A Comprehensive Analysis of Entropy-based DDoS Detection and Alternative Methods

***Abstract*—Distributed Denial of Service (DDoS) attacks pose a major threat to network security by overwhelming systems with excessive traffic, causing severe disruptions. As attacks increase in scale and complexity, traditional detection methods struggle to maintain effectiveness. This paper explores statistical techniques for DDoS detection, focusing on the Gini Index and comparing it with the Entropy-based approach. The proposed framework leverages the Gini Index for real-time anomaly detection, ensuring high precision and low latency. A comparative analysis with the Entropy-based approach evaluates the detection accuracy, response time, and computational efficiency. Enhancements such as adaptive thresholds and flow aggregation improve scalability in high-speed networks. By integrating insights from Gini Index and Entropy-based detection, this paper presents an efficient and adaptable solution for modern network security, emphasizing detection speed and computational efficiency. The findings serve as a foundation for future advancements in DDoS mitigation.**

***Index Terms*—DDoS Detection, Entropy-based approach, Gini Index, Network Security, Cybersecurity**

I. Introduction

As digital infrastructure continues to grow rapidly, ensuring strong and scalable network security is more important than ever. Among various cyber threats [1], Distributed Denial of Service (DDoS) [2] attacks remain among the most disruptive. These attacks flood servers or networks with excessive, malicious traffic, causing service outages, financial losses, and reputational harm. To ensure consistent service availability, deploying reliable and real-time DDoS detection mechanisms is essential.

Over time, several detection techniques have been proposed, each with its advantages and limitations. Signature-Based Approaches (SBA) [3] detect known attack patterns with high accuracy but fail to recognize zero-day or previously unseen threats [4]. Anomaly-Based Approaches (ABA) [5] detect unusual traffic behavior but often suffer from high false positives [6], especially in dynamic conditions.

Entropy-Based Approaches (EBA) are effective for identifying volumetric DDoS attacks by analyzing entropy fluctuations in traffic. However, they face scalability issues in high-speed networks. Machine Learning-Based Approaches (MLBA) [7] offer adaptability through learning from traffic behavior, but they require large training datasets and significant computational power. Other methods, such as Gini Index-Based Approaches (GIBA) [8], CAPTCHA-Based techniques [9], and Challenge-Response mechanisms, also offer alternatives, though GIBA remains relatively underexplored.

Despite these advancements, few studies have compared these techniques across critical performance metrics like computation speed, detection accuracy, scalability, resource efficiency, and false positives. While EBA has shown effectiveness against volumetric attacks [10], more comparative analysis with GIBA, MLBA, and newer hybrid approaches is needed.

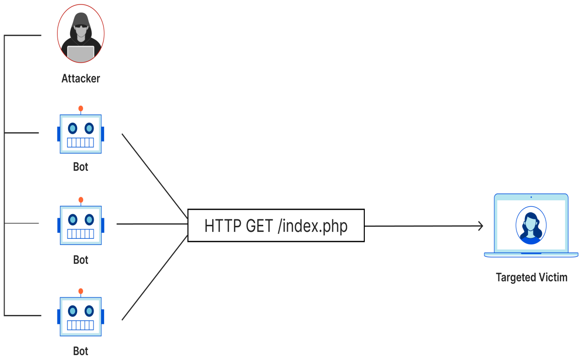


Fig. 1. How DDoS Attack Works

This study addresses these gaps by comparing EBA and GIBA using Mininet-based simulations. It evaluates both techniques across essential metrics, including detection accuracy, computational speed, resource usage, and real-time monitoring. By identifying their strengths and limitations, the study supports the development of scalable and efficient DDoS detection systems [11].

As digital networks become integral to modern infrastructure, the risk posed by advanced DDoS attacks increases [12]. Effective detection is crucial for preserving service availability in high-speed environments. EBA is responsive but challenged by computational overhead; MLBA is flexible but resource-intensive; and GIBA shows lightweight potential, though further evaluation is needed.

II. Literature Review

Numerous DDoS detection techniques have been developed, each addressing specific challenges and offering distinct strengths and limitations.

