Distributed Intrusion Detection Systems: Architecture, Challenges, And Performance Evaluation

Mr. Piyush Mahesh Tiwatne[1], Ms. Sakshi Vishwas Gade[2], Ms. Pooja Govindram Parmar[3], Mr. Laxman Jalindar Gaikwad[4], Prof. Rajashri S. Khadake[5]

[1,2,3]*Dept. of DACS, MAEER's MIT Arts, Commerce &Science College Alandi(D)*

[4,5]*Dept. of Computer Application, MAEER's MIT Arts, Commerce &Science College Alandi(D)*

doi.org/10.64643/ijirt.208480-459

***Abstract*—The rapid growth of computer networks, cloud computing, and Internet of Things (IoT) devices has increased the number and complexity of cyberattacks. Traditional Intrusion Detection Systems (IDS), which depend mainly on a central system, can face difficulties when dealing with large amounts of network traffic and monitoring different parts of a network. Distributed Intrusion Detection Systems (DIDS) offer a solution by using multiple detection systems that work together to monitor network activities and identify possible attacks. This paper provides an overview of DIDS and discusses different architectures, including hierarchical, peer to-peer, multi-agent, and cloud- and edge-based approaches. It also explains the main functions of DIDS, such as collecting network data, analysing activities, sharing information between detection nodes, and responding to detected threats. The paper discusses the main benefits of DIDS, including better scalability, improved reliability, and the ability to monitor larger and distributed networks. At the same time, several challenges are considered, such as communication between detection nodes, maintaining consistent information, managing trust, handling high network traffic, and protecting the detection system from attacks. The performance of DIDS can be measured using factors such as detection accuracy, false positive rate, detection time, throughput, and resource usage. Common datasets, including NSLKDD, CICIDS2017, and UNSW-NB15, are used in research to evaluate intrusion detection methods. The paper also discusses the balance between good detection performance and the resources required by the system. Finally, future research areas such as privacy-preserving detection, load balancing, federated learning, blockchain based trust, and edge computing are discussed as possible ways to make Distributed Intrusion Detection Systems more secure, reliable, and efficient.**

***Index Terms*—Cybersecurity, Distributed Computing, Distributed Intrusion Detection Systems, Federated Learning, Internet of Things, Intrusion Detection, Network Security, Performance Evaluation.**

I. INTRODUCTION

Modern cyberattacks are growing in scale, speed and sophistication in parallel with the growth of distributed computing infrastructures such as cloud platforms, edge networks and the Internet of Things (IoT). The current state of organizations is such that they are spread across geographically distributed data centers, multiple cloud providers, and thousands of endpoint devices, and this results in an attack surface that cannot be effectively covered by traditional centralized security monitoring [1], [2]. Intrusion Detection Systems (IDS), originally formalized by Denning's seminal model of real-time intrusion detection [3], have traditionally relied on a single monitoring point to collect and analyze traffic or host-based logs. For small and medium sized networks, centralized IDS architectures are effective, but they have a single point of failure, pose a performance bottleneck under high traffic loads, and lack visibility to correlate attacks.

Distributed Intrusion Detection Systems (DIDS) are used to overcome these limitations, by distributing the detection process to multiple cooperating nodes, sensors or agents. Each node observes a local portion of the network or system, performs local analysis and exchanges information with peer nodes or with a coordinating layer to detect distributed and coordinated attacks that would otherwise be invisible to any single monitoring point [5], [6]. This paradigm shift is akin to other trends in distributed computing where decentralization increases fault tolerance, scalability, and responsiveness.

This paper is a comprehensive review of the state of art in Distributed Intrusion Detection Systems. Specifically, it: (i) surveys the major architectural paradigms used to build DIDS, (ii) identifies and

analyses the core technical and operational challenges associated with distributed detection, (iii) reviews performance evaluation methodologies and benchmark datasets used across the literature, and (iv) outlines open research directions, including the integration of federated learning, blockchain-based trust mechanisms, and edge computing. The rest of this paper is organized as follows: Section 2 reviews related work; Section 3 addresses DIDS architectures, Section 4 analyzes the main challenges, Section 5 presents performance evaluation criteria and comparative results, Section 6 discusses open issues and future directions, and Section 7 concludes the paper.

