National University of Science and Technology
POLITEHNICA Bucharest

PhD Thesis

in Computer Science, Information Technology and System Engineering

Architecture of Distributed Reputation System for Internet Domains

Arhitectură unui sistem distribuit de reputație pentru domeniile Internet

presented by

Drd.Inf. Cristian Alexandru GHEORGHIȚĂ

supervised by

Prof.dr.ing. Florin POP

2024
Bucharest, Romania

Contents

1 | Introducere

În era digitală contemporană, internetul a devenit o parte integrantă a vieții cotidiene, facilitând comunicarea, comerțul și accesul la informații la nivel global. Totuși, această conectivitate răspândită prezintă și provocări semnificative, în special în domeniul securității cibernetice. Una dintre problemele presante este deturnarea și utilizarea abuzivă a numelor de domenii, care poate conduce la consecințe sociale și financiare grave atât pentru organizații, cât și pentru indivizi. Actorii malițioși exploatează domenii compromise pentru a distribui malware, a desfășura atacuri de tip phishing și a fura informații sensibile, precum CNP-uri, detalii ale conturilor bancare, informații despre carduri de credit și date personale. Astfel de activități nu doar că pun în pericol securitatea și confidențialitatea utilizatorilor, ci afectează și reputația domeniilor legitime implicate în aceste incidente [1].

Sistemele existente de reputație a domeniilor, inclusiv Notos [1], Exposure [2], Kopis [3] și listele publice negre și albe, precum Black Mirror [2] și Pi-hole BL [3], au fost dezvoltate pentru a identifica și gestiona credibilitatea domeniilor. Aceste sisteme utilizează diverse tehnici, cum ar fi monitorizarea în timp real, analiza caracteristicilor domeniilor și evaluarea reputației adreselor IP pentru a detecta și bloca domeniile malițioase. Deși aceste soluții au contribuit semnificativ la îmbunătățirea securității pe internet, ele se confruntă frecvent cu limitări legate de resurse, scalabilitate și capacitatea de a procesa eficient volume mari de date. Arhitecturile centralizate se luptă cu cerințele computaționale necesare pentru o analiză completă în timp real, ceea ce duce la întârzieri în detectarea amenințărilor și la o vulnerabilitate crescută la atacurile cibernetice în evoluție.

Pentru a aborda aceste provocări, această teză propune un sistem de reputație a domeniilor distribuit, inovator, care valorifică resursele computaționale neutilizate ale nodurilor voluntare, în special telefoanele mobile aflate în stare inactivă în timpul încărcării pe timpul nopții. Inspirat de proiecte precum Folding@Home [4], care utilizează calculul distribuit pentru cercetări științifice complexe, această abordare își propune să exploateze puterea de procesare colectivă a unei rețele globale de dispozitive. Prin distribuirea sarcinilor de lucru între numeroase noduri voluntare, sistemul urmărește să depășească

[1] `https://owasp.org/`, [Accessed on 11 October 2023]

[2] `https://github.com/T145/black-mirror`, [Accessed on 11 March 2023]

[3] `https://github.com/Pyenb/Pi-hole-blocklist`, [Accessed on 25 November 2023]

limitările procesării centralizate, îmbunătățind scalabilitatea, eficiența și promptitudinea evaluărilor de reputație a domeniilor.

1.1 Caracteristici principale ale sistemului propus

Caracteristicile cheie ale sistemului distribuit de reputație a domeniilor propus includ:

- **Arhitectură de calcul distribuit:** Utilizarea unei rețele de dispozitive mobile voluntare pentru a efectua sarcini de procesare a datelor la scară largă, sporind astfel capacitatea de calcul și reducând sarcina pe serverele centralizate.
- **Valorificarea Cloud Computing-ului Mobil (MCC):** Integrarea tehnologiilor web avansate și a principiilor MCC [5] pentru a facilita comunicarea și procesarea eficientă în rețeaua distribuită.
- **Monitorizare și analiză în timp real:** Implementarea unor algoritmi capabili să analizeze reputația domeniilor în timp real, permițând detectarea promptă a activităților malițioase.
- **Anonimizarea datelor și protecția confidențialității:** Asigurarea confidențialității utilizatorilor prin tehnici de anonimizare a datelor și protocoale sigure, abordând astfel preocupările legate de securitatea datelor și încrederea participanților.
- **Scalabilitate și optimizarea resurselor:** Proiectarea sistemului astfel încât să se extindă orizontal prin adăugarea mai multor noduri voluntare, optimizând utilizarea resurselor fără costuri suplimentare semnificative.

1.2 Structura cercetării

Cercetarea este structurată pentru a explora și aborda sistematic diferitele aspecte ale dezvoltării sistemului propus. Aceasta începe cu o revizuire cuprinzătoare a sistemelor existente de reputație a domeniilor și a amenințărilor de securitate cibernetică, urmată de dezvoltarea unor algoritmi avansați pentru evaluarea reputației domeniilor. Arhitectura sistemului distribuit este apoi proiectată, încorporând principii cheie și valorificând tehnologii moderne. Este realizată o implementare de tip proof-of-concept, evaluată pe baza datelor din lumea reală, concentrându-se în mod special pe baza de date a domeniilor .ro. Teza se încheie cu o analiză a rezultatelor, o discuție asupra provocărilor și limitărilor, și recomandări pentru cercetări viitoare.

1.3 Metodologie și obiective

Metodologia utilizată în această cercetare include o combinație de revizuire a literaturii, dezvoltare de algoritmi, proiectare de sistem, implementare și evaluare empirică. Obiec-

tivele principale ale acestei teze sunt:

- **Analiza sistemelor existente de reputație a domeniilor și a amenințărilor de securitate cibernetică:** Înțelegerea peisajului actual, identificarea limitărilor și stabilirea necesității unei soluții îmbunătățite.
- **Dezvoltarea de algoritmi avansați pentru stabilirea reputației domeniilor:** Crearea unor metode eficiente pentru evaluarea credibilității domeniilor pe baza diferitelor caracteristici și comportamente.
- **Proiectarea unei arhitecturi inovatoare pentru un sistem distribuit de reputație a domeniilor:** Conceptualizarea unui sistem scalabil și eficient care valorifică dispozitivele mobile voluntare.
- **Dezvoltarea și demonstrarea unei implementări proof-of-concept:** Validarea sistemului propus prin implementare practică și testare cu date reale.
- **Evaluarea sistemului:** Evaluarea sustenabilității soluției în comparație cu sistemele existente și a contribuției sale potențiale la securitatea cibernetică.

1.4 Structura tezei

Teza este organizată în următoarele capitole:

- **Introducere:** Prezintă contextul, semnificația cercetării și conturează obiectivele și structura tezei.
- **Analiză critică a lucrărilor anterioare:** Această secțiune examinează critic sistemele existente de reputație a domeniilor, inclusiv Notos, Exposure și Kopis, alături de listele negre și albe deschise utilizate frecvent. Fiecare dintre aceste soluții utilizează metodologii distincte pentru a evalua gradul de încredere al domeniilor. Această analiză subliniază necesitatea unei soluții mai adaptive și cuprinzătoare pentru reputația domeniilor. Arhitectura propusă integrează învățarea automată, analitica în timp real și algoritmi adaptivi pentru a depăși aceste limitări. Prin combinarea punctelor forte ale sistemelor existente și abordarea limitărilor lor, cadrul propus își propune să ofere o abordare scalabilă, robustă și proactivă a evaluării reputației domeniilor, oferind părților interesate instrumente mai bune pentru a mitiga amenințările cibernetice în mod eficient.
- **Amenințări de securitate cibernetică împotriva numelor de domenii:** Această secțiune explorează principalele amenințări care afectează reputația domeniilor, inclusiv ransomware-ul, care utilizează domenii compromise pentru livrarea de sarcini malițioase [6]; Fast Flux DNS, folosit pentru a ascunde infrastructura malițioasă prin schimbări rapide ale adreselor IP [7]; și abuzul de certificate SSL, unde atacatorii exploatează certificate legitime pentru a stabili încrederea în timp ce desfășoară activități frauduloase [8]. Pe baza acestor amenințări, este dezvoltat un model de amenințare cuprinzător, care descrie principalele vectori de atac, precum

phishing-ul, botnet-ul de comandă și control și furtul de credențiale. Acest model identifică vulnerabilitățile critice și informează strategiile de consolidare a sistemelor de reputație a domeniilor pentru a atenua riscurile cibernetice în evoluție.

- **Algoritmi propuși pentru stabilirea reputației domeniilor:** Acest capitol oferă o explorare cuprinzătoare a caracteristicilor evaluabile ale domeniilor în Secțiunea 4.1, dezvoltarea algoritmilor de reputație în Secțiunea 4.2 și implementarea mecanismelor de reevaluare a domeniilor în Secțiunea 4.3. Aceste componente stabilesc împreună cadrul fundamental pentru arhitectura propusă, permițând o abordare sistematică a evaluării reputației domeniilor și asigurând integritatea scorurilor reevaluate în timp.
- **Arhitectura unui sistem distribuit de reputație pentru domeniile internetului:** Această secțiune oferă o prezentare cuprinzătoare a arhitecturii propuse, detaliind principiile de proiectare fundamentale în Secțiunea 5.1 și componentele cheie care constituie sistemul în Secțiunea 5.2. Este analizată pe larg metodologia utilizată pentru colectarea datelor, cu accent pe selectarea și procesarea caracteristicilor domeniilor pentru a asigura o analiză robustă. În plus, în Secțiunea 5.5 este explorată integrarea Cloud Computing-ului Mobil (MCC) pentru a îmbunătăți scalabilitatea și eficiența, valorificând nodurile distribuite pentru sarcini computaționale. Discuția evidențiază modul în care aceste elemente se sincronizează pentru a crea un sistem rezistent și adaptiv pentru evaluarea precisă a reputației domeniilor internetului.
- **Evaluarea arhitecturii propuse:** Această secțiune prezintă o analiză detaliată a fezabilității arhitecturii, începând cu obiectivele de proiectare și cerințele de sistem în Secțiunea 6.1 care au ghidat dezvoltarea acesteia. Sunt prezentate stiva tehnologică în Secțiunea 6.2 și fluxul de implementare în Secțiunea 6.3, oferind informații despre pașii parcurși pentru a obține funcționalitatea dorită. Discuția abordează, de asemenea, provocările întâmpinate în timpul implementării proof-of-concept și strategiile folosite pentru a le depăși. În plus, sunt evidențiate caracteristicile cheie demonstrate, subliniind capacitatea sistemului de a-și îndeplini obiectivele și de a valida eficiența acestuia în contextul cadrului propus de reputație a domeniilor.
- **Studiu de caz asupra domeniilor de internet .ro:** Aplică sistemul la baza de date a domeniilor .ro, prezintă particularitățile întâmpinate în implementare referitoare la accesul la date mai detaliate despre domenii și discută provocările și limitările.
- **Concluzii:** Rezumă rezultatele cercetării, evaluează impactul arhitecturii distribuite și oferă recomandări pentru lucrări viitoare.

Această cercetare își propune să contribuie la domeniul securității cibernetice prin introducerea unei soluții noi pentru îmbunătățirea sistemelor de reputație a domeniilor. Prin valorificarea dispozitivelor mobile inactive într-un cadru de calcul distribuit, sistemul propus abordează limitările critice ale soluțiilor existente legate de scalabilitate și constrângeri de resurse. Abordarea promovează utilizarea eficientă a resurselor existente,

reduce dependența de infrastructura centralizată și încurajează implicarea comunității în eforturile de securitate cibernetică.

Beneficiile potențiale includ:

- **Detectarea îmbunătățită a domeniilor malițioase:** Capacitatea de calcul sporită permite o analiză mai cuprinzătoare și în timp util, conducând la o detectare mai rapidă a amenințărilor.
- **Scalabilitate și flexibilitate:** Sistemul se poate adapta la creșterea volumelor de date și la amenințările cibernetice în evoluție fără costuri suplimentare semnificative.
- **Împuternicirea și conștientizarea comunității:** Implicarea voluntarilor crește conștientizarea cu privire la problemele de securitate cibernetică și le oferă indivizilor posibilitatea de a contribui la mecanismele colective de apărare.
- **Soluție sustenabilă și rentabilă:** Maximizarea utilizării dispozitivelor existente în perioadele de inactivitate promovează practici de calcul sustenabile.

2 | Analiză critică a lucrărilor anterioare

Pentru organizații și indivizi deopotrivă, deturnarea numelor de domenii reprezintă amenințări semnificative sociale[1] și financiare [9]. Această activitate malițioasă poate duce la acces neautorizat la informații sensibile prin redirecționarea utilizatorilor către site-uri frauduloase sau prin distribuirea de malware. Atacatorii exploatează domeniile deturnate pentru a disemina software malițios care infectează dispozitivele, oferindu-le acces la date valoroase și private, precum numere de securitate socială, detalii ale conturilor bancare, informații despre carduri de credit, fotografii personale și alte bunuri confidențiale. Reputația domeniului compromis suferă ca rezultat, ceea ce duce la pierderea încrederii utilizatorilor și la posibile repercusiuni legale și financiare pentru părțile afectate [2].