Signature-based Approaches (SBA): SBA matches traffic patterns to known signatures, proving effective against familiar threats but ineffective for zero-day attacks [13]. Anomaly-based Approaches (ABA): ABA detects deviations from normal traffic behavior, identifying new threats but often producing false positives and requiring high computation. Entropy-based Approaches (EBA): EBA analyzes randomness in traffic for real-time detection [14], but suffers from scalability and

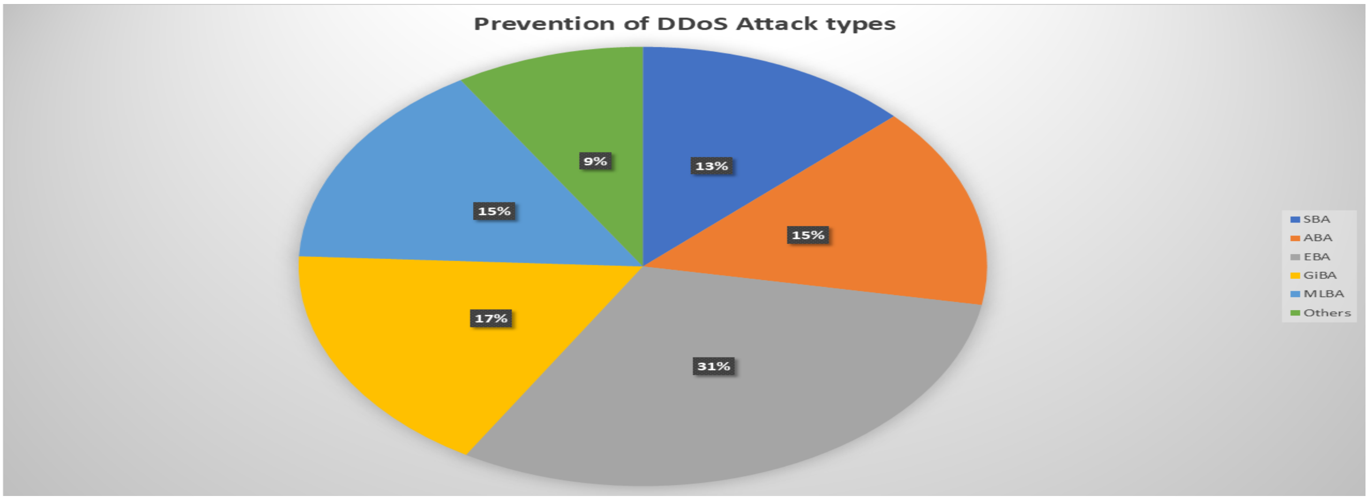


Fig. 2. Prevention of DDoS attack types

computational overhead [15]. Gini-Index-based Approaches (GIBA): GIBA detects anomalies using traffic inequality [16], but its comparative performance is not well-studied. Machine Learning-based Approaches (MLBA): MLBA classifies traffic using trained models, offering adaptability to evolving threats but requiring large datasets and significant resources. Other Techniques: Hybrid statistical methods [17], CAPTCHA-based defenses, and Challenge-Response tests can enhance accuracy but may increase latency and affect usability.

Most studies focus on individual techniques, with few comprehensive evaluations across performance metrics like detection accuracy, false positives, scalability, and resource efficiency. SBA struggles with zero-day threats, ABA with false alarms, EBA with high-speed environments, and GIBA lacks comparative validation.

Research into adaptive thresholding [18], fast Entropy calculations [19], and hybrid Machine Learning solutions is ongoing. This study contributes by systematically comparing EBA and GIBA under dynamic traffic conditions to assess their strengths and limitations.

A. Gaps and Limitations in Existing Research

- Limited Comparative Studies: Few studies have compared Entropy-based Approaches (EBA) and Gini-Index-based Approaches (GIBA) under dynamic network conditions, leading to limited insight into their real-world effectiveness [20].
- Underexplored GIBA: GIBA's potential for detecting subtle anomalies, scaling in high-speed networks, and adapting to varied environments remains underexplored.
- Real-Time Detection Challenges: EBA faces computational constraints that hinder real-time deployment. The absence of adaptive thresholding reduces its responsiveness to traffic changes.
- Lack of Optimization: Optimization techniques for improving detection speed and accuracy are insufficiently studied, limiting scalability.
- Insufficient Practical Validation: Much of the existing research is theoretical. More empirical studies are needed to validate these methods in real network settings.