II. LITERATURE REVIEW

In this section, we review the evolution of the intrusion detection research for distributed computing environments in chronological order and organized into four thematic strands: (i) basic and signature/anomalybased detection, (ii) early distributed and multiagent architectures, (iii) machine learning and dataset-based detection, and (iv) recent privacy-preserving, blockchain-based, and edge-oriented approaches. Table I presents a synthesized comparison of representative studies.

2.1. Fundamentals Signature Based and Anomaly Based Detection

The first formal model for real-time intrusion detection was proposed by Denning [3] based on statistical profiling of subject behavior, thus laying the conceptual foundation for anomaly-based detection. Thus, snort [7] popularized light-weight rule-based signature detection for network traffic, providing high accuracy against known attacks, but no defence against novel attacks [8]. Later, GarciaTeodoro et al. [5] and Chandola et al. [9] surveyed anomaly-based techniques. They argue that such methods can in principle detect zero-day attacks; however, they also typically have higher false positive rates than signature-based methods a trade-off that remains a core concern in IDS design decisions today.

2.2. Early Distributed and Multi-Agent Architectures

As single-host and single-point monitoring proved insufficient for growing enterprise networks, Snapp et al. [14] proposed one of the earliest Distributed Intrusion Detection System (DIDS) prototypes, combining host- and network level monitors under a centralized coordinator to detect attacks spanning multiple hosts. Balasubramaniyan et al. [15] extended this concept with the Autonomous Agents for Intrusion Detection (AAFID) framework, in which independent software agents monitor local segments and report to transceivers and monitors in a hierarchical structure an architecture that directly informs the hierarchical model discussed in Section 3.1. Spafford and Zamboni [25] and Ning et al. [16] further examined abstraction-based correlation across distributed agents, addressing how alerts from heterogeneous sensors could be fused into coherent attack scenarios despite differing formats and semantics a challenge that persists in modern DIDS (see Section 4.2).

2.3. Machine Learning & Benchmark Data Sets

The 1998-1999 DARPA/KDD Cup evaluations produced the KDD Cup 99 dataset [10] which, despite redundancy and bias problems documented later on, became the de-facto benchmark for MLbased IDS research. To address these limitations, Tavallaee et al. [11] proposed NSL-KDD by removing duplicate records and rebalancing the difficulty levels. Lee and Stolfo [23] were among the first to consider intrusion detection not as a purely rule-based problem, but as a data mining and feature construction problem, a view that forms the basis of most modern ML-based IDS. More recent datasets, CICIDS2017 [12] and UNSW-NB15 [13], include modern attack categories (web attacks, infiltration, botnets) on realistic background traffic and are now standard benchmarks for evaluating both centralized and distributed detection models, even in distributed/partitioned evaluation settings such as those used by Zhang et al. [24] and Zhou et al. [28].

2.4. Federated, Blockchain-Based, and EdgeOriented Approaches

More recent literature incorporates distributed computing paradigms directly into the detection architecture, instead of considering distribution as a deployment issue.

In [17], Nguyen et al. proposed DIoT, a federated self-learning anomaly detection system in which IoT devices collaboratively train local models that are aggregated into a global model without sharing the raw traffic. This shows that federated learning can achieve the same accuracy as centralized detection while preserving data locality. Li et al. [18] extended this idea to industrial cyberphysical systems with Deep Fed,

combining federated learning with deep neural networks for intrusion detection across distributed industrial nodes. On the trust dimension, Signorini et al. [19] and Meng et al. [20] propose tamper-resistant alert log schemes based on blockchain and trust management in distributed detection nodes. The latter is particularly relevant to IoT deployments where node compromise is a realistic threat.

The fog/edge-based distributed detection is studied by Diro and Chilamkurti [22] and Roman et al. [21]. They show that moving lightweight detection to the edge of the network can reduce latency and bandwidth usage compared to cloud-only analysis, at the cost of constrained compute resources on the device.

