GESTIONAREA SECURITĂȚII CIBERNETICE PRIVIND ALGORITMII DE ÎNVĂȚARE AUTOMATĂ

DIONISIE Ștefania
Facultatea de Inginerie Industrială și Robotică, Specializarea: Logistică Industrială, Anul de studii: Master II, e-mail: stefi_dio@yahoo.com

Conducător ştiinţific: Ș.l. Dr. Ing. Adrian Popescu

REZUMAT: Trecerea la o lume tot mai digitalizată câștigă din ce în ce mai multă atenție în întreaga lume și, în special, sunt efectuate studii privind inteligența artificială, big data și cloud. Riscurile de securitate cibernetică pentru dispozitivele Internet of Things (IoT) provenite de la o diversitate de furnizori și desfășurate în număr mare, cresc rapid. Totul, inclusiv informațiile despre starea tuturor obiectelor, sunt partajate în timp real și toate locațiile și obiectele sunt echipate cu senzori. Acestea acționează inteligent, cum ar fi luarea deciziilor. Deoarece senzorii sunt echipați în locații și obiecte și conectați la rețele de calculatoare, utilizatorii pot primi informații în orice moment și oriunde. Cu toate acestea, este posibil să apară probleme neașteptate din cauza complexității ridicate și a incertitudinii fabricii inteligente. Astfel, este foarte probabil să înceteze procesul de fabricație, să declanșeze defecțiuni și să transmită informații importante.

CUVINTE CHEIE: Securitate cibernetică, IoT, Fabrici inteligente, Machine learning, Deep learning

1. Introducere

Probleme multiple sunt ridicate în conformitate cu creșterea rapidă a IoT. Cu alte cuvinte, utilizatorii sunt vulnerabili la multe amenințări, cum ar fi procesarea unor cantități uriașe de date, procesarea consumului de energie, rezolvarea amenințărilor la adresa securității și criptarea / decodarea datelor masive. În consecință, utilizarea unei soluții criptate corespunzător și necesitatea criptării bazate pe inteligență artificială, inclusiv învățarea automată și învățarea profundă sunt în creștere pentru a rezolva aceste probleme atunci când numeroase instrumente sunt conectate în mediul IoT. Învățarea automată are o gamă largă de aplicații, inclusiv motoarele de căutare, diagnostice medicale, detectare de fraudă în utilizarea cardului de credit, analiză a pieței de valori, clasificarea a secventelor de ADN, recunoaștere a vorbirii și limbajului scris, jocuri și robotică.

În acest sens, s-au depus eforturi extinse pentru a aborda problemele de securitate și confidențialitate în rețelele IoT, în principal prin abordări criptografice tradiționale. Cu toate acestea, caracteristicile unice ale nodurilor IoT fac ca soluțiile existente să fie insuficiente pentru a cuprinde întregul spectru de securitate al rețelelor IoT. Tehnicile Machine Learning (ML) și Deep Learning (DL), care sunt capabile să ofere informații integrate în dispozitivele și rețelele IoT, pot fi folosite pentru a face față diferitelor probleme de securitate.

Unele sisteme de învățare automată încearcă să elimine toată nevoia de intuiție sau cunoștințe de specialitate din procesele de analiză a datelor, în timp ce alții încearcă să stabilească un cadru de colaborare între expert și computer. Cu toate acestea, intuiția umană nu poate fi înlocuită în totalitate, deoarece proiectantul sistemului trebuie să precizeze forma de reprezentare a datelor și metodele de manipulare și caracterizare a acestora.

