Programul 2 - Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare

Subprogramul 2.1 - Competitivitate prin cercetare, dezvoltare și inovare

Tip proiect: Soluții

Titlul proiectului: 4. Instrumente automate de detecție a comportamentelor anormale în rețelele de calculatoare

Coordonator: Conf. Univ. Dr. Paul Irofti

RAPORT STIINTIFICO-TEHNIC

1. Organizațiile partenere in proiect:

1.1. **UNIVERSITATEA DIN BUCUREȘTI** (***UB***), cu sediul în București, sector 5, Șoseaua Panduri, nr. 90, cod fiscal (CUI) 4505502, cont bancar IBAN: RO66TREZ70520F332000XXXX, Trezoreria Sector 5, reprezentata prin RECTOR Marian Preda și DIRECTOR DE PROIECT Paul Irofti, email: paul.irofti@fmi.unibuc.ro, în calitate de CONDUCATOR PROIECT

1.2. **NEXTGEN SOFTWARE SRL** (***NEXT***), cu sediul în București, Sectorul 1, Strada STOLNICULUI, Nr. 6-10, Etaj 5, Ap. 16, înregistrată la Registrul Comerțului sub nr J40/8038/30.06.2015, cod fiscal (CUI) RO 34720987, cont RO 75 BTRL RONC RT03 0366 6701, deschis la Banca Transilvania, reprezentata prin ADMINISTRATOR - Tiberiu Leta și Responsabil proiect Partener – dr. ing. Ana-Maria Paraschiv, în calitate de PARTENER 1

ETAPA I. Începerea proiectului și documentare soluții

ACTIVITĂȚI:

1.1. Începerea execuției serviciilor de conducere și realizare a proiectului, conform prevederilor contractului de finanțare; management proiect
1.2. Documentare și identificare de soluții pe baza ofertelor existente pe piața de profil

Termen realizare: 31.12.2021

Livrabile/Rezultate așteptate:

1.1. Kick-off meeting; demarare achiziții; scoaterea la concurs a pozițiilor vacante și angajarea personalului
1.2. Documentație - 1 Raport de cercetare (state-of-the art) - UB, incluzând: baze de date de antrenare UB/NEXT; programe scurte demonstrative ce folosesc tool-urile identificate

REZULTATELE ETAPEI I

1.1.1. Kick-off meeting: au fost realizate două întâlniri de tip kick-off meeting.

1.1.2. Au fost realizate 3 întâlniri organizatorice împreună cu beneficiarul.

1.1.3. Demarare achiziții

UB a inclus pe lista de investiții pentru anul 2022 echipamentele aferente proiectului NetAlert. Next nu are planificate achiziții pentru Etapa I. A fost reanalizat planul de achiziții pentru Etapele următoare.

1.1.4. Scoaterea la concurs a pozițiilor vacante și angajarea personalului

UB a scos la concurs toate cele 7 poziții vacante aferente echipei locale:

1. Research Gov.Ro
    - Postdoctorand: https://jobs.research.gov.ro/anunt.php?id=4760
    - Asistent cercetător (doctorand): https://jobs.research.gov.ro/anunt.php?id=4761
    - Programator (masterand): https://jobs.research.gov.ro/anunt.php?id=4762
2. Euraxxes:
    - Postdoctorand: https://www.euraxess.gov.ro/jobs/706258
    - Asistent cercetător (doctorand): https://www.euraxess.gov.ro/jobs/706259
    - Programator (masterand): https://www.euraxess.gov.ro/jobs/706260

Nextgen Software a demarat procedura de înregistrare a companiei pe site-urile https://jobs.research.gov.ro și https://www.euraxess.gov.ro, în vederea scoaterii la concurs a celor două posturi vacante în cadrul proiectului.

Concluzie: Obiectivul activității 1.1. *Organizarea unei sesiuni de kick-off meeting de către UB împreună cu echipa de implementare și revizuirea planificării acțiunilor tehnice și administrative în primele 15 zile de la semnarea contractului de finanțare (T0)* a fost îndeplinit.