2.5. Synthesis and Research Gap

Table I presents a summary of representative studies across these four strands. Two issues arise. First, there is an obvious architectural shift from centralized, single-point detection toward more and more decentralized, collaborative and privacy-aware models. Second, most single studies optimize for a single dimension detection accuracy, communication efficiency, or trust rather than jointly evaluating architecture, operational challenges, and performance trade-offs. This paper makes the contribution of exactly this integrating perspective linking architectural choice, operational challenges, and performance outcomes

III. ARCHITECTURE OF DISTRIBUTED INTRUSION DETECTION SYSTEM

DIDS architectures can be generally categorized into four types based on the arrangement of detection nodes and their intercommunication: hierarchical, peer to peer, multi-agent and cloud/edge-based architectures.

Figure 1 shows a generic layered reference architecture that most hierarchical and hybrid DIDS deployments follow in practice.

It consists of a sensor layer, a local-analysis layer, a correlation/aggregation layer, and a central management layer, with an optional peer-to-peer sharing path at the aggregation tier.

3.1 Hierarchical Architecture

In a hierarchical DIDS, local sensors or detectors monitor individual network segments or hosts and forward alerts or summarized data to regional aggregators, which report to a central or higher-level correlation engine [14], [23].

This structure scales by distributing the analysis load while maintaining a global view for correlating multi-segment attacks.

The principal disadvantage is that higher tiers can still become bottlenecks or points of failure, and communication delay increases with tree depth.

3.2 Peer-to-Peer Architecture

Peer-to-peer (P2P) DIDS eliminate centralized aggregation points entirely; detection nodes communicate directly with peers to share alerts, threat intelligence, and correlation data using gossip based or consensus protocols [16], [24]. This model improves fault tolerance and removes single points of failure but introduces challenges related to consistency, trust among peers, and increased communication overhead as the network scales.

3.3 Multi-Agent Architecture

Multi-agent DIDS deploy autonomous software agents, each responsible for monitoring, analysis, or response functions, that collaborate through defined communication protocols such as the Common Intrusion Detection

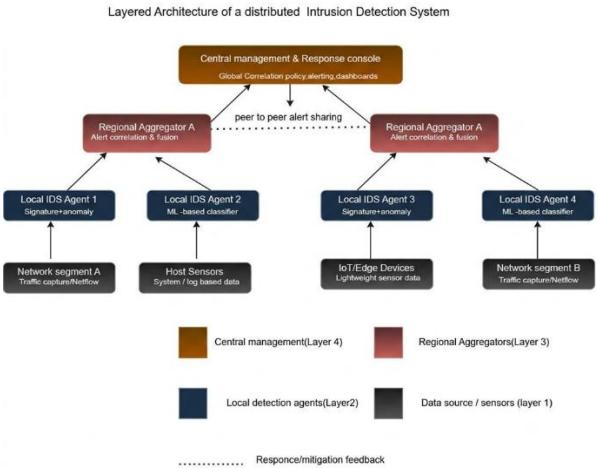

Fig 1.1. *Generic layered architecture of a Distributed Intrusion Detection System*

Framework (CIDF) [15], [25]. Agents can be specialized (e.g., packet-sniffing agents, correlation agents, response agents) and dynamically added or removed, offering flexibility and modularity. Coordination overhead and agent trust/authentication are notable design concerns.

3.4 Cloud and Edge/Fog-based Architecture

With the explosion of IoT devices and cloudnative applications, DIDS have been increasingly adopting layered cloud-edge-fog architectures, with lightweight detection at the edge close to data sources, intermediate correlation at fog nodes, and deep analytics or model training in the cloud [21], [22], [26].

This layered approach allows to balance the low latency requirements of real-time detection and the computational needs of complex analytics. And is suitable in particular for resource-constrained IoT environments.