2. Tipuri de algoritmi

Diferiți algoritmi de învățare automată sunt grupați în funcție de producția acestora. Unele tipuri de algoritmi sunt:

a) Învățarea supervizată total

Algoritmii de învățare supervizați construiesc un model matematic al unui set de date care conține atât intrările, cât și ieșirile dorite. Datele sunt cunoscute sub numele de date de antrenament și constau dintr-un set de exemple de antrenament. Fiecare exemplu de antrenament are una sau mai multe intrări și ieșirea dorite, cunoscute și sub numele de semnal de supraveghere. În modelul matematic, fiecare exemplu de antrenament este reprezentat de o matrice sau vector, uneori numit vector caracteristic, iar datele de antrenament sunt reprezentate de o matrice . Prin optimizarea iterativă a unei funcții obiective , algoritmii de învățare supervizați învață o funcție care poate fi utilizată pentru a prezice rezultatul asociat cu noile intrări. O funcție optimă va permite algoritmului să determine corect ieșirea pentru intrările care nu au făcut parte din datele de instruire. Se spune că un algoritm care îmbunătățește precizia ieșirilor sau predicțiilor sale în timp a învățat să îndeplinească acea sarcină. Tipurile de algoritmi de învățare supervizată includ învățarea activă, clasificarea și regresia. Algoritmii de clasificare sunt utilizați când ieșirile sunt restricționate la un set limitat de valori, iar algoritmii de regresie sunt folosiți atunci când ieșirile pot avea orice valoare numerică într-un interval. De exemplu, pentru un algoritm de clasificare care filtrează e-mailurile, intrarea ar fi un e-mail primit, iar ieșirea ar fi numele folderului în care se va înregistra e-mailul. [1]

Tot procesul de modelare se realizează pe un set de exemple format numai din intrări în sistem. Prin urmare, în acest caz, sistemul trebuie să fie capabil să recunoască modele, pentru a fi capabil de a eticheta noi intrări. [1]

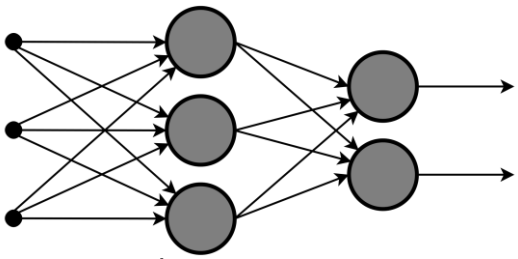

Fig. 1. Învățatare supervizată [1]

b) Învățarea supervizată parțial

Învățarea supervizată parțial se încadrează între învățarea nesupervizată (fără date de formare etichetate) și învățarea supervizată (cu date de formare complet etichetate). Unele dintre exemplele de formare lipsesc, însă mulți cercetători în învățarea automată au descoperit că datele neetichetate, atunci când sunt utilizate împreună cu o cantitate mică de date etichetate, pot produce o îmbunătățire considerabilă a preciziei învățării. [1]

c) Învățarea nesupervizată

Algoritmii de învățare nesupervizați iau un set de date care conține doar intrări și găsesc structura în date, cum ar fi gruparea punctelor de date. Prin urmare, algoritmii învață din datele de testare care nu au fost etichetate sau clasificate. În loc să răspundă la feedback, algoritmii de învățare nesupervizați identifică puncte comune în date și reacționează pe baza prezenței sau absenței unor astfel de puncte comune în fiecare bucată nouă de date. O aplicație centrală a învățării nesupervizate este în domeniul estimării densității în statistici , cum ar fi găsirea funcției densității probabilității, deși învățarea nesupervizată cuprinde alte domenii care implică rezumarea și explicarea caracteristicilor datelor. Analiza clusterului este atribuirea unui set de observații în subgrupuri (numite clustere), astfel încât observațiile din cadrul aceluiași cluster să fie similare conform unuia sau mai multor criterii predesemnate, în timp ce observațiile extrase din diferite clustere sunt diferite. Diferite tehnici fac presupuneri diferite asupra structurii datelor, adesea definite printr-o anumită metrică de similitudine și evaluate, de exemplu, prin compacitate internă sau similitudinea dintre membrii aceluiași cluster. Alte metode se bazează pe densitatea estimată și pe conectivitatea grafică. [1]

