Proiect cofinanţat din Fondul Social European prin Programul Operaţional Sectorial Dezvoltarea Resurselor Umane 2007-2013
Investeşte în oameni!

Proiect InnoRESEARCH - POSDRU/159/1.5/S/132395
Burse doctorale şi postdoctorale în sprijinul inovării şi competitivităţii în cercetare

UNIVERSITATEA **POLITEHNICA** DIN BUCUREŞTI
Facultatea Automatică şi Calculatoare
Departamentul _____

Nr. Decizie Senat _____din _____

Metode Defensive pentru Asigurarea Securităţii Cibernetice

Intelligent Defensive Techniques for Cyber Security Assurance

REZUMAT AL TEZEI DE DOCTORAT

Autor: ing. Adriana Cristina Enache
Conducător de doctorat: prof. dr. ing. Valentin Sgârciu

COMISIA DE DOCTORAT

Preşedinte	Prof. univ. dr. ing. Theodor Borangiu	de la	Universitatea Politehnica Bucureşti
Conducător de doctorat	Prof. univ. dr. ing. Valentin Sgârciu	de la	Universitatea Politehnica Bucureşti
Referent	Prof. univ. dr. ing. Victor Patriciu	de la	Academia Tehnică Militară
Referent	Prof. univ. dr. ing. Ion Bica	de la	Academia Tehnică Militară
Referent	Prof. univ. dr. ing. Dorin Cârstoiu	de la	Universitatea Politehnica Bucureşti

Bucureşti

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013
Investește în oameni!

Cuprins

UNIUNEA EUROPEANĂ
Fondul Social European POSDRU 2007-2013
Instrumente Structurale 2007-2013
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE
OIPOSDRU
Universitatea POLITEHNICA din Bucuresti

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013
Investește în oameni!

1 INTRODUCERE

Tehnologia a evoluat într-o manieră galopantă şi a catalizat ritmul de creştere al numărului de utilizatori, volumului de date sau dimensiunii reţelelor. Prin urmare, atât entităţile private cât şi cele guvernamentale trebuie să ţină pasul şi să fie capabile de a se adapta la noile tendinţe din domeniul tehnologiei informaţiei pentru a îşi asigura un loc cât mai sus în lista de potenţiali colaboratori. Securitatea cibernetică se încadrează în regula anterioară, mai ales că scopul acesteia este de a proteja aceste sisteme noi de atacurile cibernetice aflate într-un continuu proces de metamorfoză şi extindere către noile platforme. Pentru a asigura sistemelor informatice un nivel minim de securitate, trebuie în primul rând să cunoaştem contextul actual al ameninţărilor cibernetice şi să dispunem de metode şi mecanisme avansate pentru detecţia acestora.

Soluţiile pe care le propunem în cadrul acestei teze sunt bazate pe algoritmi de inteligenţă computaţională (CI), care în opinia noastră pot oferi metode pentru crearea unor mecanisme automate şi capabile de a oferi adaptabilitatea la noile transformări din ecosistemul cibernetic. În cadrul activităţii de cercetare, interesul nostru s-a îndreptat spre acei algoritmi inspiraţi din principiile şi funcţionalităţile sistemelor biologice, precum: Swarm Intelligence (SI) şi Artificial Immune Systems (AIS). Natura a reprezentat întotdeauna o sursă de inspiraţie pentru mediul ştiinţific datorită echilibrului natural al componentele sale care creează un sistem eficient, adaptabil la schimbările perpetue existente în mediul biologic şi durabil în timp.

În cadrul acestie teze sunt propuse două versiuni îmbunătăţite pentru un algoritm SI recent (the Bat Algorithm) şi o variantă modificată pentru un algoritm AIS de nouă generaţie inspirat din paradigma teoriei pericolului (Dendritic Cell Algorithm). Pentru a valida ipotezele teoretice am implementat noile versiuni, apoi le-am integrat în cadrul unor sisteme pentru detecţia intruziunilor (IDS) şi am testat modelele propuse pe baze de date publice care conţin înregistrări specifice anomaliilor şi respectiv normale. Mai mult, am propus arhitecturi pentru detecţia intruziunilor care combină algoritmi inspiraţi biologic cu algoritmi de clasificare convenţionali, demonstrând astfel că această îmbinare poate îmbunătăţi performanţele modelului. Totodată, am adresat problema de detecţie web spamming prin propunerea, implementarea şi testarea a două modele. Primul asociază SI cu clasificatorul de maşini cu suport vectorial, iar cel de al doilea apelează la cAnt-Miner pentru sarcina de clasificare.

Modelele pe care le propunem în cadrul acestei teze pot fi integrate cu uşurinţă în cadrul unor soluţii pentru asigurarea securităţii cibernetice pentru componentele de extragere a atributelor cheie sau pentru procesul de detecţie a anomaliilor. Prin urmare, rezultatele activităţii de cercetare au o gamă largă de aplicabilităţi şi pot fi valorificate sub forma unor componente modulare pentru diverse sisteme precum: security and event management (SIEM), visual analytics, sisteme pentru detecţia intruziunilor (IDS) sau alte produse pentru asigurarea securităţii sistemelor informatice şi de comunicaţii.

1.1 MOTIVAŢIA

În ziua de azi, aproape orice individ are contact cu lumea cibernetică unde sunt stocate informaţii publice, dar şi unele confidenţiale sau sensibile ce trebuie protejate. Aceste date stocate electronic au devenit motorul activităţilor şi proceselor esenţiale desfăşurate de entităţi, întrucât din aceste date se pot extrage informaţii valoroase despre clienţi, furnizori sau chiar secrete care stau la baza produselor sau serviciilor oferite. Pierderea sau deteriorarea acestor date poate duce la decredibilizarea acelei companii, pierderi financiare uriaşe sau poate determina chiar situaţii internaţionale indezirabile (ex. spionajul cibernetic dintre

guvernele unor state). Cum orice informaţie valoroasă atrage atenţia indivizilor cu intenţii maliţioase, nu este de mirare că numărul şi varietatea atacurilor cibernetice continuă să crească în fiecare zi. Termenul de „cybersecurity" a devenit un cuvânt *metaforic* în concepţia utilizatorilor de rând care nu îi înţeleg complet semnificaţia şi nu ştiu prin ce metode să se protejeze de ameninţările din mediul cibernetic, este o *corvoadă* pentru dezvoltatorii de software care trebuie să se alinieze la standardele de securitate care de multe ori reduc performanţele sistemelelor informatice, şi respectiv este o *necesitate* pentru managementul companiilor care au devenit conştienţi de importanţa securităţii informatice.

În ciuda interesului ridicat şi progreselor în domeniul securităţii informatice, furtuna de atacuri cibernetice continuă să se intensifice în rafale continue din punct de vedere al numărului şi diversităţii acestora. O dată cu evoluţiile tehnologiei informaţiei, actorii cibernetici cu intenţii maliţioase şi-au extins aria de atac către noile dispozitive sau platforme, acolo unde au găsit vulnerabilităţi şi au apelat la tehnici avansate (ex. algoritmi de criptare) pentru a îşi îmbunătăţi capabilităţile de atac. Aşadar, instrumentele de hacking au evoluat şi ele, astfel încât astăzi până şi persoanele mai puţin instruite dar cu intenţii maliţioase pot intra în „arena" cybercrime prin cumpărarea sau închirierea unor instrumente de hacking la costuri reduse de pe „piaţa neagră" a Internetului (ex. DarkWeb). Prin urmare, este clar că peisajul atacurilor cibernetice este într-o continuă expansiune din punct de vedere al *volumului* (din ce în ce mai mulţi indivizi cu intenţii maliţioase se pot „integra" uşor în ecosistemul cibernetic prin simpla închiriere a unor instrumente de hacking sau cumpărarea unor servicii de cyber-crime) şi *complexităţii* (noi tipuri de atacuri cibernetice ies la suprafaţă sau continuă să rămână ascunse în sistemele noastre informatice până la declanşarea sau descoperirea lor).

Luând în considerare toate cele prezentate anterior, am putea considera că asigurarea securităţii poate reprezenta un proces de tip cerere de cauţiune pentru utilizatorii care devin victimele sigure ale atacatorilor şi se află la mila acestora (ex. Ransomware). Aşadar, pentru a preveni aceste situaţii, trebuie asigurat un nivel adecvat de securitate pentru sistemele informatice prin implementarea unor politici de securitate şi integrarea unor soluţii eficiente de securitate. Cu toate acestea, monitorizarea şi auditarea tuturor evenimentelor din interiorul sistemelor informatice nu este o chestiune practică şi poate duce la încărcarea ineficientă a sistemelor. Mai mult, interpretarea unui astfel de volum de date este imposibil de realizat de către utilizatorii umani şi dificil de procesat de către sistemele informatice. Soluţiile pe care le propunem în cadrul acestei teze de doctorat se referă la utilizarea algoritmilor CI pentru filtrarea datelor şi extragerea acelor componente lor cheie pentru etapa de detecţie. Algoritmii de tip CI pot procesa volume mari de date într-o manieră eficientă prin utilizarea cunoaşterii şi a capacităţii de învăţare continuă; având în vedere contexul dinamic al mediului cibernetic, cele două caracteristici ale CI reprezintă atribute benefice pe termen lung şi pot ajuta la implementarea unor soluţii de securitate viabile şi cu un grad ridicat de adaptabilitate.Aceste ipoteze ne-au ghidat activitatea de cercetare şi au conturat principalele obiective ale proiectului de cercetare.

1.2 OBIECTIVELE GENERALE

Principala direcţie de cercetare a acestei lucrări se referă la propunerea şi testarea unor tehnici inovative pentru asigurarea securităţii cibernetice, care să fie bazate pe algoritmi de inteligenţă computaţională.

Principalele obiective generale ale tezei sunt enumerate mai jos:

- cercetarea algoritmilor de inteligenţă computaţională şi a aplicabilităţii acestora pentru asigurarea securităţii cibernetice;
- propunerea şi testarea unor modele defensive de tip pro-active bazate pe algoritmi de inteligenţă computaţională;

- îmbunătăţirea unor algoritmilor de inteligenţă computaţională şi aplicarea acestora pentru asigurare a securităţii cibernetice.

1.3 ORGANIZAREA ŞI STRUCTURA TEZEI

Am organizat acestă lucrare în opt capitole care urmăresc etapele principale ale activităţii de cercetare, pornind de la definiţia şi contextul actual al securităţii cibernetice şi respectiv introducerea conceptului de inteligenţă computaţională (CI). Principalele contribuţii personale se regăsesc în capitolele următoare care prezintă variantele modificate pentru doi algoritmi inspiraţi biologic (the Bat Algorithm şi Dendritic Cell Algorithm) şi respectiv modelele care integrează algoritmii îmbunătăţiţi cu cei convenţionali pentru a crea arhitecturi inovative pentru detecţia intruziunilor sau pentru detecţia web spamming.

Capitolul doi descrie conceptul de securitate cibernetică şi principalele componente ale mediului cibernetic din pespectiva autorului. În continuare sunt prezentate o serie de ameninţări cibernetice pentru a evidenţia o parte din posibilele pericole care afectează nivelul de securitate al sistemelor informatice.

Capitolul trei prezintă algoritmii de inteligenţă computaţională (CI), începând cu definiţia şi conceptele care stau la baza CI. Principalele obiective urmărite în cadrul acestui capitol se referă la introducerea algoritmilor utilizaţi în experimentele realizate. Aceşti algoritmi au fost împărţiţi în două mari categorii: convenţionali şi respectiv inspiraţi biologic. În ceea ce priveşte algoritmii inspiraţi biologic, aici ne-am axat pe două categorii: Swarm Intelligence şi Artificial Immune Systems. Am realizat o comparaţie a acestor două clase evidenţiind similitudinile şi respectiv caracterul lor complementar, din perpectiva autorului.

Capitolul patru include contribuţiile proprii referitoare la propunerea a două versiuni îmbunătăţite pentru un algoritm SI recent (the Bat Algorithm) şi respectiv o variantă modificată a unui algoritm AIS de generaţie nouă (Dendritic Cell Algorithm).

Capitolul cinci este un preambul al capitolului şase. Aici sunt prezentate istoria, definiţia şi o clasificare a modelelor de detecţie a intruziunilor (IDS).

Capitolul şase include modelele IDS propuse care au la bază algoritmi SI sau AIS pentru componente de selectare a atributelor sau pentru etapa de detecţie a anomaliilor.

Capitolul şapte se referă la detecţia WebSpamming, pentru care am propus două metode bazate pe algoritmi SI.

La final, concluziile şi contribuţiile proprii sunt incluse în capitolul opt. Tot aici sunt enumerate câteva perspective de cercetare.

1.4 ACKNOWLEDGEMENT

Rezultatele prezentate in acest articol au fost obtinute cu sprijinul Ministerului Fondurilor Europene prin Programul Operational Sectorial Dezvoltarea Resurselor Umane 2007-2013, Contract nr. POSDRU/159/1.5/S/132395.

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132395.

2 CONTEXTUL CIBERNETIC ACTUAL

Astăzi tehnologia a devenit în mod progresiv o componentă omniprezentă şi a îmbunătăţit atât viaţa personală cât şi activităţile de lucru ale oamenilor, generând astfel din ce în ce mai multe avantaje în viaţa de zi cu zi dar în acelaşi timp deschizând noi portiţe de intrare în sistemele informatice, sub forma unor vulnerabilităţi care pot fi exploatate de atacatori. În

fiecare zi dispozitivele devin mai inteligente şi mai conectate la Internet. O dată cu aceasta, valoare datelor stocate şi procesate creşte foarte rapid, generând astfel necesitatea de a proteja aceste date critice care au devenit comustibilul proceselor esenţiale desfăşurate de companiile private şi entităţile guvernamentale.

Datorită valorii datelor, acestea atrag interesul atacatorilor care vor încerca să obţină diverse avantaje de pe urma acestora. Profilul atacatorului este divers, iar motivaţia acestuia poate varia de la convingeri sociale sau afinităţi politice (hacktivism), beneficii economice (cyber-crime) până la spionaj sau cyber-warfare (susţinute de entităţi guvernamentale). În acest peisaj cibernetic dinamic, ameninţările cibernetice au devenit din ce în ce mai complexe în ultimii ani, de la simple scheme de phishing până la campanii complexe numite Advanced Persistent Threats (APTs) care au vizat instituţii financiare, companii industriale sau chiar entităţi guvernamentale. Prin urmare, pentru a putea asigura securitatea cibernetică trebuie să avem o privire de ansamblu asupra conceptului de securitate cibernetică şi respectiv despre contextul actual al ameninţărilor cibernetice.

2.1 SECURITATEA CIBERNETICĂ

Există mai multe definiţii acceptate în casta specialiştilor pentru termenul de ***securitatea cibernetică***, majoritatea împărtăşind aceeaşi piloni principali : confidenţialitatea, integritatea şi disponibilitatea. Conform ITU (Rec. ITU-T X.1205, Overview of cybersecurity, (04/2008)): "Cybersecurity is the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets. Organization and user's assets include connected computing devices, personnel, infrastructure, applications, services, telecommunications systems, and the totality of transmitted and/or stored information in the cyber environment. Cybersecurity strives to ensure the attainment and maintenance of the security properties of the organization and user's assets against relevant security risks in the cyber environment. The general security objectives comprise the following:

- Availability
- Integrity, which may include authenticity and non-repudiation
- Confidentiality."[13]

Cu alte cuvinte, securitatea cuprinde toate metodele şi tehnicile, de la simplele politici de securitate până la sistemele avansate de securitate, toate acestea conlucrând împreună pentru asigurarea unui nivel de securitate adecvat al sistemului ţintă. Un element interesant de remarcat în cadrul definiţiei date de ITU este utilizarea termenului „relevant risks", care reliefează faptul că întotdeauna vor exista riscuri reziduale, iar protejarea completă a sistemelor informatice este aproape imposibilă. În schimb, ceea ce îşi propune securitatea informatică este să prevină ameninţările cele mai probabile şi toate acestea la un cost cât mai redus.

În opinia noastră securitatea informatică este compusă din două componente principale: bunurile critice (constând în elementele hardware sau software care stochează, procesează şi transferă datele sensibile care trebuie protejate) şi jucătorii (atacatorii şi experţii de securitate). Aceste două componente reflectă câmpul de luptă din mediul cibernetic şi cursa continuă în care se află producătorii de malware cu experţii de securitate, fiecare încercând să utilizeze tehnologia informaţiei pentru propriile scopuri. Ecosistemul cibernetic este asimetric şi fără frontiere, prin urmare schemele de atac pot fi diverse şi forţa lor de penetrare se poate exinde pe suprafeţe teritoriale întinse pe mai multe continente. De la început, trebuie să stabilim faptul că panorama de ameninţări cibernetice este enormă şi are o rată de creştere

exponenţială. Prin urmare, scopul acestui capitol este de a oferi o mică parte din lumea complexă a mediului cibernetic şi a ameninţărilor existente.

2.2 EXEMPLE DE ATACURI CIBERNETICE

2.2.1 Distributed Denial of Service (DDoS)

Distributed Denial of Service (DDoS) este un tip de ameninţare în care mai multe sisteme ţintă sunt compromise în vederea obţinerii unui nivel scăzut de securitate sub forma privării acesului utilizatorilor autorizaţi la anumite resurse. Pentru a realiza aceasta, atacatorul poate inunda sistemul cu un număr mare de mesaje până la punctul în care resursa vizată devine indisponibilă. DDoS este un atac vizibil la nivelul administratorului şi de aceea este frecvent utilizat ca un instrument pentru hacktivism sau chiar pentru şantaj. Primele astfel de atacuri s-au desfăşurat în aprilie 2007 şi au avut drept ţinte organizaţiile din Estonia, inclusiv instituţiile bancare, guvernamentale şi chiar mass-media. Mărul discordiei se pare să fi fost relocarea unei statui din bronz a lui Tallinn, un simbol pentru minoritatea rusă din Estonia.

Deşi aparent simplu de realizat, atacurile de tip DDoS continuă să dăinuie în ciuda metodelor de securitate constatate, încă mai există hibe ale protocoalelor (ex. SlowLoris exploatează o vulnerabilitate a serverului Web Apache privind cererile de tip HTTP GET; aceste cereri pot include un număr limitat de linii permiţând astfel retrasmiterea lor la infinit) sau sistemelor (ex. neimplementarea unor politici minime de securitate). Mai mult, dacă până acum atacatorii aveau nevoie de o puterea de calcul majoră prin crearea unei reţele de victime (botnets), astăzi aceştia pot apela la serviciile de cloud computing în acelaşi scop şi toate la un cost redus care poate ajunge la suma de 150$ pentru o săptămână.