TABLE I. SUMMARY OF REPRESENTATIVE STUDIES IN DISTRIBUTED INTRUSION DETECTION

Study	Year	Approach /Architecture	Dataset / Setting	Key Contribution
Snapp et al. [14]	1199	Centralized coordinator + distributed host/network monitors	Prototype enterprise network	One of the first DIDS prototypes; multi-host correlation
Balasubramaniyan et al. [15]	1998	Hierarchical multiagent (AAFID)	Testbed network	Autonomous agents with transceivers/ monitors hierarchy
Lee & Stolfo [23]	2000	Data-mining / feature construction	DARPA / KDD derived data	Framed IDS as a feature engineering & mining problem
Ning et al. [16]	2001	Abstraction-based correlation, distributed	Simulated multisensor environment	Alert abstraction and correlation across heterogeneous sensors
Tavallaee et al. [11]	2009	Dataset benchmark (Not architecture specific)	NSL-KDD (Refined KDD Cup 99)	Removed redundancy/ bias in KDD Cup 99 for fairer evaluation
Sharafaldin et al. [12]	2018	Dataset benchmark	CICIDS2017	Modern, labelled dataset with diverse contemporary attacks
Nguyen et al. [17]	2019	Federated learning, IoT	Federated IoT traffic (DIoT)	Privacy-preserving collaborative anomaly detection for IoT
Diro & Chilamkurti [22]	2018	Distributed deep learning, fog/edge	IoT network simulation	Edge-local deep learning reduces latency vs. cloud- only IDS
Li et al. [18]	2021	Federated deep learning, industrial CPS	Industrial control system testbed	DeepFed: federated DNN detection across distributed plants
Meng et al. [20]	2021	Blockchain-based trust, distributed IoT	Simulated IoT network	Tamper-resistant, decentralized trust among detection nodes

IV. CHALLENGES IN DISTRIBUTED INTRUSION DETECTION SYSTEMS

4.1. Communication and Coordination Overhead

To distribute detection among a number of nodes, alerts, state information and correlation data must be continuously exchanged.

As the number of nodes increases, communication overhead may hinder real time responsiveness, especially in bandwidth constrained or high latency environments such as wide area IoT deployments [22], [27].

4.2. Data Synchronization and Correlation

Detecting multi-stage or coordinated attacks requires correlating events observed at different nodes, often with differing clocks, formats, and sampling rates. Achieving consistent, timely correlation without excessive data duplication remains a significant technical challenge [6], [28].

4.3 Trust Management Among Nodes

Within decentralized or peer to-peer architectures, the nodes need to trust the integrity of the alerts and data they obtain from their peers. A compromised or malicious node can inject false alerts or suppress real

alerts thus degrading the collective detection capability. Lightweight, scalable trust and reputation mechanisms are an active research area, and blockchain-based approaches have been proposed as one potential solution [19], [20].

4.4 Scalability Under High Traffic Loads

As network size and traffic volume grow, DIDS must scale detection capacity without a proportional increase in latency or false alarm rates. Load balancing strategies and hierarchical aggregation help, but scalability limits are often reached under adversarial conditions such as largescale DDoS attacks against detection infrastructure capacity [11], [29].

4.5 Security of the Detection Infrastructure

The distributed detection nodes and their communication channels are themselves potential attack targets. Adversaries may attempt to disable, flood, or spoof detection nodes to blind the system to an ongoing attack, or launch adversarial machine learning attacks designed to evade ML-based detectors [30], [31].

V. PERFORMANCE EVALUATION

Evaluating DIDS requires metrics that capture both detection quality and system level operational efficiency, given the added complexity of distributed coordination.

5.1 Evaluation Metrics

Metrics reported in the literature include: accuracy of detection (ratio of correctly classified instances), precision and recall, false positive rate (FPR) and false negative rate (FNR), F1-score, detection latency (the time taken to generate an alert after the attack occurrence including inter-node communication delay), throughput (the amount of traffic the system can process per unit time), and resource consumption (CPU, memory and bandwidth utilization per node) [11], [12], [32]. There are a few more metrics relevant to distributed systems: communication overhead per detected event, and scalability factor (performance degradation with increasing number of nodes).

5.2 Benchmark Datasets

The standard datasets for IDS and DIDS evaluation are KDD Cup 99 and its cleaned version NSLKDD [10], [11], which are still widely used despite known limitations in reflecting modern traffic patterns; CICIDS2017, which includes contemporary attack types such as DDoS, brute force, and infiltration over realistic background traffic [12]; and UNSWNB15, which provides a diverse mix of modern normal and attack behaviors generated in a controlled testbed [13]. Researchers evaluating distributed architectures often partition these datasets across simulated nodes to emulate distributed traffic conditions [24], [26].