Având în vedere aceste riscuri considerabile, este esențial să existe mecanisme eficiente pentru a identifica domeniile cu reputație rea, în vederea prevenirii infectării sistemelor și protejării informațiilor sensibile. De-a lungul anilor, au fost dezvoltate sisteme specializate pentru a oferi informații cuprinzătoare despre reputația unui domeniu, pe baza mai multor factori. Aceste sisteme utilizează diverse metodologii, inclusiv: Monitorizare în timp real, Liste albe și negre și Reputația IP-urilor. Acestea includ, dar nu se limitează la:

- Liste albe și negre accesibile publicului;
- Notos;
- Exposure;
- Kopis.

În acest capitol, am realizat o revizuire cuprinzătoare a sistemelor de reputație a domeniilor și resurselor existente, și anume Listele Negre și Albe deschise, Notos, Exposure și Kopis. Aceste sisteme reprezintă eforturi semnificative în lupta continuă împotriva domeniilor malițioase și a amenințărilor cibernetice. Prin analiza metodologiilor, punctelor forte și limitărilor lor, obținem informații valoroase despre stadiul actual al analizei reputației domeniilor și identificăm domeniile în care sunt necesare îmbunătățiri.

[1] `https://itp.cdn.icann.org/en/files/security-and-stability-advisory-committee-ssac\-reports/hijacking-report-12-07-2005-en.pdf`, [Accessed on 25 March 2023]

[2] `https://owasp.org/`, [Accessed on 11 October 2023]

3 | Amenințări de securitate cibernetică împotriva numelor de domenii

În peisajul digital modern, numele de domenii nu sunt doar adrese de internet; ele sunt componente fundamentale ale infrastructurii globale online care permit comunicarea, comerțul și schimbul de informații. Sistemul de Nume de Domenii (DNS) servește drept cartea de telefon a internetului, traducând numele de domenii ușor de citit de către oameni în adrese IP pe care computerele le folosesc pentru a se identifica reciproc în rețea [10]. Totuși, acest rol critic face, de asemenea, ca numele de domenii și infrastructura DNS să fie ținte atractive pentru infractorii cibernetici care doresc să exploateze vulnerabilitățile în scopuri malițioase. Amenințările de securitate cibernetică legate de domenii au devenit din ce în ce mai sofisticate și răspândite, reprezentând riscuri semnificative pentru indivizi, organizații și chiar pentru securitatea națională.

Atacatorii cibernetici folosesc numele de domenii și mecanismele DNS pentru a desfășura o gamă largă de activități dăunătoare, inclusiv phishing, distribuirea de malware, comandă și control (C2) al botnet-urilor, exfiltrarea de date și atacuri de tip denial-of-service [11]. Printre aceste amenințări, utilizarea algoritmilor de generare a domeniilor (DGAs) a devenit o tactică deosebit de provocatoare. DGAs permit malware-ului să genereze algoritmic un număr mare de nume de domenii care pot fi folosite pentru a stabili comunicarea cu serverele C2 [12]. Această tehnică permite atacatorilor să mențină controlul asupra sistemelor infectate, în timp ce evită detectarea și eforturile de eliminare, deoarece măsurile de securitate întâmpină dificultăți în a bloca sau monitoriza lista mereu în schimbare de domenii.

Prin analizarea acestor amenințări, dorim să evidențiem slăbiciunile pe care atacatorii le exploatează și să subliniem importanța unor sisteme avansate de reputație a domeniilor capabile de detectare în timp real a domeniilor malițioase. Cunoștințele obținute din această explorare servesc ca fundație pentru capitolele ulterioare, în care propunem algoritmi inovatori și o arhitectură distribuită concepută pentru a îmbunătăți detectarea și prevenirea acestor amenințări.

3.1 Modelul de amenințare

Un model de amenințare este o reprezentare structurată a potențialelor amenințări de securitate la adresa unui sistem, care ajută la înțelegerea obiectivelor atacatorului, a vulnerabilităților pe care le-ar putea exploata și a impactului acțiunilor lor. În contextul securității cibernetice legate de numele de domenii, modelul de amenințare se concentrează pe modul în care actorii malițioși pot exploata Sistemul de Nume de Domenii (DNS) și numele de domenii pentru a desfășura activități dăunătoare. Aceasta include manipularea înregistrărilor de domenii, a înregistrărilor DNS și valorificarea vulnerabilităților legate de domenii pentru a compromite sisteme, a fura date sau a perturba serviciile. Această secțiune descrie modelul de amenințare prin identificarea activelor aflate în pericol, a potențialilor adversari, a vectorilor de atac și a impacturilor posibile ale acestor amenințări.

3.1.1 Active

În contextul securității cibernetice legate de numele de domenii, mai multe active critice sunt vulnerabile la exploatarea de către actorii malițioși. Acestea sunt următoarele:

- Numele de domenii și infrastructura DNS;
- Încrederea utilizatorilor și reputația;
- Informații sensibile;
- Resursele rețelei;
- Operațiunile de afaceri.

3.1.2 Adversari potențiali

Peisajul amenințărilor pentru securitatea cibernetică legată de numele de domenii implică o gamă diversă de adversari, fiecare cu motivații și metode distincte. Infractorii cibernetici sunt printre cei mai răspândiți adversari. Acești indivizi sau grupuri organizate sunt motivați în principal de câștiguri financiare. Ei se angajează în activități precum furtul de date sensibile, fraude, distribuirea de malware și orchestrarea de atacuri de tip ransomware [13]. Infractorii cibernetici operează adesea la nivel transnațional, folosind tehnici sofisticate pentru a exploata vulnerabilitățile din numele de domenii și infrastructura DNS, și pot vinde datele furate sau accesul pe darkweb [14].

Principalii adversari potențiali sunt următorii:

- Hacktiviști;
- Entități sponsorizate de stat;

- Insideri;
- Script Kiddies;
- Competitori.

3.2 Vectori de atac

Această secțiune revizuiește vectorii semnificativi de atac care amenință securitatea domeniilor, concentrându-se pe strategii care exploatează caracteristicile domeniilor, mai degrabă decât vulnerabilitățile specifice DNS. Aceste metode subliniază necesitatea unor algoritmi sofisticați de reputație pentru a atenua riscurile legate de domenii. Metodele discutate sunt următoarele:

- Deturnarea domeniilor;
- Phishing și Typosquatting;
- Fast Flux DNS;
- Distribuirea de malware prin domenii compromise;
- Algoritmi de generare a domeniilor;
- Comandă și control (C2) prin DNS.

3.3 Concluzii

În această secțiune, am descris principalele metode prin care un domeniu poate fi compromis. Prin sintetizarea și analizarea acestor tehnici, putem construi o înțelegere mai cuprinzătoare a caracteristicilor și comportamentelor manifestate de domeniile compromise. Această înțelegere îmbunătățită servește ca fundație pentru rafinarea și optimizarea algoritmului, în vederea îmbunătățirii acurateței și fiabilității detectării compromisului domeniilor.

4 | Algoritmi propuși pentru stabilirea reputației domeniilor

În acest capitol vom crea un model pentru un algoritm de reputație a domeniilor prin identificarea caracteristicilor evaluabile ale numelor de domenii, utilizând clasificări precum: caracteristici sintactice, caracteristici de rețea și configurații de zonă, și vom stabili o metodologie de scorare pentru a obține o rată mare de pozitive adevărate și, de asemenea, pentru a obține o rată de falsuri pozitive cât mai mică posibil. În final, vom aborda partea inovatoare a algoritmului nostru, reprezentată de abilitatea de a reevalua un domeniu deja procesat pentru a ne asigura că reputația este menținută sau, în unele cazuri, că reputația a fost curățată prin reutilizarea numelui de domeniu de către un actor legitim.

4.1 Colectarea și verificarea datelor

În construirea unui sistem robust de reputație a domeniilor, fundamentul constă în colectarea cuprinzătoare și verificarea meticuloasă a datelor referitoare la numele de domenii. Aceste date servesc drept intrare critică pentru algoritmul nostru de reputație, influențând capacitatea acestuia de a evalua cu precizie gradul de încredere al domeniilor. Având în vedere numărul vast de domenii și natura dinamică a internetului, colectarea de date relevante și de înaltă calitate reprezintă atât o provocare semnificativă, cât și o necesitate vitală.

Pentru o colectare mai eficientă a datelor, similar cu Antonakis et al. [1], am împărțit caracteristicile domeniului în trei categorii: caracteristici de domeniu, caracteristici DNS și caracteristici de rețea.

- **Caracteristici de domeniu**:
  Această categorie include:
  - Vârsta domeniului,
  - Durata înregistrării,
  - Detalii despre registrant,
  - Expirarea domeniului,
  - Analiza lexicală,

  - Typosquatting,
  - Lungimea numelui de domeniu,
  - Includerea în liste negre,
  - Proprietarii anteriori,
  - Evaluări de la servicii de reputație terțe,
  - Relevanța și calitatea conținutului,
  - Rapoarte de spam de email,
  - Domenii generate algoritmic,
  - Nume de domenii care conțin entropie ridicată.

- **Caracteristici DNS**:
  Această categorie include probleme legate de configurația DNS, cum ar fi:
  - Tehnici Fast Flux,
  - Utilizarea neobișnuită a subdomeniilor,
  - Adoptarea DNSSEC,
  - Înregistrări de autentificare pentru email,
  - Servere deschise de relay pentru email.

- **Caracteristici de rețea**:
  Această categorie include probleme legate de configurația rețelei și asocieri cu IP-uri sau rețele malițioase, cum ar fi:
  - Adresa IP,
  - Configurarea SSL,
  - Securitatea infrastructurii de domeniu (tehnologii învechite, protocoale de securitate vulnerabile),
  - Redirecționări către domenii cunoscute ca fiind malițioase.

Caracteristicile de domeniu din prima categorie sunt obținute utilizând o combinație de surse de date și tehnici analitice. Această secțiune descrie metodele utilizate pentru a colecta și analiza aceste caracteristici, asigurând că datele colectate sunt precise, actualizate și relevante pentru algoritmul de reputație.

Principalele surse de date sunt următoarele:

- **WHOIS:**
  Un protocol de interogare și răspuns utilizat pe scară largă pentru obținerea de informații despre resursele Internetului, inclusiv numele de domenii, blocuri de adrese IP și numerele sistemelor autonome (ASNs);

- **Instrument de detectare DGA:**
  Un instrument bazat pe algoritmul de identificare DGA oferit de proiectul Safing Portmaster[1];

[1] `https://github.com/safing/portmaster/tree/v0.6.4/detection/dga`, [Accessed on 13 June 2023]

- **Instrument de typosquatting:**
  Un instrument dezvoltat care combină capacitatea URLInsane[2] de a identifica numele de domenii typosquatting, pe baza unui domeniu legitim, dar implementat pentru a funcționa recursiv pentru a identifica dacă un anumit nume de domeniu este un domeniu typosquatted;
- **Inspectarea caracteristicilor DNS:**
  Prin inspectarea caracteristicilor DNS, putem extrage informații despre configurația numelui de domeniu și potențialele sale vulnerabilități.

Ca o sumarizare a caracteristicilor de domeniu care vor fi incluse în algoritmul nostru, am creat Tabelul 1.

4.2 Algoritmul de reputație

Am ales să implementăm algoritmul de scor de reputație utilizând învățarea automată și, mai specific, cu ajutorul algoritmului XGBoost [15], mai exact varianta de regresie. Aceasta permite sistemului să se adapteze în timp, pe măsură ce sunt validate mai multe domenii. Rezultatele noilor domenii validate vor fi introduse ca feedback într-un algoritm de antrenare care va genera un model nou și mai precis.

4.2.1 Algoritmi de învățare automată

XGBoost, sau eXtreme Gradient Boosting, este o variantă avansată a Gradient Boosting Machines, inițial introdusă de J.H. Friedman [16]. Gradient Boosting construiește modele secvențial, fiecare model corectând erorile predecesorilor săi prin minimizarea unei funcții de pierdere cu ajutorul unor modele aliniate cu gradientul negativ al pierderii. XGBoost îmbunătățește acest cadru prin optimizări adaptate pentru o eficiență ridicată, scalabilitate și performanță, în special pe date la scară largă și cu dimensiuni ridicate.

O caracteristică distinctivă a XGBoost este funcția sa obiectivă regularizată, care combină o funcție de pierdere cu un termen de regularizare pentru a penaliza modelele excesiv de complexe, cum ar fi arborii de decizie mari. Această regularizare explicită atenuează supraînvățarea, echilibrând precizia și simplitatea modelului. În plus, XGBoost introduce un algoritm bazat pe histograme pentru divizarea caracteristicilor, care discretizează caracteristicile continue în bini. Această abordare reduce costurile computaționale menținând în același timp o precizie ridicată, permițând construirea rapidă a arborilor de decizie.