2.2.2 Advanced Persistent Threats (APTs)

Advanced Persistent Threat (APT) este un tip de ameninţare care utilizează tehnici de exploatare sofisticate pentru a pătrunde în sistemele ţintă şi a rămâne nedescoperite pentru o perioadă îndelungată de timp în care atacatorul va recolta informaţii valoroase. Spre deosebire de DDoS, aceste atacuri nu sunt vizibile imediat, iar obiectivele APTs nu sunt imedate ci din contră, urmăresc să rămână ascunse în timp ce extrag datele urmărite.

Multe dintre APTs de astăzi au fost dezvoltate în urmă cu câteva decenii şi prin urmare entităţile care se bazează pe tehnicile tradiţionale de securitate sunt mai mult ca sigur vulnerabile la aceste tipuri de atacuri. În istoria APT există mai multe astfel de exemple, printre primele se numără Moonlight Maze care a avut drept ţinte entităţi guvernamentale ale SUA, apoi Titan Rain sau Operaţiunea Aurora. Toate aceste atacuri s-au prelungit pe parcursul unei perioade îndelungate, existând în sistem fără ca utilizatorii să fie conştienţi de prezenţa acestora şi permiţând extragerea unor informaţii secrete. Probabil unul dintre cele mai răsunătoare APTs este ***Stuxnet,*** care a generat o frenezie mediatică în anul 2010. Ce face acest atac interesant este faptul că este primul vierme informatic utilizat ca armă în războiul cibernetic tacit şi care permitea obţinerea controlului şi sabotarea facilităţilor industriale prin operarea acestora în afara parametrilor autorizaţi. Acest tip de malware infecta PCLs (Programmable Logic Controllers) şi oferea posibilitatea de a fi reprogramate de către atacatori, astfel încât să obţină degradarea componentelor industriale, în timp ce utilizatorul uman primea informaţii eronate sub forma unor valori normale transmise de malware. Se pare Stuxnet a vizat acele sisteme care rulau Siemens software Step 7 şi conform rapoartelor Symatec [14] majoritatea victimelor se aflau în Iran, ceea ce a dus la incriminarea guvernelor SUA şi a celui israelian, deşi niciunul dintre acestea nu şi-au recunoscut în mod public implicarea. Un alt aspect important, este faptul că au existat mai multe versiuni ale acestui malware, care se putea actualiza, iar ciclul său de viaţă era limitat întrucât avea programată o

dată de auto-distrugere (24 iunie 2012). Deşi o parte dintre vulnerabilităţile exploatate (CVE-2010-2568) au fost soluţionate, se pare că rezolvările au fost superficiale iar o parte dintre ele au fost redescoperite în august 2015 de către un cercetător de la HP şi resoluţionate de către Microsoft. Prin urmare, sistemele Windows neactualizate încă sunt vulnerabile la aceste tipuri de ameninţări.

2.2.3 Ransomware

Ransomware este un tip de ameninţare care limitează accesul utilizatorilor autorizaţi la sistemele sau fişierele proprii, dacă nu este plătită o răscumpărare. Acest malware a fost creat în esenţă pentru a produce bani, iar tehnicile de propagare sunt diverse precum: ataşamente trimise prin e-mail, linkuri compromise integrate în site-uri web sau chiar trimise prin IM (instant messaging). Acest tip de ameninţare a evoluat în mod constant şi a pornit de la o „joacă", ajungându-se la instrumente de hacking pentru obţinerea de avantaje materiale imediate.

Primul ransomware cunoscut este PC Cyborg care a fost dezvoltat de către J. Popp în anul 1989. Acest malware ascundea fişierele şi cripta numele acestora, cerând victimei bani pentru reînnoirea licenţei. Apoi, ransomware a evoluat sub forma unor avertismente de necesitatea de a actualiza sau instala un program anti-virus şi ajungând în prezent să utilizeze metode avansate de ciptare asimetrică pentru a priva accesul utilizatorului la date. Printre cele mai cunoscute Ransomware sunt CryptoLocker (2013) şi CryptoWall (cu cele patru versini ale sale). Ambele utilizează criptare asimetrică şi reţelele TOR pentru a comunica cu centru de comandă şi control (C&C). Ceea ce face aceste ameninţări periculoase este faptul că exploatează naivitatea utilizatorilor care deschid orice ataşament sau dau un click pe un link aparent inofensiv şi ajung să fie victimele atacatorilor care le vor cripta datele şi apoi îi vor momi spre plătirea unei sume de bani în speranţa obţinerii datelor. Mai mult, algoritmii de criptare utilizaţi sunt avansaţi şi greu de spart, iar recomandările specialiştilor se referă la păstrarea unor copii de siguranţă a datelor importante.

2.3 NECESITATEA UNOR SOLUŢII INTELIGENTE

Evoluţia atacurilor cibernetice depinde de evoluţia tehnologiei şi a oamenilor care o dezvoltă. Cu alte cuvinte, noua tehnologie este o sabie cu două tăişuri, de care atacatorii pot beneficia pentru a îşi implementa propriile metode de protecţie, precum autentificarea botnet-urilor, criptarea datelor aparţinând victimelor (ex. Ransomware) sau anonimizarea plăţilor (ex. BitCoin). Prin urmare, este clar că avem nevoie de noi soluţii de securitate de tip pro-active care să permită protejarea bunurilor critice. O dată cu creşterea volumului de date generate de utilizatori, dezvoltarea uneltelor de hacking şi metamorfozarea malware-ului, soluţiile de securitate trebuie să evolueze şi ele, să fie capabile de a monitoriza şi proteja sistemele informatice într-un mod inteligent. În cadrul acestei teze prin ***soluţii inteligente*** înţelegem că sunt bazate pe algoritmi de inteligenţă computaţională (CI).

3 ALGORITMI DE INTELIGENŢĂ COMPUTAŢIONALĂ

Părintele maşinilor inteligentă este considerat Alan Turing, care în anul 1950 îşi publica lucrarea „Computing Machinery and Intelligence". Prin urmare domeniul inteligenţei artificiale este destul de nou, dacă îl comparăm cu altele precum matematica, fizica sau chiar domeniul calculatoarelor. Am menţionat inteligenţa artificială (AI), întrucât inteligenţa computaţională (CI) este o subramură a acesteia şi de multe ori cei doi termeni sunt confundaţi. Diferenţa este explicată de către Bezdek în 1994 [15] care puncta faptul că AI se

referă la procesarea simbolurilor şi raţionarea acestora, pe când CI se adresează raţionării lingvistice, numerice sau granulare.

Inspiraţia algoritmilor CI este destul de holisticăşi cuprinde domenii precum: psihologia, selecţia naturală sau colaborarea din cadrul grupurilor de animale, teoriile moderne privind logica sau chiar metode probabilistice care stau la baza luării unei decizii. Cu alte cuvinte, CI încearcă să transpună *comportamentul inteligent* în modele matematice pentru a construi algoritmi în vederea rezolvării unei game diverse de probleme. Unul dintre atributele cheie ale CI este învăţarea, care se referă de fapt la extragerea şi utilizarea cunoşterii pentru obţinerea unor informaţii sau abilităţi noi. În funcţie de *metoda de învăţare* putem avea CI supervizaţi, nesupervizaţi, semi-supervizaţi sau RL (reinforcement learning). Există mai multe astfel de clasificări ale algoritmilor CI. În ceea ce priveşte proiectul nostru de cercetare, vom împărţi algoritmii în două categorii: convenţionali (ex. arborii de decizie, maşinile cu suport vectorial sau Naive Bayes) şi cei inspiraţi din paradigme biologice (swarm intelligence şi artificial immune systems). Această clasificare ne va ajuta să distingem algoritmii bine-cunoscuţi sau convenţionali de cei inspiraţi biologic, SI şi AIS, care fac obiectul cercetării noastre. Pentru ultima categorie de algoritmi am realizat o comparaţie proprie şi am propus câteva versiuni îmbunătăţite. Prin urmare, în continuare ne vom limita la introducerea conceptelor care stau la baza algoritmilor SI şi AIS, prezentarea versiunilor originale şi cele modificate pentru doi algoritmi. Detaliile privind algoritmii convenţionali şi alţi algoritmi SI şi AIS utilizaţi în modelele propuse se regăsesc în teza autorului.

3.1 SWARM INTELLIGENCE

Swarm Intelligence (SI) *este o ramură a inteligenţei coputaţionale care se ocupă cu sistemele naturale sau artificiale compuse din mai mulţi indivizi sau agenţi care se auto-organizează şi nu necesită un punct de comandă central. Indivizii din sistem sunt relativ omogeni (sunt asemănători sau fac parte dintr-o anumită tipologie) şi interacţiunea dintre aceştia se bazează pe reguli simple ce depind de informaţiile locale schimbate cu alţi indivizi în mod direct sau prin mediu (stigmergy). Deşi nu există un punct de comandă central care să dicteze comportamentul indivizilor, interacţiunea dintre aceştia generează la nivel global un comportament inteligent al populaţiei (swarm).[3]*

Conceptul de swarm intelligence este introdus în 1989 de către Gerardo Beni şi Jim Wang în contextul sistemelor de roboţi. Pornind de la agenţi care se puteau autoorganiza, cercetătorii au extins conceptul de swarm intelligence pentru a defini algoritmii inspiraţi din acest comportament şi din inteligenţa colectivă a unor colonii de insecte sau animale. Progresul major este atins după 1990, începând cu algoritmul propus de Marco Dorigo în teza sa de docorat, care s-a inspirat din comportamentul coloniilor de furnici, algoritm intitulat ***Ant Colony Optimization***. Apoi, în 1995 un nou algoritm având la bază paradigma zborului cârdurilor de cocori (***Particle Swarm Optimization***) este propus de către J. Kennedy şi R. Eberhart. În următoarele decenii au apărut o multitudine de algoritmi inspiraţi din comportamentul diverselor insecte sau animale precum albinele (Artificial Bee Colony, 2005), licuricii (Firefly Algorithm, 2008) sau liliecii (Bat Algorithm, 2010).

În general algoritmii de swarm intelligence au la bază două componente : ***explorarea şi exploatare***, care sunt aplicate în mod diferit, de unde şi diversitatea algoritmilor. Explorarea este componenta aleatoare a algoritmului care îi va permite acestuia să abordeze o plajă cât mai variată de posibile soluţii. Cea de a doua componentă se referă la îmbunătăţirea acestei soluţii candidat când anumite condiţii sunt întrunite. Majoritatea acestor algoritmi euristici implementează exploatarea prin algoritmi de tip greedy, precum ABC, sau prin metode de căutare locală, precum ACO. În orice caz, indivizii din grup îşi vor alege următorul candidat în funcţie de experienţa personală dar şi de cea acumulată la nivelul grupului. O altă

componentă importantă este ***funcţia fitness*** (sau funcţia de performanţă), care asociază un cost fiecărei soluţii candidat. Strategia algoritmului urmăreşte îmbunătăţirea valorii funcţiei şi obţinerea unei soluţii aşa-zis optime. Prin urmare, alegerea componentelor din cadrul funcţiei poate influenţa performanţele algoritmului.

În ultimele două decenii, algoritmii SI au devenit din ce în ce mai populari. Principalele motive fiind: *simplitatea (*majoritatea algoritmilor sunt uşor de înţeles şi de implementat), *flexibilitatea* (deşi simpli, algoritmii SI oferă flexibilitate, putând fi utilizaţi pentru o multitudine de aplicaţii cum ar fi probleme de optimizare, NP etc.) şi *robusteţea* (algoritmul va funcţiona corect chiar dacă o parte dintre indivizi nu vor reuşi să îşi realizeze sarcinile). *[4]*

3.1.1 Bat Algorithm

Bat Algorithm (BA) este un algoritm de Swarm Intelligence propus în 2010 de către Yang [16], care s-a inspirat din ecolocaţia liliecilor. Când îşi caută hrana, aceştia emit vibraţii de sunete cu frecvenţe înalte care îi ajută să aproximeze diferenţa dintre un obtacol şi o ţintă urmărită. Pentru a transpune acest comportament într-un algoritm inteligent, autorul emite trei ipoteze:

- Toţi liliecii folosesc ecolocaţia pentru a aproxima diferenţa dintre un obstacol şi o ţintă.
- Liliecii zboară aleatoriu şi traiectoria lor este definită de locaţia lor în spaţiu (x_i) şi viteza de mişcare (v_i). Aceste două variabile sunt calculate în funcţie de frecvenţa ($freq_i$), amplitudinea (A_i) şi vibraţia emisiilor sonore (r_i) ale individului. Mai mult, în funcţie de proximitatea prăzii, individul îşi va ajusta amplitudinea şi vibraţiile emisiilor.
- Ultima regulă presupune că amplitudinea variază de la o valoare maximă (A_0) la o valoare minimă (A_{min}).

Practic algoritmul se bazează pe un grup de *indivizi identici* care au înscrise în memoria internă următoarele date: *locaţia* (x_i) (care este de fapt soluţia candidat propusă de individ), *viteza* (v_i) şi *frecvenţa* ($freq_i$) pe care şi le va actualiza la fiecare iteraţie a algoritmului conform ecuaţiilor următoare:

$$freq_i = freq_{min} + (freq_{max} - freq_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + \left(x_i^{t-1} - x_{best}\right) freq_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

Unde $\beta \in [0, 1]$ este un număr aleator extras dintr-o distribuţie uniformă. Cu cât individul se apropie de ţintă, îşi va ajusta *amplitudinea* (A_i)şi *vibraţiile sunetelor* (r_i) conform :

$$A_i^{t+1} = \alpha\, A_i^t \quad (4)$$

$$r_i^{t+1} = r_i^0\, [1 - e^{-\gamma t}] \quad (5)$$

Unde α ($0 < \alpha < 1$) şi γ ($\gamma > 0$) sunt variabile constante, alese de utilizator la rularea algoritmului.

Pentru a îmbunătăţi soluţia candidat, autorul mai adaugă o etapă de căutare locală bazată pe random walks şi astfel noua soluţie devine:

$$x_{new} = x_{old} + \delta\, A_t^* \quad (6)$$

Unde $\delta \in [-1, 1]$ este un număr aleator şi A_t^* este media amplitudinii tuturor indivizilor din grup la iteraţia t. Pseudocodul lui BA este ilustrat mai jos.

Algoritmul 1 - Bat Algorithm

```
1.  for i <=  1 to  SN do
2.  f(x_i)  :=  objective function
3.  x_i; v_i :=  initialize
4.  A_i; r_i :=  initialize
5.  freq_i  :=  initialize
6.  end for
```

```
7.  while t < MAX_IT do
8.  for i  <= 1 to SN do
9.  Generate new solution cf. eq. (1)(2)(3)
10.    if rand > r_i then
11.    Get best solution
12.    Local search for sol. cf. eq. (6)
13.    end if
14.    if rand < A_i AND f(x_i) < f(x_new) then
15.    Accept new solution
16.    r_i;A_i  := update cf. eq. (4)(5)
17.    end if
18.    end for
19.    t  := t + 1
20.    best :=   f(x_best)
21.    end while
```

Ca orice algoritm SI, BA are două componente : explorarea şi exploatarea. Toţi indivizii din grup vor zbura aleatoriu şi vor folosi random walks pentru a îşi diversifica locaţia. După ce şi-a stabilit o zonă a soluţiei candidat, individul o va exploata prin ajustarea amplitudinii şi a vibraţiilor sonore. Mai mult, această trecere de la explorare la exploatare se face în mod dinamic. Autorul susţine că acestă proprietate va permite crearea unor aplicaţii eficiente, în sensul în care cel care implementează aplicaţia va putea decide când trebuie să se facă această trecere în funcţie de cerinţele şi specificul aplicaţiei.

3.1.2 Scurtă discuţie despre SI

Algoritmii de SI sunt bazaţi pe agenţi simplii care imită comportamentul altor indivizi din grup sau au trasate sarcini precise, însă niciunul nu are privirea de ansamblu asupra problemei. Deşi nu există un lider desemnat al grupului, iar fiecare individ realizează sarcini simple, grupul are un comportament global inteligent datorită comunicării existente la nivelul populaţiei. În opinia noastră principalele concepte care stau la baza algoritmilor SI includ: agenţii (indivizii care fac parte din grup), explorarea (oferă diversitatea soluţiilor), exploatarea (realizată după explorarea spaţiului de căutare pentru alegerea unei zone de interes), comunicarea (sub forma colaborării dintre indivizii care fac parte din grup) şi funcţia fitness (valoarea sa va cuantifica calitatea soluţiilor candidate).
În tabelul următor vom realiza o comparare a algoritmior SI pe care i-am utilizat în modelele propuse, care are în vedere conceptele principale ale SI enumerate anterior.

Tabelul 1. Compararea algoritmilor SI

Denumire algoritm	**Problema**	**Tipologii de Indivizi**	**Explorarea**	**Exploatarea**	**Comunicarea**	**Fitness function**
PSO	Dinamica mişcării grupului (cârdurile de cocori)	identici	Parametrii aleatori (r1, r2)	Parametrii de învăţare (c1, c2)	Directă la nivel de grup	În funcţie de calitatea locaţiei
ACO	Găsirea celui mai scurt drum dintre muşuroi şi hrană	identici	Funcţia euristicăşi cantitatea de fero nom	Rata de evaporare a cantităţii de fe romon	Indirectă (stigmergy)	În funcţie de calitatea drumului găsit

Denumire algoritm	Problema	Tipologii de Indivizi	Explorarea	Exploatarea	Comunicarea	Fitness function
ABC	Găsirea celei mai bune surse de hrană	3 categorii: lucrători, privitori şi cercetaşi	Random walks (albinele cercetaşe)	Algoritm de tip greedy (albinele lucrătoare)	Directă prin dansul albinelor lucrătoare	În funcţie de calitatea sursei de hrană
BA	Găsirea ţintei (prada vânată)	identici	Random walks, ajustarea frecvenţei	Adaptarea intensităţii şi ratei de emisie sonoră	Directă la nivel de grup	În funcţie de locaţia ţintei
BAL	Găsirea ţintei (prada vânată)	identici	Lévy flights, ajustarea frecvenţei	Adaptarea intensităţii şi ratei de emisie sonoră	Directă la nivel de grup	În funcţie de locaţia ţintei
BA(E)	Găsirea ţintei (prada vânată)	identici	Random walks, distanţa Euclideană faţă de vecin, ajustarea frecvenţei	Adaptarea intensităţii şi ratei de emisie sonoră	Directă la nivel de grup	În funcţie de locaţia ţintei

3.2 ARTIFICIAL IMMUNE SYSTEMS

Artificial Immune System (AIS) *este o ramură a inteligenţei computaţionale care se referă la algoritmi inspiraţi biologic, ce transformă principiile, funcţiile şi modelele din biologie în modele matematice care pot fi implementate în vederea rezolvării unei game variate de probleme* [17]. Sistemul imunitar uman (NIS – Natural Immune System) protejează organismul de diversele ameninţări din exterior, sarcinile acestuia fiind divizate între cele două componente principale ale sale: nativă şi adaptivă, care împreună reuşesc să construiască o arhitectură defensivă multi-nivel. Acest sistem complex este caracterizat de un grad ridicat de paralelizare şi distributivitate adaptabilă obţinute prin intermediul grupurile de agenţi imunitari care utilizează reţeaua chimică de trimitere a mesajelor şi care iau decizii la nivel local sau chiar global.