Addressing these limitations is vital for advancing DDoS detection. This research aims to bridge these gaps by comparing EBA and GIBA, introducing optimization strategies, and validating results through simulation.

III. Proposed Work

This research focuses on the comparative analysis and implementation of Distributed Denial of Service (DDoS) detection techniques, with a particular emphasis on the Gini Index-based method alongside Entropy-Based Approaches (EBA). The proposed work involves building a real-time DDoS detection system using the Gini Index, tested through traffic simulation in Mininet. The goal is to evaluate detection performance, scalability, and adaptability under various network conditions. By comparing Gini and Entropy-based methods, this study highlights their individual strengths, limitations, and suitability for high-speed, large-scale networks.

The Gini Index-based method is assessed against EBA across several key aspects. Detection accuracy is compared by analyzing true and false positive rates. Computation speed is evaluated through latency and responsiveness. Adaptability considers how each method handles evolving attack patterns. Ease of implementation reflects deployment complexity and practicality. Real-time monitoring examines continuous traffic analysis efficiency. Resource use involves memory and processing overhead. Scalability is measured under different traffic loads. Lastly, robustness against false positives is judged by how accurately each method identifies real attacks while reducing false alerts.

By analyzing, optimizing, and validating these detection mechanisms, this research aims to improve DDoS defense strategies and support the development of efficient, real-time protection systems.

IV. Implementations

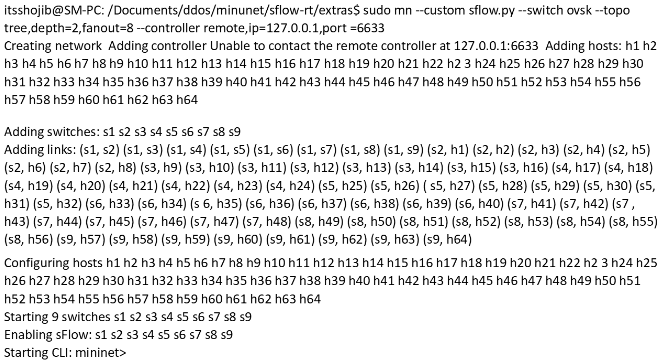
The DDoS detection system integrates Entropy-based and Gini index-based techniques within a Linux environment using a Software-Defined Networking (SDN) architecture. This setup allows for flexible traffic control, real-time monitoring, and adaptive mitigation. The main objective is to evaluate and compare the computational efficiency, detection accuracy, responsiveness, and adaptability of both techniques under different traffic conditions in a controlled and reproducible testbed.

A. Tools and Technologies

Several tools and technologies were used to build and test the system. Mininet was employed to emulate a virtual SDN network with configurable hosts, switches, and links. sFlow-RT provided real-time flow analytics and computed both Entropy and Gini Index values for traffic monitoring. Python was used for scripting traffic behavior, processing collected data, and visualizing trends. The POX Controller managed flow entries, handled detection logic, and enforced mitigation rules. `iperf` generated both normal and attack traffic streams to simulate realistic scenarios, while Wireshark was used for packet-level inspection and to verify the correctness of the traffic flows and detection results.

B. Network Setup

The virtual network topology created by Mininet consisted of multiple hosts connected to a single SDN-enabled switch, which was controlled by the POX controller. This architecture enabled precise control over the flow of traffic and supported the simulation of diverse attack scenarios. Normal and malicious hosts were configured to test detection sensitivity and reaction accuracy (see Figure 3).