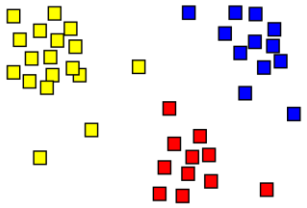

Fig. 2. Clustere (Machine learning, 2021)

d) Auto-învățarea

Auto-învățarea a fost introdusă împreună cu o rețea neuronală capabilă de auto-învățare numită crossbar adaptive array (CAA). Este o învățare fără recompense externe și fără sfaturi externe ale profesorului. Algoritmul de auto-învățare CAA calculează, într-o manieră transversală, atât deciziile despre acțiuni, cât și emoțiile (sentimentele) cu privire la situațiile de consecință. Sistemul este condus de interacțiunea dintre cogniție și emoție. Algoritmul de auto-învățare actualizează o matrice de memorie W = || w (a, s) || astfel încât în fiecare iterație să execute următoarea rutină de învățare automată:

În situația s efectuați o acțiune a;
Primiți situația de consecință;
Calculați emoția de a fi în situația de consecință v (s ');
Actualizați memoria transversală w '(a, s) = w (a, s) + v (s'). [1]

Este un sistem cu o singură intrare, situație s și o singură ieșire, acțiune (sau comportament) a. Nu există nici o intrare separată. Valoarea propagată înapoi (întărirea secundară) este emoția față de situația de consecință. CAA există în două medii, unul este mediul comportamental, iar celălalt este mediul genetic, din care primește emoții inițiale despre situațiile care trebuie întâlnite în mediul comportamental. [1]

e) Învățarea robotului

În robotica de dezvoltare , algoritmii de învățare a robotului își generează propriile secvențe de experiențe de învățare, cunoscute și sub denumirea de curriculum, pentru a dobândi cumulativ noi abilități prin explorarea autoghidată și interacțiunea socială cu oamenii. Acești roboți folosesc mecanisme de îndrumare, cum ar fi învățarea activă, maturizarea, sinergiile motorii și imitația. [1]

3. Învățare profundă (Deep learning)

Învățarea profundă (cunoscută și sub numele de învățare profundă structurată) face parte dintr-o familie mai largă de metode de învățare automată bazate pe rețele neuronale artificiale cu învățare prin reprezentare. [2]

Rețelele neuronale artificiale (ANN) au fost inspirate din prelucrarea informațiilor și noduri de comunicații distribuite în sistemele biologice. Rețelele neuronale tind să fie statice și simbolice, în timp ce creierul biologic al majorității organismelor vii este dinamic (plastic) și analog. [2]

Adjectivul „deep” în învățarea profundă se referă la utilizarea mai multor straturi în rețea. Învățarea profundă este o variație modernă care se referă la un număr nelimitat de straturi de mărime mărginită, care permite aplicarea practică și implementarea optimizată, păstrând în același timp universalitatea teoretică în condiții ușoare. În învățarea profundă, straturilor li se permite, de asemenea, să fie eterogene și să se abată pe larg de la modelele conexioniste informate biologic, din motive de eficiență, formabilitate și înțelegere, de unde partea „structurată”. [2]

În învățarea profundă, fiecare nivel învață să-și transforme datele de intrare într-o reprezentare puțin mai abstractă și mai compusă. Într-o aplicație de recunoaștere a imaginii, intrarea brută poate fi o matrice de pixeli; primul strat reprezentativ poate abstractiza pixelii și codifica marginile; al doilea strat poate compune și codifica aranjamente ale muchiilor; al treilea strat poate codifica un nas și ochi; iar al

patrulea strat poate recunoaște că imaginea conține o față. Important, un proces de învățare profundă poate învăța ce caracteristici să plaseze în mod optim în ce nivel pe cont propriu. [2]

Pentru sarcinile de învățare supravegheate , metodele de învățare profundă elimină ingineria caracteristicilor, prin traducerea datelor în reprezentări intermediare compacte asemănătoare componentelor principale și derivă structuri stratificate care elimină redundanța în reprezentare. [2]