După cum am precizat, sistemul imunitar este de tip multi-nivel, organizat astfel în: imunitatea nativă şi imunitatea adaptivă. Sarcina principală a NIS este de a proteja corpul uman împotriva contaminării cu agenţi precum viruşi sau patogeni. O clasă de celule non-proprii sunt cunoscute sub denumirea de antigen. Astfel, când un patogen este identificat în corp, atacul iniţial este realizat de componenta nativă a sistemului imunitar, iar componenta adaptivă este utilizată pentru acei patogeni care nu au fost distruşi de către cea nativă [18].

- ***Imunitatea nativă*** – este acea componentă a NIS care constă în celulele şi mecanismele utilizate în vederea prevenirii infecţiilor cauzate de alte organisme. Oferă un proces iniţial de imunizare prin componenta nativă şi nu garantează protecţia totală [18].
- ***Imunitatea adaptivă*** – permite corpului să identifice şi să contracareze orice microb pintr-un receptor antigen, chiar dacă nu a mai întâlnit acest tip de invadator. Are un nivel de complexitate mai ridicată decât componenta nativă. La bază, foloseşte cele două tipuri de globule albe (limfocitele : T-cell şi B-cell). Un proces aleator de generare a receptorilor antigenilor are loc, iar limfocitele sunt apoi responsabile cu identificarea şi ditrugerea antigenilor [18].

3.2.1 Transpunerea NIS în AIS

Pentru a reprezenta aceste componente biologice în domeniul calculatoarelor, Castro şi Timmis [17] au definit un framework multi-nivelar sub forma unui şablon, în vederea descrierii structurii paradigmelor AIS. În funcţie de domeniul de aplicare al AIS, avem de definit trei componente:

- ***Reprezentarea componentelor*** – pentru a reprezenta componentele funcţionale din sistem. Spre exemplu putem avea o mulţime S, cu două tipuri de componente: antigeni (Ag) şi anticorpi (Ab). Acestea sunt definite ca vectori d-dimensionali, tipul acestora depinzând de tipul formei spaţiului care poate include [17]:
  - ***Valori reale*** – atributele sunt reprezentate ca valori ale numerelor reale,
  - ***Hamming*** – atributele sunt elemente ale unui set finit dintr-un alfabet,
  - ***Simboluri*** – atributele sunt de tip nominal, reprezentate ca simboluri.
- ***Măsurarea afinităţii*** – pentru a cunatifica interacţiunile dintre componentele sistemului (în general dintre detector şi antigen). Calculează distanţa metrică dintre Ag şi Ab, care poate fi distanţa Euclidiană, Hamming etc. Pe baza valorii acestei funcţii se vor lua deciziile algorimului (eliminarea/generarea detectorilor etc).
- ***Algoritmul imunitar*** – pentru a determina dinamica sistemului şi a componentelor acestuia.

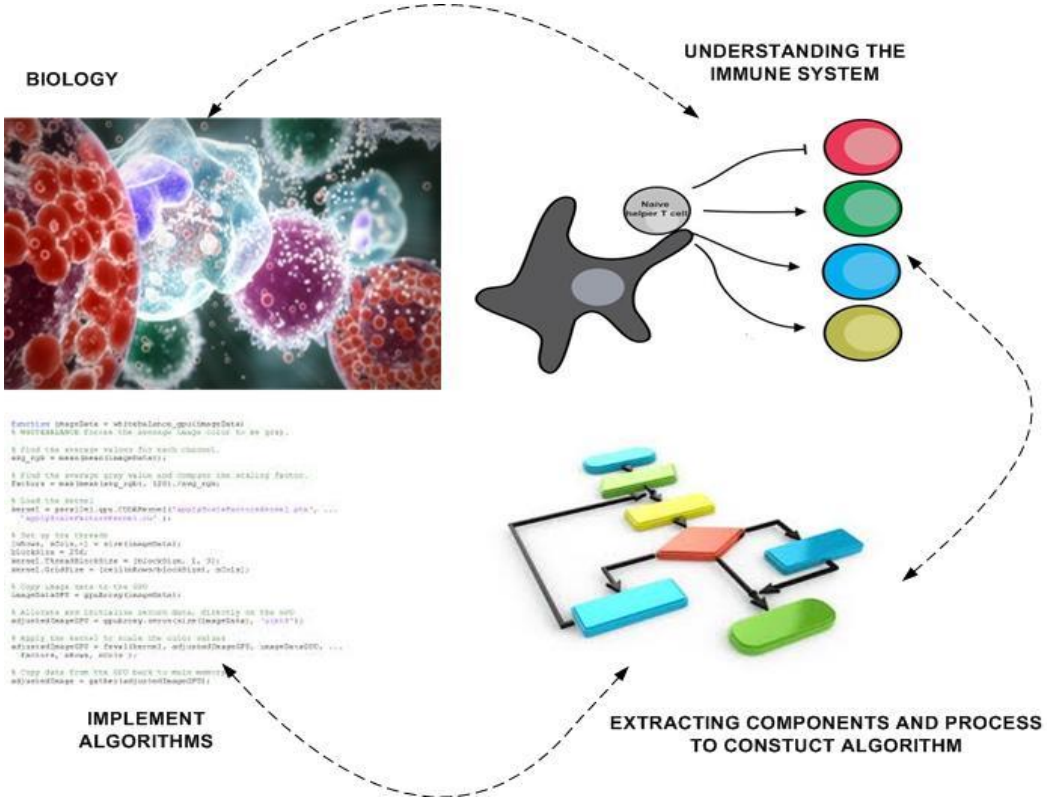

Figura1. Transpunerea NIS în AIS

3.2.2 Dendritic Cell Algorithm

Dendritic Cell Algorithm (*DCA)* este un algoritm AIS de generaţie recentă care are la bază principiile din teoria pericolului propusă de Matzinger, referitoare la rolul şi funcţiile celulelor dendritice (DC-cells). Autorii îmbină două concepte: *semnalele* (care evaluează

contextul) şi *antigenii* (care sunt datele de intrare ce trebuie evaluate), pentru a calcula o variabilă MCAV (mature context antigen value) ce are o valoare cuprinsăîn intervalul 0 şi 1 [12].În cadrul acestei teze ne vom referi la varianta deterministă a algoritmului DCA (dDCA).

Algoritmul are la bază o populaţie de celule DC, iar fiecare celulă se poate afla într-una din următoarele stări [12][10] :

- ***Imaturitate*** **–** este starea iniţială a oricărei DC-cell, în care colectează antigen şi procesează semnalele,
- ***Semi-maturitate*** – sunt celule imature care pot decide dacă un semnal este sigur,
- ***Maturitate*** – sunt celule mature care pot decide dacă semnalele locale reprezintă un pericol.

Toate celulele sunt iniţial imature apoi, pe parcursul colectării de antigeni ele vor deveni semi-mature sau mature, în funcţie de tipu lsemnalelor pe care le întâlnesc/colectează pe parcursul vieţii (lifespan). Sunt definite trei tipuri de semnale, respectiv [23]:

- ***PAMP(Pathogenetic Associated Molecular Patterns)***– are o influenţă considerabilă în vederea transformării unei celule DC din starea imatură într-o stare de maturitate; atributele care fac parte din această categorie de semnale indică, în general, pericolul.
- ***Danger (Pericol)*** – reflectă potenţiale anomalii, cu cât valoarea acestora creşte, proporţional va creşte şi status-ul de anormalitate al sistemului monitorizat.
- ***Safe (Sigur)*** **–** creşte în valoare o dată cu observarea unui comportament normal al sistemului, acesta este un indicator de încredere al sistemului; cu alte cuvinte, exprimă un comportament normal al sistemului.

Tabelul 2. Matricea de ponderi pentru DCA

	PAMP (S_1)	**Danger (S_2)**	**Safe (S_3)**
CSM (O_1)	W1	W1 / 2	W1 * 1.5
Semi-mature(O_2)	0	0	W2
Mature (O_3)	W3	W3 / 2	-W3 * 1.5

DCA are trei etape principale, respectiv:

a. ***Pre-procesare*** **–** este etapa în care analistul trebuie să stabilească metoda de reprezentare a componetelor de tip antigen (ex. ID-ul unui proces); tot în această etapă se aleg acele atribute relevante pentru detecţie, care vor fi incluse în matricea de semnale şi acestea vor fi grupate în cele trei categorii. Împărţirea atributelor pe categorii de semnale se poate face pe baza cunoştinţelor unui expert în domeniu sau prin procese automatizate (ex. Information Gain, Principal Component Analysis etc.).

b. ***Detecţia*** **–** entităţile DC sunt create şi încep să colecteze date (antigenii care trebuie analizaţi). Pentru fiecare antigen se va calcula un semnal de ieşire O_i :

$$O_i = \sum_{i=1}^{3} W_{i,j}\, S_i$$

unde S_i reprezintă semnalul de intrare (PAMP, Danger sau Safe) şi $W_{i,j}$ reprezintă ponderea care este prestabilită pentru fiecare categorie de semnal, conform tabelului 2. Fiecare entitate DC are un timp de viaţă, iar o dată cu procesarea antigenilor, durata sa va scădea cu o valoare egală cu cea a semnalului CSM. O dată ce CSM va creşte atât de mult încât va depăşi pragul de migrare prestabilit pentru entitatea DC, celula va trece într-una din cele două stări de maturitate (semi-maturitate sau maturitate). Din

acest moment se începe evaluarea componentelor semi-mature şi mature ale semnalului O. Componenta care are valoarea mai mare va determina polaritatea, K, pentru acel tip de antigen. O dată ajunsă la maturitate, entitatea DC îşi va trimite mai departe informaţiile analizelor de antigeni şi îşi va încheia ciclul de viaţă. Din acest moment grupul activ de entităţi DC este reînnoit cu o nouă entitate DC care îi va lua locul celei vechi care a murit.

c. ***Analiza*** **–** este etapa finală, în care algoritmul va decide clasa antigenului. Pentru aceasta se va calcula valoarea MCAV pentru fiecare antigen în parte. Valoarea variabilei MCAV este dată de rezultatul fracţiei dintre numărul de entităţi care au catalogat antigenul ca anomalie şi frecvenţa de colectare a acelui tip de antigen de către grupul de celule DC.

Pe parcursul ultimului deceniu, au existat mai multe variante ale algorimului DCA (pDCA-prototype, lbDCA-libtissue, rDCA etc.), însă cel mai frecvent utilizat este cel determinist (dDCA-deterministic DCA), obţinând rezultate promiţătoare şi fiind previzibil din punct de vedere al rezultatelor obţinute. În pseudocodul de mai jos este redat dDCA.

Algoritmul 2 – Dendritic Cell Algorithm [23]

```
input : antigen and signal instances
output: antigen types and anomaly metric MCAV
       set DC population size;
       initialise DCs;
while data do
       if antigen then
              agCounter++;
              cellIndex = agCounter % populationSize;
              DC of cellIndex assigned antigen;
              update DC's antigen pro_le;
       end
       if signal then
              calculate csm and k;
              foreach DC do
                     DC.lifespan -= csm;
                     DC.sumK += k;
                     if DC.lifespan <= 0 then
                            record antigens and DC.sumK;
                            reset DC;
                     end
              end
       end
end
foreach antigen type do
       calculate MCAV;
end
```

3.2.3 Scurtă discuţie despre AIS

Spre deosebire de algoritmii SI, AIS nu are un model arhetipal drept şablon, în schimb există patru mecanisme din teoria sistemelor imunitare din care autorii s-au inspirat: selectarea

negativă, principiul clonării selective, reţele imune idiotipice şi teoria pericolului [19]; aceştia sunt detaliaţi în tabelul următor.

Tabelul 3. Algorimi AIS

Componenta NIS	Mecanismul imunitar	Algoritmul AIS	Datele pentru antrenare	Aplicabilitatea în domeniul calculatoarelor
Imunitatea adaptivă	*Mecanismul de selectare negativă (limfocitele B-cell şi T-cell)*	Negative Selection Algorithms (Forrest et. al. 1994 [20])	Celulele proprii	Detecţia erorilor, detecţia anomaliilor etc.
	Principiul Clonării Selective (limfocitele B-cell)	Clonal Selection Algorithms (CLONALG 2002 - Castro şi Von Zuben [21])	Celulele non-proprii	Metode de optimizare sau căutare.
	Teoria reţelelor idiotipice (limfocitele B-cell)	Immune Network Algorithms (aiNet 2001 Castro et. al. [22])	Celulele non-proprii	Probleme de clasificare
Imunitatea nativă	*Teoria pericolului (DC-cell)*	DC algorithm Greensmith et. al. 2007 [23]	Celulele proprii şi non-proprii	Detecţia anomaliilor
	Teoria pericolului (DC-cell şi T-cell)	TLR algorithm Twycross et. al. 2007 [24]	Celulele proprii	Detecţia anomaliilor

3.3 SWARM INTELLIGENCE ŞI ARTIFICIAL IMMUNE SYSTEMS. SCURTĂ DISCUŢIE.

În opinia noastră SI şi AIS prezintă similarităţi dar în acealaşi timp oferă perpective complementare. Ambele categorii de algritmi sunt inspiraţi din natură (grupuri de animale sau sistemul imunitar format din grupuri de celule) şi au la bază o populaţie de agenţi care au de îndeplinit diverse sarcini pentru buna-funcţionare a sistemului. Din punct de vedere biolgic SI are ca obiectiv supravieţuirea grupului, pe când AIS are ca scop supravieţuirea gazdei. Pentru a îşi îndeplini scopul SI va avea în vedere soluţiile candidate cele mai bune, în funcţie de valoare funcţiei fitness. În schimb, AIS va menţine o populaţie de celule active în care vor fi incluse doar cele performante, restul fiind eliminate din grupul curent de lucru. Sistemul imunitar poate fi văzut ca un oragnism de grupuri (swarm) de celule care vor interacţiona pentru a îşi proteja gazda.

În primul rând compararea celor două clase de algoritmi va avea în vedere definiţiile acestora enunţate de Dorigo şi Biratti (SI) şi Castro şi Timmis (AIS). Din acestea putem deduce că noţiunile care stau la baza SI sunt bine conturate, pe când definiţia AIS este destul de generală şi aruncă mingea în terenul teoriei imonologiei, lasând în acelaşi timp la latiudinea autorului algoritmului selectarea principiilor şi a funcţiilor. În opinia noastră, principalul motiv pentru lipsa unei definiţii complete se datorează complexităţii domeniului imunologic care încă mai are elemente şi principii ascunse chiar şi pentru specialiţii din domeniul biologiei.

În continuare vom enumera câteva elemente şi principii care se regăsesc în SI şi AIS, detalierea acestora fiind realizată în cadrul tezei: agenţii, explorarea, exploatarea, comunicarea, funcţia fitness, adaptabilitatea, învăţarea, robusteţea, paralelismul şi auto-organiarea. Elementul cheie prezent în ambele clase de algoritmi este colaborarea, care în

opinia noastră este implementat diferit. SI apelează la soluţia optimă la nivel individual sau global, pe când AIS apelează la o populaţie de agenţi optimi care sunt supuşi unui proces de selecţie.

4 PROPUNERILE DE ÎMBUNĂTĂŢIRE PENTRU ALGORITMI

După introducerea algoritmilor CI, în acest capitol sunt prezentate analiza, problemele sesizate şi propunerile pentru îmbunătăţire a doi algoritmi: the Bat Algorithm şi Dendritic Cell Algorithm. Pentru detaliile privind analiza algoritmilor, vom face referire la teza autorului.

4.1 DENDRITIC CELL ALGORITHM

4.1.1 Algoritmul DCA din Perspectiva Şablonului AIS

Fiind un algoritm de tip AIS, DCA se încadrează în şablonul stabilit de Castro şi Timmis [17], după cum urmează:

- ***Reprezentarea componentelor*** – în cazul algoritmului DCA, vom sabili două componente: Antigenii (Ag) şi Semnalele (S). Astfel, anticorpii din şablon sunt înlocuiţi de semnale care au imprimate matricile de anomalii sau normalitate ale antigenilor. Cele două componente sunt reprezentate sub forma unor vectori de dimensiune d, unde cardinalitatea acestora este definită ca fiind : $\langle S \rangle = \langle S_{PAMP} \rangle + \langle S_{Danger} \rangle + \langle S_{Safe} \rangle = \langle Ag \rangle = d$. Mecanismul de reprezentare al spaţiului în cazul DCA este distinct, întrucât avem de a face cu trei categorii de semnale ale căror valoare poate activa sau inhiba zona de pericol definită în preajma antigenilor. De fapt, ne interesează semnalul de ieşire, sau mai precis polaritatea (K) care va marca zona de detecţie. Această polaritate trebuie să coincidă în cazul mai multor entităţi DC, iar populaţia de celule DC acţionează asemenea unui forum decizional în care fiecare are o pondere egală în luarea deciziei finale referitoare la clasificarea antigenului. Pentru a înţelege mai uşor ne vom referi la reprezentarea geometrică a algoritmului DCA din figura 2.

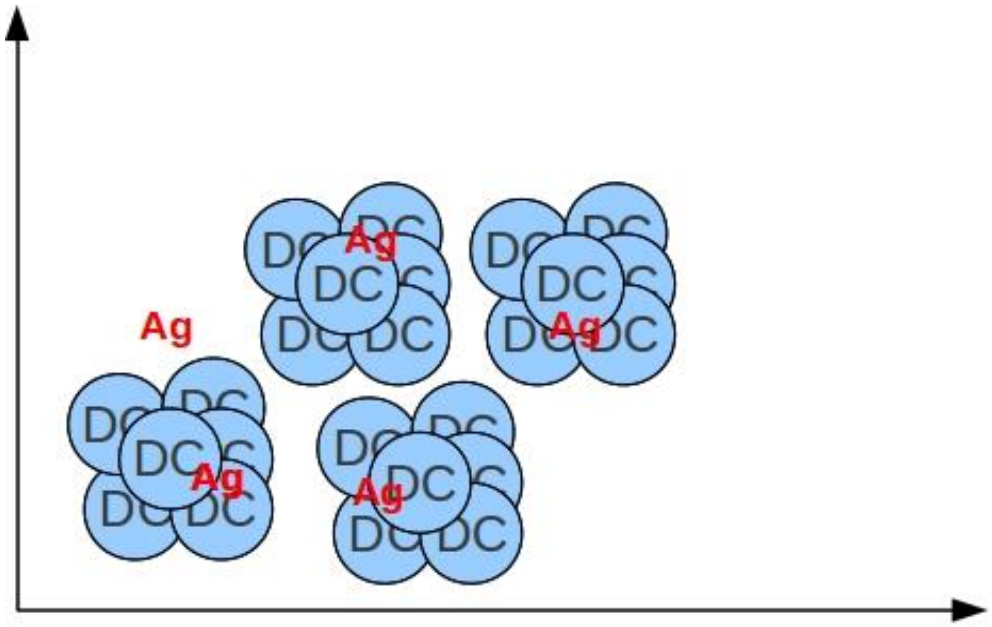

Figura 2 Reprezentarea componentelor în cazul algoritmului DCA

Zona de determinarea a antigenilor clasificaţi ca anomalii este delimitată de intersecţia dintre entităţile DC care au o polaritate spre maturitate (semnalul lor de maturitate este mai mare decât cel de imaturitate).