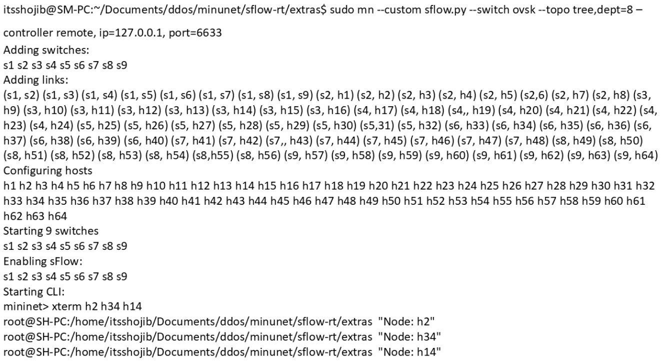
```
itsshojib@SM-PC: /Documents/ddos/minunet/sflow-rt/extras$ sudo mn --custom sflow.py --switch ovsk --topo
tree,depth=2,fanout=8 --controller remote,ip=127.0.0.1,port =6633
Creating network  Adding controller Unable to contact the remote controller at 127.0.0.1:6633  Adding hosts: h1 h2
h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h2 3 h24 h25 h26 h27 h28 h29 h30
h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56
h57 h58 h59 h60 h61 h62 h63 h64

Adding switches: s1 s2 s3 s4 s5 s6 s7 s8 s9
Adding links: (s1, s2) (s1, s3) (s1, s4) (s1, s5) (s1, s6) (s1, s7) (s1, s8) (s1, s9) (s2, h1) (s2, h2) (s2, h3) (s2, h4) (s2, h5)
(s2, h6) (s2, h7) (s2, h8) (s3, h9) (s3, h10) (s3, h11) (s3, h12) (s3, h13) (s3, h14) (s3, h15) (s3, h16) (s4, h17) (s4, h18)
(s4, h19) (s4, h20) (s4, h21) (s4, h22) (s4, h23) (s4, h24) (s5, h25) (s5, h26) ( s5, h27) (s5, h28) (s5, h29) (s5, h30) (s5,
h31) (s5, h32) (s6, h33) (s6, h34) (s 6, h35) (s6, h36) (s6, h37) (s6, h38) (s6, h39) (s6, h40) (s7, h41) (s7, h42) (s7 ,
h43) (s7, h44) (s7, h45) (s7, h46) (s7, h47) (s7, h48) (s8, h49) (s8, h50) (s8, h51) (s8, h52) (s8, h53) (s8, h54) (s8, h55)
(s8, h56) (s9, h57) (s9, h58) (s9, h59) (s9, h60) (s9, h61) (s9, h62) (s9, h63) (s9, h64)
Configuring hosts h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h2 3 h24 h25
h26 h27 h28 h29 h30 h31 h32 h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51
h52 h53 h54 h55 h56 h57 h58 h59 h60 h61 h62 h63 h64
Starting 9 switches s1 s2 s3 s4 s5 s6 s7 s8 s9
Enabling sFlow: s1 s2 s3 s4 s5 s6 s7 s8 s9
Starting CLI: mininet>
```

Fig. 3. Host Setup for DDoS Detection in an SDN Environment

C. Traffic Simulation and Baseline Metrics

To establish baseline behavior, UDP traffic was generated using `iperf`. During normal traffic conditions, the entropy values remained around ± 1.5, while the Gini Index stayed close to 0.5, indicating well-distributed and stable traffic flows. These metrics served as reference thresholds for detecting anomalies (Figure 4).