Un alt avantaj critic este gestionarea nativă a valorilor lipsă de către XGBoost. În timpul antrenării, acesta învață automat divizări optime pentru datele lipsă, fără a necesita pre-

[2] `https://github.com/ziazon/urlinsane`, [Accessed on 13 June 2023]

Feature	Data Type	Description
Domain Age	Categorical	Age of the domain in years (<1 year, 1 - 5, 5 - 10, 10 - 20, >20)
Domain Expiration (years)	Numeric	Days until domain expiration
Domain Length	Numeric	Number of characters in the domain name
DGA Suspect	Boolean	1 if suspected to use Domain Generation Algorithm, else 0
Typosquatting Suspect	Binary	1 if suspected typosquatting, else 0
DNS TTL (Time To Live)	Categorical	One of (<300, 300 - 1800, 1800 - 86400, 86400 - 604800, >604800)
DNSSEC	Boolean	1 if DNSSEC is enabled, else 0
Fast Flux Suspect	Boolean	1 if suspected of fast flux DNS technique, else 0
SSL	Boolean	1 if SSL is configured, else 0
SSL Certificate Issuer	Categorical	Categorical encoding of the SSL issuer
SSL Certificate Valid From	Numeric	Days since SSL certificate was issued
SSL Certificate Validity	Numeric	Duration in days that the SSL certificate is valid
Registrar	Categorical	Categorical encoding of the domain registrar
IP Address	Numeric	Numerical representation of the IP address
AS Number	Numeric	Autonomous System Number
Is Blacklisted	Boolean	1 if the domain is blacklisted, else 0
Is IP Blacklisted	Boolean	1 if associated IP is blacklisted, else 0
TLS Version	Categorical	Categorical encoding of the TLS version
Strict-Transport-Security	Boolean	1 if HSTS is enabled, else 0
Content-Security-Policy	Boolean	1 if CSP is implemented, else 0
Has Email Server	Boolean	1 if the domain has an email server configured, else 0
Is Email Server Blacklisted	Boolean	1 if the email server is blacklisted, else 0
Email Has DMARC	Boolean	1 if DMARC is configured, else 0
Email Has SPF	Boolean	1 if SPF is configured, else 0
Email Has DKIM	Boolean	1 if DKIM is configured, else 0
Email Server Is Open Relay	Boolean	1 if the email server is an open relay, else 0

Table 1: Features Used for Domain Reputation Assessment.

procesare sau imputare. Această capacitate este deosebit de valoroasă în procesul nostru de colectare a caracteristicilor de domeniu, unde golurile de date ar putea compromite integritatea modelului. Prin abordarea valorilor lipsă direct în timpul construcției arborelui, XGBoost asigură robustețea și fiabilitatea, minimizând efectele negative ale seturilor de date incomplete și îmbunătățind performanța generală a modelului predictiv.

4.2.2 Structura datelor de antrenament

După cum se arată în Tabelul 1, caracteristicile de domeniu utilizate în acest proces constau într-o combinație de valori numerice, booleene și categorice. Valorile categorice sunt în mod inerent mai complexe de procesat, deoarece reprezintă informații discrete, non-numerice, cum ar fi Numele Registratorului sau clasificările domeniilor. Aceste valori necesită tehnici de codificare pentru a le transforma în formate care pot fi interpretate de algoritm. O abordare comună este codificarea One-Hot [17], unde fiecare categorie este reprezentată ca o caracteristică binară separată. Alternativ, se poate utiliza codificarea etichetelor, atribuind o valoare numerică unică fiecărei categorii, deși aceasta poate introduce relații ordinale nedorite.

Registrar Name	**Encoded Representation (One-Hot)**
GoDaddy	[1, 0, 0, 0]
Namecheap	[0, 1, 0, 0]
Google Domains	[0, 0, 1, 0]
Romarg	[0, 0, 0, 1]

Table 2: One-Hot Encoded Representation of Registrars.

Conform documentației XGBoost, codificarea One-Hot este recomandată deoarece asigură absența relațiilor ordinale implicite, ceea ce este esențial atunci când nu există o ordine inerentă între categorii. Gestionarea corectă a acestor caracteristici asigură faptul că algoritmul poate procesa eficient diferite tipuri de date, îmbunătățindu-și astfel capacitatea de a produce scoruri de reputație precise și nepartinitoare.

4.2.3 Antrenarea modelului

Pentru antrenarea modelului de reputație a domeniilor, a fost pregătit un set de date în format CSV, care include atât domenii benigne, cât și malițioase, pentru a asigura diversitatea și capacitățile robuste de generalizare. Pentru domeniile benigne, a fost utilizat un subset din setul de date Alexa's Top 1 Million Domains [3], selectând primele 100.000 de domenii pentru o procesare eficientă. Pentru domeniile malițioase, a fost utilizată lista neagră Black Mirror [4], un depozit dinamic și frecvent actualizat. Aplicația personalizată de extracție a caracteristicilor domeniilor, implementată în Go, a facilitat extragerea caracteristicilor pentru 100.000 de domenii benigne și 100.000 de domenii potențial malițioase. Acest set de date echilibrat a oferit o bază solidă pentru antrenare, minimizând riscurile de supraînvățare.

Setul de date a fost structurat pentru a include o coloană suplimentară care indică originea domeniului—fie benign, fie din lista neagră. Ulterior, acesta a fost împărțit în

[3] `https://www.kaggle.com/datasets/cheedcheed/top1m`, [Accessed on 20 June 2023]
[4] `https://github.com/T145/black-mirror`, [Accessed on 11 March 2023]

subseturi de antrenament și de testare, permițând evaluarea performanței și testarea generalizării. Pentru a spori rigoarea evaluării, a fost utilizată validarea încrucișată, în special metodologia k-fold [18]. Prin împărțirea datelor în k subseturi și desemnarea iterativă a unui subset pentru testare, în timp ce restul erau folosite pentru antrenare, a fost obținută o evaluare cuprinzătoare a performanței, atenuând riscurile de supraînvățare. Rezultatele acestor iterații au fost agregate pentru o evaluare robustă a modelului.

Modelul finalizat a fost serializat într-un format portabil de fișier pentru distribuire către nodurile mobile de lucru din sistemul distribuit. Acest lucru a asigurat uniformitatea și compatibilitatea în diferite medii, permițând nodurilor de lucru să efectueze predicții sau clasificări eficiente. Prin descentralizarea modelului antrenat, sistemul a valorificat resursele distribuite ale rețelei pentru o analiză scalabilă și eficientă a domeniilor, asigurând o redundanță și o reziliență ridicate. Această abordare a subliniat adaptabilitatea și scalabilitatea sistemului, susținând diverse configurații hardware și optimizând capacitatea de calcul în cadrul rețelei de calcul distribuit.

4.2.4 Evaluarea modelului

Pentru a evalua performanța modelului inițial, un subset de domenii, care include atât exemple legitime, cât și malițioase, a fost exclus din procesul de antrenare și rezervat pentru scopuri de testare. Aceste domenii, denumite L1 și L2 pentru domeniile legitime și M1 și M2 pentru domeniile malițioase, au fost selectate cu atenție pentru a asigura o reprezentare echilibrată a diversității datelor.

Evaluarea a fost realizată folosind aplicația de lucru dezvoltată special pentru implementarea pe nodurile de calcul distribuit. Această aplicație a procesat domeniile de probă rezervate, permițându-ne să evaluăm capacitatea modelului de a clasifica date care nu au fost văzute anterior.

Analiza caracteristicilor extrase ale domeniilor evidențiază indicatori cheie ai domeniilor suspecte, vârsta domeniului fiind un factor semnificativ, deoarece domeniile mai noi sunt adesea legate de activități malițioase datorită naturii lor tranzitorii. Setările Time-to-Live (TTL) servesc, de asemenea, ca metrică de încredere, valorile neobișnuit de mici semnalând adesea un comportament malițios. În plus, statutul de includere în liste negre—fie pentru adrese IP, servere de email sau domeniul însuși—oferă dovezi substanțiale ale potențialelor amenințări. Deși certificatele SSL erau cândva un indicator de încredere, fiabilitatea lor a scăzut din cauza ușurinței de a obține certificate de scurtă durată, folosite frecvent de actorii malițioși. Aceste constatări subliniază importanța unei abordări multifacetate, care integrează multiple caracteristici ale domeniilor pentru a spori precizia detectării și a asigura o evaluare cuprinzătoare a gradului de încredere al domeniilor.

Rezultatele evaluării, rezumate în Tabelul 4, demonstrează capacitatea algoritmului de predicție de a clasifica domeniile pe baza scorurilor de reputație și a acurateței asoci-

Feature	**L1**	**L2**	**M1**	**M2**
Domain Age	>20	>20	<1 year	<1 year
Domain Expiration	917	1282	194	256
Domain Length	5	4	7	29
DGA Suspect	0	0	0	1
Typosquatting Suspect	0	0	0	0
DNS TTL	1800 - 86400	1800 - 86400	<= 300	<= 300
DNSSEC	1	0	0	0
Fast Flux Suspect	0	0	0	0
SSL	1	1	1	1
SSL Issuer	ISSUER1	ISSUER2	ISSUER3	ISSUER4
SSL Certificate Valid From	304	116	29	53
SSL Certificate Valid To	92	255	61	392
Registrar	REG1	REG2	REG3	REG3
IP Address	3231846509	783192862	3259220266	2890123802
AS Number	AS3233	AS47388	AS214231	AS13335
Is Blacklisted	0	0	1	1
Is IP Blacklisted	0	0	1	0
TLS Version	TLSv1.2	TLSv1.3	TLSv1.3	TLSv1.3
Strict-Transport-Security	0	0	0	0
Content-Security-Policy	0	0	0	0
Has Email Server	1	1	0	0
Is Email Server Blacklisted	0	0	0	0
Email has DMARC	0	1	0	0
Email has SPF	1	1	0	0
Email has DKIM	0	1	0	0
Email Server is Open Relay	0	0	0	0

Table 3: Feature Comparison Across Domains.

Domain	**Reputation Score**	**Accuracy**
L1	0.93	0.87
L2	0.89	0.91
M1	0.21	0.82
M2	0.15	0.84

Table 4: Domain Reputation Score and Accuracy.

ate. Scorurile ridicate de reputație (0,93 și 0,89) pentru domeniile legitime, cu valori ale acurateței de 0,87 și 0,91, indică o performanță puternică a modelului, deși cu o anumită variabilitate care sugerează îmbunătățiri potențiale în ponderarea caracteristicilor sau diversitatea setului de date. În schimb, scorurile scăzute de reputație (0,21 și 0,15)

pentru domeniile malițioase, însoțite de valori ale acurateței de 0,82 și 0,84, evidențiază eficiența modelului în identificarea amenințărilor, dar expun și provocări în distingerea cazurilor limită în care caracteristicile domeniilor malițioase și benigne se suprapun.

Caracteristici cheie, cum ar fi vârsta domeniului, TTL DNS, statutul de includere în liste negre și configurațiile de email, influențează puternic predicțiile modelului, subliniind relevanța acestora în evaluarea reputației domeniilor. Totuși, caracteristici precum atributele SSL și politicile de securitate prezintă o variabilitate limitată, sugerând oportunități de rafinare pentru a spori puterea de discriminare. Setul de date echilibrat de 100.000 de domenii benigne și 100.000 de domenii malițioase contribuie la generalizare, deși includerea unor tipuri suplimentare de domenii, a datelor DNS în timp real și a analiticii comportamentale ar putea optimiza și mai mult performanța. În general, modelul demonstrează capacități solide de clasificare, evidențiind în același timp direcții pentru îmbunătățiri viitoare care să abordeze limitările rămase și să sporească fiabilitatea.

4.3 Reevaluarea domeniului și efectul acesteia asupra scorului curent

Sistemul a fost conceput pentru a încorpora un mecanism de degradare a scorului de reputație pentru domeniile evaluate anterior. Această funcționalitate abordează două provocări critice: menținerea validității scorului de reputație în timp și prioritizarea domeniilor pentru reevaluare.

După ce un domeniu a fost evaluat, este rezonabil să presupunem că statutul său va rămâne relativ stabil imediat după evaluare, menținându-și astfel scorul de reputație. Această presupunere este valabilă în special pentru domeniile cu o istorie îndelungată și cu actualizări minime, deoarece stabilitatea lor are o greutate mai mare în comparație cu domeniile mai noi, care suferă schimbări frecvente. Acești factori sunt luați în considerare pentru a optimiza procesul de programare și pentru a îmbunătăți acuratețea metricilor de reputație.

Având în vedere aceste considerații, a fost dezvoltată o formulă pentru a introduce o "degradare" controlată în scorul de reputație al unui domeniu după evaluarea sa. Acest mecanism de degradare asigură că, pe măsură ce trece timpul, domeniile cu evaluări mai vechi sunt prioritizate în mod natural pentru reevaluare, menținând fiabilitatea și actualitatea metricilor sistemului.

$$\text{new\_score} = \text{old\_score} \cdot \left(1 - \frac{\lambda}{\text{domain\_age\_days} + \text{revalidation\_days}}\right)^{\text{days}}$$

Această versiune a formulei echilibrează dinamica degradării, asigurând corectitudinea între domenii de vârste și istorii de validare variate, menținând în același timp integritatea și fiabilitatea sistemului de scorare a reputației.