1.2.1. Raport de cercetare (state-of-the art)

Universitatea din București (Coordonator) și Nextgen Software SRL (Partener 1)

Evoluția rapidă a tehnologiilor digitale de comunicație conduce, în contextul zilelor noastre, la confruntări cu volume imense ale datelor de orice natură (e.g. video, voce, text, senzori etc.) și de asemenea la expansiunea rețelelor de calculatoare. Nevoia securității, confirmată de creșterea numărului de atacuri din ultimii 6 ani [19], a motivat cercetările în domeniul Sistemelor de Detecție a Intruziunilor (SDI). Un SDI reprezintă un cadru software proiectat cu scopul detecției unui atac asupra unui nod (sau mai multor noduri) din rețea și alertarea corespunzătoare a operatorului de sistem.

Un atac definește o operație realizată cu scopul de a compromite rețeaua. Structura slabă a rețelei, neglijența utilizatorilor sau configurări greșite ale componentelor software și hardware aduc, de cele mai multe ori, vulnerabilități semnificative asupra nodurilor [23]. În [23], autorii disting următoarele atacuri:

- *Denial of Service (DoS)*: un tip de atac bazat pe obținerea de drepturi asupra resurselor unei rețele sau gazde cu scopul de a întrerupe mediul ambiental de calcul și a opri un anume serviciu.
- *Probe*: operație folosită pentru a colecta date și informații legate de rețea, cum ar fi numărul, tipul de mașini, tipurile de software sau aplicații folosite. Determină adesea doar un pas premergător atacului propriu-zis.
- *User to Root(U2R)*: atac lansat pentru obține acces ilegal la resurse sau conturi. De exemplu, folosind rețelele sociale, atacatorul accesează un cont de utilizator obișnuit, prin intermediul căruia accede la un nivel superior de autorizare.
- *Remote to User (R2U)*: în acest caz, accesul obținut ilegal la conturile de utilizator, de pe o mașină locală, este folosit pentru a transmite mesaje în rețea.

În general, tehnicile de detecție statistică presupun elaborarea unui estimator antrenat pe un set de date cunoscut, cu scopul de a realiza performanțe similare de detecție pe seturi noi de date necunoscute. Detecția de intruziuni este des interpretată în literatură ca o problemă de detecție (statistică) de anomalii într-un set de date, care în particular reflectă stările rețelei (e.g. flux de pachete) la diferite momente de timp. Principalele tipuri de anomalii identificate în literatură sunt:

- *Anomalii singulare*: o anomalie singulară reprezintă un punct individual anormal într-un set de date. De exemplu, o tranzacție ilegală în detecția de fraude sau imaginea unui produs deteriorat în lanțul de producție. Acesta este cu siguranță cel mai comun tipar de anomalii analizat în literatură.
- *Anomalii contextuale*: o anomalie condițională sau contextuală este o instanță a datelor ce se poziționează anormal în timp, spațiu sau ansamblul conexiunilor dintr-o rețea. Prețul de 1 dolar per acțiune Apple a fost normal în 1997, însă în zilele noastre ar ilustra o anomalie. Alte exemple includ anomalii spațiale, spațio-temporale sau de tip graf.
- *Anomalii colective*: anomalii de grup sau contextuale ilustrează un subset anormal de date (interdependente). Atacurile în rețea care provoacă comportamente anormale într-o subregiune a acesteia este un exemplu de anomalie colectivă.

Tip atac \ Tip anomalie	**DoS**	**Probe**	**U2R**	**R2U**
Singulară	-	-	✓	✓
Contextuală	-	✓	-	-
Colectivă	✓	-	-	-

Tabel 1: Asocierea atacurilor cu clasele de anomalii

Literatura bogată a SDI se poate structura în multiple moduri, iar clasificări ale diferitelor paradigme existente pot fi găsite în [18-25, 27]. Mai departe prezentăm o succintă taxonomie, care nu este sub nici o formă completă, însă poate oferi o vedere de ansamblu asupra domeniului.