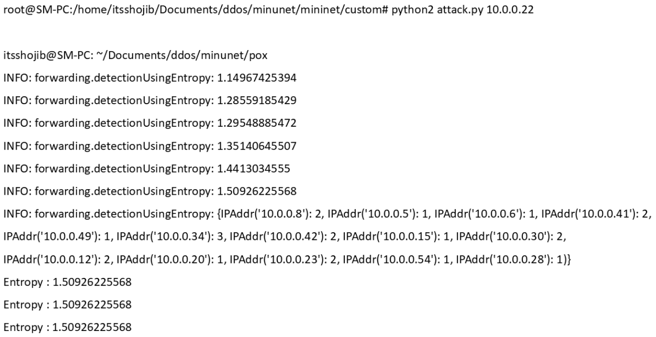
```
itsshojib@SM-PC:~/Documents/ddos/minunet/sflow-rt/extras$ sudo mn --custom sflow.py --switch ovsk --topo tree,dept=8 –

controller remote, ip=127.0.0.1, port=6633
Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9
Adding links:
(s1, s2) (s1, s3) (s1, s4) (s1, s5) (s1, s6) (s1, s7) (s1, s8) (s1, s9) (s2, h1) (s2, h2) (s2, h3) (s2, h4) (s2, h5) (s2,6) (s2, h7) (s2, h8) (s3,
h9) (s3, h10) (s3, h11) (s3, h12) (s3, h13) (s3, h14) (s3, h15) (s3, h16) (s4, h17) (s4, h18) (s4,, h19) (s4, h20) (s4, h21) (s4, h22) (s4,
h23) (s4, h24) (s5, h25) (s5, h26) (s5, h27) (s5, h28) (s5, h29) (s5, h30) (s5,31) (s5, h32) (s6, h33) (s6, h34) (s6, h35) (s6, h36) (s6,
h37) (s6, h38) (s6, h39) (s6, h40) (s7, h41) (s7, h42) (s7,, h43) (s7, h44) (s7, h45) (s7, h46) (s7, h47) (s7, h48) (s8, h49) (s8, h50)
(s8, h51) (s8, h52) (s8, h53) (s8, h54) (s8,h55) (s8, h56) (s9, h57) (s9, h58) (s9, h59) (s9, h60) (s9, h61) (s9, h62) (s9, h63) (s9, h64)
Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h23 h24 h25 h26 h27 h28 h29 h30 h31 h32
h33 h34 h35 h36 h37 h38 h39 h40 h41 h42 h43 h44 h45 h46 h47 h48 h49 h50 h51 h52 h53 h54 h55 h56 h57 h58 h59 h60 h61
h62 h63 h64
Starting 9 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9
Enabling sFlow:
s1 s2 s3 s4 s5 s6 s7 s8 s9
Starting CLI:
mininet> xterm h2 h34 h14
root@SH-PC:/home/itsshojib/Documents/ddos/minunet/sflow-rt/extras  "Node: h2"
root@SH-PC:/home/itsshojib/Documents/ddos/minunet/sflow-rt/extras  "Node: h34"
root@SH-PC:/home/itsshojib/Documents/ddos/minunet/sflow-rt/extras  "Node: h14"
```

Fig. 4. Connecting Hosts for Traffic Simulation

D. DDoS Attack Simulation and Detection

In the attack scenario, additional hosts generated high-volume UDP traffic to simulate a DDoS event. This led to a noticeable drop in Entropy (below 0.5), reflecting a concentration of traffic from a few sources. Simultaneously, the Gini Index rose above 0.9, highlighting the unequal distribution of traffic. These shifts indicated a clear deviation from the baseline and successfully triggered the detection mechanism (Figure 5).



```
root@SM-PC:/home/itsshojib/Documents/ddos/minunet/mininet/custom# python2 attack.py 10.0.0.22


itsshojib@SM-PC: ~/Documents/ddos/minunet/pox
INFO: forwarding.detectionUsingEntropy: 1.14967425394
INFO: forwarding.detectionUsingEntropy: 1.28559185429
INFO: forwarding.detectionUsingEntropy: 1.29548885472
INFO: forwarding.detectionUsingEntropy: 1.35140645507
INFO: forwarding.detectionUsingEntropy: 1.4413034555
INFO: forwarding.detectionUsingEntropy: 1.50926225568
INFO: forwarding.detectionUsingEntropy: {IPAddr('10.0.0.8'): 2, IPAddr('10.0.0.5'): 1, IPAddr('10.0.0.6'): 1, IPAddr('10.0.0.41'): 2,
IPAddr('10.0.0.49'): 1, IPAddr('10.0.0.34'): 3, IPAddr('10.0.0.42'): 2, IPAddr('10.0.0.15'): 1, IPAddr('10.0.0.30'): 2,
IPAddr('10.0.0.12'): 2, IPAddr('10.0.0.20'): 1, IPAddr('10.0.0.23'): 2, IPAddr('10.0.0.54'): 1, IPAddr('10.0.0.28'): 1)}
Entropy : 1.50926225568
Entropy : 1.50926225568
Entropy : 1.50926225568
```

