Proiect integrator de cercetare în securitatea calculatoarelor

Detecție a atacurilor DDoS prin algoritmi cu architectură AI

Nuțeanu Dorin

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Introducere

În contextul creșterii exponențiale a amenințărilor cibernetice, atacurile de tip Distributed Denial of Service (DDoS) au devenit unele dintre cele mai distructive metode utilizate de atacatori pentru a perturba funcționarea infrastructurilor digitale critice. Aceste atacuri generează un volum masiv de trafic malițios asupra unei rețele sau a unui server, cu scopul de a-i satura resursele și a împiedica accesul utilizatorilor legitimi. Detectarea și prevenirea atacurilor DDoS reprezintă o provocare majoră pentru specialiștii în securitate cibernetică, având în vedere caracterul dinamic și evolutiv al acestor amenințări.

Acest proiect integrator de cercetare își propune dezvoltarea și evaluarea unor modele de detecție a atacurilor DDoS bazate pe algoritmi de inteligență artificială (AI). Scopul principal este de a analiza eficiența diferitelor tehnici de învățare automată și rețele neuronale în identificarea traficului anormal, utilizând baze de date cu trafic real și simulat, care conțin atacuri malițioase.

Prin explorarea algoritmilor avansați de clasificare și detecție a anomaliilor, proiectul va investiga performanța unor modele capabile să analizeze traficul de rețea și să identifice rapid și precis un atacurDDoS. Analiza va include preprocesarea datelor, selecția caracteristicilor relevante și metricilor de evaluare pentru a asigura o detecție robustă și scalabilă.

Capitolul 1
Ce este un atac DDoS

Un atac de tip Distributed Denial of Service (DDoS) este un tip de atac cibernetic care are scopul de a perturba funcționarea normală a unui sistem, serviciu sau rețea țintă prin supraîncărcarea acestuia cu un volum masiv de trafic. Spre deosebire de un atac Denial of Service (DoS) tradițional, care provine dintr-o singură sursă, un atac DDoS este lansat din mai multe dispozitive compromise, adesea formând un botnet, o rețea de calculatoare infectate, controlate de la distanță de un atacator.Atacurile DDoS reprezintă una dintre cele mai distructive amenințări în domeniul securității cibernetice, afectând companii, guverne și indivizi. Motivele din spatele acestor atacuri variază de la șantaj financiar până la război cibernetic, iar atacatorii folosesc multiple metode pentru a suprasolicita țintele. Pe măsură ce atacurile DDoS continuă să evolueze, organizațiile trebuie să adopte strategii avansate de detecție și atenuare, inclusiv tehnici bazate pe învățare automată (Machine Learning), pentru a se proteja eficient împotriva acestor amenințări.

O rețea botnet este o colecție de dispozitive compromise, cunoscute sub numele de boți, care sunt controlate de la distanță de un atacator numit botmaster. Aceste dispozitive, ce pot include computere, telefoane mobile, servere sau dispozitive IoT, sunt infectate prin diverse metode, cum ar fi e-mailuri de tip phishing, exploatarea vulnerabilităților software sau descărcarea de fișiere malițioase. Odată infectate, ele comunică cu un server de comandă și control (C&C), fie centralizat sau descentralizat prin rețele peer-to-peer (P2P), pentru a primi instrucțiuni privind atacurile cibernetice, precum DDoS, spam, furt de date sau minare ilegală de criptomonede. Pentru a evita detectarea, botnet-urile folosesc tehnici avansate, cum ar fi criptarea comunicațiilor, rootkits și mutarea frecventă între servere C&C. Botnet-urile pot fi clasificate în două categorii principale: centralizate și descentralizate. Cele centralizate sunt coordonate printr-un server unic C&C, utilizând protocoale precum IRC, HTTP sau HTTPS pentru a transmite instrucțiuni boților. Acest model are un dezavantaj major, deoarece, odată ce serverul C&C este identificat și eliminat, întreaga rețea devine inutilă. Pe de altă parte, botnet-urile descentralizate P2P funcționează fără un punct unic de control, permițând boților să comunice direct între ei, ceea ce le face mai greu de detectat și eliminat. Acest model le oferă o rezistență sporită împotriva acțiunilor de contracarare. Exemple notabile de botnet-uri P2P sunt ZeroAccess și Gameover Zeus, care au fost utilizate pentru atacuri cibernetice de mare amploare.