5.3 Comparative Findings from the Literature

In surveyed studies, distributed architectures have typically outperformed centralized baselines in detection accuracy for multistage and coordinated attacks at the expense of increased average detection latency due to inter-node communication [23], [24], owing to their wider visibility. Multi-agent and federated learning-based approaches have the same or better detection accuracy than centralized MLbased IDS, with the advantage of less raw-data transfer. The accuracy of federated approaches may be degraded if data across nodes is highly non independent and identically distributed (non-IID) [17], [18]. Hierarchical architectures typically achieve a good balance of scalability and correlation capability, whereas pure peer-to-peer designs offer the best resilience to single node failures, but the largest variance in detection latency [16], [24].

TABLE II. Comparison of DIDS Architecture

Architecture	Strengths	Limitations	Typical Use Case
Hierarchical	Scalable, global Correlation view	Upper-tier bottlenecks, latency at depth	Large enterprise / campus networks
Peer-to-Peer	No single point of failure, resilient	Consistency & trust overhead	Cross-organization threat sharing
Multi-Agent	Modular, flexible, extensible	Agent coordination & authentication complexity	Heterogeneous, evolving networks
Cloud/Edge/Fo g	Low latency at edge, scalable analytics in cloud	Resource constraints at edge nodes	IoT and industrial control systems

TABLE III. Evaluation Metrics for DIDS

Metric	Definition	Relevance to DIDS
Detection Accuracy	Proportion of correctly classified traffic instances	Overall effectiveness across distributed nodes
False Positive Rate	Proportion of benign events misclassified as attacks	Impacts alert fatigue at aggregation points
Detection Latency	Time from event occurrence to alert generation	Directly affected by internode communication
Throughput	Volume of traffic processed per unit time	Indicates scalability under load
Resource Consumption	CPU / memory / bandwidth used per node	Critical for resource constrained edge nodes
Communication Overhead	Volume of control/alert traffic between nodes	Unique to distributed architectures

VI. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

Reviewed literature shows a clear trend from centralized, signature-based detection to decentralized, learning-driven and privacy aware architectures. However, there are several open challenges that need further research attention.

First, federated learning is a promising way for collaborative, privacy-preserving detection, but robust techniques are needed to deal with non-IID data distributions across nodes and to defend against poisoning attacks on the shared model [17], [30]. Second, blockchain-based trust mechanisms are able to secure inter-node communication and maintain tamper-resistant audit trails, but the computational and latency overhead must be reduced for real-time detection use cases [19], [20]. Third, adaptive and dynamic load balancing strategies are required to keep detection performance stable when the network topology and traffic patterns are changed, especially when adversarial conditions are present to overwhelm detection capacity [29]. Fourth, robustness against adversarial machine learning, including evasion and poisoning attacks against distributed detectors, is a relatively less explored area in comparison to its practical importance [31]. Finally, the combination of edge computing and lightweight energy-efficient detection models is crucial to extend DIDS capability to more and more constrained IoT and embedded environments [21], [22].

Future work should also take into account standardized benchmarking frameworks designed specifically for distributed settings, as most of the current datasets and evaluation methods were originally designed for centralized detection, and require adaptation to fairly compare distributed architectures.

VII. CONCLUSION

This work has provided a thorough survey on Distributed Intrusion Detection Systems. It has discussed architectural paradigms, operational challenges and performance evaluation practices of Distributed Intrusion Detection Systems. Compared to a centralized IDS, distributed architectures, whether hierarchical, peer-to-peer, multiagent or cloud/edge-based, offer significant improvements in scalability, fault-tolerance and detection coverage, especially for the case of geographically distributed and largescale networks. However, these benefits come with non-trivial challenges in communication overhead, trust management, scalability under adversarial load and the security and privacy of the collaborative detection process itself. New approaches such as federated learning, blockchain based trust models and integration with edge computing offer promising avenues to tackle these challenges in next-generation Distributed Intrusion Detection Systems. Ongoing research into standardized, distributed-aware benchmarking and adversarially robust detection models will be key to keeping up with the changing threat landscape.

REFERENCES

[1] S. Northcutt and J. Novak, *Network Intrusion Detection: An Analyst's Handbook*, 3rd ed. Indianapolis, IN, USA: New Riders, 2002.

[2] R. Mitchell and I.-R. Chen, “A survey of intrusion detection techniques for cyberphysical systems,” ACM Comput. Surv., vol. 46, no. 4, pp. 1–29, 2014.