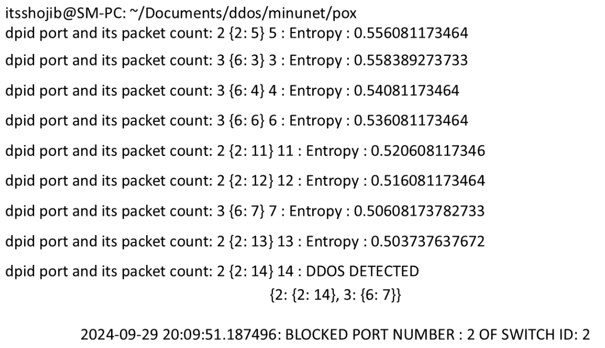
Fig. 5. Impact of DDoS Attack on Entropy and Gini Index

E. DDoS Mitigation and Response

Upon detecting the attack, the POX controller dynamically applied mitigation policies to block malicious sources by modifying the flow rules on the switch. This action restored the network to stable operation and ensured uninterrupted service for legitimate users. The effectiveness of this response was validated through real-time monitoring and packet analysis (Figure 6).

V. Results and Discussions

Efficient computational performance is essential for real-time DDoS detection. Figure 7 shows that the Entropy-based method consistently detects DDoS attacks faster than the Gini Index, with lower computational overhead. Entropy's



```
itsshojib@SM-PC: ~/Documents/ddos/minunet/pox
dpid port and its packet count: 2 {2: 5} 5 : Entropy : 0.556081173464
dpid port and its packet count: 3 {6: 3} 3 : Entropy : 0.558389273733
dpid port and its packet count: 3 {6: 4} 4 : Entropy : 0.54081173464
dpid port and its packet count: 3 {6: 6} 6 : Entropy : 0.536081173464
dpid port and its packet count: 2 {2: 11} 11 : Entropy : 0.520608117346
dpid port and its packet count: 2 {2: 12} 12 : Entropy : 0.516081173464
dpid port and its packet count: 3 {6: 7} 7 : Entropy : 0.50608173782733
dpid port and its packet count: 2 {2: 13} 13 : Entropy : 0.503737637672
dpid port and its packet count: 2 {2: 14} 14 : DDOS DETECTED
                                              {2: {2: 14}, 3: {6: 7}}

          2024-09-29 20:09:51.187496: BLOCKED PORT NUMBER : 2 OF SWITCH ID: 2
```

Fig. 6. DDoS Detection and Automated Mitigation in SDN

lightweight calculations make it well-suited for real-time applications, while the Gini Index introduces higher latency.

- Entropy-based detection enables faster processing, reducing latency and improving mitigation speed.
- The Gini Index offers greater granularity in specific attack patterns, especially in precise traffic distribution shifts.
- Entropy adapts more effectively to dynamic traffic, whereas the Gini Index struggles with frequent fluctuations.
- Entropy yields lower false positive rates, distinguishing traffic surges from attacks better than the Gini Index, which shows a slightly higher rate under congestion.

These findings highlight Entropy-based detection's strengths in minimizing latency and computational overhead, making it ideal for high-speed networks where timely response is crucial.

A. Comparison of Detection Techniques

The following summary compares both detection methods across key performance metrics:

- Computation Speed: Entropy offers faster response due to simpler calculations, while the Gini Index introduces more latency, reducing its real-time efficiency.
- Detection Accuracy: Both methods achieve high accuracy, but Entropy performs better in detecting volumetric attacks due to its sensitivity to traffic variation.
- Adaptability: Entropy is more adaptive to changing traffic conditions, while the Gini Index is less effective under variable patterns.
- Ease of Implementation: Entropy is easier to integrate and requires fewer resources, whereas the Gini Index is more complex and computationally heavier.
- Real-Time Monitoring: With lower overhead, Entropy supports continuous monitoring, while the Gini Index may introduce delays due to processing demands.
- False Positives: Entropy better distinguishes between normal traffic changes and attacks, reducing false alerts more effectively than the Gini index.