Capitolul 2
Cum funcționează atacurile DDoS

Atacurile DDoS presupun supraîncărcarea unui sistem țintă cu un volum mare de solicitări, depășind capacitatea acestuia de a le gestiona și ducând la refuzul serviciului pentru intreaga retea de utilizatori.

2.1 Atacuri bazate pe volum

Aceste atacuri au scopul de a satura lățimea de bandă a țintei prin trimiterea unui număr foarte mare de pachete de date, blocând astfel accesul utilizatorilor legitimi. Un exemplu comun este supraincarcarea UDP (UDP Floods), unde atacatorii trimit un număr mare de pachete User Datagram Protocol (UDP) către porturi aleatorii, forțând sistemul țintă să consume resurse pentru a le gestiona. În mod similar, inundațiile ICMP (Ping Floods) implică trimiterea unui volum excesiv de cereri Internet Control Message Protocol (ICMP), epuizând astfel lățimea de bandă și capacitatea de procesare a sistemului atacat.

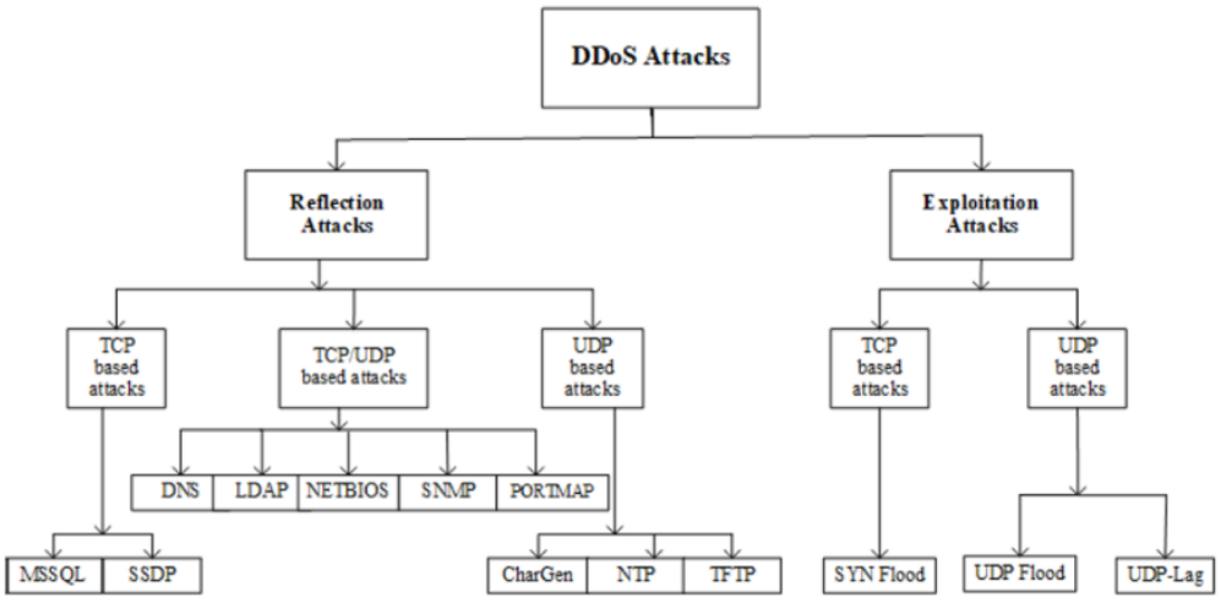

Figura 2.1: Tipuri de atacuri DDoS, prezente in baza de date CIC-DDoS2019

O metodă mai sofisticată este amplificarea DNS, unde atacatorii exploatează servere DNS

vulnerabile pentru a amplifica traficul. Aceștia trimit cereri mici care generează răspunsuri mult mai mari, reușind astfel să copleșească resursele țintei cu un trafic disproporționat de mare.

2.2 Atacuri bazate pe protocoale

Aceste atacuri exploatează vulnerabilitățile protocoalelor de rețea, epuizând resursele serverului prin manipularea proceselor de comunicare.

Printre cele mai utilizate metode se numără atacurile SYN Flood, care afectează procesul de handshake al protocolului TCP. Atacatorii trimit un număr mare de cereri de conexiune (SYN requests), dar nu finalizează procesul, lăsând resursele serverului blocate și incapabile să răspundă altor solicitări.

O variantă similară este atacul ACK Flood, unde atacatorii trimit un număr mare de pachete ACK (Acknowledgment), suprasolicitând astfel capacitatea serverului de a procesa conexiunile existente.