Algoritmii de învățare profundă pot fi aplicați sarcinilor de învățare nesupravegheate. Acesta este un beneficiu important, deoarece datele neetichetate sunt mai abundente decât datele etichetate. [2]

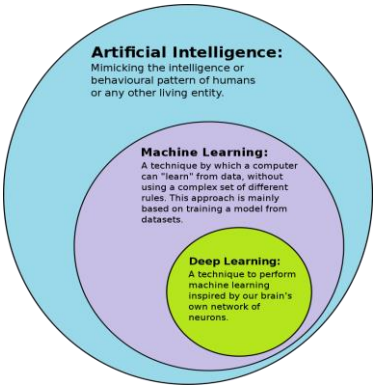

Fig. 3. Învățarea profundă este un subset de învățare automată și învățarea automată este un subset de inteligență artificială (AI) [2]

4. Analiza malware-ului

Pentru analiza malware-ului, învățarea automată s-a dovedit benefică și a fost utilizată de cercetătorii în domeniul securității și de companiile antivirus. Operațiunea generalizată a pașilor de învățare automată pentru analizele malware este prezentată în Figura 4.

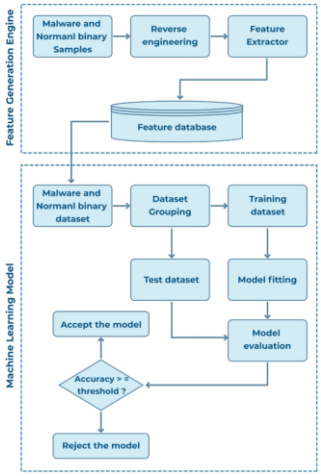

Fig. 4. Pași în învățarea automată pentru analizele malware [2]

Perspectiva de învățare automată a analizelor malware implică două componente majore: motorul de generare a caracteristicilor și modelul de învățare automată. Motorul de generare a funcțiilor începe cu colectarea de programe malware și eșantion normal. Apoi, aplicăm diverse tehnici de inginerie inversă, cum ar fi analiza și depanarea binare folosind coduri de program personalizate (codul cercetătorului folosind instrumente Linux, cum ar fi ObjDump sau alte biblioteci disponibile pentru alte sisteme de operare), surse deschise (analizor PE, Angr) sau instrumente comerciale (IdaPro) . Ingineria inversă este procesul invers al încercării de a realiza ceea ce programul este destinat să facă și să cunoască

structura programului. Hackerii îngreunează sarcina de inginerie inversă prin aplicarea diferitelor tehnici de ofuscare. Rezultatul etapei de inginerie inversă oferă unele date brute, care sunt preprocesate de componenta extractor de caracteristici. Aceasta oferă setul de date cu caracteristici curate. Cea de-a doua componentă, dar vitală, este modelul de învățare automată, al cărui flux de intrare este malware și set de date binar normal obținut din faza 1. Setul de date este împărțit în două subseturi ca set de date de antrenament și test. Această divizie se bazează pe alegerea analistului de program sau a echipei de cercetare. Validarea încrucișată K-fold și împărțirea de la 60 la 40% sunt cele mai frecvent utilizate. Setul de date de instruire se încadrează în modelul de învățare automată. Algoritmii de învățare automată implementați aici pot fi supravegheați sau nesupravegheați în funcție de natura setului de date. Modelul este apoi evaluat cu setul de date de testare. Precizia trebuie să fie peste valoarea pragului definită. Modelul de învățare automată este acceptat dacă acuratețea depășește valoarea pragului, altfel experimentul este reluat cu parametrii de reglare, remodelarea sau schimbarea abordării utilizate sau algoritmii utilizați. [2]