Modele de clasificare. În [6] sunt analizate performanțele a 3 metode de extragere de atribute (nesupervizate): Analiza Componentelor Principale (PCA), Autoencoder (AE) și Arbori de Izolare (IF) pentru identificarea tiparelor anormale în traficul de rețea. Testele prezentate folosesc două seturi de date generate sintetic, în care s-au simulat anomalii de pierdere, duplicare și reordonare de pachete. Prin intermediul unui software dedicat, autorii simulează traficul de rețea (între 2 sau mai multe mașini) și agregă setul de date folosind analiza Tstat. Metodele clasice de reducție dimensională PCA și AE nu sunt suficiente pentru a scoate în evidență anomaliile plantate, însă metoda IF detectează pachetele reordonate cu eroare de 0.05%. Aparent celelalte tipuri de anomalii au rezistat detecției celor 3 metode abordate.

Algoritmii de clasificare cu clasa unică au fost folosiți cu succes în detecția pachetelor anormale [7,26]. După etapa de extragere de atribute, care generează o familie de descriptori ai setului de date, autorii din [7] antrenează de asemenea familii (*ensemble*) de clasificatori OC-SVM pe descriptori diferiți. Testele confirmă îmbunătățirea performanțelor de robustețe după agregarea rezultatelor generate de fiecare membru din familie. De asemenea, în [26] modelul SVDD este adaptat la paradigma de învățare activă, unde observațiile cu statut difuz sunt etichetate de un expert și apoi procesul de antrenare este reluat luând în calcul noile etichete. În teste sunt folosite date ce ilustrează traficul http, peste care au fost plantate atacuri sintetice (e.g. buffer overflow, cross-site scripting, code injection etc.), iar varianta activă a SVDD detectează un număr superior de atacuri comparativ cu metoda clasică.

Modele de reconstrucție. În [6,9,37] tehnica reducției dimensionale de tip PCA este utilizată pentru a crea un clasificator nesupervizat. Mai pe larg, în [9] sunt folosite funcții de clasificare compuse din primele *p* componente principale majore și ultimele *q* componente minore din descompunerea spectrală a setului de date. În cazul în care scorul standard asociat acestor componente depășește un prag ales empiric, observația face parte din clasa normală. Testele pe setul de date KDD99 (conține celor 4 tipuri de atac din Tabelul 1) confirmă o precizie de detecție de aprox. 99%. Rezultatele din [37] confirmă, de asemenea, eficiența reducției dimensionale (de tip PCA) pe setul de date *etichetat* NSL-KDD. Aici, problema detecției de anomalii este redusă la una de învățare supervizată. După reducția celor 41 de dimensiuni, autorii aplică metoda Arborilor Aleatori (Random Forests) pentru a finaliza etapa de antrenare, păstrând un număr redus de atribute. Observând că varianța optimală este atinsă pentru reducția la 10 dimensiuni, autorii prezintă diferențe neglijabile de acuratețe între detecția pe datele originale (cu 41 de atribute) și cea pe datele rezultate din reducția optimală, în ambele cazuri acuratețea de clasificare depășind 99%.

Modele probabilistice. Modelele bazate pe mixturi de distribuții gaussiene (GMM) apar în [14], unde este dezvoltat un sistem Bayesian de detecție a intruziunilor în rețea. Aici, în locul unei diferențieri binare a traficului : normal - anormal, parametrii asociați distribuțiilor estimate permit alocarea unei clase pentru observațiile normale și clase multiple pentru cele anormale. Evaluarea empirică folosește seturi de date mai recente (comparativ cu lucrările precedente): KYOTO 2006 și ISCX. În [12,38], se analizează clasificarea (sau clustering) profilelor de trafic după aplicația de proveniență. De exemplu, Thunderbird utilizează protocoale de trafic specifice agenților de e-mail (SMTP, POP etc.), însă sub influența intruziunilor, poate genera și alte protocoale (application-layer) precum HTTP. Modelul GMM este optimizat utilizând algoritmul

de Maximizare a Expectanței, iar parametrii estimați servesc la clasificarea ierarhizată de trafic. Testele evidențiază rezultate favorabile ale GMM pentru detecția intruziunilor.