- ***Măsurarea afinităţii*** **–** este intersecţia dintre componente. Acest mecanism este înlocuit cu cel de vot majoritar realizat de entităţile DCA în stadiul de analiză. Mai mult clasificarea unui tip de antigen este rezultatul conlucrării întregului grup de celule DC. Dacă pragul critic prestabilit pentru valoarea MCAV este depăşit, atunci acel antigen este detectat ca o anomalie.
- ***Algoritmul imunitar*** **–** se referă la necesitatea de a înţelege conceptele şi principiile biologice ale sistemului imunitar înainte de a îi extrage conceptele şi mecanismele pentru a le trasnpune în modele matematice. DCA preia din caracteristicile şi funcţiile celulelor DC din cadrul sistemului imunitar, entităţi care participă în mod activ la detectarea anomaliilor având la bază procesarea semnalelor şi a antigenilor.

4.1.2 Enunţarea Problemei şi Propunerea Soluţiei

DCA, propus în 2008 de către Greensmith [23], este unul din algoritmii de ultimă generaţie din clasa AIS. Algoritmul se bazează pe o populaţie de celule identice care colaborează în vederea obţinerii unei analize pertinente referitoare la antigenii colectaţi. Mecanismul de colaborare este implementat sub forma unui vot majoritar al celulelor DC, care fac parte din grupul de decizie. Fiecare entitate DC are propria perspectivă asupra clasificării tipului de antigen (întrucât fiecare celulă DC are propriul ciclu de viaţă diferit) şi are o putere egală în votul exprimat (la final fiecare analiză a unui tip de antigen are o pondere egală). Algoritmul calculează valoarea MCAV şi o va ataşa antigenului pentru a îl clasifica drept anomalie sau normal, în funcţie de pragul de decizie prestabilit.

O posibilă problemă ar fi stabilirea acestui prag de decizie care ar putea influenţa performanţele de clasificare ale algoritmului DCA. Pentru a adresa această problemă voi face referire la reprezentarea geometrică din subcapitolul anterior (figura 2), pe care o vom transforma prin reprezentarea interseţiei dintre celule (zona de anomalii delimitată). Pentru simplitate vom alege ilustrarea zonei de pericol pentru un tip de antigen, în cazul mai multor antigeni se va face multiplicarea acestora în spaţiul de reprezentare al componentelor. Din figura 3 deducem că valoarea MCAV redă aria spaţiului de decizie, care se împarte în două zone: de normalitate şi respectiv de anomalii.

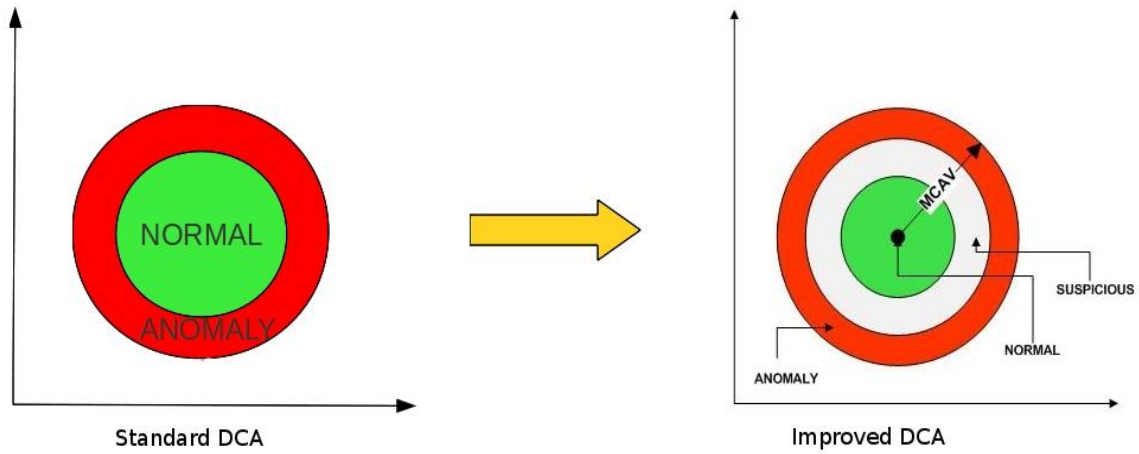

Figura 3 Reprezentarea componentelor algoritmului DCA şi pespectiva de îmbunătăţire

Penru a îmbunătăţi algoritmul ne vom axa pe componenta de decizie, respectiv pragul ce stabileşte frontiera dintre anomalii şi zona de normal. Soluţia pe care o propunem este de a crea o zonă de suspiciuni prin adăugarea unui prag suplimentar (vezi figura 3) pentru a delimita acele înregistrări pentru care atribuirea clasei este uşor incertă. Paradigma de vot majoritar este destul de comună şi este utilizată de mulţi algoritmi din domeniul calculatoarelor însă, ceea ce vrem să evidenţiem în acest caz este faptul că unele dezbateri pot fi la limita dintre normalitate şi anomalie. Prin urmare vom crea acestă zonă de confort care va face subiectul unei analize ulterioare mai detaliată în vederea luării deciziei finale privind tipul antigenului. Adăugarea acestui nou prag nu are influenţe majore asupra algoritmului întrucât doar adaugă o operaţie suplimentară de comparare. Pentru analiza setului de date suspicioase putem alege opinia unui expert în domeniu sau un alt algoritm de clasificare.

4.2 THE BAT ALGORITHM

Bat Algorithm (BA) este un algoritm SI recent care imită comportamentul liliecilor ce utilizează ecolocaţia pentru a identifica prada sau obstacolele din drumul său. Structura algoritmului se pliază pe şablonul agoritmilor SI incluzând agenţi identici, explorarea, exploatarea şi funcţia fitness pentru evaluarea soluţiilor.BA poate fi considerat o evoluţie a lui PSO sub o formă mai generalizată şi care oferă un grad mai ridicat de flexibilitate datorită saltului dinamic dintre explorare şi exploatare. Algoritmul poate fi formalizat sub forma unei probleme de optimizare (a se vedea teză pentru mai multe detalii).

4.2.1 Enunţarea Problemei şi Propunerea Soluţiei

Toţi algoritmii SI sunt bazaţi pe ***explorarea (diversificare) şi exploatare (intensificare)***, iar echilibrul acestor două componente joacă un rol important în găsirea soluţiei. Prima se referă la diversificarea soluţiei prin realizarea unor operaţiuni de căutare la nivel global şi lărgirea ariei de căutare cât mai departe de soluţia curentă astfel încât să permită acoperirea unui spaţiu cât mai mare din aria posibilă. Prin urmare, această componentă oferă libertatea de explorare a cât mai multor soluţii candidat. Avantajul evident este faptul că nu va permite blocarea căutării în jurul soluţiilor optime locale. Cu toate acestea, un grad ridicat de explorare poate determina căutarea oarbă a unor soluţii şi un nivel scăzut de convergenţă. Exploatarea se referă la realizarea unor căutari la nivel local. Această componentă exploatează o anumită regiune de soluţii având în vedere informaţiile locale. Acest proces este opus diversficării şi va duce la convergenţa algoritmului.

Din testele pe care le-am realizat cu algoritmul BA, am sesizat că acesta are un început promiţător, însă pierde desul de repede din diversitatea soluţiilor. Prin urmare, cu cât numărul de iteraţii creşte, algoritmul pierde din gradul de explorare întrucât condiţia rand (0, 1) > r este dificil de îndeplinit întrucât valoarea lui r creşte exponenţial. Mai mult, chiar dacă această condiţie este satisfăcută noua soluţie este generată în jurul candidatului cel mai bun găsit la nivel global şi cu o abatere mică datorită amplitudinii scăzute. Prin urmare, algoritmul s-ar putea bloca în minimele locale. Pentru a rezolva această problemă am prpus două îmbunătăţiri prezentate în subcapitolele următoare.

Zborurile Lévy

Îmbunătăţirea algoritmului se referă la componenta algoritmului de *căutare locală* prin aplicarea *zborurile Lévy* [4][6]. Cu alte cuvinte, ecuaţia (7) va deveni:

$$x_{new} = x_{old} + t^{-\delta} A_t^* \qquad (9)$$

Unde $t^{-\delta}$ este distibuţia Lévy şi $1 < \delta \leq 3$ este o constantă. Noua versiune a algoritmului poartă denumirea BAL.

Zborurile Lévy sunt procese Markov, iar cercetările recente au arătat că aceste distribuţii pot descrie "metodele" de căutare a hranei pentru unele specii de animale (albatroşii sau alte animale prădătoare) [20]. De fapt, aceste distribuţii sunt tot random walks, dar al căror pas este descris de o distribuţie Lévy. Traiectoria acestora descriu puncte apropiate, pentru o căutare în vecinătate, care apoi se transformăîn întoarceri bruște de 90^0.

Principalul motiv pentru înlocuirea distribuţiei este de a îmbunătăţi componenta aleatorie a algoritmului, astfel încât acesta să nu rămână blocat în minimele locale. Astfel prin schimbarea distribuţiei, am avea posibilitatea de a "acoperi" un spaţiu cât mai mare pentru primii paşi ai iteraţiilor în vedereaalegerii spaţiului de exploatat. În acest mod, vom păstra logica algoritmului, modificând doar elementul de explorare locală.

Distanţa Euclidiană

În această a doua abordare, vom îmbunătăţi componenta de *căutare locală*prin includerea unui termen adiţional în ecuaţia de generare a noii soluţii. Noua soluţie va fi calculată conform următoarei formule:

$$x_{new} = x_{old} + u\sqrt{\sum_{i=1}^{d}\left(x_{old,i} - x_{j,i}\right)^2} + \delta\, A_t^* \qquad (10)$$

Unde $0 < u \leq 1$ este un număr aleatoriu şi $x_j$este un poziţia vecinului j care are o soluţie mai bună decât cea a individului curent pentru care se calculază noua soluţie. Dacă un vecin cu o soluţie mai bună nu este identificat într-un număr limitat de încercări, atunci se va păstra ecuaţia algoritmului original. Prin urmare liniile 10-13 din algoritmul 1 devin:

Algoritmul 3 - BA(E)

```
1.  if rand > ri then
2.  trial = SN * 2; j= rand (1, SN)
3.  While trial <> 0 AND f(xj) <= f(xi) do
4.    j= rand(1, SN); trial = trial – 1;
5.  end while
6.  if f(xj) > f(xi) then
7.     x_new = Improve_candidate_sol(x_old, xj) cf. (10)
8.  else
9.  x_new = Improve_candidate_sol(x_old) cf. (7)
10.    end if
11.    Accept new solution
12.    end if
```

În noua versiune a algoritmului, denumită BA(E), se va asigura convergenţa operaţiunii de căutare într-o zonă unde soluţiile au valori ale funcţiei fitness mai bune. Din punct de vedere al complexităţii algoritmului, păstrăm numărul de evaluări ale funcţiei fitness însă creştem puţin timpul de execuţie datorită operaţiunii de căutare a unui vecin cu o soluţie de o calitate superioară. Mai mult, forma originală a algoritmului este păstrată cu singura diferenţă că vom adăuga un termen suplimentar în ecuaţia de căutare locală a noii soluţii.

5 SISTEMELE DE DETECŢIE A INTRUZIUNILOR

5.1 NOŢIUNI DE BAZĂ

Sistemele de detecţie a intruziunilor (IDS) *monitorizează evenimentele din sistem şi decid dacă acestea sunt posibile intruziuni sau reprezintă operaţiuni legitime.*În general, IDS sunt clasificate în două categorii, în funcţie de metoda de detecţie utilizată, astfel:**IDS bazat pe semnături**şi **IDS bazat pe anomalii**.

Prima categorie identifică intruziunile pe baza corespondenţei datelor monitorizate cu descrieri predefinite ale unui comportament intruziv (precum semnături ale codurilor maliţioase). Această metodă permite identificarea eficientă a intruziunilor, generând în acelaşi timp un număr scăzut de alarme false. Din acest motiv IDS bazate pe semnături sunt cele mai răspândite. Pe de altă parte, intruziunile pot fi polimorfe sau pot evolua continuu (criptarea viruşilor informatici şi recriptarea acestora după un timp; bombele logice pot fi identificate abia după ce se lansează atacul etc.). Detecţia pe bază de semnături nu va putea identifica aceste tipuri de intruziuni necunoscute. O metodă de a rezolva această problemă este de a actualiza periodic baza de date cu semnături, manual (consumatoare de timp) sau automat (cu ajutorul unui algoritm de inteligenţă artificială de tip supervizat). Din păcate, elaborarea acestor seturi de date este destul de costisitoare, întrucât necesită analizarea intruziunilor respective şi identificarea unor patternuri. O altă soluţie pentru această problemă este oferită de Denning, care propune detecţia anomaliilor bazate pe deviaţii de la profilul normal.

Detecţia anomaliilor este ortogonală detecţiei bazate pe semnături. Această metodă porneşte de la ideea că anomaliile din sisteme sunt rare şi diferite de profilul normal. Aşadar, va construi un profil normal al sistemului şi va considera deviaţiile de la acesta ca fiind intruziuni. Această metodă permite identificarea atacurilor noi şi necesită doar date normale pentru a construi profilul. Cu toate acestea, un mare dezavantaj constă indentificarea limitei dintre profilul normal şi cel anormal, datorită lipsei acestuia din setul de date pentru stagiul de învăţare (putem simula o serie de atacuri, dar nu putem şti toate tipurile de atacuri).

În funcţie de tipul de procesare utilizat de modelul "comportamental" al sistemului putem grupa tehnicile de detecţie bazate pe anomalii în trei mari categorii [26]:

- **Statistice** – comportamentul sistemului este analizat folosind metode sau modele statistice. Pentru analiză se folosesc datele de audit colectate şi în funcţie de metrica stabilită (numărul unor evenimente, intervale de timp, supraîncărcarea unor resurse etc.) modelul statistic poate determina dacă datele observate sunt anomalii. Ca modele statistice se pot folosi : modelul operaţional (compară datele observate cu un prag – thershold - prestabilit), procesele Markov (bazate pe probabilitatea tranziţiei), time series (ia în calcul ordinea şi timpul dintre colectarea observaţiilor) etc..Principalul dezavantaj este faptul că atacatorul poate simula anumiţi parametrii, astfel încât datele să "pară" normale (ex. Atacul SlowLoris de tip DDoS prin simularea unui trafic de date normal în reţea, dar cu cereri Http incomplete care pot păstra serverul ocupat, facându-lastfel indisponibil).
- **Cognitive (de cunoaştere)**–încearcă să contureze comportamentul normal al sistemului. Printre metodele cel mai des întâlnite se numără : maşina automată cu stări finite (Finite State Machine – care modelează sistemul sub forma unor stări, tranziţii şi acţiuni), modelul specialist (foloseşte cunoştinţele specialiştilor umani pentru a rezolva probleme de inceritudine) etc.. Avantajele metodei sunt robusteţea şi flexibilitatea, însă pentru crearea unui astfel de model avem nevoie de expertiza unor persoane avizate, iar procesul de implementare este unul anevoios.
- **Bazate pe cunoaştere (AI)**–folosesc algoritmi de inteligenţă artificială pentru a detecta anomaliile (arborii de decizie, reţelele neuronale artificiale, maşinile cu suport vectorial

etc.). Modelele de IDS bazate pe AI sunt flexibile şi au un grad ridicat de adaptabilitate. Dacă metodele clasice, reprezentate de clasificatorii tradiţionali, precum SVM, reţelele neuronale sau arborii decizionali, sunt mai degrabă orientaţi spre clasificarea şi nu spre detecţia anomaliilor. Prin urmare, aceste modelesunt destul de „costisitoare”, dacă avem în vedere necesitatea de antrenare sau complexitatea modelelor matematice care stau la baza acestora, rezultând implementarea unor sisteme de monitorizareconsumatoare de resurse. În general, performanţele oferite de algoritmii tradiţionali sunt acceptabile, dacă avem în vedere FAR şi ADR, însă performanţele acestora scad în condiţiile unui mediu variabil şi distribuit. Metodele inspirate biologic sunt mai simple şi prezintă anumite proprietăţi care le recomandă pentru detecţia anomaliilor.

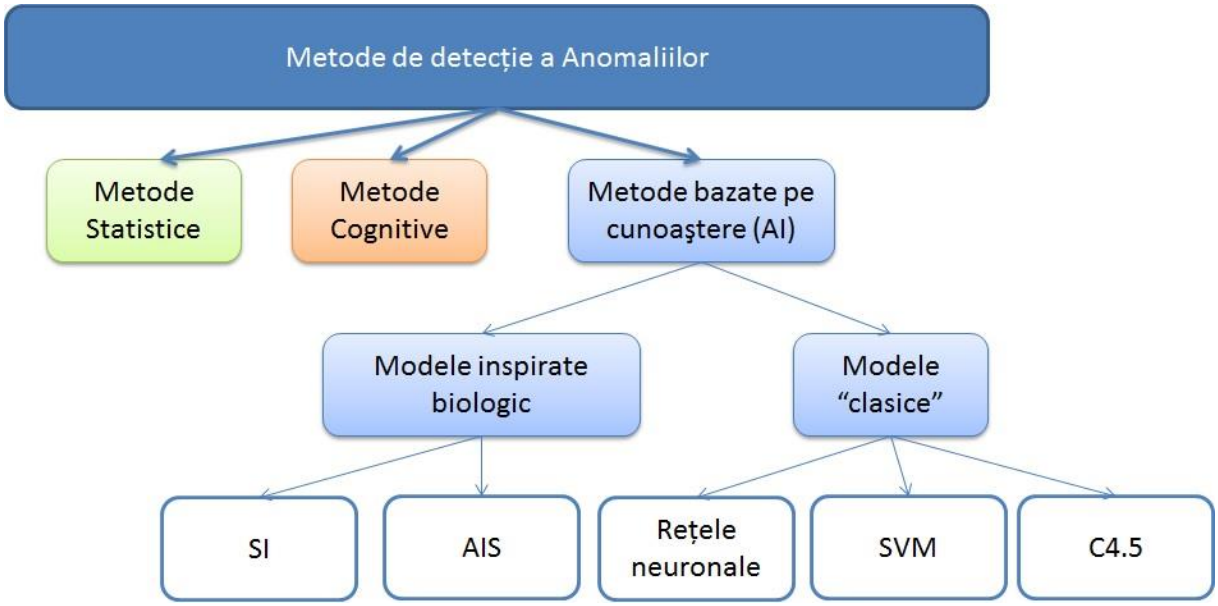

Figura 4. Clasificarea IDS bazate pe anomalii

5.2 FUNCŢIONALITĂŢILE ŞI EVALUAREA PERFORMANŢELOR IDS

Principalele patru funcţionalităţi ale IDS sunt :

- ***Colectarea datelor*** – se referă la colectarea datelor din sursele diverse şi omogenizarea acestora pentru procesul de interpretare.
- ***Reducerea numărului de atribute –*** unele atribute din setul de date pot fi redundante sau pot afecta negativ performanţele etapei de detecţie. Aşadar, pentru a micşora setul de atribute, putem utiliza diverse metode de reducere a atributelor, păstrând astfel doar setul relevant.
- ***Detecţia –*** datele colectate şi omogenizate trebuie analizate, rezultând informaţii cu privire la posibilele anomalii din sistem.
- ***Răspunsul –*** materializeazărezultatul etapei de detecţieprin generarea unor alarme sau chiar răspunsuri active (ex. blocarea sesiunii suspicioase sau eliminarea pachetelor).