5. Perspectivele big data

Lumea crește cu o cantitate imensă de date denumită big data. Datele cresc adesea odată cu creșterea serviciilor și resurselor utilizate de diferiți indivizi și entități dintr-o organizație sau o companie. Un site de rețele sociale, un blog, istoricul navigării clienților, urmărirea comerțului electronic, traficul în rețea, tranzacții financiare, date medicale, toate se adaugă în fiecare secundă, producând tone imense de date. Acest lucru vine cu provocarea de gestionare a datelor și deschide ușa hackerilor și altor adversari. În această secțiune, discutăm provocările care au condus la o abordare de date mari pentru analiza malware-ului, cadrele de bază de date mari, cum sunt depășite provocările de către diferiți cercetători și tehnicile utilizate pentru a depăși provocările. Fiecare persoană fizică, companie sau organizație dorește să reducă sau să prevină daunele cauzate de atacurile malware. Vor să detecteze și să împiedice atacurile malware cât mai curând posibil. Acest lucru ar fi la îndemână cu surse mici de date, dar avem o cantitate imensă de date inevitabilă. Tehnicile primitive cu resurse limitate și capacități de procesare nu sunt capabile să gestioneze big data. Apache Hadoop și Apache Spark au făcut ca sarcina de analiză a datelor mari să fie convenabilă și eficientă. Printre aceste două cadre mari distribuite de date, cercetătorii folosesc în principal Apache Spark, deoarece susține modelul de învățare automată și procesarea în timp real pentru sarcina lor de analiză malware. A. Abordarea detecției În această secțiune, discutăm abordarea generală a detecției pentru analizele malware folosind cadre de date mari. Figura 5 prezintă elementele de bază pentru abordarea de detectare. [3]

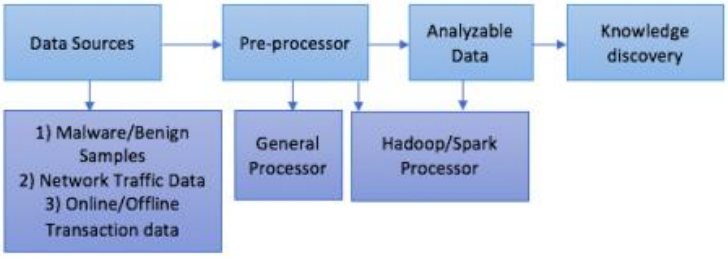

Fig. 5. Analize malware folosind cadre Big Data [3]

Sursele de date pot fi malware / fișiere executabile benigne, date despre traficul de rețea sau date despre tranzacții online / offline de unde dorim să detectăm o anomalie sau un model neobișnuit. Componenta preprocesorului este cea mai puternică componentă a acestei arhitecturi. Sursele de date intră în unele sarcini de pre-procesare utilizând capacități normale de procesare de calcul sau realizate exclusiv folosind cadrele Hadoop / Spark. Rezultatul este o dată curată și analizabilă în care aplicăm diferiți algoritmi de învățare automată sau de extragere a datelor pentru a găsi un model util, denumit și descoperirea cunoștințelor. [3]

6. Analiza programelor malware în IoT

Unul dintre cele mai notorii domenii de atac este inserarea și executarea unui virus pe dispozitivele IoT prin exploatarea vulnerabilităților existente în dispozitivele IoT. Înainte de a intra în detaliile malware-ului, este important să înțelegem tipurile de malware care pun în pericol securitatea IoT. Un malware este o amenințare care persistă ca urmare a vulnerabilităților menționate anterior și executată printr-o serie de atacuri. Tipurile comune de malware includ, dar nu se limitează la, bot, spyware, ransomware, adware, troian și virus, pentru a numi câteva. S-a descoperit prin multe studii că există dispozitive inteligente care sunt conectate la internet fără o protecție adecvată de securitate, care nu numai că reprezintă amenințări pentru dispozitivul în sine, ci permit, de asemenea, atacatorilor să utilizeze resurse pentru atacuri la scară masivă, precum DDoS. Tipurile de programe malware care au reușit să perturbe funcționalitatea normală a organizației, aplicației sau entităților țintă includ, dar nu se limitează la, NotPetya, Stuxnet, Cryptlocker, Red October, Night Dragon și așa mai departe. Acestea sunt atacurile generice de malware, în timp ce există familii optimizate de atacuri malware care vizează în special dispozitivele IoT. Astfel de atacuri includ WanaCry, Cryptlocker, Mirai, Stuxnet și așa mai departe. Acestea sunt atacurile malware care au costat industria sume uimitoare și alte pierderi, cum ar fi imaginea publică a companiei. [4]