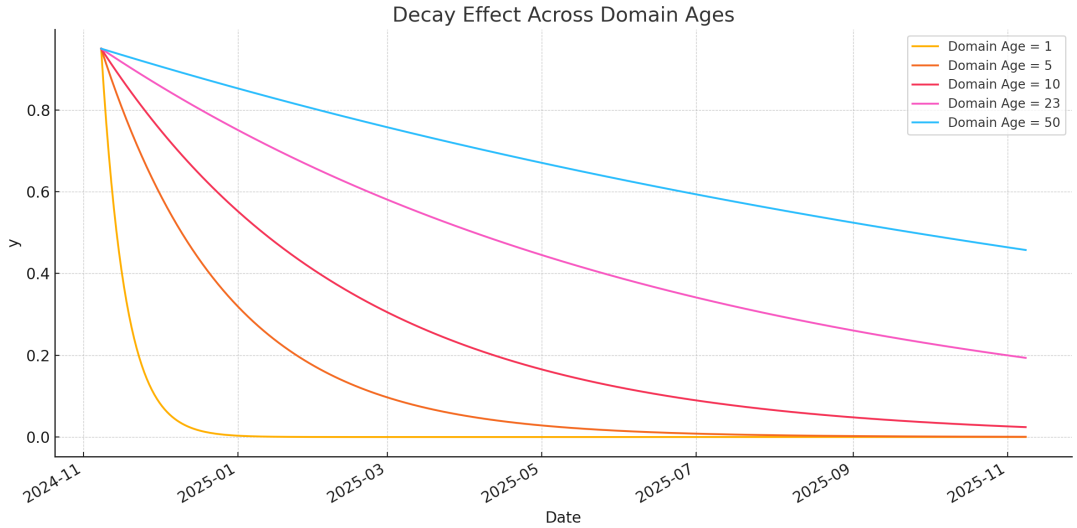

Figure 1: Decay Effects with Final Formula.

Implementarea acestei formule de degradare a scorului de reputație asigură că domeniile mai noi și mai vulnerabile sunt reevaluate mai frecvent decât domeniile mai vechi și mai bine stabilite. Rata cu care se înregistrează noi domenii [Monitoring the Initial DNS Behavior of Malicious Domains. Shuang Hao] face dificilă evaluarea și monitorizarea tuturor acestora. Considerăm că abordarea noastră îmbunătățește această problemă prin oferirea unei lățimi de bandă mai mari pentru validarea acestor domenii, în favoarea celor deja reputate.

4.4 Concluzii

Acest capitol a introdus un model cuprinzător pentru evaluarea reputației domeniilor prin identificarea caracteristicilor evaluabile ale acestora, stabilirea metodologiilor de scorare și integrarea unui mecanism inovator de reevaluare. Prin clasificarea caracteristicilor în grupuri sintactice, de rețea și de configurare a zonelor, modelul surprinde natura complexă a comportamentului și securității domeniilor. Validată printr-o analiză riguroasă a extragerii caracteristicilor, a codificării datelor și a evaluării cu învățare automată, abordarea a demonstrat o acuratețe și o fiabilitate ridicată în distingerea domeniilor legitime de cele malițioase. Abordând provocări precum înregistrarea rapidă a domeniilor, configurațiile DNS dinamice și practicile de securitate inconsistente, modelul oferă un cadru scalabil și adaptiv cu reevaluare în timp real pentru a asigura evaluări actualizate și corecte. Acest capitol fundamental pregătește calea pentru implementarea practică și rafinarea modelului, stabilind scena pentru arhitectura sistemului, evaluarea și aplicațiile sale în îmbunătățirea securității cibernetice și a încrederii.

5 | Arhitectura unui sistem distribuit de reputație pentru domeniile internetului

În peisajul în continuă schimbare al securității cibernetice, evaluarea precisă și eficientă a reputației domeniilor este esențială. Sistemele centralizate se confruntă adesea cu limitări în ceea ce privește scalabilitatea, latența și disponibilitatea resurselor atunci când analizează natura vastă și dinamică a internetului. Acest capitol introduce o arhitectură inovatoare distribuită pentru reputația domeniilor, care valorifică calculul distribuit și cloud computing-ul mobil (MCC) [5] pentru a îmbunătăți scalabilitatea, a reduce latența și a crește acuratețea. Prin descentralizarea sarcinilor de procesare într-o rețea de noduri voluntare, inclusiv dispozitive mobile, sistemul abordează provocările fundamentale în evaluarea reputației domeniilor.

Arhitectura propusă funcționează prin partiționarea seturilor mari de date și distribuirea sarcinilor computaționale, cum ar fi extracția caracteristicilor sau analiza preliminară, către noduri voluntare. Rezultatele sunt agregate de un server central, care rafinează scorurile de reputație și redistribuie sarcinile. Acest design pune accent pe scalabilitate, toleranță la erori și utilizarea eficientă a resurselor, inspirându-se din cadrele de calcul distribuit de succes, cum ar fi Folding@home. Beneficiile cheie includ:

- **Scalabilitate:** Sistemul poate gestiona volume de date în creștere prin recrutarea mai multor noduri voluntare, evitând blocajele procesării centralizate;
- **Reducerea latenței:** Distribuirea sarcinilor între mai multe noduri sau procesarea datelor mai aproape de sursa lor reduce timpul necesar pentru scorarea reputației;
- **Optimizarea resurselor:** Utilizarea dispozitivelor subutilizate maximizează utilizarea resurselor fără investiții suplimentare semnificative;
- **Implicarea comunității:** Implicarea voluntarilor promovează un sentiment de efort colectiv, încurajând securitatea cibernetică ca responsabilitate comună, similară cu Folding@home.

Prin abordarea provocărilor de scalabilitate inerente sistemelor centralizate, această abordare distribuită permite procesarea eficientă a seturilor mari de date, favorizând o soluție colaborativă și scalabilă pentru nevoile reale de securitate cibernetică.

5.1 Principii și presupuneri cheie

În proiectarea arhitecturii pentru un sistem distribuit de reputație a domeniilor, mai multe principii cheie și presupuneri fundamentale ghidează dezvoltarea și implementarea. Aceste principii asigură că sistemul este robust, scalabil, eficient și sigur, în timp ce presupunerile stabilesc contextul operațional în care funcționează sistemul. Această secțiune analizează aceste principii directoare și presupunerile care stau la baza arhitecturii sistemului.

Următoarele sunt principiile cheie care ne-au ghidat în proiectarea arhitecturii propuse:

- Scalabilitate;
- Fiabilitate;
- Eficiență;
- Securitate.

Proiectarea unui sistem distribuit de reputație a domeniilor este ghidată de mai multe presupuneri cheie, care stabilesc contextul de bază pentru implementarea și funcționarea sa. Aceste presupuneri abordează aspectele tehnice, operaționale și de reglementare care sunt esențiale pentru funcționalitatea și succesul sistemului. Următoarele sunt principalele presupuneri luate în considerare în proiectarea arhitecturii propuse:

- Participarea nodurilor;
- Conectivitatea rețelei;
- Disponibilitatea resurselor;
- Conformitatea cu confidențialitatea datelor.

Prin definirea explicită și abordarea acestor presupuneri, proiectarea sistemului asigură că riscurile potențiale sunt atenuate, parametrii operaționali sunt realiști, iar arhitectura se aliniază cu standardele tehnice și de reglementare. Aceste presupuneri fundamentale nu doar ghidează procesul de dezvoltare, ci servesc și ca un punct de referință pentru evaluarea fezabilității și scalabilității sistemului în implementări din lumea reală.

5.2 Componente Propuse ale Arhitecturii și Interacțiunile Lor

Arhitectura propusă este compusă din două componente principale:

- **Nexus:** Principalul server de control și comandă responsabil pentru:
  - Programarea domeniilor pentru validare.
  - Validarea răspunsurilor de la lucrătorii mobili.

  - Versiunarea și stocarea rezultatelor în baza de date.
  - Furnizarea de statistici și rapoarte pentru domeniile analizate.
  - Gestionarea interogărilor pentru domenii specifice.
- **BOINC:** [referință aici] Un program de planificare a procesării voluntare a muncii însărcinat cu:
  - Administrarea dispozitivelor voluntare conectate.
  - Alocarea lucrărilor pe care acestea sunt însărcinate să le efectueze.
  - Colectarea și gestionarea rezultatelor obținute de la lucrători.

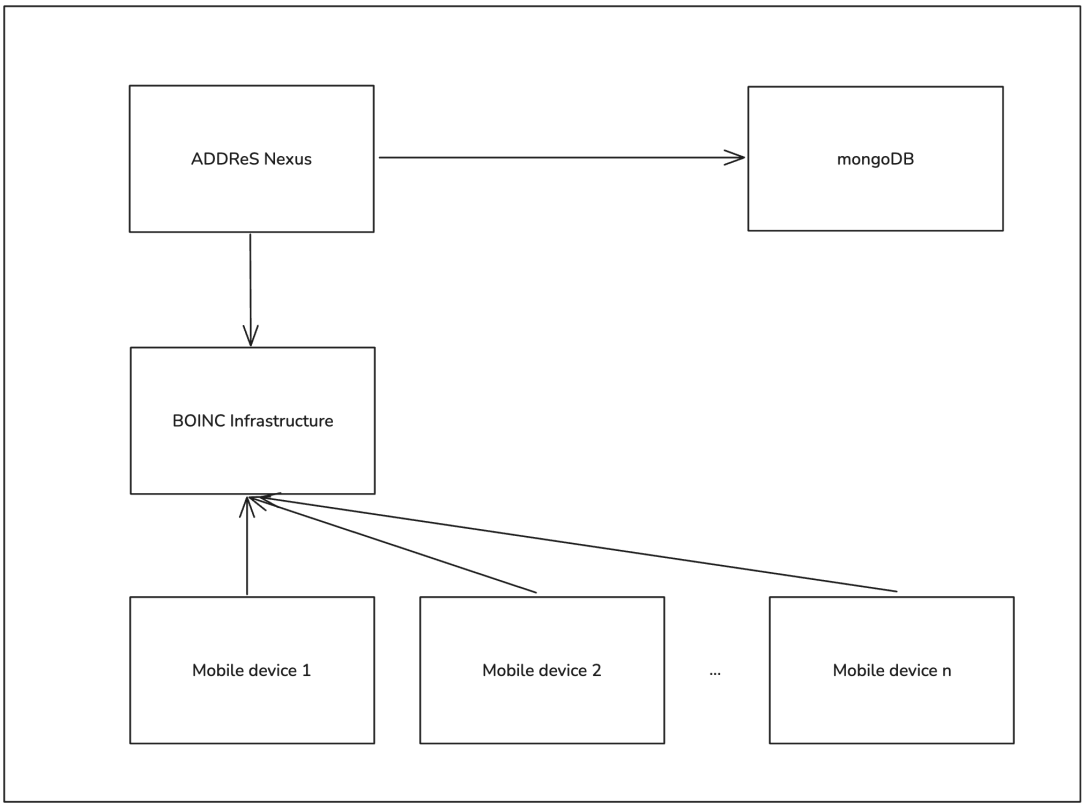

Figure 2: Componentele arhitecturii ADDReS.

5.3 Sistemul de Comandă și Control - Nexus

Chiar dacă majoritatea părților mobile ale sistemului sunt decentralizate, pentru a proteja confidențialitatea și securitatea sistemului, am preferat să păstrăm factorul decizional ca un serviciu închis. Astfel, mitigăm posibilele încercări de manipulare a rezultatelor evaluărilor domeniilor prin distribuția aleatoare a lucrărilor și consens asupra rezultatelor descoperite.

5.3.1 Serviciul Web

Serviciul web acționează ca o poartă de intrare în sistem prin furnizarea de instrumente care permit gestionarea acestuia. Utilizează un sistem de Management al Utilizatorilor care suportă diferite roluri bazate pe nivelul de acces al operatorilor. Permite operatorilor să programeze domenii specifice pentru validare și să interacționeze cu rezultatele obținute din evaluările domeniilor.

5.3.2 Programatorul de Evaluare a Domeniilor

Programatorul de domenii este responsabil de programarea domeniilor noi și vechi pentru evaluare. Folosim termenii "noi" și "vechi" deoarece sistemul este proiectat să reevalueze domenii care au fost evaluate anterior în funcție de nivelul de decădere a scorului. Cu cât prezintă un scor mai mare de decădere, cu atât este mai probabil ca domeniul să fie reevaluat. Această metodă permite sistemului să revizuiască domenii care sunt mai susceptibile să devină suspicioase și să realizeze monitorizarea aproape în timp real a domeniilor.