În cadrul activităţii de cercetare am avut în vedere doar două funcţionalităţi ale IDS: reducerea atributelor şi respectiv etapa de detecţie. Funcţionalităţile IDS sunt ilustrate în figura 5.

Implementarea unor sisteme de detecţie a intruziunilor eficiente este dificilăîntrucât acestea trebuie să ofere un nivel scăzut de **alarme false(FAR)** şi în acelaşi timp să

poatădetecta atacurile din sisteme cu o **precizie** cât mai mare (ADR). Totodată, este de preferat ca IDS să ofere **răspunsuri în timp real.**

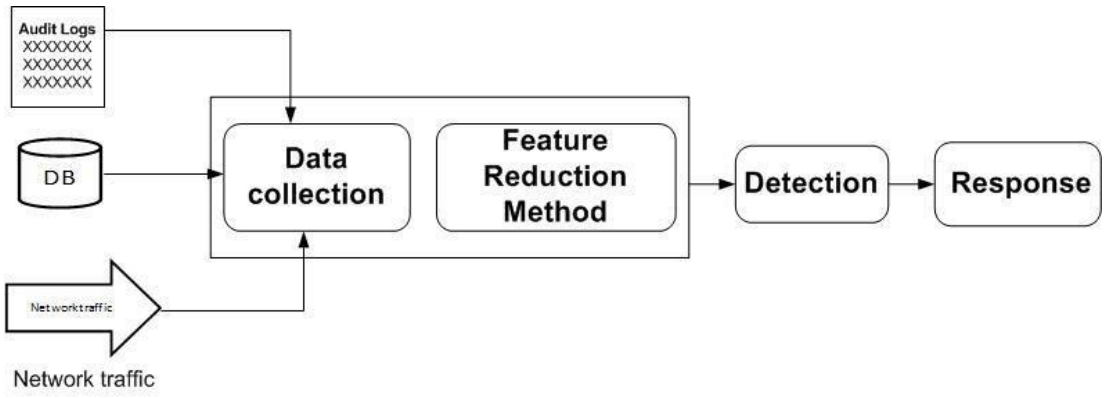

Figura 5. Funcţionalităţile IDS

În contextul actual, în care sutele de aplicaţii şi echipamente eterogene existente în infrastructură generează un volum considerabil de loguri, sistemele de monitorizare sunt suprasolicitate, iar uneori numărul mare de alarme false este dificil de interpretat de către operatorul uman. De cele mai multe ori, logurile includ date redundante sau irelevante pentru detecţia intruziunilor, iar reducerea acestora a devenit aproape o necesitate.

Pentru a ***evalua îmbunătăţirea performanţei IDS*** avem nevoie de metrici şi cum modelelepropuse sunt bazate pe algoritmi de inteligenţă computaţională, ne vom referi la : matricea de erori şi curba ROC.

a) **Matricea de erori (Confusion Matrix)** conţine informaţii referitoare la clasa prezisă de clasificator şi respectiv clasa reală. În general, evaluarea sistemelor este sintetizată sub forma unei matrici de erori care va permite determinarea cu uşurinţă a parametrilor de performanţă.

Tabelul 4. Matricea de erori

		Clasa Prezisă	
		Clasa negativă (Normal)	Clasa pozitivă (Atac sau Intruziune)
Clasa Reală	Clasa Negativă (Normal)	TN	FP
	Clasa Pozitivă (Atac)	FN	TP

Pe baza acestei matrici putem calcula următorii parametrii de performanţă:

- Specificitatea (TNR) = $\frac{TN}{TN+FP}$
- Senzitivitatea sau Rata de Detecţie (DR) (TPR) = $\frac{TP}{TP+FN}$
- Rata Alarmelor False (FAR) = 1-senzitivitatea = $\frac{FP}{TN+FP}$
- Acurateţea = $\frac{TN+TP}{TN+TP+FN+FP}$
- Precizia = $\frac{TP}{TP+FP}$

b) **Curbele ROC** – provin din tehnicile de detecţie a semnalelor radar dezvoltate pe timpul celui de al doilea război mondial. Acestea sunt folosite pentru a indica rata de detecţie în funcţie de numărul de alarme false pentru un canal de comunicaţii

zgomotos. În domeniul detecţiei intruziunilor, metoda este utilizată pentru evaluarea diferitelor scheme de învăţare. Un IDS optim trebuie să maximizeze DR şi să minimizeze FAR. Curba ROC descrie grafic DR pe axa Y şi FAR pe axa X pentru parametrii diferiţi.

Dintre acestea, cel mai des sunt utilizate DR, care indică dacă IDS poate detecta intruziunile, FAR care măsoară numărul de alarme false generate de IDS ca urmare a clasificării logurilor normale ca intruziunişi respectiv acurateţea care arată dacă IDS este capabil să genereze alarme atunci când trebuie.

Avantajele şi stadiul actual privind utilizarea algorimilor SI şi AIS în arhitecturile IDS se regăsescîn teză.În continuare ne vom axa asupra prezentării modelele IDS propuse.

6 SISTEME DE DETECŢIE A INTRUZIUNILOR PROPUSE BAZATE PE ALGORITMI DE INTELIGENŢĂ COMPUTAŢIONALĂ

6.1 SETUL DE DATE

Pentru testarea modelelor IDS propuse am folosit setul de date NSL-KDD [27]. Acest set este o versiune îmbunătăţită a lui KDD-Cup, în sensul că nu conţine date redundante sau duplicate iar complexitatea acestora este mai scăzută. Fiecare înregistrare din setul de date conţine 41 de atribute şi este identificată ca activitate normală ori atac. Aceste atribute pot fi grupate în trei categorii principale: informaţii despre conexiune (10 atribute), informaţii despre timp (9 atribute), informaţii referitoare la conţinut (13 atribute) şi atribute de bază (9 atribute). Totodată, atacurile incluse în setul de date aparţin uneia dintre următoarele categorii:

- ***DoS (Denial of Service)***– atacatorul încearcă săîmpiedice accesul la date al utilizatorilor autorizaţi.
- ***R2L (Remote to Local)*** – atacatorul obţine accesul de la distanţă la resursele locale ale sistemului.
- ***U2R (User to Root)*** - atacatorul obţine drepturi de administrator pentru resursele sistemului.
- ***Probing*** – atacatorul încearcă să obţină accesul la resursele sistemului prin utilizarea unor puncte vulnerabile cunoscute (port-sweep, IP-weep etc.).

Mai multe detalii privind atributele setului de date se pot regăsi în anexa tezei de doctorat.

6.2 MODELE IDS BAZATE PE SI

6.2.1 Componentele IDS

Diagrama modelului IDS este redatăîn figura 6, iar opţinile privind algoritmii aleşi pentru completarea blocurilor din diagramă sunt enumeraţi în tabelul5.
Sumarul din tabelul 5împarte modelele IDS propuse în trei categorii principale:

- optimizarea procesului de alegere al parametrilor pentru maşinile cu support vectorial (SVM),
- metode de selectare a stributelor, utilizând versiunile binare ale algoritmilor SI îmbunătăţiţi (BBAL - Binary Bat Algorithm cu zborurile Lévy, BBA(E) - Binary Bat Algorithm cu distanţa Euclidiană),
- varianta care combină metodele de selectare a atributelor cu procesul de alegere a parametrilor pentru clasificatorul SVM.

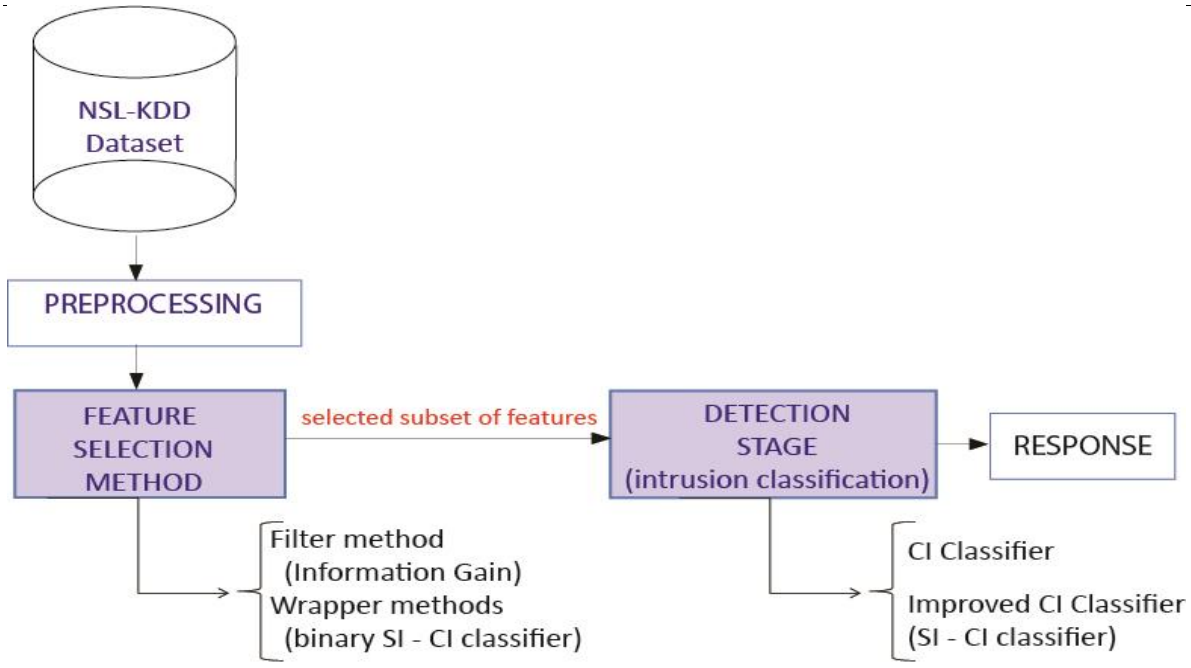

Figura 6. Diagrama modelelor IDS propuse

Tabelul 5. Sumarul modelelor IDS propuse care se bazează pe SI

Model IDS	Metodă pentru selectarea atributelor	Etapa de detecţie
Setup 1 **Îmbunătăţirea procesului de selecţie a parametrilor pentru SVM**	**Metode de tip filter** **Information Gain**	**SI*-SVM** **SI*є PSO(pt. comparare)** **SI*є PSO** **SI*є BA [7]** **SI*є ABC[2]** **SI*є BAL[11]**
Setup 2 **Metode pentru selectarea atributelor folosind versiunea binară a SI**	**Metode de tip wrapper** **SI*-CI** **SI*є BPSO(pt. comparare)** **SI*є BBA (pt. comparare)** **SI*є BBA(E) − CI є {SVM, C4. 5}[9]** **SI*є BBAL − CI є {SVM, NB}[6]**	**Clasificator CI** **CI є {SVM, C4. 5, NB}** **CI є {SVM, C4. 5, NB}** **CI є {SVM, C4. 5}** **CI є {SVM, NB}**
Setup 3 **Combinate**	**Metode de tip wrapper** **SI*-SVM** **SI*є BBA(E)** **SI*є BBA(E)** **SI*є BBA(E)** **SI*є BBAL** **SI*є BBAL**	**SI*-SVM** **PSO-SVM (pt. comparare)** **BA-SVM (pt. comparare)** **BA(E)-SVM [8]** **PSO-SVM (pt. comparare)** **BA-SVM [4]**

Toate modelele IDS au fost implementate şi testate pe o platformă cu procesor Intel Core 2 Duo, 1.80GHz, cu memorie RAM de 2GB şi sistem de operare Ubuntu 10.04.4.. Algorimii de Swarm Intelligence, în variantele lor binare sau standard, au fost implemetaţi în

Java.PentruInformation Gain şi algoritmii de clasificare convenţionali(SVM, C4.5, NB) am utilizat biblioteca Weka versiunea 3.6.10 [32].

6.2.2 Optimizarea parametrilor SVM

6.2.2.1 Parametrii SVM

SVM (Suppor Vector Machines, Maşinile cu suport vectorial) sunt clasificatori binari liniari care îşi construiesc un hiperplan pentru a separa cele două clase, în cazul nostru anomaliile de traficul de reţea normal. Punctele care aparţin hiperplanului se numesc vectori suport iar distranţa dintre aceştia poartă denumirea de margine. Datele nu sunt întotdeauna perfect liniar separabile şi de aceea putem defini o margine soft, adică putem accepta unele erori în etapa de antrenare a modeluluimaterializate subforma unor înregistrări clasificate incorect. Mai mult, putem transforma setul de date într-un spaţiu multidimensional de atribute în care datele să fie liniar separabile. Pentru aceasta, putem folosi **funcţii kernel (funcţii nucleu)**, cum ar fi funcţia polinomială, funcţia radial-basis (RBF) sau funcţia sigmoidă, care sunt printre cele mai frecvent utilizate. Dintre acestea vom alege RBF întrucât are mai puţini parametrii ce trebuie selectaţi şi oferă tranformări nonlineare bune.

Cu alte cuvinte algoritmul SI va căuta doi parametrii [4][2] :

- C – este parametrul de regularizare care controlează flexibilitatea marginii. O valoare mare pentru C va determina crearea unui model cu o acurateţe ridicatăşi cu o margine îngustă. Pe de altă parte, o valoare mică poate genera multe erori şi va crea o margine largă. Valorile trebuie alese astfel încât să nu se creeze un model cu o acurateţe care să influenţeze negativ proprietăţile de generalizare, sau invers.
- σ - este constanta funcţiei nucleu, care va influenţa proprietăţile de overfitting şi underfitting ale modelului. Acest parametru arată corelaţia dintre vectorii suport care construiesc hiperplanul.

6.2.2.2 Modele Propuse

În figura de mai jos se regăseşte diagrama modelului IDS, conform şablonului din figura 6, dar în plus blocurile modelului sunt completate. Pentru îmbunătăţirea procesului de alegere a parametrilor SVM am ales $SI \in \{PSO, ABC, BA, BAL\}$.

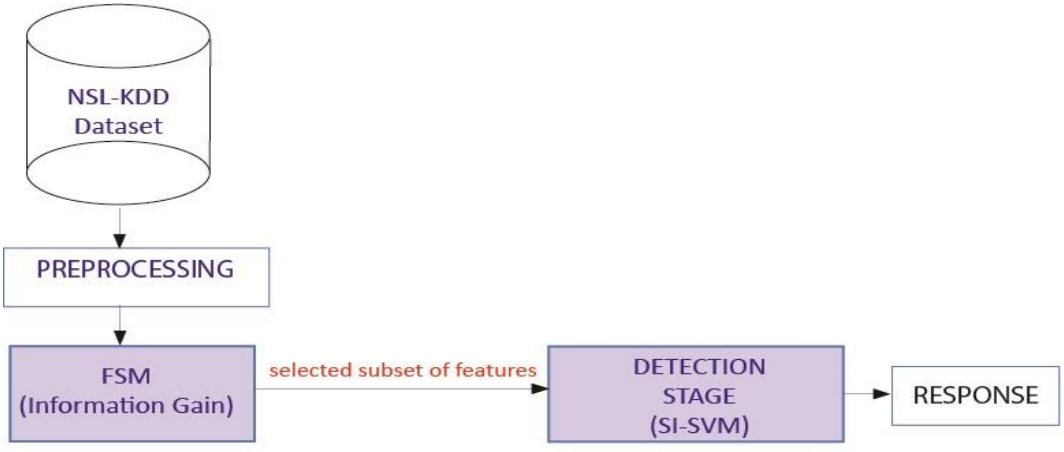

Figura 7. Diagrama modelelor IDS propuse pentru problema selectării parametrilor SVM

Pentru testarea modelului am selectat în mod aleator 9566 de înregistrări şi le-am împărţit în două fişiere unul pentru antrenareşi al doilea pentru testare. De asemenea, am transformat valorile simbolice în valori numerice, asociind valoarea 1 pentru atacuri şi valoarea 0 pentru traficul normal.

Tabelul 6. Rezultatele testelor pentru modelele IDS propuse care se bazează pe SI

Sistemul	***Nr. de caracteristici***	***ADR***	***FAR***	***Acurateţea***
SVM	41	87.00%	0.265	89.30%
IG-SVM	26	85.00%	0.286	88.10%
IG-PSO-SVM	26	98.27%	0.044	98.68%
IG-ABC-SVM [2]	**26**	**98.53%**	**0.037**	**98.89%**
IG-BA-SVM [7]	**26**	**99.05%**	**0.030**	**99.15%**
IG-BAL-SVM [11]	**26**	**99.15%**	**0.019**	**99.38%**

Rezulatele testelor sunt ilustrateîn tabelul 6. Acestea relevă faptul că algoritmii de SI pot fi utilizaţi pentru a îmbunătăţi performanţa clasificatorului SVM. Deşi metoda de selectare a atributelor duce la performanţe mai scăzute, totuşi va reduce nivelul de complexitate al clasificatorului. Acest lucru este important întrucât IDS trebuie să proceseze seturi de date multidimensionale şi cu un nivel ridicat de zgomot. Se poate observa că PSO obţine rezultatele cele mai slabe, însă convergenţa rapidăşi simplitatea algoritmului pot reprezenta avantaje în cazul unor aplicaţii care necesită o execuţie rapidăşi acceptă mici inadvertenţe de performanţă.