2.3 Atacuri la nivel de aplicație

In cadrul acestori atacuri se vizează direct aplicațiile, mai degrabă decât infrastructura rețelei, încercând să epuizeze resursele software ale serverului țintă.

Un exemplu frecvent este supraincarcarea HTTP (HTTP Floods), unde atacatorii trimit un număr mare de cereri HTTP către un server web, forțându-l să proceseze un volum excesiv de trafic. Acest lucru poate duce la încetinirea sau blocarea serviciilor web.

O altă tehnică periculoasă este atacul Slowloris, care implică deschiderea unui număr mare de conexiuni către un server, fără a le finaliza. Acest lucru împiedică serverul să accepte noi conexiuni legitime, blocând astfel accesul utilizatorilor reali.

Capitolul 3
Realizarea sistemului de detectie

In realizarea sistemului de detectie a unui atac de tip DDoS prin metode automatizate care sa aiba la baza inteligenta artificiala a fost necesar utilizarea sau crearea unei baze de date relevante in continut si potrivita in dimensiune. Din punctul de vedere a domeniului Machine Learning aceast proiect este unul de de clasificare a unei informatii in N-clase de atacuri asupra retelei noastre. Pentru a privi cu aplicabilitate aceasta abordare trebuie luat in considere ca metodele de protectie trebuie sa analizeze in timp real datele pe care acesta le primeste si sa semnaleze detectia unui atac. Astfel dezvoltarea trebuie sa aiba ca scop secundar dimensiunea scazuta a modelului si eficienta acestuia pentru a fi fezabil.

3.1 Baza de date

CICDDoS2019 dezvoltată de Canadian Institute for Cybersecurity (CIC) este o bază de date complexă, care include fluxuri de trafic de rețea atât benigne, cât și malicioase, acoperind o gamă variată de scenarii de atacuri DDoS. Aceasta conține următoarele caracteristici esențiale:

1. Diverse tipuri de atacuri DDoS – Baza de date include multiple variante de atacuri DDoS, clasificate în funcție de vectorul de atac, cum ar fi:

    - Atacuri bazate pe volum (Volume-Based): UDP Flood, ICMP Flood, DNS Amplification.
    - Atacuri bazate pe volum (Volume-Based): UDP Flood, ICMP Flood, DNS Amplification.
    - Atacuri la nivel de protocol (Protocol-Based): SYN Flood, ACK Flood.
    - Atacuri la nivel de aplicație (Application Layer-Based): HTTP Flood, Slowloris

2. Captură de trafic în timp real – Datele din CICDDoS2019 sunt colectate într-un mediu realistic de testare (testbed) folosind pcap (packet capture) și apoi extrase în formate NetFlow și CSV, permițând analiza aprofundată a pachetelor și fluxurilor de trafic.

3. Set extins de caracteristici (Features) – Baza de date conține peste 80 de caracteristici extrase din traficul de rețea, incluzând:

    - Caracteristici legate de fluxul de date (durata sesiunii, numărul de pachete, rata de transmitere a pachetelor).

- Caracteristici legate de transport (protocole TCP/UDP, numărul de SYN/ACK).
- Caracteristici temporale (timpul mediu dintre pachete, latența conexiunii).

4. Date etichetate pentru învățare supravegheată – Setul de date conține etichete pentru fiecare pachet și sesiune de trafic, ceea ce permite antrenarea algoritmilor de învățare supravegheată pentru clasificarea atacurilor DDoS versus trafic benign.

Deși CICDDoS2019 este o resursă valoroasă, există și câteva limitări:

1. Lipsa scenariilor distribuite din rețele reale – Baza de date este generată într-un mediu controlat și poate să nu reflecte toate complexitățile atacurilor DDoS din rețele reale.

2. Necesitatea preprocesării datelor – Setul de date conține valori lipsă și redundante, ceea ce necesită o etapă semnificativă de curățare a datelor pentru a îmbunătăți performanța modelelor ML.

3. Posibila dependență de caracteristicile dataset-ului – Modelele ML antrenate exclusiv pe CICDDoS2019 pot suferi de overfitting, necesitând testare pe alte dataset-uri pentru validare (ex. UNSW-NB15, CICIDS2017).