Table I summarizes the key differences between the Entropy-based and Gini Index-based detection techniques, reinforcing the advantages and limitations of each approach.

TABLE I
COMPARISON OF ENTROPY-BASED AND GINI INDEX-BASED DETECTION

Aspect	**Detection Method**	
	Entropy-based	*Gini Index-based*
Computation Speed	Fast, real-time detection.	Slower, higher latency.
Detection Accuracy	High, effective for volumetric attacks.	High, but less effective for extreme cases.
Adaptability	Adapts well to dynamic traffic.	Struggles with traffic variations.
Ease of Implementation	Simple, low resources.	Complex, needs more resources.
Real-Time Monitoring Capability	Excellent real-time detection.	Moderate, with delays.
Scalability	Efficient in large networks.	Limited by computation load.
Robustness Against False Positives	Very low.	Moderate.

VI. CONCLUSION AND FUTURE SCOPE

This study compared Entropy-based and Gini Index-based methods for DDoS detection, with Entropy-based detection showing better results in speed, adaptability, and ease of implementation. It identified traffic anomalies quickly and was easier to integrate into real-time systems. While both methods showed high accuracy, Entropy-based detection proved more efficient for high-speed networks.

These findings highlight the need to choose detection methods based on accuracy, resource usage, and response time. Understanding these trade-offs is key to building scalable and reliable DDoS detection systems.

Future research can explore advanced Entropy techniques to make detection faster and more accurate. Developing hybrid systems that combine Entropy methods with machine learning can help adapt to new and complex attack patterns. Using dynamic thresholding may improve detection by adjusting to different types of network traffic. Testing these methods in real-world environments will also help improve their reliability. Enhancing resilience against advanced threats will make networks stronger and more secure.

REFERENCES

[1] K. S. Wilson and M. A. Kiy, "Some fundamental cybersecurity concepts," *IEEE access*, vol. 2, pp. 116–124, 2014.

[2] N. Z. Bawany, J. A. Shamsi, and K. Salah, "Ddos attack detection and mitigation using sdn: methods, practices, and solutions," *Arabian Journal for Science and Engineering*, vol. 42, pp. 425–441, 2017.

[3] P. Szynkiewicz, "Signature-based detection of botnet ddos attacks," in *Cybersecurity of Digital Service Chains: Challenges, Methodologies, and Tools*. Springer, 2022, pp. 120–135.

[4] H. Al-Rushdan, M. Shurman, S. H. Alnabelsi, and Q. Althebyan, "Zero-day attack detection and prevention in software-defined networks," in *2019 international arab conference on information technology (acit)*. IEEE, 2019, pp. 278–282.