Modele de clustering (și co-clustering). Tehnicile de clustering nesupervizate scot în evidență în mod natural observațiile neconforme cu majoritatea. Prin definiție, ele partiționează un set de date în subseturi cu membri similari (clusters). Pentru a îmbunătăți performanțele, această partiționare se realizează nu doar după observații, ci și după atribute. În felul acesta, într-un set de date, algoritmii de clustering partiționează coloanele, iar schemele de co-clustering partiționează liniile [8,10,11,27,28]. În [8] se arată utilitatea algoritmilor de clustering/co-clustering pentru identificarea anomaliilor colective în traficul de rețea (e.g. flooding DoS). De obicei, succesul acestora necesită un număr semnificativ de anomalii în setul de date. Pentru a elimina această presupunere, autorii din [8] folosesc parametrul Hurst pentru a genera un scor și a oferi o ordonare pe cluster-e, iar cluster-ul cu cel mai mic scor este considerat anomalie. Testele constată o îmbunătățire a ratei TPR comparativ cu alte metode de clustering. În [28] este prezentat un SDI în 4 module, bazat pe o strategie de clustering ierarhizat. După extracția de cluster-e pure, modulele permit rafinarea acestora pentru a evidenția tiparele necunoscute și, mai mult, pentru a clasifica aceste anomalii. Strategia este testata pe setul de date KDD '99.

Modele de învățare profundă. Modelele reprezentative folosite curent în tehnicile de învățare profundă sunt rețelele neurale cu un număr sporit de straturi ascunse. O dată cu succesul recent al rețelelor neurale profunde în multe dintre domeniile inginerești, utilizarea lor în componența SDI-urilor a prezentat un interes major [15,16,17,19]. O vedere limitată asupra contribuțiilor învățării profunde la SDI se poate găsi în [19,24,27]. De asemenea, o analiză comparativă prezentată în [36] evaluează SDI bazate pe diferite metode de clasice și profunde implementate pe arhitecturi GPU. Seturile de date de test considerate de autori include NSL-KDD, iar rezultatele experimentale arată ca LSTM și Rețelele Neurale Convoluționale Profunde ating o acuratețe mai mare comparativ cu alte modele.
În [19], pe lângă o listare amănunțită a rezultatelor de acuratețe obținute în literatură de către metode profunde precum Autoencoder și Rețele Neurale Artificiale (RNA), autorii realizează o comparație de performanță pe principalele seturi de date redate în lista de mai jos. În evaluare, dintre metodele profunde, agregarea metricilor (precizie, acuratețe, timp antrenare, timp test) recomandă performanța RNA ca superioară în raport cu restul metodelor. Adâncimea RNA nu se remarcă decisiv în evaluare. Surprinzător, performanțe similare sunt atinse doar de metoda clasică a Arborilor Aleatori (Random Forests), însă timpii acesteia la testare sunt semnificativ mai mari decât în cazul RNA.

O listare extensivă a diverselor modele de învățare profundă moderne, folosite pentru crearea de SDI, apare de asemenea în [27]. Un SDI bazat pe Rețele Neurale Recurente (RNR) a fost propus de [35] în contextul clasificării binare și multi-clasă a setului de date NSL-KDD. Modelul a fost testat variind numărul de noduri ascunse și rata de învățare. Rezultatele arată că acuratețea este oarecum influențată de acești 2 factori, iar modelul propus arată performanțe superioare celor clasice și altor metode RNR. Principalul dezavantaj este costul crescut de calcul, care aduce un timp mare de antrenare, și ratele scăzute de detecție pentru clasele de atacuri singulare precum U2R.