Analiza malware bazată pe ML în IoT: S-a investigat detectarea și propagarea malware-ului în sistemul multimedia wireless (WMS) bazat pe IoT. Autorii au propus o abordare bazată pe cloud, pentru a detecta potențialele malware și propagarea acestora și au folosit joc diferențial bazat pe stări pentru a suprima malware-urile. După realizarea echilibrului Nash, autorii încearcă să găsească strategii optime pentru ca WMS să se apere împotriva malware-ului. În mod similar, a fost propusă o tehnică liniară bazată pe SVM pentru clasificarea malware-ului în IoT bazat pe Android. Deși SVM implică mai mult timp de clasificare datorită eliminării caracteristicilor inutile; cu toate acestea, este favorabil datorită complexității sale mai mici și a preciziei mai bune. Pentru a evalua acuratețea detecției modelului de detectare, autorii au luat în considerare diferite tipuri de malware și caracteristicile acestora. Rezultatele raportate de autori arată că SVM funcționează relativ mai bine decât alte clasificatoare pentru majoritatea malware-urilor investigate, unde rezultatul este peste 99%. În mod similar, se utilizează SVM și PCA pentru a detecta inserarea falsă de date în rețeaua inteligentă. Aplicarea acestei tehnici ar putea fi ușor încorporată în IoT. Autorii au folosit două metode. În prima metodă, datele etichetate sunt utilizate pentru învățarea supravegheată pentru instruirea SVM, în timp ce în a doua metodă, nu se folosește nicio formare. În plus, autorii au folosit învățarea nesupravegheată. Aceste tehnici ML sunt folosite pentru a izola datele manipulate de datele normale și pentru a detecta astfel atacurile. Rezultatele au arătat eficiența metodelor ML pentru detectarea defectuoasă a datelor care ar putea fi rezultatul fie al malware-ului, fie al altor tipuri de atacuri. [4]

Soluțiile de securitate bazate pe ML din rețelele IoT au anumite limitări: ML este utilizat pentru a crea modele, care sunt utilizate pentru proiectarea, testarea și instruirea seturilor de date. Acești algoritmi ML sunt utilizați pentru a identifica posibile modele și asemănări în seturi de date mari și pot face predicții în noile date obținute. Cu toate acestea, observăm că limitarea fundamentală a abordărilor ML este că, în mare parte, are nevoie de seturi de date pentru a învăța, iar apoi modelul învățat este aplicat datelor reale. Este posibil ca acest fenomen să nu cuprindă întreaga gamă de caracteristici și proprietăți ale datelor. În plus, datele pentru instruirea unui model reprezintă alte provocări de securitate și atacuri cibernetice. În acest sens, tehnicile DL au fost folosite pentru a aborda limitările tehnicilor ML. Abordând limitele ML, algoritmii DL au devenit cheia succesului în industria actuală. Remarcăm, de asemenea, că modelele DL au fost utilizate de o serie de giganți tehnologici, cum ar fi Apple folosește mecanismul DL în proiectul său Siri, Microsoft folosește mecanisme DL în Cortana, Amazon folosește algoritmi DL în Alexa și în mod similar Google Photos, Spotify și Grammarly, sunt toate conduse prin algoritmii DL. În plus, DL este, de asemenea, utilizat în domenii industriale, cum ar fi industria financiară pentru prezicerea prețului acțiunilor, industria de îngrijire a sănătății pentru refacerea medicamentelor testate pentru boli. Algoritmii DL câștigă avânt, dar aduc noi limitări: RL și DRL sunt unele dintre domeniile de cercetare promițătoare, folosite pentru extragerea automată a caracteristicilor complexe din