Programatorul începe prin verificarea resurselor disponibile ale sistemului și le aloca în mod egal între noile domenii și cele vechi. Dacă nu există domenii vechi de revalidat, resursele sunt realocate către coada de domenii noi. La început, echilibrul va fi înclinat către domeniile noi datorită revalidărilor mai puține sau deloc, dar pe măsură ce trece timpul și mai multe domenii sunt validate, echilibrul se va restabili și noile și vechile domenii vor primi lățime de bandă alocată.

```
for domain in existing_old_domains:
    domain_obj = new Domain(
        domain_name=domain.name,
        decay_value=domain.decay_value,  # decay_value > 0
        last_evaluated=domain.last_evaluated
    )
    old_pq.insert(domain_obj, priority=domain_obj.decay_value)

while True:
    R = get_total_resources()
    R_new = R * 0.5
    R_old = R - R_new  # Handle odd R

    for i from 1 to R_new:
        if new_queue is not empty:
            domain = new_queue.dequeue()
            schedule_evaluation(domain)
        else:
            R_old += (R_new - i + 1)
            break

```

```
    for i from 1 to R_old:
        if old_pq is not empty:
            domain = old_pq.extract_max()
            schedule_evaluation(domain)
        else:
            R_new += (R_old - i + 1)
            break

    new_domains = get_new_domains()
    for domain in new_domains:
        domain_obj = new Domain(
            domain_name=domain.name,
            decay_value=0,
            last_evaluated=null
        )
        new_queue.enqueue(domain_obj)
    wait_until_next_cycle()
```

Listing 5.1: Scheduling Algorithm for Domain Evaluation.

5.3.3 Distribuitorul de Evaluări ale Domeniilor

Distribuitorul de evaluări ale domeniilor este responsabil pentru distribuirea sarcinilor către nodurile lucrătoare voluntare într-un mod care să asigure securitatea rezultatelor. Fiind un serviciu deschis și permițând oricui să participe la procesul de evaluare a reputației unui domeniu, acest lucru creează oportunitatea ca actori rău intenționați să încerce să influențeze rezultatele pentru domeniile pe care doresc să le aprobe. Pentru a atenua această problemă, am conceput o formulă prin care sistemul determină numărul de noduri redundante ce trebuie selectate pentru o anumită sarcină. Abia după ce sistemul ajunge la consens, rezultatele sunt validate și stocate.

Având în vedere că numărul de noduri participante în sistem este variabil, am proiectat o formulă care oferă numărul recomandat de noduri pentru distribuirea sarcinilor.

Am conceput următoarea formulă:

$$\sum_{k=\lceil \frac{n}{2} \rceil}^{n} C_k^n (1-p)^k p^{n-k} \geq c$$

care permite calcularea numărului minim de noduri redundante n necesar pentru a atinge un nivel de încredere dorit C că majoritatea nodurilor care evaluează un domeniu sunt oneste, având în vedere o probabilitate de compromitere a unui nod p. Astfel, pentru o probabilitate de compromitere a unui nod de 0.1 (10%) și un nivel de încredere dorit de 0.95 (95%), vom avea nevoie de cel puțin 3 noduri.

5.3.4 Consensul Rezultatelor Evaluării

Când rezultatele finale ale nodurilor de lucru sunt recuperate, este important să se stabilească un consens între rezultate pentru a:

- Compensa variațiile obișnuite ale scorurilor cauzate de:
  - Diferențe în momentul colectării datelor: Nodurile pot avea perspective ușor diferite asupra datelor din cauza actualizărilor care au loc între evaluări.
  - Variații locale ale datelor: Nodurile pot avea acces la subseturi diferite de date, cum ar fi stările cache.
  - Latența rețelei: Întârzierile în recuperarea datelor pot afecta actualitatea datelor utilizate în evaluare.
- Compensa posibilitatea rezultatelor eronate generate de dispozitive malițioase.

Pentru a contracara aceste diferențe, am stabilit un mecanism de consens bazat pe aplicarea Intervalului Intercuartil (IQR)[19] asupra listei de scoruri generate de noduri. Am ales această metodă datorită capacității sale de a reduce efectiv influența valorilor extreme (outliers), lăsând să fie înregistrate doar scorurile legitime.

5.3.5 Aplicatorul de Decădere a Scorului Domeniilor

O componentă importantă al procesului de reevaluare este reprezentat de aplicatorul de decădere a scorului domeniilor (DSDA), care este programat să ruleze zilnic pentru a ajusta corespunzător factorul de decădere al domeniilor evaluate. Scorurile decăzute generate de Aplicatorul de Decădere a Scorului Domeniilor sunt utilizate de Planificatorul de Evaluare a Domeniilor pentru a prioritiza domeniile pentru reevaluare. Domeniile cu scoruri mai mici (indicând o decădere mai mare) primesc o prioritate mai ridicată, asigurând astfel alocarea resurselor către domeniile cu cea mai mare incertitudine.

```
DECAY_CONSTANT = 0.5
MIN_SCORE = 0.0
MAX_SCORE = 1.0

for domain in domain_repository.get_all_domains():
    # Calculate effective domain age
    effective_domain_age = max(domain.domain_age, 1)

    # Calculate decay denominator and decay factor
    decay_denominator = effective_domain_age + domain.
       revalidation_count
    decay_factor = 1 - (DECAY_CONSTANT / decay_denominator)

    # Ensure decay factor is between 0 and 1
```

```
    decay_factor = max(min(decay_factor, 1), 0)

    # Update domain score with decay factor
    decayed_score = domain.score * (decay_factor ** domain.
        days_since_validation)
    decayed_score = max(min(decayed_score, MAX_SCORE), MIN_SCORE)

    # Update the domain score and save changes
    domain.score = decayed_score
    domain_repository.update(domain)
```

Listing 5.2: Domain Score Decay Algorithm.

5.4 Sistemul de Stocare

Pentru aspectul de stocare al sistemului, am avut nevoie de o soluție care să permită inserarea rapidă a datelor și, de asemenea, să ofere o structură flexibilă pentru datele ingerate. Dintre multiplele soluții disponibile, am ales MongoDB[1] datorită performanțelor sale și capacității sale de scalare.

5.5 Rețea de Calcul Distribuit

Propunem un sistem inovator de evaluare a reputației domeniilor bazat pe o arhitectură de calcul distribuit. În mod specific, abordarea noastră valorifică resursele computaționale ale dispozitivelor mobile din cadrul unei rețele distribuite. Această arhitectură reduce dependența de resursele centralizate de calcul de înaltă performanță, distribuind sarcina de lucru pe o rețea extinsă de dispozitive voluntare. Prin utilizarea capacității colective de calcul a dispozitivelor mobile, sistemul propus are ca scop furnizarea unei soluții scalabile, eficiente din punct de vedere al costurilor și eficiente pentru evaluarea în timp real a reputației domeniilor.

Gestionarea unei astfel de arhitecturi distribuite necesită, de asemenea, o infrastructură scalabilă pentru gestionarea înregistrării dispozitivelor, alocarea sarcinilor și colectarea rezultatelor, precum și metode sigure pentru protejarea confidențialității datelor și prevenirea accesului neautorizat. În ciuda acestor provocări, beneficiile potențiale ale unui sistem distribuit, inclusiv eficiența costurilor, scalabilitatea și reziliența, îl fac o abordare promițătoare pentru sarcinile computaționale de mare anvergură, cum ar fi evaluarea reputației domeniilor.

Componentele principale ale sistemului sunt:

- Serverul Central;

[1] `https://www.mongodb.com/`, [Accessed on 02 October 2023]

- Nodurile de Lucru;
- Distribuția Sarcinilor;
- Mecanismele de Consens;
- Asimilarea Rezultatelor.

5.6 Concluzii

Rețeaua propusă de calcul distribuit abordează eficient limitările sistemelor centralizate în evaluarea reputației domeniilor, prin utilizarea nodurilor distribuite pentru a spori scalabilitatea, eficiența și rentabilitatea, reducând în același timp dependența de infrastructura de înaltă performanță.

6 | Evaluarea Arhitecturii Propuse

Acest capitol prezintă obiectivele, cerințele, alegerile tehnologice, fluxul de implementare, provocările întâmpinate și caracteristicile demonstrate cu succes printr-un Proof of Concept (PoC). Acesta oferă o bază concretă pentru validarea designului sistemului și deschide calea pentru dezvoltări și implementări ulterioare.

6.1 Obiectivele Designului

Prin implementarea sistemului de tip proof of concept, am avut ca scop validarea mai multor aspecte critice ale sistemului distribuit propus pentru evaluarea reputației domeniilor. Obiectivele specifice ale designului au fost:

- Implementarea infrastructurii pentru a evalua cerințele sistemului și documentarea procesului.
- Activarea extracției caracteristicilor domeniilor și a predicției prin învățare automată pe dispozitive Android, fără a modifica platforma de bază.
- Compilarea și implementarea wrapper-ului BOINC și a Aplicației Worker pe dispozitivele Android.
- Implementarea unui mecanism robust de validare pentru replicarea sarcinilor pe mai multe noduri, pentru a asigura securitatea rezultatelor.
- Integrarea rezultatelor validate în sistemul de management ADDReS Nexus.
- Demonstrarea funcționalității end-to-end a sistemului distribuit.
- Abordarea provocărilor tehnice și documentarea soluțiilor.
- Evaluarea performanței sistemului și a utilizării resurselor.

6.2 Tehnologii Utilizate

În această secțiune, prezentăm o examinare detaliată a tehnologiilor selectate pentru implementarea proof-of-concept-ului. De asemenea, elucidăm raționamentul din spatele fiecărei decizii, evidențiind modul în care aceste alegeri se aliniază cu obiectivele de design și cerințele operaționale ale sistemului.

- **ADDReS Nexus:**

  - **Componenta Web**: Oferă o interfață vizuală pentru interacțiunea cu sistemul. Aceasta include mai multe niveluri de acces, permițând utilizatorilor să interacționeze cu datele despre domenii stocate conform permisiunilor lor.
  - **Scripturi**: Include următoarele scripturi cheie utilizate de sistem:
    * *Domain Scheduler*: Gestionează programarea evaluărilor de domenii.
    * *Score Decay Applicator*: Implementează mecanismul de diminuare a scorurilor reputației domeniilor.
    * *Work Validator*: Asigură integritatea și acuratețea rezultatelor de la nodurile distribuite.
    * *Work Assimilator*: Integrează rezultatele validate în baza de date a sistemului.
  - **Stocare**: Responsabilă pentru stocarea datelor esențiale ale sistemului, incluzând:
    * Credențiale utilizatori și token-uri de acces.
    * Caracteristicile domeniilor extrase.
    * Scorurile de evaluare ale reputației domeniilor.

- **Infrastructura de Calcul Distribuit:**

  - **Client BOINC**: Software-ul instalat pe dispozitivele voluntare pentru a primi și executa unități de lucru.
  - **Server BOINC**: Serverul central care gestionează distribuirea unităților de lucru și colectarea rezultatelor.
  - **Aplicații de Proiect BOINC**: Aplicații specifice care definesc sarcinile executate de clienții BOINC.
  - **Unități de Lucru și Fișiere de Rezultate**: Unități de lucru trimise clienților și fișierele de rezultate corespunzătoare returnate după execuție.
  - **Scheduler**: O componentă care atribuie unități de lucru clienților disponibili și asigură distribuirea eficientă a sarcinilor.
  - **Manager BOINC (Interfață Utilizator)**: O aplicație pentru utilizatori care permite monitorizarea și gestionarea contribuțiilor lor la sistemul distribuit.
  - **Instrumente de Administrare și Comunitate Bazate pe Web**: Instrumente online pentru gestionarea proiectului BOINC, implicarea comunității și furnizarea statisticilor proiectului.

6.2.1 Fundamentul Proof of Concept

Pentru a dezvolta un proof-of-concept reproductibil și distributibil, a fost selectată platforma de containerizare ușoară Docker[1]. Această alegere ne permite să asigurăm reproducerea și distribuirea fiabilă a sistemului. Docker funcționează prin împachetarea tuturor elementelor necesare unei aplicații—precum codul, runtime-ul, bibliotecile și setările—în containere care se comportă constant pe orice mediu. Acest lucru face ca Docker să fie o fundație ideală pentru construirea sistemelor PoC compuse din multiple servicii.

6.3 Fluxul de Implementare

Implementarea proof-of-concept-ului (PoC) a implicat o serie de pași metodici pentru a depăși provocările tehnice și a atinge obiectivele de design prezentate anterior. Această secțiune oferă o relatare detaliată a procesului de dezvoltare, evidențiind activitățile și etapele cheie care au condus la realizarea cu succes a sistemului distribuit de evaluare a reputației domeniilor utilizând BOINC și dispozitive Android.

6.3.1 Dezvoltarea Serviciului Web ADDReS Nexus

Pentru a gestiona programarea domeniilor și validarea rezultatelor, MongoDB a fost integrat folosind PyMongo[2], în timp ce administrarea utilizatorilor a continuat să se bazeze pe baza de date SQLite implicită a Django. Această abordare hibridă ne-a permis să beneficiem de flexibilitatea și performanța MongoDB pentru gestionarea datelor legate de domenii, menținând în același timp simplitatea și robustețea caracteristicilor de autentificare și administrare a utilizatorilor oferite de Django.