Principalele seturi de date utilizate pentru evaluarea SDI:

- **KDD '99 [29]**: Unul dintre cele mai populare și utilizate seturi de date pentru SDI. Conține aproximativ 5 mil. de instanțe pentru antrenare și 2 mil. instanțe pentru test. Fiecare instanță are 41 de atribute și este etichetată ca normală sau atac. Corpusul conține atacurile din Tabelul 1.
- **NSL-KDD [30]**: Set obținut prin rafinarea setului KDD '99 și eliminarea anumitor redundanțe.
- **KYOTO 2006+ [31]**: Set de date creat prin înregistrarea traficului de rețea de la Universitatea din Kyoto. Ultima versiune include traficul dintre anii 2006-2015. Fiecare instanță are 24 atribute, din care 14 sunt derivate din atributele specifice setului KDD '99, iar restul de 10 sunt atribute adiționale.
- **UNSW-NB15 [32,33]**: Set de date creat de Australian Center for Cyber Security. Conține aprox. 2 mil. de instanțe cu 49 atribute fiecare. Tipurile de atacuri aflate în corpus extinde lista Tabelului 1: Worms, Shellcode, Port Scans, Exploits, Fuzzers, DoS, Probe.
- **CIC-IDS2017 [34]**: Set de date creat de Canadian Institute of Cyber Security în 2017. Atributele pe fiecare instanță includ amprente de timp, adrese IP ale sursei și destinației, protocoale etc.
- **CSE-CIC-IDS2018 [34]**: Acest set de date este o actualizare a CIC-IDS2017 cu profile de utilizatori care conțin reprezentări abstracte ale evenimentelor.

Alte biblioteci Python pentru învățare automată și detecție de anomalii:

- Python Outlier Detection (PyOD) [1]
- Accelerating Large-scale Unsupervised Heterogeneous Outlier Detection (SUOD)[2]
- Skyline real time anomaly detection system (Skyline) [3]
- Automated Time-series Outlier Detection System (TODS) [4]
- Python Streaming Anomaly Detection (PySAD) [5]
- Graphomaly: bibliotecă de detecție a anomaliilor în grafuri

Primele teste de colectarea datelor și testarea sistemelor de stocare și procesare pentru sistemele de învățare automată au fost realizate.

În cadrul acestui proiect, au fost realizate deja primele măsurători ale traficului de rețea. În această primă fază a proiectului se extrag datele pentru:

- Physical layer;
- Data link layer;
- Network layer transport layer.

Toate datele colectate sunt interpretate, parsate, organizate și populează câmpurile următoare în setul de date pentru algoritmii de învățare:

- Pentru physical layer :
  - HeaderLength : lungimea pachetului efectiv
  - HeaderCaptureLength: lungimea pachetului capturat
- Pentru data link layer:
  - SrcMAC: adresa fizică (MAC) sursă
  - DestMAC: adresa fizică (MAC) destinație

  - EthernetType: tipul de date prezent (ex: 0x0800=ipv4)

- Pentru network layer :
  - SrcPort: portul sursă
  - DestPort: portul destinație
  - Protocol: protocolul folosit

- Pentru transport layer :
  - SrcIP: adresa IP sursă în format ipv4 sau ipv6 dacă este prezent
  - DestIP: adresa IP destinație în format ipv4 sau ipv6 dacă este prezent în funcție de protocol
  - UpperLayerChecksum: valoare Checksum pentru headerul transport (UDP/TCP)
  - TCPSequence: TCP squence number (în cazul in care face parte dintr-un pachet mai mare). Prezent doar în cazul în care transport layer este TCP
  - TCPack: TCP ACK (parte din headerul TCP) prezent doar în cazul în care transport layer este TCP
  - TCPFlags: TCP flags, valoare numerică 0-255 octet cu fiecare bit reprezentând unul din flag-uri prezente. Prezent doar în cazul în care transport layer este TCP

În plus, la nivelul pachetului este trimisă și data în format utc_iso8601 cu rezoluție de nanosecunde pe câmpul UTC_ISO8601, și câmpul Content care conține maxim 100 de carectere din conținutul mesajului de tip UDP (codificat Base64).

În prezent dispunem de 16 câmpuri în total care descriu proprietăți ale pachetelor capturate. În următoarele teste de rețea ne așteptăm să putem captura mai multe câmpuri, dar aceste sunt deja garantate la colectarea actuală. Numărul de câmpuri este extrem de important deoarece afectează calitatea rezultatelor generate de algoritmii de învățare. Atenție specială va fi dedicată colectării și generării de proprietăți/câmpuri pentru pachete colectate (*feature collection and engineering*).