cantități mari de date nesupravegheate cu dimensiuni mari. În ciuda faptului că cercetările recente din aceste domenii au arătat o performanță extraordinară, există încă unele domenii în care sunt necesare îmbunătățiri mai optimizate, concentrate și specifice IoT. De asemenea, merită menționat faptul că aceste capacități de calcul combinate ale RL și DL implică cheltuieli de calcul și stocare. Prin urmare, în ciuda performanțelor lor, este posibil ca aceste metode să nu fie adecvate pentru dispozitivele IoT constrânse de resurse. [4]

6.1. Tehnici în curs de dezvoltare pentru ML în securitatea IoT

Rețelele Adversare Generative (GAN) și învățarea distribuită (Federated Learning) au câștigat recent atenție din partea comunității de cercetare.

1) Securitate IoT bazată pe Rețeaua Adversarială Generativă (GAN): GAN-urile au revoluționat metodele ML folosindu-le într-un cadru contradictoriu. În esență, două rețele neuronale sunt desfășurate într-un cadru în care concurează între ele într-un joc cu suma zero și ajung în cele din urmă la un echilibru. Cele două rețele neuronale componente includ un generator și un discriminator. Există un set de instruire pe care este instruit un discriminator. Apoi, rețeaua generatorului generează datele candidate, în timp ce discriminatorul evaluează datele fie clasificate corect, fie nu (de exemplu, maparea anumitor caracteristici pe etichetele lor respective). Scopul generatorului este de a „păcăli" discriminatorul, în timp ce scopul discriminatorului este de a se face mai puternic împotriva generatorului, rezultând în cele din urmă o convergență a rețelei. Cu alte cuvinte, generatorul încearcă să facă discriminatorul să creadă că datele de intrare au provenit din eșantion, mai degrabă decât din generator, în timp ce discriminatorul încearcă să afle dacă datele provin dintr-un eșantion real sau dintr-un generator. Structura generală a unui GAN este prezentată în Fig. 6. În contextul securității IoT, GAN-urile au produs rezultate remarcabile pentru a rezolva diferite probleme de securitate în rețelele IoT. [4]

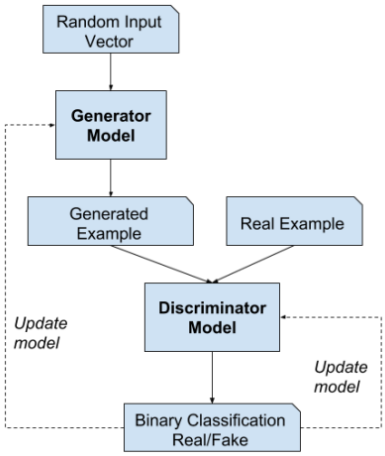

Fig. 6. Arhitectura generală a Rețelei Adversare Generative (GAN) [4]

2) Înclinare asociată și securitate IoT: Acest tip este un nou cadru de dezvoltare a modelului AI, distribuit pe dispozitive mobile. Oferă modele foarte personalizate și sigure, menținând confidențialitatea clientului / utilizatorului. În esență, în dezvoltarea modelelor FL, instruirea și evaluarea se fac fără acces direct la datele utilizatorului. Odată cu progresele recente în proiectarea cipurilor și a tehnologiei celulare, smartphone-urile (Samsung S9, Apple X) au o capacitate de calcul semnificativă și sunt echipate cu caracteristici AI. Prin urmare, majoritatea modelelor ML sunt capabile să ruleze pe aceste dispozitive mobile inteligente. Aceste dispozitive (ca parte a arhitecturii de calcul FL) pot descărca un model, care rulează local pe aceste dispozitive, iar modelul este îmbunătățit în continuare prin învățarea din datele locale stocate în aceste dispozitive. Aceste actualizări ale modelului îmbunătățit sunt rezumate de obicei sub forma parametrilor modelului și a ponderilor corespunzătoare. Aceste actualizări sunt criptate și trimise către dispozitivul principal (sau către cloud / server central). Ulterior, toate aceste actualizări sunt