3.2 Impartirea si preprocesarea datelor

In cadrul fisierului dataset.py am descarcat baza de date disponibila pe kaggle.com si am m-am folosit de pachetele .parquet deja etichetate pentru antrenare si testare.

```
Train data size: (120065, 78)
Test data size: (38973, 78)
Train labels distribution: Label
Syn        48840
Benign     42007
UDP        18090
MSSQL       8523
LDAP        1906
NetBIOS      644
UDPLag        55
Name: count, dtype: int64
Test labels distribution: Label
Benign           10847
DrDoS_UDP        10420
UDP-lag           8872
DrDoS_MSSQL       6212
DrDoS_LDAP        1440
DrDoS_NetBIOS      598
Syn                533
WebDDoS             51
Name: count, dtype: int64
```

Figura 3.1: Dimensiunea si continutul bazei de date CICDDoS2019

In urma evaluarii am observat un dezechilibru in clasele din grupul de testare.

Clasa "WebDDos" nu este prezenta in antrenare. Astfel in fisierul preprocess.py am sters din testare aceasta clasa. Pe langa aceasta am mai implementat preprocesari la nivel structural al datasetului, pentru a reduce informatia irelevanta pentru model.

In continuare in fisierul preprocess.py am realizat:

- Maparea coloanelor: Se realizează maparea numelor coloanelor între seturile de date de antrenare și testare, asigurându-se că acestea sunt consecvente. Numele coloanelor din setul de testare sunt redenumite pentru a se potrivi cu cele din setul de antrenare.

- Gestionarea valorilor nule și a duplicatelor: Se verifică existența valorilor nule sau a valorilor duplicate. Nu au fost găsite valori nule, iar orice duplicate din setul de date sunt eliminate.

- Eliminarea coloanelor cu o singură valoare unică: Coloanele care conțin o singură valoare unică sunt eliminate, deoarece nu oferă informații relevante pentru clasificare.

- Eliminarea coloanelor puternic corelate: Coloanele care au un coeficient de corelație de 0.8 sau mai mare sunt eliminate pentru a reduce multicolinearitatea și a îmbunătăți performanța modelului.

3.3 Selecția modelului

În clasificarea multi-clasă, diverse algoritmi de învățare automată sunt folosiți pentru a antrena și evalua performanța modelului. Fiecare model are propriile avantaje și dezavantaje, în funcție de complexitatea setului de date și de tipul caracteristicilor utilizate. Am ales utilizarea modelelor disponibile din libraria sklearn si sunt implementate in fisierul model.py

- Random Forest

  Un algoritm de învățare supravegheată bazat pe arborele de decizie. Acesta construiește mai mulți arbori de decizie (de unde și numele de "pădure") și combină predicțiile lor pentru a obține un rezultat final mai stabil și mai precis.

- K-Nearest Neighbors

  Algoritm de clasificare bazat pe distanță, care atribuie o etichetă unui punct nou pe baza celor mai apropiați K vecini din setul de antrenament.

- Extra Trees Classifier

  O variantă a Random Forest, dar cu mai multă randomizare în alegerea punctelor de divizare în arbori de decizie.

- Multi-Layer Perceptron (MLP) Classifier

  MLP este un tip de rețea neuronală artificială, care conține un strat de input, unul sau mai multe straturi ascunse și un strat de output. Este un model puternic pentru clasificare non-liniară.

- XGBoost

  XGBoost este un algoritm bazat pe boosting, care construiește mai mulți arbori de decizie secvențial, fiecare corectând greșelile arborilor anteriori.

Fiecare model are avantaje și dezavantaje și este ales în funcție de natura setului de date și a problemei. Dar decizia finala se va face pe baza scorurilor generale pentru fiecare metrica aleasa.

3.4 Evaluarea Modelului

Evaluăm performanța modelelor utilizând diverse metrice de performanță, cum ar fi acuratețea, precizia, recall-ul, F1-score și ROC AUC. Curbele ROC sunt trasate pentru a compara performanța fiecărui model pe diferite clase. Se creează un tabel cu scorurile fiecărui model pentru o comparație ușoară si coerenta.

Acuratețea (Accuracy)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Unde:

- TP (True Positives) – Predicții corecte pentru clasa pozitivă.
- TN (True Negatives) – Predicții corecte pentru clasa negativă.
- FP (False Positives) – Predicții incorecte pentru clasa pozitivă.
- FN (False Negatives) – Predicții incorecte pentru clasa negativă.

Precizia (Precision)

Măsoară proporția instanțelor corect prezise ca fiind pozitive din toate instanțele prezise ca pozitive.