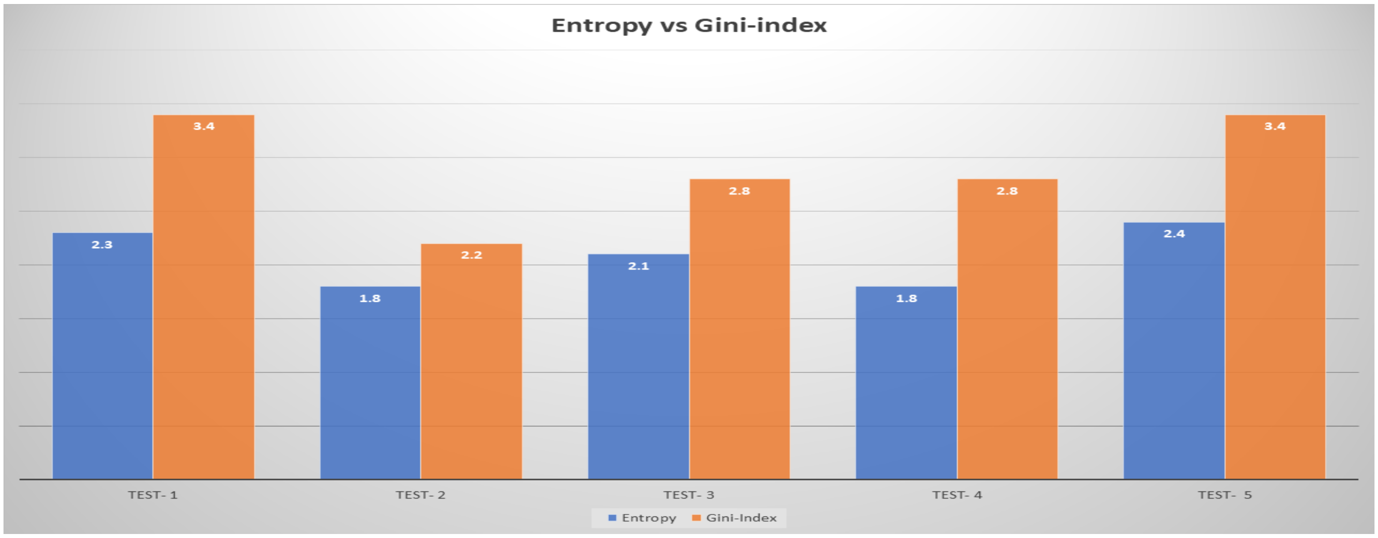


Fig. 7. Computational Time Comparison for DDoS Detection: Entropy vs. Gini Index

[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[6] O. Abouabdalla, H. El-Taj, A. Manasrah, and S. Ramadass, "False positive reduction in intrusion detection system: A survey," in *2009 2nd IEEE International Conference on Broadband Network & Multimedia Technology*. IEEE, 2009, pp. 463–466.

[7] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[8] M. Dilshad, M. H. Syed, and S. Rehman, "Efficient distributed denial of service attack detection in internet of vehicles using gini index feature selection and federated learning," *Future Internet*, vol. 17, no. 1, p. 9, 2025.

[9] M. Mehra, M. Agarwal, R. Pawar, and D. Shah, "Mitigating denial of service attack using captcha mechanism," in *Proceedings of the international conference & workshop on emerging trends in technology*, 2011, pp. 284–287.

[10] M. Goldstein, "Some thermodynamic aspects of the glass transition: Free volume, entropy, and enthalpy theories," *The Journal of Chemical Physics*, vol. 39, no. 12, pp. 3369–3374, 1963.

[11] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Computer networks*, vol. 62, pp. 122–136, 2014.

[12] F. Yan, Y. Jian-Wen, and C. Lin, "Computer network security and technology research," in *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*. IEEE, 2015, pp. 293–296.

[13] A. Singh and B. B. Gupta, "Distributed denial-of-service (ddos) attacks and defense mechanisms in various web-enabled computing platforms: issues, challenges, and future research directions," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 18, no. 1, pp. 1–43, 2022.

[14] J. Jeong, S. Park, J. Lim, J. Kang, D. Shin, and D. Shin, "A study on network anomaly detection using fast persistent contrastive divergence," *Symmetry*, vol. 16, no. 9, p. 1220, 2024.

[15] M. E. Zainodin, Z. Zakaria, R. Hassan, and Z. Abdullah, "Entropy based method for malicious file detection," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 4, pp. 856–861, 2022.

[16] M. A. Bouke, A. Abdullah, S. H. ALshatebi, M. T. Abdullah, and H. El Atigh, "An intelligent ddos attack detection tree-based model using gini index feature selection method," *Microprocessors and Microsystems*, vol. 98, p. 104823, 2023.

[17] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to ddos attack detection and response," in *Proceedings DARPA information survivability conference and exposition*, vol. 1. IEEE, 2003, pp. 303–314.

[18] G. No and I. Ra, "Adaptive ddos detector design using fast entropy computation method," in *2011 fifth international conference on innovative mobile and internet services in ubiquitous computing*. IEEE, 2011, pp. 86–93.

[19] J. David and C. Thomas, "Ddos attack detection using fast entropy approach on flow-based network traffic," *Procedia Computer Science*, vol. 50, pp. 30–36, 2015.

[20] A. S. Saud, S. Shakya, and B. Neupane, "Analysis of depth of entropy and gini index based decision trees for predicting diabetes," *Indian Journal of Computer Science*, vol. 6, no. 6, pp. 19–28, 2021.