calculate pentru a îmbunătăți modelul partajat. Această distribuție de analize și calcule grele pe dispozitive inteligente, spre deosebire de sistemul de calcul centralizat, va avea ca rezultat diverse beneficii. De exemplu, implementarea mai rapidă a modelului și economisirea timpului (răspuns mai rapid la schimbarea continuă a comportamentului clientului) în dezvoltarea unor motoare uriașe de recomandare (personalizate). De asemenea, îmbunătățește confidențialitatea utilizatorului (deoarece nu există acces direct la date brute în timpul dezvoltării și instruirii modelelor), iar actualizările individuale sunt neidentificate în cloud și în serverul central în timpul actualizărilor modelului. Aceste caracteristici principale fac din FL o alegere excelentă pentru rețelele mobile și distribuite (de exemplu, rețelele IoT) în ceea ce privește păstrarea confidențialității și o eficiență îmbunătățită; cu toate acestea, aduce câteva provocări. Este posibil ca rețelele IoT practice și în timp real să nu fie statice, iar configurația rețelei să se schimbe în continuare. Ca urmare, este posibil ca toate dispozitivele să nu participe complet până la convergența modelului FL. Întrucât modelele FL învață iterativ și se bazează pe dispozitivele participante, calitatea învățării FL poate fi pusă în pericol dacă puține dispozitive sunt abandonate în mijlocul procesului de învățare. [4]

7. Concluzii

Securitatea și confidențialitatea IoT sunt de o importanță capitală și joacă un rol esențial în comercializarea tehnologiei IoT. Soluțiile tradiționale de securitate și confidențialitate suferă de o serie de probleme care sunt legate de natura dinamică a rețelelor IoT. ML și mai precis tehnicile DL și DRL pot fi utilizate pentru a permite dispozitivelor IoT să se adapteze mediului lor dinamic. Aceste tehnici de învățare pot sprijini operațiunea de auto-organizare și, de asemenea, pot optimiza performanța generală a sistemului prin învățarea și procesarea informațiilor statistice din mediu (de exemplu, utilizatorii umani și dispozitivele IoT). Cu toate acestea, seturile de date necesare pentru algoritmii ML și DL sunt încă puține, ceea ce face ca evaluarea comparativă a eficienței soluțiilor de securitate bazate pe ML și DL să fie o sarcină dificilă. În această lucrare, am discutat despre tipuri de algoritmi, DL, precum și pericolele malware și perspectivele big data împreună cu tehnici de dezvoltare ML. Pentru a atenua unele dintre neajunsurile abordărilor de învățare automată a securității IoT, fundamentele teoretice ale DL și DRL vor trebui consolidate, astfel încât performanțele modelelor DL și DRL să poată fi cuantificate pe baza parametrilor precum complexitatea calculului, învățarea eficientă, precum și strategii de reglare a parametrilor.

8. Bibliografie

[1] „Machine learning,” 10 Mai 2021. [Interactiv]. Available: https://en.wikipedia.org/wiki/Machine_learning.

[2] „Deep learning,” 09 Mai 2021. [Interactiv]. Available: https://en.wikipedia.org/wiki/Deep_learning.

[3] S. Poudyal, Z. Akhtar, D. Dasgupta și K. D. Gupta, „Malware Analytics: Review of Data Mining, Machine Learning and Big Data Perspectives,” 20 Februarie 2020. [Interactiv]. Available: https://ieeexplore.ieee.org/document/9002996 .

[4] F. Hussain, R. Hussain, S. A. Hassan și E. Hossain, „Machine Learning in IoT Security: Current Solutions and Future Challenges,” 08 Aprilie 2020. [Interactiv]. Available: https://ieeexplore.ieee.org/document/9060970 .