Performanța modelelor este exaluata utilizând diverse metrice de performanță, cum ar fi acuratețea, precizia, recall-ul, F1-score și ROC AUC.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Indică cât de multe dintre predicțiile pozitive sunt cu adevărat corecte.

Recall (Sensibilitatea sau Rata de Detectare a Pozitivelor)

Măsoară proporția instanțelor pozitive detectate corect.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

Indică cât de bine modelul identifică clasa pozitivă.

F1-Score

Este media armonică dintre precizie și recall, oferind un echilibru între cele două.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

Este utilă când există un dezechilibru între clase, penalizând modele care favorizează o clasă în detrimentul alteia.

ROC AUC (Receiver Operating Characteristic - Area Under Curve)

ROC (Receiver Operating Characteristic) este o curbă care arată raportul dintre Rata Fals Pozitivă (FPR) și Rata Adevărat Pozitivă (TPR) la diferite praguri de decizie. AUC (Area Under Curve) măsoară aria de sub curba ROC, oferind o măsură globală a performanței modelului.

$$TPR = \frac{TP}{TP + FN} \tag{3.5}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.6}$$

Capitolul 4
Rezultate

Prin antrenarea secvențiala și testarea modelului, cod prezent în fișierul model.py am putut extrage rezultatele pe baza cărora sa alegem un model. Fiecare model fiind salvat sub forma unor checkpoint-uri pentru a fi utilizat ulterior. In momentul în care fișierul model.py este rulat se poate vedea progresul de antrenare.

Figura 4.1: Rularea antrenare

Când antrenarea este gata, testarea are loc împreuna cu salvarea rezultatelor în funcție de metricile alese.

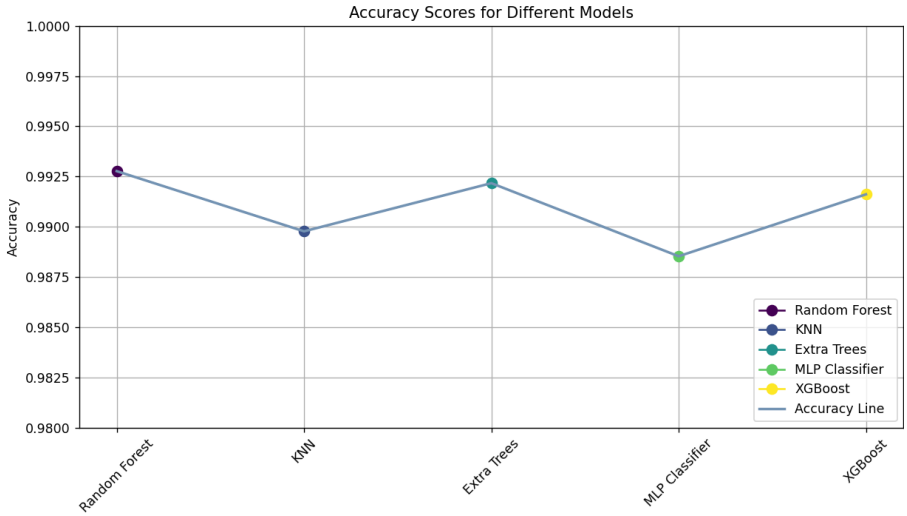

Figura 4.2: Reprezentare acurateți pentru model

In urma afișării acurateții se poate observa performantele crescute ale modelelor RF și Extra

Trees. Totuși daca analizam curba ROC se poate observa cum clasificarea cu MLP si XGboost sunt cele mai performante. Din aceasta cauza decizia finala a celui mai bun model se va lua în urma tabelului cu toate rezultatele indiciilor de performanta.