Această configurație a permis sistemului să obțină avantaje din ambele paradigme. MongoDB a oferit scalabilitatea și flexibilitatea necesare pentru gestionarea programării și validării domeniilor la scară largă, în timp ce interfața de administrare Django și SQLite au asigurat o experiență sigură și integrată pentru gestionarea utilizatorilor. De asemenea, această abordare a poziționat sistemul pentru scalabilitate viitoare, permițând componentelor bazate pe MongoDB să evolueze independent, susținând operațiuni sau integrații de date mai complexe, fără a perturba aplicația centrală Django.

În final, am creat un fișier Dockerfile care descrie toate specificațiile necesare pentru mediu, versiunile de dependențe și am împachetat aplicația într-o imagine Docker împreună cu dependențele și fișierele de configurare. Aceasta ne permite să integrăm cu ușurință serviciul web Nexus alături de restul componentelor infrastructurii.

[1] `https://docs.docker.com`, [Accessed on 29 March 2024]

[2] `https://pymongo.readthedocs.io/en/stable/`, [Accessed on 29 March 2024]

6.3.2 Configurarea Mediilor Nexus și BOINC cu Docker

Mediul fundamental al sistemului este stabilit prin implementarea serviciului Docker pe o mașină virtuală bazată pe Linux. Distribuția specifică de Linux nu este relevantă, deoarece Docker operează direct cu kernelul, care oferă abstracțiile necesare la nivel de sistem pentru containerizare. Ulterior, descărcăm infrastructura Docker pentru BOINC[3] și modificăm fișierul de configurație docker-compose.yml pentru a integra platforma Nexus. Această abordare permite serviciilor să funcționeze pe aceeași rețea și, în același timp, consolidează proiectul sub aceeași configurație.

După rularea comenzii `docker compose up` din utilitarul Docker Compose, obținem infrastructura implementată. Dacă procesul a fost realizat cu succes, putem interacționa acum cu serverul web Nexus pe portul 80 și cu consola de administrare BOINC pe portul 8081.

Utilizarea Docker a oferit un mediu izolat care a simplificat procesul de configurare și a asigurat consistența pe diferite sisteme. Containerele Docker pot fi ușor pornite, oprite și scalate, oferind flexibilitate în timpul dezvoltării și testării.

6.3.3 Dezvoltarea Aplicației Worker

În această secțiune vom explora procesul de dezvoltare al aplicației worker, provocările întâmpinate la încercarea de a implementa prima versiune pe dispozitivul mobil Android și soluțiile găsite pentru a depăși aceste limitări.

Versiunea inițială a aplicației worker a fost implementată în Python, valorificând ecosistemul robust de biblioteci pentru sarcini precum interogări WHOIS, obținerea de informații DNS, analiză de rețea și colectarea de date despre domenii, esențiale pentru extragerea caracteristicilor. De asemenea, integrează un model de învățare automată XGBoost pre-antrenat, gestionat cu Scikit-learn, pentru a prezice scorurile de reputație ale domeniilor. Deși flexibilitatea și suportul pentru unelte din Python sunt ideale pentru dezvoltare, dependența acestuia de un interpret prezintă o provocare pe dispozitivele Android, care nu dispun de un mediu Python implicit. Construirea unui interpret Python pentru Android ar intra în conflict cu principiul de proiectare al sistemului, care presupune accesibilitatea acestuia fără a necesita modificări pe dispozitivele utilizatorilor.

Limbajul de programare Go a fost selectat pentru acest proiect datorită simplității, eficienței și capacității sale de a îndeplini cerințele unice ale sistemului. Una dintre cele mai convingătoare caracteristici ale Go este abilitatea sa de a compila toate bibliotecile și codul necesar într-un singur fișier binar portabil. Această caracteristică elimină necesitatea dependențelor externe sau a mediilor de rulare complexe, simplificând semnificativ implementarea pe diverse platforme. Mai mult, Go suportă cross-compilation,

[3] `https://github.com/marius311/boinc-server-docker`, [Accessed on 12 April 2024]

permițând construirea aplicațiilor pentru diferite arhitecturi și sisteme de operare, inclusiv Android. Această compatibilitate cross-platform este deosebit de avantajoasă pentru proiectele care vizează medii eterogene sau platforme mobile.

Pentru a construi cu succes aplicația, este imperativ să descărcați Android NDK[4], un set de instrumente care permite implementarea unor părți din aplicație în cod nativ. În cazul nostru, acesta ne va permite să construim un fișier binar compatibil cu arhitectura aarch64 a Android.

Executarea următorului script bash va produce un fișier binar compatibil:

```
#!/bin/bash

export NDK=/Users/dev/Library/Android/sdk/ndk/23.1.7779620
export CC=$NDK/toolchains/llvm/prebuilt/darwin-x86_64/bin/aarch64-linux
   -android21-clang
env GOOS=android GOARCH=arm64 CGO_ENABLED=1 CC=$CC go build -o
   worker_arm64
```

Listing 6.1: Bash script for building Android worker application.

După încărcarea în emulatorul Android și rularea programului nostru, putem confirma că aplicația worker se execută cu succes.

6.3.4 Wrapper-ul Aplicației BOINC

BOINC oferă un API C++ pentru unitățile de lucru care interacționează cu infrastructura sa, gestionând sarcini precum pornirea/oprirea proceselor, raportarea progresului, accesul la fișiere și raportarea erorilor. Totuși, integrarea directă a acestui API în aplicația worker o cuplează strâns cu platforma BOINC, complicând astfel migrarea viitoare către sisteme alternative. Pentru a menține flexibilitatea, a fost adoptat modelul de "gridificare", așa cum a fost propus de Mateos et al.[20], folosind o aplicație wrapper care încorporează API-ul BOINC și acționează ca un intermediar de comunicație între aplicația worker și serverul BOINC.

Deoarece BOINC nu furnizează binare pentru arhitectura aarch64 sau Android, wrapper-ul a fost compilat în mod încrucișat folosind uneltele Clang din Android NDK. Pentru a pregăti aplicația worker și wrapper-ul său pentru o distribuție sigură, acestea au fost încărcate într-un director structurat pe serverul BOINC și semnate folosind chei criptografice. Acest proces a implicat generarea cheilor, semnarea binarelor și actualizarea configurației serverului pentru a face aplicația accesibilă nodurilor voluntare. Aceste măsuri au asigurat conformitatea cu cerințele de securitate ale BOINC, menținând integritatea sistemului și permițând o implementare eficientă.

[4] `https://developer.android.com/ndk`, [Accessed on 22 April 2024]

6.3.5 Trimiterea Tarelor la Distanță (Remote Job Submission)

"Remote job submission" se referă la procesul prin care sarcinile sunt trimise către un server BOINC de către scripturi sau programe care operează extern, fără a necesita acces direct (login) pe server. Acest lucru este facilitat prin utilizarea de Web RPC-uri (Remote Procedure Calls), care permit crearea de loturi (batches) și sarcini (jobs), precum și gestionarea fișierelor de intrare și ieșire. Această metodă de trimitere se distinge de trimiterea locală, care implică interacțiunea directă cu serverul.

Trimiterea la distanță introduce câteva diferențe cheie față de trimiterea locală. În primul rând, cei care trimit sarcini trebuie să dețină un cont de utilizator pe proiectul BOINC. Trimiterile sunt legate de acest cont, iar cei care trimit trebuie să-și furnizeze acreditările. În plus, utilizatorii trebuie să aibă drepturi de acces și limite (quotas) corespunzătoare pentru a crea sarcini.

Sarcinile sunt organizate și trimise în "batches" (loturi), chiar și atunci când este nevoie de o singură sarcină; în astfel de cazuri, se creează un lot cu un singur element. Spre deosebire de sarcinile locale, trimiterile la distanță nu utilizează un asimilator pentru a procesa fișierele de ieșire. În schimb, operațiunea "retire batch" este folosită pentru a indica faptul că fișierele și înregistrările din baza de date ale lotului pot fi curățate în siguranță.

Procesul de trimitere a sarcinilor către serverul BOINC este gestionat de scriptul Nexus Domain Scheduler. Acest script este responsabil de determinarea următorului set de domenii care necesită validare sau revalidare, pe baza algoritmului de programare prezentat anterior. Odată ce domeniile sunt identificate, scriptul le organizează într-un batch, care servește drept intrare pentru procesările ulterioare. Prin automatizarea acestei sarcini, Nexus Domain Scheduler asigură o gestionare eficientă și sistematică a fluxurilor de lucru pentru validarea domeniilor în cadrul infrastructurii BOINC.

6.3.6 Procesarea Sarcinilor pe Dispozitive Android

Sarcinile sunt trimise dispozitivelor Android cu un factor de replicare de trei, asigurând astfel că fiecare unitate de lucru este procesată de cel puțin trei dispozitive. Această configurare se aliniază cu calculele noastre statistice, indicând că, la o probabilitate de compromitere a unui nod de 10%, atribuire a sarcinilor către trei dispozitive atinge un nivel de încredere de 95% că majoritatea dispozitivelor care procesează orice sarcină dată sunt oneste.

Pentru a implementa acest lucru, am configurat setările proiectului BOINC astfel încât să necesite un cvorum minim de trei rezultate valide (min_quorum=3) înainte de a considera o unitate de lucru finalizată. De asemenea, am stabilit corespunzător numărul maxim de rezultate cu erori și numărul total de rezultate, pentru a gestiona eventualele eșecuri sau

discrepanțe.

Am instanțiat trei dispozitive Android emulate utilizând [Emulator Name], fiecare cu profiluri hardware unice și conectate la proiectul BOINC folosind conturi de utilizator individuale. Acest lucru a asigurat că fiecare emulator a acționat ca un participant distinct în rețea. Clientul BOINC a fost instalat pe fiecare emulator, iar setările de rețea au fost ajustate pentru a permite comunicarea cu serverul BOINC.

După implementare, dispozitivele s-au înregistrat cu succes în infrastructura proiectului și au început să primească unități de lucru. Serverul BOINC a distribuit sarcinile astfel încât fiecare unitate de lucru să fie trimisă către cel puțin trei dispozitive. Dispozitivele au procesat sarcinile folosind aplicația worker bazată pe Go, executată prin intermediul wrapper-ului BOINC, și au returnat rezultatele către server.

6.3.7 Validarea Rezultatelor

În procesarea volumelor de lucru pot apărea discrepanțe din cauza diferențelor de arhitectură a mașinilor, sisteme de operare, topologii de rețea sau chiar a interferențelor malițioase. Pentru a remedia această situație, se utilizează un factor de replicare, asigurând agregarea rezultatelor care sunt statistic mai apropiate. Aplicațiile worker returnează scoruri de reputație împreună cu caracteristicile de domeniu în format JSON, care sunt validate folosind metoda Intervalului Intercuartil (IQR) pentru filtrarea valorilor extreme. Media scorurilor validate este înregistrată ca scor de reputație final în baza de date Nexus, asigurând fiabilitate și reprezentativitate. Extracția caracteristicilor de domeniu aplică, de asemenea, o abordare bazată pe consens, reținând datele consistente și eliminând informațiile conflictuale, asigurând acuratețea caracteristicilor extrase.

Variațiile în caracteristicile de domeniu pot apărea și din cauza diferențelor de localizare a nodurilor, ISP-uri, configurații proxy sau liste negre în hardware-ul de comunicație. Un mecanism de filtrare bazat pe consens atenuează aceste discrepanțe, păstrând doar datele validate pentru antrenarea și rafinarea modelului de învățare automată, care este ulterior actualizat pentru volume de lucru viitoare. Toate procesele, inclusiv validarea datelor, agregarea și stabilirea consensului, sunt gestionate de un script Python integrat în configurația proiectului BOINC. Acest script asigură o validare robustă și consecventă în timpul fazei de consens, sporind fiabilitatea sistemului de calcul distribuit.

6.3.8 Asimilare

Sarcina de asimilare este acoperită de scriptul de trimitere a sarcinilor utilizând metoda API-ului BOINC `get_output_files_url` pentru a recupera URL-urile fișierelor de ieșire care sunt stocate pe serverul BOINC și `file_get_contents` pentru a obține rezultatele stocate în fișiere. Structura rezultatului este descrisă mai jos:

```
{
    "domain_age": 5,
    "domain_expiration_days": 365,
    "domain_length": 12,
    "dga_suspect": 0,
    "typosquatting_suspect": 0,
    "dns_ttl": "1800-86400",
    "dnssec": 1,
    "fast_flux_suspect": 0,
    "ssl": 1,
    "ssl_certificate_issuer": "Let's Encrypt",
    "ssl_certificate_valid_from": 120,
    "ssl_certificate_validity": 365,
    "registrar": "Namecheap",
    "ip_address": "192.0.2.1",
    "as_number": "AS12345",
    "is_blacklisted": 0,
    "is_ip_blacklisted": 0,
    "tls_version": "TLSv1.3",
    "strict_transport_security": 1,
    "content_security_policy": 1,
    "has_email_server": 1,
    "is_email_server_blacklisted": 0,
    "email_has_dmarc": 1,
    "email_has_spf": 1,
    "email_has_dkim": 1,
    "email_server_is_open_relay": 0
}
```

Listing 6.2: Sample worker response strcuture.