După cum era de aşteptat ABC are nivelul cel mai ridicat de complexitate, întrucât este construit dintr-un grup de trei tipologii de indivizi, iar uneori aceştia trebuie să se aştepte între ei. Deşi obţine rezultate mai bune decât PSO, algoritmul ABC este consumator de resurse din punct de vedere al timpului de execuţie şi al memoriei utilizate.

Cele mai bune rezultate sunt obţinute de BAşi BAL. Deşi diferenţele de performană nu sunt atât de mari, ele pot deveni totuşi importante în cazul unor sisteme reale (aici testele au fost realizate pe un set redus de date). Din punct de vedere al complexităţii, algoritmii BAşi BALsunt mai simplii decât ABC şi sunt comparabili cu PSO din punct de vedere al timpului de execuţie şi al memoriei utilizate. Mai mult, performanţele obţinute de algoritmul BAL sunt uşor superioare algoritmului original (număr mai mic de alarme false şi o rată de detecţie mai mare). De aceea putem concluziona că BAşi BALsunt candidaţii preferaţi, cu menţiunea că rezultatele sunt valide numai pentru acest set de date ales aleator din NSL-KDD.

6.2.3 Metode de Selectare a Atributelor

6.2.3.1 Transformarea algorimilor SI în versiuni binare

Versiunea binară, denumită Binary Bat Algorithm (BBA), presupune că individul (liliacul) din grup va căuta într-un spaţiu multidimensional binar. Prin urmare, locaţia individului, sau soluţia, devine un şir de zero şi unu. Pentru aceasta autorii folosesc funcţia sigmoidă [28]:

$$S(v_{i,j}) = \frac{1}{1 + e^{-v_{i,j}}} \qquad (11)$$

Aşadar coordonatele fiecărui individ din grup devin:

$$x_{i,j} = \begin{cases} -1\,, dacă\ S(v_{i,j}) > \delta \\ 0, \qquad\quad altfel \end{cases} \tag{12}$$

Unde δ este un număr aleator între zero şi unu. Astfel, soluţia devine un şir de zero şi unu. Putem abstractiza fiecare coordonată, astfel încât aceasta să reflecte prezenţa atributului în set, dacă coordonata are valoarea unu, sau absenţa atributului din set, dacă coordonata are valoarea zero. Cea mai bună locaţie găsită la nivelul grupului este*soluţia problemei de opimizat*, în cazul de faţă este setul optim de atribute. Pentru a determina calitatea soluţiei propuse, algoritmul are la bază o *funcţie de performanţă*, iar îmbunătăţirea valorii acesteia reflectă o soluţie mai bună. Prin urmare, este important să alegem componentele corespunzătoare şi proporţiile corecte pentru a defini acestă funcţie de a cărei valoare depinde calitatea soluţiei determinate.

6.2.3.2 Modele Propuse

În figura 8 se regăseşte diagrama modelului IDS, conform şablonului din figura 6, dar în plus blocurile modelului sunt completate. Pentru metodele de selectare a atributelor am ales $binary\ SI\ \in \{BPSO, BBA, BBAL, BBA(E)\}$.

În acest caz avem două setări pentru funcţiile de performanţă :

Opţiunea 1 $$fitness = 90\ \%\ Accuracy + 10\ \%\ \frac{1}{nbFeat} \tag{13}$$

Opţiunea 2 $$fitness = 60\%\ ADR + 30\ \%\frac{1}{FAR} + 10\ \%\ \frac{1}{nbFeat} \tag{14}$$

Rezultatele testelor şi algoritmii CI utilizaţi pentru testarea modelelor sunt redate în tabelele următoare.

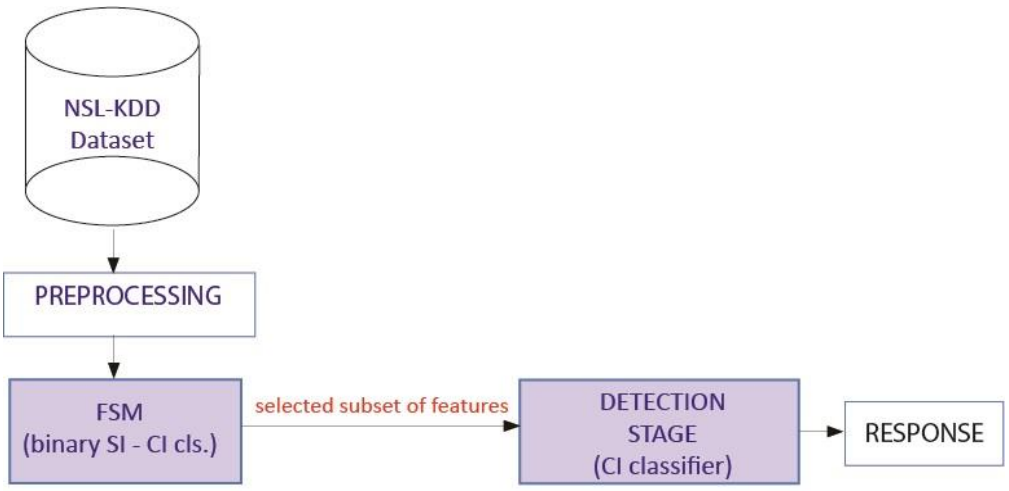

Figura 8. Diagrama modelelor IDS propuse – metode de selectare a atributelor

Din tabelul 7 putem deduce că BBAL este un candidat mai bun decât BBA sau BPSO pentru alegerea atributelor, întrucât necesită mai puţine iteraţii pentru a ajunge la rezultate similareşi

chiar uneori superioare; acest lucru fiind mai vizibil în cazul clasificatorului Naive Bayes. Mai mult, introducerea numărului de atribute ca element al funcţiei de performanţă determină alegerea unui număr mai mic de atribute (comparabil cu cele alese în lucrarea anterioară[4]).

Tabelul 7. Rezultatele testelor pentru modelele IDS – opţiunea 1

	Setul de date pentru antrenare				***Setul de date pentru testare***		
Alg. SI	**Nr. Atr.**	**ADR**	**FAR**	**Nr. Iteraţii**	**ADR**	**FAR**	**Timp (ms.)**
BBAL-NB	15	95.91	2.95	53	91.62	5.73	764
BBA-NB	18	94.62	5.30	70	89.73	7.24	793
BPSO-NB	22	93.73	6.37	80	89.44	7.86	829
NB Simplu	41	91.46	8.35		90.53	6.66	1019
	Setul de date pentru antrenare				**Setul de date pentru testare**		
Alg. SI	**Nr. Atr.**	**ADR**	**FAR**	**Nr. Iteraţii**	**ADR**	**FAR**	**Timp(ms.)**
BBAL-SVM	16	99.07	0.86	4	95.78	2.89	68768
BBA-SVM	22	99.00	0.96	6	95.03	3.28	78251
BPSO-SVM	23	98.89	1.07	10	94.03	4.01	80726
SVM Simplu	41	98.93	1.05		89.64	6.88	82603

Rezultatele din tabelul 8 arată că BBA(E) este un candidat mai bun decât BBA sau BPSO pentru alegerea atributelor, întrucât necesită mai puţine iteraţii pentru a ajunge la rezultate similare; Mai mult introducerea numărului de alarme false ca element al funcţiei de performanţă determină valori mai reduse ale FAR (comparabil cu modelul anterior din tabelul 7).

Tabelul 8. Rezultatele testelor pentru modelele IDS – opţiunea 2

	Setul de date pentru antrenare					***Setul de date pentru testare***	
Alg. SI	**Nr. Indivizi**	**ADR**	**FAR**	**Nr. Atr.**	**Nr. Iteraţii**	**ADR**	**FAR**
BBA(E)-C4.5	2	99.76	0.23	15	60	96.02	2.75
BBA- C4.5	2	99.76	0.23	18	80	96.01	3.20
BPSO- C4.5	2	99.36	0.56	20	87	96.66	2.62
C4.5 Simplu		99.26	0.73	41		95.67	3.02
	Setul de date pentru antrenare					**Setul de date pentru testare**	
Alg. SI	**Nr. Indivizi**	**ADR**	**FAR**	**Nr. Atr.**	**Nr. Iteraţii**	**ADR**	**FAR**
BBA(E)-SVM	2	99.76	0.23	15	60	96.02	2.75
BBA-SVM	2	99.76	0.23	18	80	96.01	3.20

	Setul de date pentru antrenare					*Setul de date pentru testare*	
Alg. SI	**Nr. Indivizi**	**ADR**	**FAR**	**Nr. Atr.**	**Nr. Iteraţii**	**ADR**	**FAR**
BPSO-SVM	2	99.36	0.56	20	87	96.66	2.62
BBA(E)-SVM	5	99.49	0.42	16	13	97.23	1.3
BBA-SVM	5	99.48	0.44	16	18	91.23	5.5
BPSO-SVM	5	99.54	0.40	17	25	97.11	1.5
SVM simplu		89.81	7.28	41		89.64	6.88

6.2.4 Modele Combinate

În figura 9 se regăseşte diagrama modelului IDS, conform şablonului din figura 6, dar în plus blocurile modelului sunt completate. În acest caz există două setări, însă în cadrul acestui rezumat vom face referire doar la una dintre acestea, cea de-a doua fiind detaliată în cadrul tezei.

În acest caz avem două setări pentru funcţiile de performanţă, în funcţie de modulul în care sunt utilizate :

Metoda de selectare a atributelor (FSM)

$$fitness = 90\ \%\ Accuracy + 10\ \%\ \frac{1}{nbFeat} \quad (13)$$

Selectarea parametrilor pentru SVM

$$fitness = 60\%\ ADR + 30\ \%\frac{1}{FAR} + 10\ \%\ \frac{1}{nbFeat} \quad (14)$$

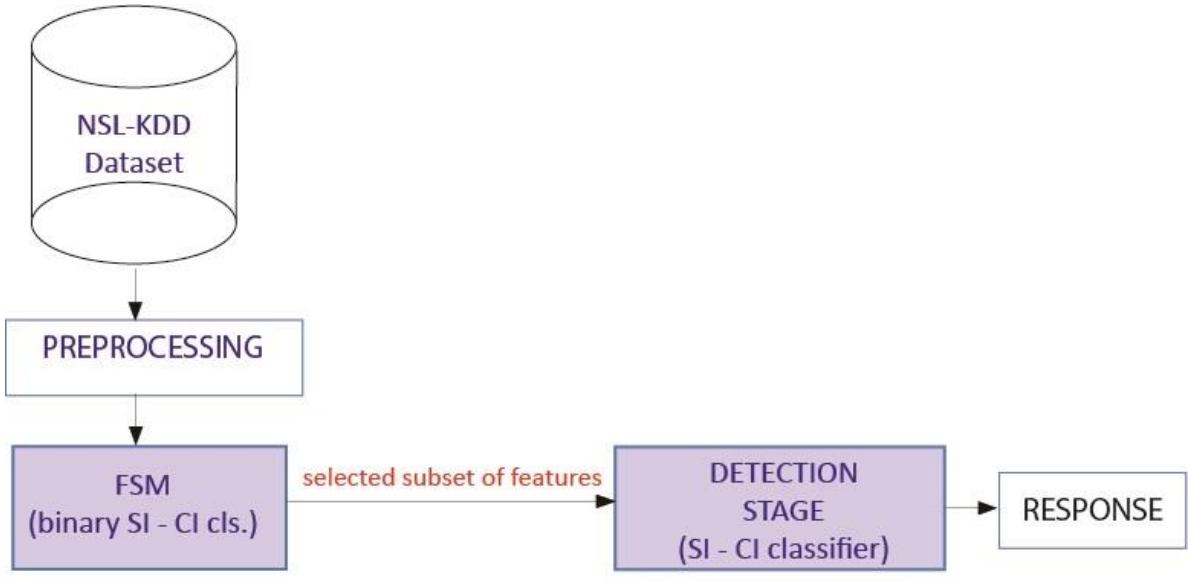

Figura 9. Diagrama modelelor IDS propuse – modelul combinat

Tabelul 9. Rezultatele testelor pentru modelele IDS – combinate (FSM şi selectarea parametrilor)

	Setul de date pentru antrenare					**Setul de date pentru testare**	
Alg. SI	***Nr. Indivizi***	***ADR***	***FAR***	***Nr. Atr.***	***Nr. Iteraţii***	***ADR***	***FAR***
BBA(E)-SVM	*2*	*99.29*	*0.61*	*19*	*50*	*97.17*	*1.6*
BBA- SVM	*2*	*99.38*	*0.52*	*19*	*80*	*90.25*	*5.3*
BPSO- SVM	*2*	*99.36*	*0.56*	*21*	*100*	*96.88*	*1.8*
BBA(E)-SVM	*5*	*99.49*	*0.42*	*16*	*13*	*97.57*	*1.3*
BBA- SVM	*5*	*99.48*	*0.44*	*16*	*18*	*91.23*	*5.5*
BPSO- SVM	*5*	*99.54*	*0.40*	*17*	*25*	*97.11*	*1.5*
SVM simplu		*98.93*	*1.05*	*41*		*89.64*	*6.88*
Alg. SI	***C***	***σ***	***ADR***	***FAR***	***Timp***	***Nr. Iteraţii***	***Nr. Indivizi***
BA(E)-SVM	*1*	*0.001*	*97.44*	*1.51*	*72.89*	*6*	*5*
BA- SVM	*1.26*	*0.001*	*96.75*	*2.0*	*70.39*	*11*	*5*
PSO- SVM	*1479.82*	*0.001*	*96.61*	*2.57*	*43.70*	*13*	*5*
SVM simplu	*1*	*0.5*	*93.8*	*4.97*	*77.45*		

Rezultatele din tabelul 9 arată că BBA(E)-SVM obţine cele mai bune performanţe şi necesită un numărmai mic de iteraţii decât BPSO sau BBA. Totodată, se observă că dacă scădem numărul de indivizi din grup, numărul de iteraţii pentru a obţine performanţe similare va creşte; fapt care era de aşteptat având în vedere că mai puţini indivizi înseamnă mai puţini candidaţi cu soluţii.
În cazul selectării atributelor am utilizat doar setul ales de BBA(E)-SVM. Rezultatele testelor arată faptul că pentru performnaţe similare BA şi PSO necesită un număr mai mare de iteraţii, uneori numărul acestora fiind chiar dublu.

În concluzie, putem spune că ipotezele noastre privind BA(E)şi BAL au fost confirmate, ambele versiuni depăşind uşor performanţele algoritmului original, fapt vizibil în numărul mai mic de iteraţii necesare pentru obţinerea unor performanţe similare. Am realizat şi o comparare a celor doi algoritmi propuşi (BA(E)şi BAL), însă putem spune că performanţele lor sunt similare cu diferenţa că BAL poate depăşi BA(E)în unele cazuri, întrucât BA(E) trebuie să caute vecinul cu o funcţie de cost superioară soluţiei curente.

6.3 MODELE IDS BAZATE PE AIS

6.3.1 Componentele IDS

Diagrama modelului IDS este redatăîn figura 10, iar opţinile privind algoritmii aleşi pentru completarea blocurilor din diagramă sunt enumeraţi în tabelul 10.

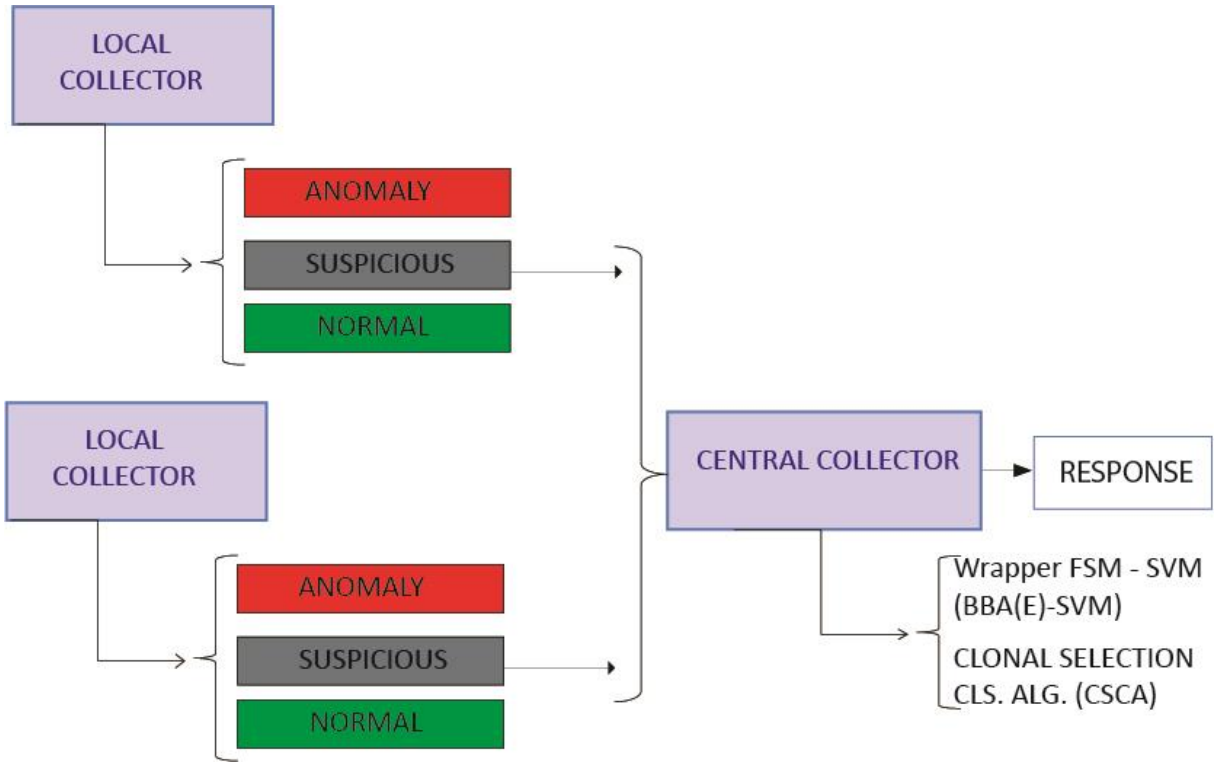

Figura 10. Diagrama modelelor IDS propuse

Sumarul din tabelul 10 împarte modelele IDS propuse în funcţie de versiunea DCA utilizată şi algoritmul de clasificare utilizat pentru colectorul central. În cadrul acestui raport sunt prezentate rezultatele publicate în lucrarea autorului [12].