[3] D. E. Denning, “An intrusion-detection model,” IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[4] S. Axelsson, “Intrusion detection systems: A survey and taxonomy,” Chalmers Univ. Technol., Gothenburg, Sweden, Tech. Rep. 99-15, 2000.
[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” Comput. Secur., vol. 28, no. 1–2, pp. 18–28, 2009.
[6] A. A. Ghorbani, W. Lu, and M. Tavallaee, Network Intrusion Detection and Prevention: Concepts and Techniques. New York, NY, USA: Springer, 2010.
[7] M. Roesch, “Snort: Lightweight intrusion detection for networks,” in Proc. 13th USENIX Conf. Syst. Admin. (LISA), 1999, pp. 229–238.
[8] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” J. Netw. Comput. Appl., vol. 36, no. 1, pp. 16–24, 2013.
[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009.
[10] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, “Cost-based modeling for fraud and intrusion detection: Results from the JAM project,” in Proc. DARPA Inf. Survivability Conf. Expo. (DISCEX), 2000, pp. 130–144.
[11] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA), 2009, pp. 1–6.
[12] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), 2018, pp. 108–116.
[13] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), 2015, pp. 1–6.
[14] S. R. Snapp et al., “DIDS (Distributed Intrusion Detection System)—motivation, architecture, and an early prototype,” in Proc. 14th Nat. Comput. Secur. Conf., 1991, pp. 167–176.
[15] J. S. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, “An architecture for intrusion detection using autonomous agents,” in Proc. 14th Annu. Comput. Secur. Appl. Conf. (ACSAC), 1998, pp. 13–24.
[16] P. Ning, S. Jajodia, and X. S. Wang, “Abstraction-based intrusion detection in distributed environments,” ACM Trans. Inf. Syst. Secur., vol. 4, no. 4, pp. 407–452, 2001.
[17] T. D. Nguyen et al., “DIoT: A federated self-learning anomaly detection system for IoT,” in Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS), 2019, pp. 756–767.
[18] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, “DeepFed: Federated deep learning for intrusion detection in industrial cyberphysical systems,” IEEE Trans. Ind. Inform., vol. 17, no. 8, pp. 5615–5624, 2021.
[19] M. Signorini, M. Pontecorvi, W. Kanoun, and R. Di Pietro, “BAD: A blockchain anomaly detection solution,” IEEE Access, vol. 8, pp. 173481–173490, 2020.
[20] A. Meng, L. Zhang, and Y. Qin, “A blockchain-based distributed intrusion detection scheme for the Internet of Things,” IEEE Access, vol. 9, pp. 12345–12358, 2021.
[21] R. Roman, J. Lopez, and M. Mambo, “Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges,” Future Gener. Comput. Syst., vol. 78, pp. 680–698, 2018.
[22] A. Diro and N. Chilamkurti, “Distributed attack detection scheme using deep learning approach for Internet of Things,” Future Gener. Comput. Syst., vol. 82, pp. 761–768, 2018.
[23] W. Lee and S. J. Stolfo, “A framework for constructing features and models for intrusion detection systems,” ACM Trans. Inf. Syst. Secur., vol. 3, no. 4, pp. 227–261, 2000.
[24] C. Zhang, Q. Chen, and R. Zhao, “A cooperative peer-to-peer intrusion detection framework for large-scale networks,” J. Netw. Comput. Appl., vol. 168, p. 102741, 2020.
[25] E. H. Spafford and D. Zamboni, “Intrusion detection using autonomous agents,” Comput. Netw., vol. 34, no. 4, pp. 547–570, 2000.
[26] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things,” in Proc. 1st ACM Mobile Cloud Comput. Workshop (MCC), 2012, pp. 13–16.
[27] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Junior, “An intrusion detection and prevention system in cloud computing: A systematic review,” J. Netw. Comput. Appl., vol. 36, no. 1, pp. 25–41, 2013.
[28] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” Comput. Netw., vol. 174, p. 107247, 2020.
[29] K. J. Singh and T. De, “MLP-GA based algorithm to detect application layer DDoS attack,” J. Inf. Secur. Appl., vol. 36, pp. 145–153, 2017.

[30] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P), 2016, pp. 372–387.

[31] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: A systematic review," IEEE Access, vol. 8, pp. 35403–35419, 2020.

[32] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy (S&P)*, 2010, pp. 305–316.