Conținutul fișierului este stocat în baza de date Nexus, asigurându-se astfel că sunt disponibile pentru referințe viitoare și pentru recalibrarea modelului XGBoost cu un set de date extins, îmbunătățind astfel acuratețea predictivă pe măsură ce trece timpul.

Cu procesul de asimilare integrând cu succes rezultatele validate în baza de date Nexus, am finalizat implementarea proof of concept-ului sistemului distribuit de reputație a domeniilor. Această implementare cuprinzătoare a inclus dezvoltarea aplicației worker, configurarea infrastructurii BOINC, procesarea sarcinilor pe dispozitive Android și asigurarea integrității datelor prin validare și asimilare.

După ce au fost stabilite componentele fundamentale ale sistemului, următorul pas critic este evaluarea performanței și eficienței sale. În secțiunea următoare, vom explora evaluarea sistemului, prezentând metodologia utilizată, detaliind configurația simulării și analizând rezultatele obținute. Această evaluare are ca scop evaluarea capacității sistemului de a atinge obiectivele de design prezentate anterior, examinarea scalabilității și eficienței sale și identificarea eventualelor domenii de îmbunătățire.

6.4 Evaluarea Sistemului

Evaluarea sistemului distribuit de reputație a domeniilor este esențială pentru validarea designului și funcționalității sale. Datorită provocărilor de a implementa sistemul cu un număr mare de participanți reali, au fost realizate simulări pentru a modela comportamentul acestuia în condiții variate. Această evaluare a implicat analiza performanței, scalabilității și eficienței în atingerea obiectivelor de design. În mod specific, simularea a evaluat 10.000 de domenii pe o rețea distribuită ce cuprinde 100 de noduri, împărțite în mod egal între dispozitive desktop și Android. Nodurile desktop, cu puterea lor de procesare mai mare și conexiunile stabile la rețea, au servit ca referință de performanță, în timp ce nodurile Android au oferit informații despre participarea dispozitivelor mobile, reflectând condițiile eterogene ale mediului real.

Configurația simulării a permis o analiză comparativă a utilizării resurselor și eficienței finalizării sarcinilor pe tipuri de noduri, demonstrând adaptabilitatea sistemului în medii diverse. Pe lângă evaluarea nodurilor worker, a fost monitorizată și încărcătura serverului central datorită rolului său esențial în distribuirea sarcinilor, validarea rezultatelor și asimilarea în baza de date. Aceste procese asigură integritatea și consistența datelor, menținând în același timp capacitatea de răspuns la interogările primite. Prin identificarea potențialelor puncte de blocaj în operațiunile serverului, evaluarea a furnizat informații pentru optimizări care să îmbunătățească scalabilitatea și fiabilitatea, susținând astfel abilitatea sistemului de a gestiona sarcini de calcul distribuit în condițiile realității.

6.4.1 Impactul asupra CPU și Memoriei

Evaluarea utilizării CPU și memoriei pe dispozitivele desktop și Android a demonstrat un consum minim de resurse în timpul execuției sarcinilor, cu timpuri medii de finalizare de 100 de milisecunde pentru desktopuri și 200 de milisecunde pentru dispozitivele Android, reflectând capabilitățile de procesare specifice platformei. Uneltele de profilare integrate în aplicația worker au arătat diferențe neglijabile în utilizarea CPU între cele două categorii de dispozitive și o alocare mică a memoriei, cu o medie de 0.4 MB, confirmând designul ușor și eficient al aplicației. Deși timpii de răspuns ai domeniilor extinși pot conduce ocazional la perioade de execuție prelungite aproape de limita de timeout, aplicația rămâne inactivă în aceste perioade, consumând resurse minime. Optimizarea timpilor de răspuns ai domeniilor sau îmbunătățirea gestionării timeout-ului ar putea îmbunătăți și mai mult eficiența globală.

6.4.2 Impactul asupra Rețelei

Impactul rețelei asupra nodurilor în timpul procesului de evaluare este minim datorită protocoalelor ușoare utilizate, cum ar fi WHOIS, DIG și HTTP, care necesită doar lățime

de bandă redusă pentru operațiuni de rutină. Cea mai semnificativă utilizare a rețelei are loc în timpul configurării inițiale, când clientul BOINC descarcă fișierele esențiale, inclusiv binarul worker, aplicația wrapper și fișierul XML de configurare. Această configurare este un proces unic; sarcinile ulterioare implică utilizarea minimă a rețelei, deoarece clientul BOINC reutilizează fișierele stocate local dacă nu sunt necesare actualizări, reducând astfel transferurile de date redundante și optimizând eficiența generală a sistemului.

Pentru serverul BOINC, activitatea de rețea se concentrează în principal pe descărcarea fișierelor de sarcini de către nodurile worker și încărcarea rezultatelor. În timp ce simulările experimentale au arătat activitate sincronizată a rețelei datorită operațiunilor concurente, se așteaptă ca implementările din lumea reală să aibă un trafic mai distribuit pe măsură ce dispozitivele se înregistrează și solicită sarcini asincron. Această analiză subliniază necesitatea pregătirii serverului pentru a gestiona posibilele vârfuri de activitate în rețea, cum ar fi în perioadele de înregistrare intensă sau completarea simultană a sarcinilor. Anticiparea acestor fluctuații este esențială pentru menținerea stabilității serverului și asigurarea funcționării continue a sistemului de calcul distribuit. Proof of concept-ul validează, de asemenea, fezabilitatea arhitecturii distribuite, demonstrând scalabilitatea, eficiența și capacitatea de a gestiona evaluarea reputației domeniilor în timp real prin valorificarea puterii computaționale descentralizate pe dispozitive diverse.

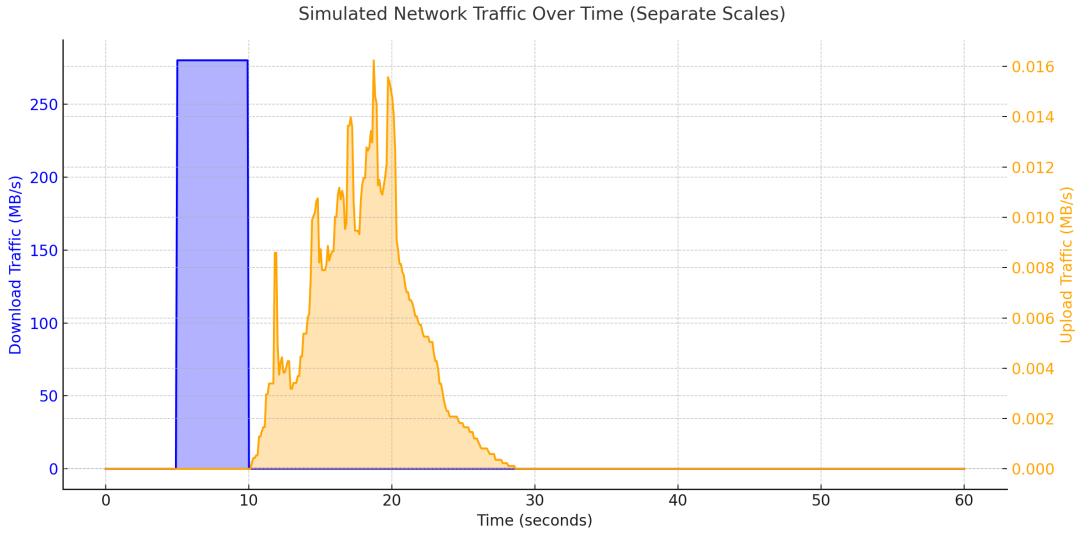

Figure 3: Simulated Network Traffic Over Time.

7 | Studiu de Caz pe Domeniile .ro

Până în prezent, experimentele noastre au fost realizate dintr-o perspectivă externă, utilizând doar seturi de date disponibile publicului. Deși această abordare a permis colectarea unui subset de caracteristici ale domeniilor, ea a restricționat în mod inerent domeniul analizei noastre la informațiile accesibile publicului. Totuși, prin accesul la o entitate privilegiată, cum ar fi un registru de Domenii de Nivel Superior (TLD), am extins semnificativ adâncimea și amploarea cercetării noastre. Acest acces ne-a dezvăluit informații despre caracteristicile domeniilor private, inclusiv validarea contactelor și istoricul domeniilor, care anterior erau inaccesibile. Prin integrarea acestor puncte de date suplimentare, am obținut o înțelegere mai cuprinzătoare a comportamentului și dinamicii domeniilor, îmbunătățind atât acuratețea, cât și robustețea analizelor noastre. Acest lucru ne-a permis să extindem și mai mult modelul nostru de algoritm de învățare automată, obținând o acuratețe mai mare în prezicerea scorului de reputație al unui domeniu.

7.1 Colectarea Informațiilor la Nivel TLD

Protocolul WHOIS pentru domeniile .ro este reglementat de legislația românească[1], care impune redactarea oricăror informații personale referitoare la domeniile deținute de persoane fizice. Reglementarea asigură conformitatea cu standardele de protecție a datelor, permițând afișarea doar a următoarelor informații:

- **Tipul Persoanei**
- **Data Înregistrării**
- **Data Expirării**
- **Starea Domeniului**
- **Numele Registrarului**
- **Nameserverele**

Această restricție limitează semnificativ cantitatea de informații accesibile pentru validarea unui domeniu .ro dintr-o perspectivă publică, ceea ce creează provocări pentru analizele și verificările cuprinzătoare ale domeniilor.

[1] `https://eur-lex.europa.eu/legal-content/RO/TXT/PDF/?uri=CELEX:32016R0679`, [Accessed on 12 May 2024]

Flagul de validare a contactului, o componentă a informațiilor private, este stabilit în urma unei proceduri concepute pentru a verifica identitatea deținătorului domeniului. Pentru persoanele fizice, acest proces implică, de obicei, depunerea unui document de identificare personală pentru a confirma că detaliile de contact furnizate sunt corecte și legale. În cazul companiilor, riscurile sunt comparativ mai scăzute, deoarece procesul de înregistrare a domeniului include verificarea detaliilor de identificare ale companiei cu datele provenite din instituții autorizate. Totuși, companiile trebuie să urmeze aceeași procedură de verificare pentru a asigura conformitatea.

Înregistrările istorice menținute de registru reprezintă o resursă esențială pentru înțelegerea ciclului de viață al unui domeniu, cuprinzând înregistrarea inițială, reînnoirile, transferurile de proprietate și ștergerea eventuală. Aceste înregistrări oferă informații cuprinzătoare despre trecutul domeniului, permițând evaluări mai precise ale comportamentului și reputației sale. Dintr-o perspectivă publică, totuși, accesul la astfel de informații este limitat. Când un domeniu devine disponibil pentru înregistrare, principala indicație a înregistrării sale anterioare este prezența sa pe listele negre publice. Această lipsă de transparență duce adesea la provocări în timpul validării domeniilor șterse.

O versiune simplificată a arhitecturii propuse a fost implementată, fiind în principal limitată de considerațiile de confidențialitate asociate cu implicarea participanților voluntari. Implementarea a fost realizată utilizând un mediu Kubernetes ușor, Minikube[2], care a servit ca infrastructură pentru găzduirea atât a serverelor Nexus, cât și a celor BOINC. În plus, a facilitat operarea nodurilor worker responsabile de calcularea scorurilor de reputație ale domeniilor pe baza caracteristicilor extrase ale domeniilor. Această abordare a permis o evaluare controlată și eficientă a arhitecturii, respectând în același timp standardele de confidențialitate și protecția datelor.

Obiectivul a fost de a demonstra eficiența modelului de învățare automată prin îmbunătățirea performanței acestuia prin integrarea unor informații suplimentare furnizate de domeniul de nivel superior (TLD). Caracteristicile de domeniu nou incluse, extrase ca parte a acestui proces de îmbunătățire, sunt detaliate în Tabelul 5.

Integrând noile caracteristici extrase în modelul XGBoost, am îmbunătățit semnificativ capacitatea sa predictivă, obținând o îmbunătățire de 5% în distingerea domeniilor malițioase de cele legitime. Printre caracteristicile adăugate, flagul de validare al registrantului a avut cel mai mare impact, având un rol deosebit de eficient ca discriminator între domeniile legitime și cele potențial malițioase.

7.2 Concluzii

Implementarea acestei versiuni simplificate a arhitecturii în cadrul TLD-ului românesc a facilitat colectarea unor informații valoroase despre caracteristicile domeniilor care nu

[2] `https://minikube.sigs.k8s.io/docs/start/`, [Accessed on 23 June 2024]

Feature	Data type	Description
Is contact valid?	Boolean	This flag
Nameserver changes	Numeric	Nameserver changes in the past year or since registration
Registrar transfer	Numeric	Number of times a domain had been transfered from a registrar to another
Ownership transfer	Numeric	Number of times a domain had it's ownership changed
Resolved domains	Numeric	The number of domains that the are being resolved by a nameserver hosted

Table 5: TLD advanced domain features.

sunt accesibile publicului. Aceasta demonstrează o oportunitate semnificativă pentru operatorii TLD de a adopta un rol mai proactiv în combaterea criminalității cibernetice, prin valorificarea avantajului de a deține informații mai detaliate și mai granulare. Această capacitate îmbunătățită poziționează TLD-urile ca jucători critici în dezvoltarea unor sisteme robuste de reputație a domeniilor.