Tabelul 10. Sumarul modelelor IDS propuse care se bazează pe SI

Model IDS	*Colector Local*	*Colector Central*
Setup 1 [10]	*DCA* cu multiplicare de antigen*	*BBA(E)-SVM*
Setup 2 [12]	*DCA* cu segmentare de antigen*	*CSCA*

Modelul propus [12] este un IDS bazat pe anomalii cu răspuns activ, construit pe baza a două componete, care îmbină rapiditatea DCA cu precizia clasificatorului CSCA. Arhitectura sistemului este construită pe baza următoarelor componente:

- *Componenta Locală* – care va colecta logurile local de la host şi va realiza o detecţie rapidă utilizând DCA. Înregistrările vor fi clasificate în funcţie de valoarea lui MCAV

astfel: anomalii ($MCAV > 0.85$), suspicioase ($MCAV \epsilon [0.65, 0.85]$) sau normale ($MCAV < 0.65$).

- *Componenta Centrală* – care va colecta toate înregistrările suspicioase de la toate componetele locale din sistemul monitorizat. Pentru detecţia finală, vom utiliza clasificatorul CSCA. Mai mult, pentru a compara performanţele obţinute de CSCA vom avea în vedere rezultatele obţinute de alţi doi algoritmi de clasificare tradiţionali: SVM (maşinile cu suport vectorial) şi C4.5, în cazul setului de date suspect.

Sistemul IDS propus va permite o detecţie rapidă, aproape în timp real, pentru componenta locală, care va bloca conexiunile clasificate ca anomalii şi suspicioase, urmând ca analiza avansată să fie realizată la nivelul central, care va lua decizia privind veridicitatea datelor considerate suspicioase.Componenta locală are la bază algoritmul DCA determinist în versiunea prezentatăîn capitolul 4, cu multiplicare de antigenşi respectiv cea cu segmentare de antigeni. Dacăîn primul caz detecţia se va realiza după analiza unui set mare de date, în cazul segmentării, rezultatul detecţiei va fi întors după analiza unui număr determinat de înregistrări. Reamintim căîmbunătăţirea algoritmului constăîn introducerea unui prag suplimentar. Am considerat utilă introducerea unei limite suplimentare, întrucât procentajul rezultat din valoarea MCAV poate reflecta un nivel de "indecizie" sau decizie la limită, prin urmare pentru acel segment de înregistrări vom apela la algoritmi clasici pentru a obţine răspunsul.

Testele au fost realizate pe o platformă cu sistemul de operare Windows 8.1, procesor Intel i7 şi 8GB de memorie RAM.

a. Configurarea Componentei Locale

Prima etapă constăîn alegerea acelor atribute din setul de date care vor constitui semnalele. Pentru aceasta alegem un proces automat bazat pe *information gain (IG)*şi astfel semnalele vor fi clasificate în funcţie de scorul obţinut după cum urmează:

PAMP: serror_rate, srv_serror_rate, same_srv_rate, dst_host serror, dst_host_rerror_rate;
Danger: count, srv_count;
Safe: logged_in, srv_different_host_rate, dst_host_count.

Tabelul 11Parametrii pentru componenta locală

Versiunea dDCA	***Numărul de celule DC***	***Multiplicarea de antigen***	***Intervalul ciclului de viaţă***	***Segmentarea antigenului***
AM1	100	10	ϵ [100, 300]	
AM2	20	5	ϵ [20, 60]	
AS1	100	10	ϵ [100, 300]	ϵ{100, 500, 950}
AS2	20	5	ϵ [20, 60]	ϵ{100, 500, 950}
AS2*	20	5	ϵ [20, 60]	ϵ{20, 50}

Pentru testarea modelului vom alege algoritmul DCA determinist cu multiplicare de antigen şi respectiv cel determinist cu segmentare de antigen, fiecare având mai multe configurări enumerate în tabelul 11. Am implementat în Java versiunea dDCA cu multiplicare de antigen (AM) şi pe cea cu segmentarea de antigen (AS). În cazul segmentării vom testa mai multe opţiuni de dimensiuni de segmente pentru a identfica influenţa acesteia asupra performanţelor obţinute. Aceste variante sunt identificate ca fiind : AS_{i_1}(100 de antigeni per segment), AS_{i_5}(500 de antigeni per segment)şi AS_{i_9}(950 de antigeni per segment), unde i

corespunde scenariului de testare al modelului şi are valorile 1 sau 2. În cel de al doilea scenariu am mai inclus două dimensiuni de segmente, în cazul unui număr mai mic de celule DC şi le-am notat ca fiind : AS_{2_2}(20 de antigeni per segment)şi AS_{2_0}(50 de antigeni per segment).

b. <u>Configurarea Componentei Centrale</u>

Pentru clasificatorii CSCA (`clonalg.CSCA`), SVM (`functions.SMO`) şi C4.5 (`trees.J48`) am utilizat implementarea lor implicităîn Java din cadrul bibliotecii de algoritmi Weka versiunea 3.6.10[32].

Rezultatele din tabelul 12 arată faptul că DCA reuşeşte să obţină rezultate bune în ceea ce priveşte ADR, FAR şi timpul de execuţie (în cazul dDCA cu segmentare am avut în vedere timpul de procesare al unui singur segment, întrucât acest interval de timp este necesar pentru a obţine un răspuns de la componenta locală). Rezultatele testelor din tabelul 12 au în vedere doar datale normale şi cele clasificate ca anomalii, dar nu şi pe cele considerate suspicioase întrucât acestea urmează a fi analizate de către componenta centrală. Multiplicarea de antigeni este integrată în vederea îmbunătăţirii performanţelor algoritmului DCA, avantajul oferit fiind acela că un antigen va fi proliferat şi va fi analizat de mai multe celule DC. Prin urmare, clasificarea antigenului nu va fi rezultatul singular al unei celule ci va fi obţinut sub forma unui vot majoritar al mai multor celule care au cicluri de viaţă diferite. Mai mult, segmentarea poate produce mai multe seturi de rezultate, nu ca în cazul multiplicării de antigeni. Prin urmare, am repetat testele de 20 de ori şi am luat în calcul valorile ADR şi FAR obţinute după ce toate segmentele au fost prcesate.

În cazul primului scenariu, am configurat DCA cu valori mai mari pentru ciclul de viaţăşi respectiv pentru factorul de multiplicare al antigenilor. În consecinţă, un antigen ar putea fi testat de mai multe ori decât în cazul celui de al doilea scenariu. AM1 obţine rezultate slabe întrucât timpul de excuţie (10 minute) este destul de îndelungat, iar numărul de înregistrări suspicioase reprezintă aproximativ 85% din setul de date original. Segmentarea de date reduce setul de suspiciuni la 45% din setul iniţial, în cazul AS_{1_1}şi îmbuntăţeşte ADR cu 13% în cazul lui AS_{1_5}, dacă le comparăm cu opţiunea AM1. Cu cât creşte numărul de antigeni incluşi în segmente, ADR are o evoluţie descendentă, dacă numărul de celule DC este mai mare decât numărul de antigeni din cadrul unui segment. Această evoluţie este mai apoi regăsită în rezultatele obţinute de AM1, care are numărul de segmente egale cu numărul de înregistrări. Dacă numărul de celule DC este egal sau mai mic decât dimensiunea segmentului atunci, procesul de multiplicare este cumva păcălit, întrucât ar putea fi aceeaşi celulă care va analiza un anumit tip de antigen. Această ipoteză explică rezultatele slabe obţinute de AS_{1_1}, în comparaţie cu AS_{1_5} care are un număr mai mare de antigeni. Pe de altă parte FAR are o evoluţie uşor descendentă o dată cu creşterea dimensiunii segmentului, întrucât cu cât numărul de antigeni trimişi spre analiză este mai mic DCA devine mai susceptibil la FP (false positive).

În ceea ce priveşte cel de al doilea scenariu, reducerea numărului de celule, al factorului de multiplicare şi respectiv al pragului pentru ciclul de viaţă al celulei DC, poate determina rezultate mai performante şi cu un număr de date suspicioase mai redus. AM2 obţine performanţe mai bune şi un număr mai mic de înregistrări suspicioase. Cu toate acestea, timpul de execuţie este destul de îndelungat (16 secunde). Prin adăugarea segmentării de antigeni, modelul obţine valori mai ridicate pentru ADR (cu o îmbunătăţire de aproximativ 12% dacă avem în vedere rezultatele obţinute de AM2) şi un timp de execuţie mai redus (12 ms în cel mai bun caz). Modificarea numărului de antigeni care fac parte din segment, aduce

îmbunătăţiri ale ADR, comparabil cu AM2. Dacă numărul de celule este mai mic decât numărul de antigeni din cadrul segmentului, atunci ADR va creşte până când numărul de antigeni va depăşi numărul de celule DC care fac parte din grup. În acest ultim caz, un număr mai mare de antigeni va determina scăderea valorii ADR. Ca şi în scenariul anterior, un număr mai mic de antigeni va afecta numărul de alarme false.

Tabelul 12. Rezultatele testelor obţinute de componenta locală

Versiunea dDCA	***ADR (%)***	***FAR (%)***	***Timpul de execuţie (ms)***	***Nr. de înregistrări suspicioase***
AM1	61 (+0.2/-0.3)	4.1 (+/-0.12)	642 X 10^3	8071 (+/-10)
AS_{1_1}	68 (+0.2/-0.34)	3 (+0.4/-0.2)	[200, 270]	4250 (+30/-9)
AS_{1_5}	79 (+1.5/-0.5)	3.3 (+/-0.21)	[2600, 2705]	6900 (+20/-5)
AS_{1_9}	72 (+1.2/-0.2)	4 (+0.11/-0.10)	[6120, 7050]	6500 (+/-20)
AM2	71 (+0.9/-0.4)	5 (+/-0.25)	16 X 10^3	2250 (+/-30)
$AS_{2_2}^*$	78 (+1.2/-0.34)	8 (+0.9/-0.2)	[12, 16]	2250 (+40/-3)
$AS_{2_0}^*$	80 (+3.1/-0.41)	6 (+0.6/-0.4)	[12, 16]	2220 (+30/-5)
AS_{2_1}	78 (+1.2/-0.34)	5 (+0.6/-0.4)	[12, 16]	2200 (+34/-2)
AS_{2_5}	76 (+1.5/-0.5)	5 (+/-0.31)	[15, 227]	2070 (+50/-5)
AS_{2_9}	75 (+1.2/-0.1)	5.1 (+0.11/-0.1)	[225, 409]	2010 (+40/-3)

În ambele scenarii segmentarea va influenţa pozitiv rata de detecţie a atacurilor, în timp ce numărul de alarme false va avea un trend uşor crescător dar cu o valoare neglijabilă. Pe când, multiplicarea în segmente va creşte rata de detecţie a atacurilor.

Tabelul 13. Rezultatele testelor obţinute de componenta centrală

Algoritmul	***Setul de date suspicioase***	***ADR***	***FAR***
Setul de date generat de AM1			
CSCA	8071	94.01	5.31
SVM	8071	98.22	1.52
C4.5	8071	99.11	0.08
Setul de date generat de AS_{1_9}			
CSCA	6949	94.37	4.99
SVM	6949	91.60	3.67
C4.5	6949	9.19	0.07
Setul de date generat de AS_{1_1}			
CSCA	4251	93.92	4.96
SVM	4251	90.40	6.90
C4.5	4251	99.04	0.077
Setul de date generat de AM2			
CSCA	2250	98.50	6.50
SVM	2250	81.32	62.03
C4.5	2250	99.64	0.155

În cele din urmă, ultima etapă a experimentului am avut în vedere componenta centrală care a analizat datele "suspicioase" colectate de la componenta locală. Vom avea în vedere doar o parte din seturile de date suspicioase obţinute în etapa iniţială de testare a componetei

locale. Pentru a evalua performanţele clasificatorilor am utilizat cross validare (10 folds cross validation). CSCA obţine performanţe bune, depăşindu-le pe cele obţinute de SVM şi fiind apropiate de cele ale binecunoscutului C4.5. Din rezultatele slabe obţinute de SVM, am putea deduce faptul că acesta are nevoie de seturi mai mari de date pentru procesul de antrenare, pe când CSCA şi C4.5 reuşesc să obţină performanţe ridicate chiar şi în cazul unor seturi de date de mici dimensiuni.

7 MODELE PROPUSE PENTRU DETECŢIA WEBSPAMMING

Astăzi internetul a devenit indispensabil pentru majoritatea oamenilor. De obicei căutările pe internet încep prin acesarea unui motor de căutare (ex. Google, Bing etc.). Principalele procese derulate de acesta sunt : web crawling, indexarea şi căutarea. În funcţie de un algoritm interior propriu, motorul de căutare va ierarhiza paginile de internet returnate utilizatorului. În mod evident, ordinea de apariţie a paginilor este importantă pentru companii sau chiar pentru hackeri care pot utiliza acest mecanism pentru a porni un atac cibernetic.

Termenul de ***Web Spamming*** *(spamindex, search engine spam etc.) a fost introdus în 1996 [29] şi se referă la încercările deliberate de a vicia algoritmii de ierarhizare din motoarele de căuare pe internet,având ca scop clasarea pe primele poziţii a anumitor pagini web.*

În general tehnicile de web spamming se pot clasifica în următoarele categorii *[30]*:

- *Content spam* – inserarea unor cuvinte frecvent căutate de utilizatori, dare care nu au legătură cu conţinutul paginii.
- *Link spam* – crearea unor linkuri în paginile web pentru a creşte scorul bazat pe link-uri.
- *Cloacking* – spammer-ul va întoarce un content diferit web crawler-ului faţă de ceea ce vede utilizatorul.

Pentru detecţia web spamming am propus două modele care au fost publicate în două lucrări ale autorului care utilizează PSO-SVM [1]şi respectiv cAnt-Miner[3] pentru detecţia content spam. Dintre acestea, în cadrul acestui rezumat vom prezenta doar modelul PSO-SVM.

7.1 MODELUL PROPUS

Modelul propus combină o versiune paralelizată a algoritmului PSO şi clasificatorul SVM. Algoritmul PSO va fi utilizat pentru selectarea celor doi parametrii SVM (C şi σ). Întrucât setul de date pentru testare este destul de redus (3766 de înregistrări) şi dorim să păstrăm logica algoritmului PSO, am implementat o versiune paralelizată de tip sincronizată. Mai mult, platforma de testare nu dispune de un număr mare de procesoare, prin urmare implementarea variantei asincrone ar putea determina creşterea nivelului de load balance între core-urile procesorului.Calcularea funcţiei fitness este operaţiunea cea mai consumatoare de resurse. Aşadar aceasta va fi componenta paralelizată (vezi figura 11).

7.1.1 Setul de Date

Pentru testarea modelului am utilizat setul de date WEBSPAM-UK2011[31] ce conţine înregistrări din domeniul .uk. Setul de date include atribute de tip conten based şi conţine: 1998 de înregistrări de tip spam şi 1768 de înregistrări de tip non-spam. Fiecare înregistrare are 11 atribute. Întrucât setul de date este echilibrat, vom utiliza nivelul de acurateţe pentru funcţia fitness a algoritmului PSO. Atributele sunt enumerate în tabelul 14.

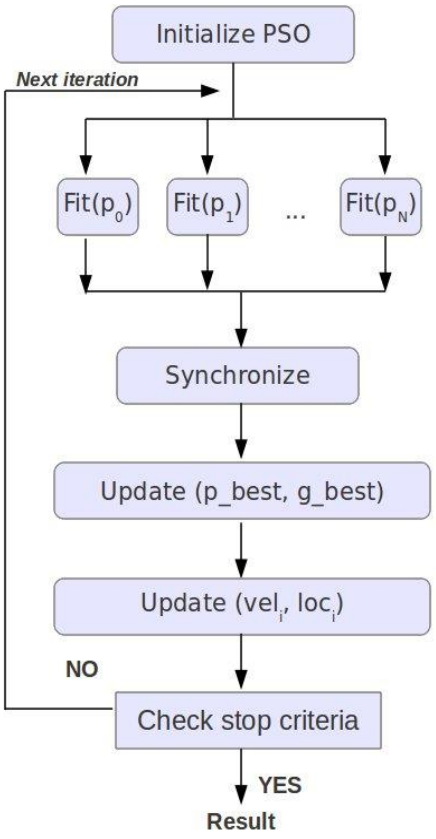

Figura 11. Versiunea sincronă pentru paralelizarea algoritmului PSO

Tabelul 14. Atributele din setul de date WEBSPAM-UK2011

Atributele din setul de date WEBSPAM-UK2011[31]	
amount_of_anchor_text; meta_char; meta_word; unique_word; total_word;	max_length; min_length; average_length; title_words; comp_ratio; img.

Testele au fost realizate pe o platformă cu 6GB de RAM, sistem de operare Windows 8 şi procesor Intel Core i7-4700HQ (4 core). Versiunea secvenţială şi cea paralelizată a algoritmului PSO au fost implementate în Java. Pentru clasificatorul SVM am utilizat librăria Weka 3.6.10 [32].Algoritmul PSO are 10 indivizi în grup, prin urmare avem 10 fire de execuţie. Parametrii C şi σ variază astfel : $C \in [1, 10^8]$, $\sigma \in [0.001, 36]$; aceste două intervale dictează ariile de căutare pentru soluţiile bidimensionale ale algoritmului PSO. Pentru testare am realizat două abordări : în prima am utilizat cross validare (10 folds cross validation), iar în a doua am creat două fişiere separate pentru învăţare şi testare.Rezultatele testelor sunt redateîn tabelele următoare. După cum era de aşteptat versiunea paralelizată a algoritmului PSO se execută de 3,5 ori mai repede decât versiunea secvenţială, ceea ce era de aşteptat având în vedere arhitectura procesorului pe care a fost testat algoritmul.

Tabelul 15. Timpul de execuţie obţinut de algoritmul PSO pentru selectarea parametrilor SVM

Sistem	***Paralel***	***Secvenţial***
PSO-SVM-10 (10 fold cross-validation)	*44.6 min*	*156.8 min*
PSO-SVM-1 (fişiere separate pentru antrenare şi testare)	*3.3 min*	*11.2 min*

Totodată, am comparat rezultatele obţinute cu performanţele altor algoritmi de clasificare precum Naive Bayes sau arborii de decizie (C4.5, algoritmul J48 din Weka). Pentru web spamming este importantă detecţia acestuia, prin urmare am avut în vedere acurateţea şi recall (TPR) pentru metricile de performanţă. Rezultatele din tabelul 15 arată că modelul propus de noi este mai performant decât ceilalţi algoritmi de clasificare. Mai mult, modelul nostru obţine rezultate mai bune decât cel propus de Silva et. al. [33].