Datele sunt colectate inițial în format JSON iar apoi parsarea, completarea câmpurilor lipsă/goale și transformarea/normalizarea datelor are loc. Datele sunt organizate într-o formă tabelară de tipul CSV. Acesta este setul de date utilizat pentru algoritmii de învățare automată (biblioteci python) descriși mai devreme.

Concluzie: Obiectivul activității 1.2. *Realizarea unei documentații state-of-the-art în perioada de timp T0+3 luni, de către membrii echipei de implementare* a fost îndeplinit

Referințe

[1] Y. Zhao, Z. Nasrullah and Z. Li, (2019). *PyOD: A Python Toolbox for Scalable Outlier Detection*. Journal of machine learning research (JMLR), 20(96), pp.1-7.

[2] Y. Zhao, X. Hu, C. Cheng, C. Wang, C. Wan, W. Wang, J. Yang, H. Bai, Z. Li, C. Xiao, Y. Wang, Z. Qiao, J. Sun, L. Akoglu (2021), *SUOD: Accelerating Large-Scale Unsupervised Heterogeneous Outlier Detection*, Proceedings of Machine Learning and Systems 3.

[3] https://github.com/earthgecko/skyline

[4] K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez and X. Hu (2021), *TODS: An Automated Time Series Outlier Detection System*, Proceedings of the AAAI Conference on Artificial Intelligence, 35(18), 16060-16062.

[5] F.S. Yilmaz and S. Kozat (2020), *PySAD: A Streaming Anomaly Detection Framework in Python*, arXiv preprint arXiv:2009.02572.

[6] M. Kiran, C. Wang, G. Papadimitriou, A. Mandal and E. Deelman (2020), *Detecting anomalous packets in network transfers: investigations using PCA, autoencoder and isolation forest in TCP*. Machine Learning, 109(5).

[7] R. Perdisci, G. Gu and W. Lee (2006), *Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems*. In IEEE Sixth International Conference on Data Mining (ICDM'06) (pp. 488-498).

[8] M. Ahmed (2018), *Collective anomaly detection techniques for network traffic analysis*. Annals of Data Science, *5*(4), 497-512.

[9] M. L. Shyu, S.C. Chen, K. Sarinnapakorn and L. Chang (2003), *A novel anomaly detection scheme based on principal component classifier*. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

[10] M. Ahmed and A. N. Mahmood (2014), *A novel approach for outlier detection and clustering improvement*. In 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA) (pp. 577-582).

[11] I. Syarif, A. Prugel-Bennett and G. Wills (2012), *Unsupervised clustering approach for network anomaly detection*. In International conference on networked digital technologies (pp. 135-145). Springer, Berlin, Heidelberg.

[12] H. Alizadeh, A. Khoshrou and A. Zuquete (2015), *Traffic classification and verification using unsupervised learning of Gaussian Mixture Models*. In *2015* IEEE international workshop on measurements & networking (M&N), 1-6.

[13] De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., and Prieto, B. (2015), *PCA filtering and probabilistic SOM for network intrusion detection*, Neurocomputing, 164, 71-81.

[14] Alhakami, W., Alharbi, A., Bourouis, S., Alroobaea, R., and Bouguila, N. (2019), *Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection*. IEEE Access, *7*, 52181-52190.

[15] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., and Venkatraman, S. (2019), *Deep learning approach for intelligent intrusion detection system*, IEEE Access, *7*, 41525-41550.

[16] Shone, N., Ngoc, T. N., Phai, V. D., and Shi, Q. (2018), *A deep learning approach to network intrusion detection*, IEEE transactions on emerging topics in computational intelligence, *2*(1), 41-50.

[17] Diro, A., and Chilamkurti, N. (2018), *Leveraging LSTM networks for attack detection in fog-to-things communications*, IEEE Communications Magazine, *56*(9), 124-130.

[18] Di Mauro, M., Galatro, G., Fortino, G., and Liotta, A. (2021). *Supervised feature selection techniques in network intrusion detection: A critical review*. Engineering Applications of Artificial Intelligence, 101, 104216.

