KKP06] S.B. Kotsiantis, D. Kanellopoulos, and P.E. Pintelas, *Data Preprocessing for Supervised Learning*, World Academy of Science, Engineering and Technology, *International Journal of Computer and Information Engineering*, vol. 1, no. 1, Jan. 2006, pp. 111-117, ISSN: 1306-4428
- [Hea15] J. Heaton, *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks*, Heaton Research Inc, CreateSpace Independent Publishing Platform, Dec. 2015, pp. 263, ISBN: 978-1505714340

- [BF00] I. Bruha and A. Famili, *Postprocessing in Machine Learning and Data Mining*, ACM SIGKDD Explorations Newsletter, vol. 2, no. 2, Dec. 2000, pp. 110-114, DOI:10.1145/380995.381059
- [Cri23] M.C. Cristescu, *Benchmarking a Smart Framework for Reducing the Coverage Closure Time in ASIC Functional Verification*, University POLITEHNICA of Bucharest, Scientific Bulletin Series C - Electrical Engineering and Computer Science, 2023, ISSN: 2286-3540 (în proces de recenzare)
- [UCI12] *Unified Coverage Interoperability Standard (UCIS)*, Version 1.0, 2012, Accellera Systems Initiative, Available at: [https://www.accellera.org/images/downloads/standards/ucis/UCIS\\_Version\\_1.0\\_Final\\_June-2012.pdf](https://www.accellera.org/images/downloads/standards/ucis/UCIS_Version_1.0_Final_June-2012.pdf) (URL link Jun. 2023)
- [Sta17] C. Stan, *CoverageLens 2.0 Release*, AMIQ Consulting, Dec. 2017, Available at: <https://github.com/amiq-consulting/CoverageLens/> (URL link Jun. 2023)
- [Goo15] *TensorFlow*, Google Brain, 2015, Available at: <https://github.com/tensorflow/tensorflow> (URL link Jun. 2023)
- [CC21] M.C. Cristescu and D. Ciupitu, *Stimuli Redundancy Reduction for Nonlinear Functional Verification Coverage Models Using Artificial Neural Networks*, IEEE, 2021 International Semiconductor Conference (CAS), Oct. 2021, pp. 217-220, DOI:10.1109/CAS52836.2021.9604141
- [Bir15] N.Ş. Birman, *Functional Coverage Patterns: Bitwise Coverage*, AMIQ Consulting, Sep. 2015, Available at: <https://www.amiq.com/consulting/2015/09/18/functional-coverage-patterns-bitwise-coverage/> (URL link Jun. 2023)
- [Acc00] Accellera Systems Initiative, Available at: [www.accellera.org](http://www.accellera.org) (URL link Jun. 2023)
- [VDC23] A. Vintilă, S. Dudă, and M.C. Cristescu, *An analysis on the impact of AI on digital IC verification coverage closure*, Design and Verification Conference (DVCon) Europe 2023, contribuție acceptată pe 23 iunie 2023, va fi prezentată la conferință pe 14 noiembrie 2023.
- [VD23] A. Vintilă and S. Dudă, *AMIQ ECTB Framework*, Available at: <https://www.amiq.com/consulting/2023/01/30/amiq-ectb-externally-controlled-testbench-architecture-framework/> (URL link Jun. 2023)
- [Rep1] M.C. Cristescu, *Tehnici de Machine Learning pentru îmbunătățirea metricilor de performanță în verificarea funcțională – O analiză a literaturii*, Scientific Research Report no. 1, University POLITEHNICA of Bucharest, June 2020, echivalat folosind manustrisul articolului [Cri21].

- [Rep2] M.C. Cristescu, *Flexible Framework for Stimuli Redundancy Reduction in Functional Verification Using Artificial Neural Networks*, Scientific Research Report no. 2, University POLITEHNICA of Bucharest, June 2021, echivalat folosind manustrisul articolului [CB21].
- [Rep3] M.C. Cristescu, *Stimuli Redundancy Reduction for Nonlinear Functional Verification Coverage Models Using Artificial Neural Networks*, Scientific Research Report no. 3, University POLITEHNICA of Bucharest, June 2021, echivalat folosind manustrisul articolului [CC21].
- [Rep4] M.C. Cristescu, *Benchmarking a Smart Framework for Reducing Coverage Closure Time in Functional Verification*, Scientific Research Report no. 4, University POLITEHNICA of Bucharest, June 2022, echivalat folosind manustrisul articolului [Cri23].