Tabelul 16. Rezultatele testelor pentru detecţia web spamming

Sistemul	***Acurateţea***	***TPR***
PSO-SVM-10	**93.5 %**	**90.19%**
PSO-SVM-1	**72.86%**	**66.18 %**
SVM	56.21 %	54.79%
C4.5	86.98 %	87.47 %
Naive Bayes	56.85 %	55.26 %
SVM [33] – WebSpamUK2006	71.1 %	60.06 %
SVM [33] – WebSpamUK2007	70.6 %	49.8 %

8 CONCLUZII. CONTRIBUŢII PERSONALE. DIRECŢII VIITOARE DE CERECETARE

Peisajul atacurilor cibernetice este într-o continuă expansiune din punct de vedere al volumului, dacă avem în vedere actorii cibernetici sau datele gestionate în mediul cibernetic, dar şi din punct de vedere al complexităţii ameninţărilor cibernetice. Atacurile cibernetice au fost generate din nevoia utilizatorilor cu intenţii maliţioase de a obţine un avantaj sub forma unui renume social, beneficiu economic sau chiar pentru a realiza spionaj informatic. Prin urmare, este clar că atât timp cât informaţii valoroase vor fi vehiculate în mediul informatic, acestea vor atrage interesul atacatorilor cibernetici.Contracararea atacurilor cibernetice este similară procesului de stingere a unui incendiu, întrucât experţii de securitatea trebuie să ofere o soluţie cât mai rapid şi trebuie să protejeze sistemele care încă nu au devenit victimele atacatorilor. Aşadar, este necesară dezvoltarea unor instrumente noi de securitate capabile de a se adapta la procesele dinamice din spaţiul cibernetic. În opinia noastră, aceste soluţii noi trebuie săpermită procesarea unor volume mari de date fără a consuma multe resurse şi respectiv fără a supraîncărca sistemele protejate.

Am început activitatea de cercetare prin studierea algoritmilor de inteligenţă computaţională, axându-mă pe algoritmii inspiraţi biologic. Natura reprezintă o sursă de inspiraţie pentru multe soluţii ştiinţifice şi tehnologice;principalele motive pentru alegerea acestora în vederea dezvoltării unor modele matematice pentru diverşi algoritmi includ simplitatea şi eficienţa dovedităîn timp, care a asigurat existenţa şi supravieţuirea acestor sisteme în cadrul unui mediu care este obiectul unor schimbări continue. Din paleta largă de

algoritmi inspiraţi biologic am ales să studiez a două clase: swarm intelligence şi artificial immune systems.

În acelaşi timp am început să cercetăm peisajul curent al ameninţărilor cibernetice. Concluziile fiind că, în ciuda eforturilor constante ale experţilor de securitate, atacurile cibernetice continuă să prolifereze în contextul unor produse malware metamorfozate şi a unor unelete de hacking care au devenit mult mai accesibile.Aşadar, o dată cu evoluţia tehnologiei intrumentele de hacking au evoluat şi ele. Monitorizarea şi auditarea continuă a evenimentelor din sistem generează volume mari de date, care trebuie apoi analizate şi din care trebuie extrase acele elemente de interes pentru stabilirea stării de normalitate/anormalitate a sistemulelor protejate.Având în vedere proprietăţile algoritmilor SI şi AIS, am considerat că aceştia ar putea oferi soluţii viabile pentru asigurarea securităţii cibernetice.

Din clasa de algoritmi SI am ales algoritmul Bat Algorithm, pentru care am propus două versiuni îmbunătăţite, respectiv dintre algoritmii AIS am selectat un algoritm de generaţie nouă, Dendritic Cell Algorithm, pentru care am modificat componenta de clasificare.Toate aceste ipoteze teoretice au fost validate prin implementarea unor modele de detecţie a intruziunilor care utilizează aceşti algoritmi pentru componentele de detecţie sau pentru selectarea atributelor.

8.1 CONTRIBUŢIILE PERSONALE

Soluţiile pe care le-am propus în cadrul acestei teze sunt bazate pe algoritmi de inteligenţă computaţionalăşi ar putea ajuta la implementarea unor soluţii de securitate capabile de a învăţa şi a oferi adaptabilitate la schimbările permanente din ecosistemul cibernetic.

Contribuţiile personale sunt enumerateîn cele ce urmează:

- Am propus **două modele** de detecţie a intruziunilor care îmbină Information Gain pentru metoda de selectare a atributelor cu clasificatorul SVM îmbunătăţit cu un algoritm SI pentru selectarea parametrilor. Dintre algoritmii SI am ales : Artificial Bee Colony (ABC)[2] şi the Bat Algorithm (BA)[7], pe care i-am comparat cu binecunoscutul PSO.
- Am propus **două versiuni modificate** ale algoritmului BA care îmbunătăţesc procesul de căutare locală. Prima soluţie pe care am numit-o BAL înlocuieşte random walks cu zborurile Levy. Cea de a doua variantă include un termen suplimentar reprezentat de distanţa Euclidiană dintre soluţia curentă a individului şi o soluţie cu o funcţie de performanţă superioară soluţiei curente. Propunerile teoretice au fost validate empiric în modele de detecţie a intruziunilor care utilizează algoritmii sub forma lor binară pentru implementarea unor metode de selectare a atributelor sau în forma lor standard pentru problema îmbunătăţirii procesului de selectare a parametrilor clasificatorului SVM. Mai mult, aceste modele utilizează funcţii de performaţă diferite şi totodată combină diverşi algoritmi de clasificare pentru a confirma ierarhia algoritmilor SI. Toate aceste modele au fost publicate în ***şase lucrări ale autorului*** [4][5][6][8][9][11].
- Am realizat o analizăşi o comparaţie proprie a algoritmilor SI, evidenţiind proprietăţile cheie ale acestor algoritmişi metodele diferite prin care acestea se regăsesc în diverşi algoritmi (vezi tabelul 1).
- Am oferit o perspectivă personală asupra similitudinilor şi naturii complementare exitente între algoritmii de Swarm Intelligence şi algoritmii Artificial Immune Systems (vezi capitolul 3).
- În cadrul acestei teze am extins analiza teoretică a algoritmului Dendritic Cell Algorithm (DCA) şi am oferit propria persectivă asupra încadrării acestui algoritm în clasa Artificial Immune System, conform şablonului propus de Castro şi Timmis [17].

- Am propus o variantă modificată a algoritmului DCA . Îmbunătăţirea algoritmului se referă la modificarea funcţiei de clasificare prin includerea unui prag suplimentar, care creează astfel o regiune de suspiciuni. Aceste înregistrări suspicioase fac obiectul unei analize detaliate care poate fi realizată de către experţi de securitate sau prin intermediul altor algoritmi de clasificare. Propunerile teoretice au fost validate empiric prin includerea algoritmului DCA modificat în două variante (DCA-AM, cu multiplicare de antigen şi DCA-AS, cu segmentare de antigen) sub forma unei componente locale în cadrul unor modele pentru detecţia intruziunilor. Pentru analiza detaliată am utilizat un algoritm AIS (CSCA) şi respectiv o versiune a clasificatorului SVM îmbunătăţită (BBA(E)-SVM). Ambele modele fiind publicate în cadrul a ***două lucrări ale autorului*** [10][12].
- Am abordat problema detecţiei WebSpamming, pentru care am propus două modele de detecţie bazate pe algoritmi SI. Primul combină o variantă paralelizată PSO cu clasificatorul SVM, iar a doua soluţie utilizează cAnt-Miner pentru sarcina de clasificare. Ambele modele fac obiectul a ***două lucrări ale autorului*** [1][3].

8.2 PERSPECTIVE DE CERCETARE

Viitoarele perspective de cercetare ar putea include următoarele aspecte:

- Aplicarea modelelor propuse în alte domenii sau pentru alte probleme de securitate informatică. În opinia noastră modelele propuse pot fi uşor integrate în alte soluţii de securitate pentru componente de detecţie sau extragere a atributelor cheie.
- Integrarea librăriei de algoritmi în diverse instrumente de securitate (ex. AlienVault). Totodată, integrarea unei componente de colectare şi omogenizare a datelor colectate ar permite mai multe simulări în cazul unor scenarii variate.
- În activitatea noastră de cercetare am abordat doar o parte din paleta largă de algoritmi SI şi AIS. Prin urmare, testarea altor algoritmi pentru comparaţii este de interes.

Mediul cibernetic este într-un continuu proces de transformare şi de aceea de cele mai multe ori ceea ce era valabil astăzi mâine poate fi permiat. În opinia noastră, pentru a putea crea uneltele de securitate compatibile cu proprietăţile ecosistemului cibernetic, acestea trebuie să utilizeze cunoaşterea şi să permită adaptarea la schimbările permanente printr-un proces de învăţare. Din acest punct de vedere CI ar putea oferi o speranţă în acest sens. Mai mult, se pare că nu suntem singulari, dovadă fiind proiectele şi concursurile care încurajează aplicarea acestor algoritmi pentru soluţii de securitate informatică (ex. DARPA Cyber GrandChallenge to create an automatic tool based on Artificial Intelligence for detecting security
aws: https://www.cybergrandchallenge.com/; http://www.csail.mit.edu/System predicts 85 percent of cyber attacks using input from human experts;).

8.3 LISTA DE LUCRĂRI PUBLICATE

1) *A.-C. Enache and V. Sgarciu. Spam host classification using pso-svm. In 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, pages 1-5, 2014.****[ISI Thomson Proceedings, IEEE Xplore]***

2) *A.-C. Enache and V. V. Patriciu. Intrusions detection based on support vector machine optimized with swarm intelligence. In 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2014, Timisoara, Romania, May 15-17, 2014, pages 153-158, 2014. .****[ISI Thomson Proceedings, IEEE Xplore, dblp][citat în cadrul unui articol indexat ISI]***

3) *A.-C. Enache and V. V. Patriciu. Spam host classi_cation using swarm intelligence. In 10th International Conference on Communications (COMM), 2014, pages 1-4, 2014.****[ISI Thomson Proceedings, IEEE Xplore]***

4) *A.-C. Enache and V. Sgarciu. Enhanced intrusion detection system based on bat algorithmsupport vector machine. In SECRYPT 2014 - Proceedings of the 11th International Conference on Security and Cryptography, Vienna, Austria, 28-30 August, 2014, pages 184-189, 2014. .**[ISI Thomson Proceedings, IEEE Xplore, dblp]***

5) *A.-C. Enache and V. V. Patriciu. Comparative study on intrusion detection systems using support vector machines improved with swarm intelligence. Scientific bulletin of Politehnica University of Timisoara, Transactions on Automatic Control and Computer Science, 59(2): 141-147, December 2014**[BDI]***

6) *A.-C. Enache, Valentin Sgarciu, and A. Petrescu-Nita. Intelligent feature selection method rooted in binary bat algorithm for intrusion detection. In 10th IEEE Jubilee International Symposium on Applied Computational Intelligence and Informatics, SACI 2015, Timisoara, Romania, May 21-23, 2015, pages 517-521, 2015. .**[ISI Thomson Proceedings, IEEE Xplore, dblp]***

7) *A.-C. Enache and V. Sgarciu. Anomaly intrusions detection based on support vector machines with bat algorithm. In 18th International Conference on System Theory, Control and Computing, October 17-19, 2014, Sinaia, Romania, pages 856-861, 2014.**[IEEE Xplore]***

8) *A.-C. Enache and V. Sgarciu. An improved bat algorithm driven by support vector machines for intrusion detection. In International Joint Conference - CISIS'15 and ICEUTE'15, 8th International Conference on Computational Intelligence in Security for Information Systems - 6th International Conference on European Transnational Education, Burgos, Spain, 15-17 June, 2015, pages 41-51, 2015.**[ISI Thomson Proceedings, dblp, publicaţie Springer]***

9) *A.-C. Enache and V. Sgarciu. A feature selection approach implemented with the binary bat algorithm applied for intrusion detection. In 38th International Conference on Telecommunications and Signal Processing, TSP 2015, Prague, Czech Republic, July 9-11, 2015, pages 11-15, 2015.**[ISI Thomson Proceedings, IEEE Xplore, dblp]***

10) *A.-C. Enache, M. Ionita, and V. Sgarciu. An immune intelligent approach for security assurance. In 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, United Kingdom, June 8-9, 2015, pages 1-5, 2015. **[ISI Thomson Proceedings, IEEE Xplore, dblp]***

11) *A.-C. Enache and V. Sgarciu. Anomaly intrusions detection based on support vector machines with an improved bat algorithm. In 2015 20th International Conference on Control Systems and Computer Science, Bucuresti, Romania, pages 317-321, 2015. .**[ISI Thomson Proceedings, IEEE Xplore]***

12) *A.-C. Enache and V. Sgarciu. Designing real-time anomaly intrusion detection through artificial immune systems, In 2016 15th European Conference on Cyber Warfare and Security, Munich, Germany, pages 331-341, 2016.**[trimis spre indexare ISI Thomson Proceedings]***

9 BIBLIOGRAFIE

[1] A.-C. Enache and V. Sgarciu. Spam host classification using pso-svm. In 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, pages 1-5, 2014.

[2] A.-C. Enache and V. V. Patriciu. Intrusions detection based on support vector machine optimized with swarm intelligence. In 9th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2014, Timisoara, Romania, May 15-17, 2014, pages 153-158, 2014.

[3] A.-C. Enache and V. V. Patriciu. Spam host classification using swarm intelligence. In 10th International Conference on Communications (COMM), 2014, pages 1-4, 2014

[4] A.-C. Enache and V. Sgarciu. Enhanced intrusion detection system based on bat algorithm – support vector machine. In SECRYPT 2014 - Proceedings of the 11th International Conference on Security and Cryptography, Vienna, Austria, 28-30 August, 2014, pages 184-189, 2014.

[5] A.-C. Enache and V. V. Patriciu. Comparative study on intrusion detection systems using support vector machines improved with swarm intelligence. Scientific bulletin of Politehnica University of Timisoara, Transactions on Automatic Control and Computer Science, 59(2):141-147, December 2014

[6] A.-C. Enache, Valentin Sgarciu, and A. Petrescu-Nita. Intelligent feature selection method rooted in binary bat algorithm for intrusion detection. In 10th IEEE Jubilee International Symposium on Applied Computational Intelligence and Informatics, SACI 2015, Timisoara, Romania, May 21-23, 2015, pages 517-521, 2015.

[7] A.-C. Enache and V. Sgarciu. Anomaly intrusions detection based on support vector machines with bat algorithm. In 18th International Conference on System Theory, Control and Computing, October 17-19, 2014, Sinaia, Romania, pages 856-861, 2014

[8] A.-C. Enache and V. Sgarciu. An improved bat algorithm driven by support vector machines for intrusion detection. In International Joint Conference - CISIS'15 and ICEUTE'15, 8th International Conference on Computational Intelligence in Security for Information Systems - 6th International Conference on EUropean Transnational Education, Burgos, Spain, 15-17 June, 2015, pages 41-51, 2015

[9] A.-C. Enache and V. Sgarciu. A feature selection approach implemented with the binary bat algorithm applied for intrusion detection. In 38th International Conference on Telecommunications and Signal Processing, TSP 2015, Prague, Czech Republic, July 9-11, 2015, pages 11-15, 2015

[10] A.-C. Enache, M. Ionita, and V. Sgarciu. An immune intelligent approach for security assurance. In 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, United Kingdom, June8-9, 2015, pages 1-5, 2015.

[11] A.-C. Enache and V. Sgarciu. Anomaly intrusions detection based on support vector machines with an improved bat algorithm. In 2015 20th International Conference on Control Systems and Computer Science, Bucuresti, Romania, pages 317-321, 2015.

[12] A.-C. Enache and V. Sgarciu. Designing real-time anomaly intrusion detection through artificial immune systems, In 2016 15th European Conference on Cyber Warfare and Security, Munich, Germany, pages 331-341, 2016.

[13] ITU. Definition of cybersecurity. http://www.itu.int/en/ITU-T/studygroups/com17/Pages/cybersecurity.aspx, 2016. [Online; accessed 23-May-2016].

[14] E. Chien N. Falliere, L. Murchu. W32.stuxnet dossier. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf, 2011. [Online; accessed 1-August-2016].

[15] James C. Bezdek. Computational Intelligence: Imitating Life, chapter What is computational intelligence ?, pages 1-11. IEEE Press, New York, 1994.

[16] X. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm", Proc. of the Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), Studies in Computational Intelligence, Springer Berlin, 284, Springer, 2010, pp. 65-74.

[17] L. N. de Castro and J. Timmis, „Artificial Immune Systems: A New Computational Intelligent Approach", Springer-Verlag, 2002.

[18] P. K.Mishra, M. Bhusry, „Artificial Immune System: State of the Art Approach", International Journal of Computer Applications (0975 – 8887) Volume 120 – No.20, Iunie 2015.
[19] Fen Gu. Theoretical and Empirical extensions of the Dendritic Cell Algorithm. PhD thesis, Department of Computer Science, University College London, 2011.
[20] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer", In Proceedings of the IEEE Symposium on Research Security and Privacy, pages 202-212, 1994.
[21] L. N. de Castro and F. J. Von Zuben, "Learning and optimisation using the clonalslection principle", IEEE Transactions on Evolutionary Computation, 6(3):239-251, 2002.
[22] Leandro N. de Castro and Fernando J. Von Zuben. aiNet: An artificial immune network for data analysis. In Hussein A. Abbass, Ruhul A. Sarker, and Charles S. Newton, editors, Data Mining: A Heuristic Approach, chapter 12, pages 231-259. Idea Group Publishing, 2001.
[23] J. Greensmith. The Dendritic Cell Algorithm. PhD thesis, Department of Computer Science, University of Nottingham, 2007.
[24] J. Twycross. Integrated innate and adaptive artificial immune systems applied to process anomaly detection. PhD thesis, Department of Computer Science, University of Nottingham, 2007.
[25] Xin-She Yang, "Random walks and lévy flights", In Nature-Inspired Metaheuristic Algorithms Second Edition, pages 11–20. Luniver Press, 2010.
[26] P. Garcia- Teodoroa, J. Diaz-Verdejoa, G. Macia-Fernandez, E. Vazquez, "Anomaly based network intrusion detection; technique, systems and challenges", , Compuers and Security 28, 18–28, 2009.
[27] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications, pages 1-6. IEEE, 2009.
[28] K. Costa D. Rodrigues J. Papa R. Nakamura, L. Pereira and X. S. Yang,"BBA: a binary bat algorithm for feature selection", In Proceedings of the 25th Conference on Graphics, Patterns and Images (SIBGRAPI '12)s, pages 291–297, 2012.
[29] E. Convey. Porn sneaks way back on web, 1996.
[30] Nikita Spirin and Jiawei Han. Survey on web spam detection: principles and algorithms.SIGKDD Explorations, 13(2):50-64, 2011.
[31] H. Wahsheh, I. Abu Doush, M. Al-Kabi, I. Alsmadi, and E. Al-Shawakfa. Using machinelearning algorithms to detect content-based arabic web spam. International Journal of Information Assurance and Security (JIAS), 7(1):14-24, 2012.
[32] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, andIan H. Witten. The weka data mining software: an update. SIGKDD Explor. Newsl., 11:10-18, 2009
[33] R. M. Silva, T. A. Almeida, and A. Yamakami. Machine learning methods for spamdexing detection. International Journal of Information Security Science, 2(3):1-22, 2013