[19] Gamage, S., and Samarabandu, J. (2020). *Deep learning methods in network intrusion detection: A survey and an objective comparison*. Journal of Network and Computer Applications, 169, 102767.

[20] Abaimov, S., and Bianchi, G. (2021). *A survey on the application of deep learning for code injection detection.* Array, 11, 100077.

[21] Bhuyan, M. H., Bhattacharyya, D. K., and Kalita, J. K. (2012). Survey on incremental approaches for network anomaly detection. *arXiv preprint arXiv:1211.4493*.

[22] Wang, J., Rossell, D., Cassandras, C. G., and Paschalidis, I. C. (2013). *Network anomaly detection: A survey and comparative analysis of stochastic and deterministic methods.* In 52nd IEEE Conference on Decision and Control (pp. 182-187). IEEE.

[23] Ahmed, M., Mahmood, A. N., and Hu, J. (2016). *A survey of network anomaly detection techniques.* Journal of Network and Computer Applications, 60, 19-31.

[24] Chou, D., and Jiang, M. (2021). *A Survey on Data-driven Network Intrusion Detection*. ACM Computing Surveys (CSUR), 54(9), 1-36.

[25] Moustafa, N., Hu, J., and Slay, J. (2019). *A holistic review of network anomaly detection systems: A comprehensive survey.* Journal of Network and Computer Applications, 128, 33-55.

[26] Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. (2009). *Active learning for network intrusion detection.* In Proceedings of the 2nd ACM workshop on Security and artificial intelligence (pp. 47-54).

[27] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2021). *Network intrusion detection system: A systematic study of machine learning and deep learning approaches.* Transactions on Emerging Telecommunications Technologies, 32(1).

[28] H. Yao, D. Fu, P. Zhang, M. Li and Y. Liu (2019), *MSML: A Novel Multilevel Semi-Supervised Machine Learning Framework for Intrusion Detection System*, in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1949-1959, doi: 10.1109/JIOT.2018.2873125.

[29] Bay S. The UCI KDD Archive [http://kdd. ics. uci. edu]. Irvine, CA: University of California, Department of Computer Science; 1999.
[30] Tavallaee M., Bagheri E., Lu W. and Ghorbani AA (2009). *A detailed analysis of the KDD CUP 99 data set*. Paper presented at: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications. Ottawa, ON, Canada: IEEE; 1-6.
[31] Song J., Takakura H., Okabe Y., Eto M., Inoue D. and Nakao K. (2011), *Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation*. Paper presented at: Proceedings of the 1stWorkshop on Building Analysis Datasets and Gathering Experience Returns for Security. Salzburg Austria; pp:29-36.
[32] Moustafa N. and Slay J. (2015) , *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*. Paper presented at: Proceedings of the Military Communications and Information Systems Conference (MilCIS). Canberra, ACT,
Australia: IEEE; pp:1-6.
[33] Moustafa N. and Slay J. (2016), *The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set*. Inf Sec J A Global Perspect.; 25(1-3):18-31. https://doi.org/10.1080/19393555.2015.1125974
[34] Sharafaldin I., Lashkari A.H. and Ghorbani A.A. (2018) , *Toward generating a new intrusion detection dataset and intrusion traffic characterization*. Paper presented at: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP). Madeira, Portugal; pp:108-116
[35] Yin, C., Zhu, Y., Fei, J., and He, X. (2017). *A deep learning approach for intrusion detection using recurrent neural networks*. IEEE Access, 5, 21954-21961.
[36] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., and Han, K. (2018). *Enhanced network anomaly detection based on deep neural networks.* IEEE access, 6, 48231-48246.
[37] Vasan, K. K., and Surendiran, B. (2016). *Dimensionality reduction using principal component analysis for network intrusion detection*. Perspectives in Science, 8, 510-512.
[38] Alizadeh, H. and Zúquete, A. (2016). *Traffic classification for managing applications' networking profiles*. Security and Communication Networks, 9(14), 2557-2575.