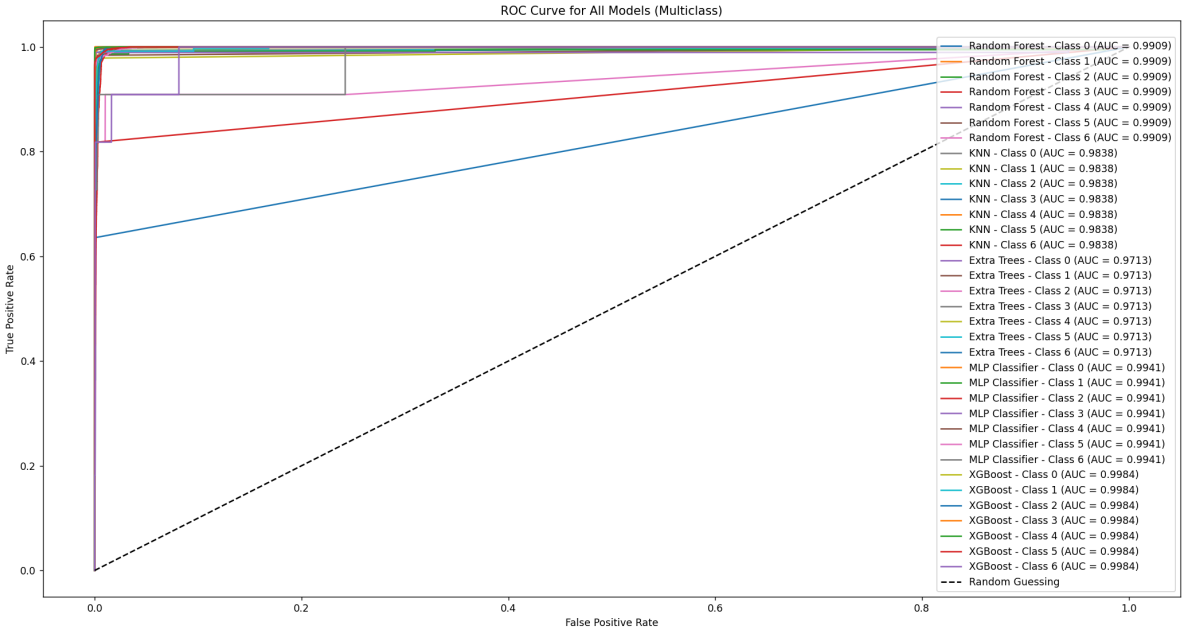

Figura 4.3: Reprezentare a rezultatelor pentru fiecare clasa

Din acest tabel se extrag performantele corespunzătoare RF care arata cum acesta este cel mai stabil pentru toți indicii și care are cele mari șanse sa întoarcă rezultate corecte în proporția cea mai mare.

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	CV Score
0	Random Forest	0.992770	0.992864	0.992770	0.992700	0.990892	0.993636
1	KNN	0.989775	0.989959	0.989775	0.989797	0.983762	0.990417
2	Extra Trees	0.992171	0.992197	0.992171	0.992158	0.971325	0.993048
3	MLP Classifier	0.988534	0.988400	0.988534	0.988418	0.994088	0.989037
4	XGBoost	0.991615	0.991619	0.991615	0.991595	0.998359	0.992021

Figura 4.4: Lista metrici pentru fiecare model

4.1 Concluzii

Prin acest proiect, am analizat mai multe modele de învățare automată pentru detectarea atacurilor DDoS, având ca obiectiv identificarea celui mai eficient model în clasificarea diferitelor tipuri de atacuri. Rezultatele obținute oferă o perspectivă asupra performanței fiecărui model și pot contribui la dezvoltarea unor sisteme de securitate cibernetică mai robuste, capabile să detecteze și să prevină atacurile DDoS.

Bibliografie

1. Dhoogla, *CICDDoS2019 Dataset*, 2019. Available at: https://www.kaggle.com/datasets/dhoogla/cicddos2019/code.

2. Canadian Institute for Cybersecurity (CIC), *CIC DDoS 2019 Dataset*, 2019. Available at: https://www.unb.ca/cic/datasets/ddos-2019.html.

3. Labellerr, *DDoS Attack Detection: A Guide to Machine Learning Techniques*, 2023. Available at: https://www.labellerr.com/blog/ddos-attack-detection/.

4. ScienceDirect, *DDoS Attack Detection Using AI: A Comprehensive Study*, *Journal of Cybersecurity Advances*, 2024. Available at: https://www.sciencedirect.com/science/article/pii/S2665917424000138?ref=pdf_download&fr=RR-2&rr=909461e79904e437.

5. Kaggle User, *Starter: DDoS Botnet Attack on IoT*, 2024. Available at: https://www.kaggle.com/code/kerneler/starter-ddos-botnet-attack-on-iot-a71a0b42-4.

6. Kaggle, *KaggleHub: Download Datasets Easily*, 2024. Available at: https://github.com/Kaggle/kagglehub/blob/main/README.md#download-dataset.

7. Sina Ahmadi, *AI in the Detection and Prevention of Distributed Denial of Service (DDoS) Attacks*, *National Coalition of Independent Scholars (NCIS)*, 2024, Seattle, USA.