8 | Concluzii

Pe măsură ce încheiem această lucrare, este un moment oportun pentru a sintetiza concluziile cheie, a evalua contribuțiile aduse domeniului și a imagina direcțiile pentru viitoare explorări și inovații.

8.1 Contribuții

Arhitectura propusă introduce o abordare nouă de valorificare a potențialului computațional al dispozitivelor mobile inactivate pentru stabilirea reputațiilor domeniilor. Prin valorificarea principiilor de calcul distribuit, sistemul demonstrează fezabilitatea transformării resurselor descentralizate, subutilizate, într-o soluție coezivă și scalabilă pentru abordarea sarcinilor compute intensive. Deși această lucrare se concentrează în principal pe prezentarea arhitecturii și validarea principiilor sale fundamentale, rezultatele obținute pe parcursul acestei teze oferă dovezi convingătoare ale practicabilității și eficienței acesteia. Sperăm că această lucrare va inspira cercetări și dezvoltări suplimentare în acest domeniu, încurajând atât eforturile academice, cât și pe cele industriale, pentru a rafina, implementa și extinde arhitectura pentru a-și atinge întregul potențial în aplicațiile din lumea reală.

O altă contribuție semnificativă este dezvoltarea funcției de decădere a scorului de reputație, care introduce o abordare nouă pentru evaluarea și reevaluarea periodică a domeniilor. Această metodologie subliniază importanța oferirii unei a doua oportunități pentru domeniile care au fost înregistrate de utilizatori legitimi, facilitând înlăturarea acestora din listele negre. Această abordare urmărește să echilibreze nevoia de securitate cibernetică cu potențialul de utilizare legitimă a domeniilor anterior compromise sau marcate.

Mai mult, această lucrare a demonstrat că un limbaj de programare modern și sigur, precum Go, poate fi utilizat eficient pentru dezvoltarea de unități de lucru complexe compatibile cu platformele cu resurse limitate, cum ar fi sistemul de operare Android. Acest lucru subliniază versatilitatea Go și adecvarea sa pentru a aborda provocările compatibilității cross-platform și utilizarea eficientă a resurselor, în special în mediile de calcul distribuit.

8.2 Listă de Publicații

1. Dragoș Smada, Mihail Dumitrache, Carmen-Ionela Rotună, Cristian-Alexandru Gheorghiță - The impact of internet domain name status on their reputation (Article, RRIA);

2. Mihail Dumitrache, Carmen Ionela Rotună, Alexandru Gheorghiță, Adrian Victor Vevera, Ionuț Sandu, Dragoș Smada - A Domain Reputation System Architecture Description Using TOGAF (Article, SIC, ISI);

3. Cristian-Alexandru Gheorghiță, Dragoș Smada, Adrian-Victor Vevera, Mihail Dumitrache, Ionuț-Eugen Sandu, Carmen-Ionela Rotună - Blacklists and whitelists in the framework of a domain reputation system (Article, RRIA);

4. Blockchain-based Decision Support System for Water Management - Bogdan-Ionuț Pahonțu, Diana-Andreea Arsene, Alexandru Predescu,Mariana Mocanu, Alexandru Gheorghiță (Article, SIC, ISI);

5. Carmen Ionela Rotună, Alexandru Gheorghiță, Ionut Sandu, Mihail Dumitrache, Meda Udroiu, Dragoș Smada - A Generic Architecture for Building a Domain Name Reputation System (Article, SIC, ISI);

6. Mihail Dumitrache, Ionuț-Eugen Sandu, Adriana-Meda Udroiu, Cristian-Alexandru Gheorghiță - Theoretical considerations about establishing the Internet domain reputation (Article, RRIA);

7. Alexandru Gheorghiță, Ionuț Petre - Securely Driving IoT by Integrating AIOps and Blockchain (Article, ROCYS);

8. Carmen Rotună, Alexandru Gheorghiță, Alin Zamfiroiu, Dragoș Smada - Smart City Ecosystem Using Blockchain Technology (Article, Informatică Economică);

9. Radu Boncea, Ionuț Petre, Victor Vevera, Alexandru Gheorghiță - Machine Learning Based Methods Used for Improving Scholar Performance (Article, eLearning & Software for Education);

10. Carmen Rotună, Carmen Cîrnu, Alexandru Gheorghiță - Implementing smart city solutions: Smart city map and city drop (Article, Calitatea Vieții);

11. Carmen Rotuna, Dragoș Smada, A Gheorghiță - Smart city applications built on big data technologies and secure IoT (Article, Ecoforum Journal);

12. Alexandru Gheorghita, Monica Anghel - Serious Games: An Oxymoron? (Proceedings, ICVL, ISI);

8.3 List de Proiecte

1. Core Program "Advanced Research Based on Emerging and Disruptive Technologies - Support for the Society of the Future," Objective: The Internet of the Future, Communications, Mobile Technologies; PN 2338 02 01 - "Architecture – Intelligent Monitoring Platform for Internet Domains through the Development of a Dynamic Reputation Assessment System (TLDRep)" - Project Team Member; Deliverable Contribution;

2. Core Program "Advanced Technologies and Services for the Development of the Information Society – TEHSIN," Objective: 01 – Advanced Technologies for e-Services; PN 09 23 01 10 – "Electronic Services Based on Cloud Computing Infrastructure" - Project Team Member; Deliverable Contribution;

3. Core Program "Methods for Analyzing the Impact of Social Media on Governance Processes"; PN 09 23 06 01 – "Electronic Services Based on Cloud Computing Infrastructure" – Project Team Member; Deliverable Contribution;

4. Sectoral Program "Methods for Analyzing the Impact of Social Media on Governance Processes"; 145/2015 – "Proposals for Solutions in Implementing the European Interoperability Framework at the National Level – Examples of Good Practices from EU Member States" – Project Team Member; Deliverable Contribution;

5. PN.802 - Expansion of Services Provided by the Online Platform for Evaluating Technical-Scientific Literature;

6. Online e-Participation System for Smart City Project Initiatives, PN 603;

7. Study on Adaptive Systems for Early Recognition of Cyber Attacks on State Resources - CS 76/19.06.2018;

8. New Research in Modeling and Optimizing Complex Systems with Applications in Industry, Business Environment, and Cloud Computing - PN101/2018;

9. PN101/2019 - Research on Advanced Policies and Solutions for Securing Critical Infrastructures Against Cyber Attacks;

10. PN 301/2019 - Non-Invasive Monitoring and Evaluation System for the Health of Elderly People in an Intelligent Environment Ro - SmartAgeing;

8.4 Viitoare Lucrări

Pentru lucrările viitoare, intenționăm să extindem domeniul de aplicare al sistemului prin explorarea includerii unor categorii suplimentare de dispozitive, cum ar fi dispozitivele Internet of Things (IoT), în pool-ul de calcul. Această extindere are ca scop valorificarea și mai mult a ecosistemului în expansiune al dispozitivelor conectate, sporind scalabilitatea, diversitatea și capacitatea generală de calcul a sistemului.

În plus, eforturile viitoare se vor concentra pe abordarea provocării legate de dimensiunea modelului de învățare automată prin implementarea unei alternative mai ușoare, și anume LightGBM[1]. LightGBM este proiectat special pentru a reduce utilizarea memoriei, menținând în același timp performanțe ridicate, oferind antrenamente mai rapide și mai eficiente ale modelului. Se așteaptă ca această adaptare să îmbunătățească compatibilitatea sistemului cu dispozitivele cu resurse limitate și să sporească eficiența operațională generală fără a compromite acuratețea predictivă.

În final, ne propunem să explorăm posibilitatea descentralizării sistemului de stocare prin implementarea unei soluții bazate pe blockchain pe nodurile participante. O astfel de abordare nu doar că ar elimina dependența de un server centralizat care gestionează o bază de date tradițională, dar ar oferi și un cadru imuabil și inerent mai sigur pentru stocarea datelor. Prin valorificarea tehnologiei blockchain, sistemul ar beneficia de o transparență îmbunătățită, rezistență la manipulare și consens descentralizat, abordând potențialele vulnerabilități asociate centralizării și întărind astfel robustețea generală și încrederea în arhitectura acestuia.

8.5 Lecții Învățate

Una dintre cele mai profunde lecții învățate pe parcursul acestei lucrări este sintetizată într-o citat de Bjarne Stroustrup, împărtășit cu mine de un coleg: "Dacă crezi că este simplu, atunci ai înțeles greșit problema." Această frază răsună profund și captează esența provocărilor întâmpinate pe parcursul fazelor de documentare și implementare ale acestui proiect.

Sarcinile care inițial păreau simple, cum ar fi compilarea unui binar pentru un sistem de operare derivat din Linux, precum Android, s-au dovedit a fi mult mai complexe decât am

[1] `https://lightgbm.readthedocs.io/en/stable/`, [Accessed on 12 May 2024]

anticipat. Aceste provocări subestimate nu au servit doar ca obstacole, ci și ca oportunități valoroase de creștere, dezvoltând o înțelegere mai profundă a complexităților implicate în dezvoltarea sistemului. Astfel de obstacole, deși solicitante, îmbogățesc călătoria și inspiră o continuare a căutării cunoașterii și explorării în acest domeniu în continuă evoluție.

Bibliography

[1] Landers, Joerg Weissmann, L Steinbrinker, Uwe Kraemer, Stefan Weinert, and H.-J Lüddeke. Notos: an electronic questionnaire structures individualised diabetes management. In *NOTOS: an electronic questionnaire structures individualised diabetes management*, 09 2013.

[2] Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. Exposure: A passive dns analysis service to detect and report malicious domains. *ACM Transactions on Information and System Securi*, 16, 04 2014. doi: 10.1145/2584679.

[3] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, and David Dagon. Detecting malware domains at the upper dns hierarchy. In *Detecting malware domains at the upper DNS hierarchy*, pages 27–27, 08 2011.

[4] Michael Shirts and Vijay Pande. Screen savers of the world unite. *Science (New York, N.Y.)*, 290:1903–4, 01 2001. doi: 10.1126/science.290.5498.1903.

[5] Hitesh Bheda. Application processing approach for smart mobile devices in mobile cloud computing. *International Journal of Software Engineering and Knowledge Engineering*, 3:1046–1054, 08 2013.

[6] Joshua Jaffe and Luciano Floridi. Ransomware: Why it's growing and how to curb its growth. *Applied Cybersecurity & Internet Governance*, 11 2024. doi: 10.60097/ACIG/192959.

[7] Elahe Soltanaghaei and Mehdi Kharrazi. Detection of fast-flux botnets through dns traffic analysis. *Scientia Iranica*, 22, 01 2016.

[8] Valentin Manescu, Neghina Alexandra, Andreea Barbu, Ganciu Rodica, and Gheorghe Militaru. Analysis of ssl certificates trends and extended validation ssl usage for e-commerce websites and internet of things. *UPB Scientific Bulletin, Series C: Electrical Engineering*, 83, 12 2021.

[9] HuffPost. Domain theft: Who owns your website?, 2014. URL `https://www.huffpost.com/entry/domain-theft_n_5877510`. Accessed: 2023-01-25.

[10] Paul Mockapetris and Kevin J Dunlap. Development of the domain name system. In *Symposium proceedings on Communications architectures and protocols*, pages 123–133, 1988.

[11] Aminollah Khormali, Jeman Park, Hisham Alasmary, Afsah Anwar, Muhammad Saad, and David Mohaisen. Domain name system security and privacy: A contemporary survey. *Computer Networks*, 185:107699, 2021.

[12] Deepak Kumar Vishwakarma. Domain name generation algorithms. *Masaryk University*, 2017.

[13] Craig Beaman, Ashley Barkworth, Toluwalope David Akande, Saqib Hakak, and Muhammad Khurram Khan. Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & security*, 111:102490, 2021.

[14] Dimitrios Georgoulias, Jens Myrup Pedersen, Morten Falch, and Emmanouil Vasilomanolakis. A qualitative mapping of darkweb marketplaces. In *2021 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–15. IEEE, 2021.

[15] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL `http://dx.doi.org/10.1145/2939672.2939785`.

[16] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[17] Lean Yu, Rongtian Zhou, Rongda Chen, and Kin Keung Lai. Missing data preprocessing in credit classification: One-hot encoding or imputation? *Emerging Markets Finance and Trade*, 58(2):472–482, 2022.

[18] Santhanam Ramraj, Nishant Uzir, R Sunil, and Shatadeep Banerjee. Experimenting xgboost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, 9(40):651–662, 2016.

[19] Jiawei Han and Micheline Kamber. Data mining: Concept and techniques (s˜ ond edition), 2006.

[20] Cristian Mateos, Alejandro Zunino, and Marcelo Campo. A survey on approaches to gridification. *Software: Practice and Experience*, 38(5):523–556, 2008.