computers

MDPI

Review

Overview on Intrusion Detection Systems for Computers Networking Security

Lorenzo Diana [1], Pierpaolo Dini [2,*] and Davide Paolini [2]

[1] Independent Researcher, 56100 Pisa, Italy; ldiana.res@libero.it
[2] Department of Information Engineering, University of Pisa, Via G. Caruso 16, 56100 Pisa, Italy; davide.paolini@ing.unipi.it
* Correspondence: pierpaolo.dini@unipi.it

Abstract: The rapid growth of digital communications and extensive data exchange have made computer networks integral to organizational operations. However, this increased connectivity has also expanded the attack surface, introducing significant security risks. This paper provides a comprehensive review of Intrusion Detection System (IDS) technologies for network security, examining both traditional methods and recent advancements. The review covers IDS architectures and types, key detection techniques, datasets and test environments, and implementations in modern network environments such as cloud computing, virtualized networks, Internet of Things (IoT), and industrial control systems. It also addresses current challenges, including scalability, performance, and the reduction of false positives and negatives. Special attention is given to the integration of advanced technologies like Artificial Intelligence (AI) and Machine Learning (ML), and the potential of distributed technologies such as blockchain. By maintaining a broad-spectrum analysis, this review aims to offer a holistic view of the state-of-the-art in IDSs, support a diverse audience, and identify future research and development directions in this critical area of cybersecurity.

Keywords: intrusion detection systems (IDSs); network security; cybersecurity; anomaly detection; signature-based detection; ML; AI; scalability; zero-day attacks; cloud computing; IoT

Academic Editors: Hooman Alavizadeh and Ahmad Salehi Shahraki

Received: 23 December 2024
Revised: 24 February 2025
Accepted: 28 February 2025
Published: 3 March 2025

Citation: Diana, L.; Dini, P.; Paolini, D. Overview on Intrusion Detection Systems for Computers Networking Security. *Computers* **2025**, *14*, 87. https://doi.org/10.3390/computers14030087

1. Introduction

1.1. Needs for Intrusion Detection Systems

With increasingly advanced and pervasive cyber threats, it has become crucial to implement solutions that can identify and combat unauthorized activity within networks. Intrusion Detection Systems (IDS) play a key role in modern security architectures, as they provide a proactive line of defense. Unlike more classic preventive measures, such as firewalls and antivirus software, which aim to block known threats, IDSs can dynamically monitor and analyze network traffic, identifying patterns of anomalous or suspicious behavior that can reveal both known and new attacks. The use of IDSs is justified by some fundamental requirements:

- Adaptability to Emerging Threats: Cyber adversaries are constantly developing new techniques and attack vectors to bypass standard defenses, exploiting unknown or zero-day vulnerabilities. IDSs must be able to adapt to these emerging threats and detect zero-day attacks that could not be identified by security measures based solely on known attack signatures (or patterns) [1–4].

- Scalability and Performance: As traffic volumes increase in modern networks, IDSs must be able to scale, efficiently and quickly handle large amounts of data without compromising detection accuracy. This is especially important in enterprise or cloud networks, where traffic spikes can be extreme and unpredictable [5–7].
- Reduction in False Positives and False Negatives: An effective IDS must minimize the number of false positives, which are security alerts generated by legitimate activity that are interpreted as malicious, and false negatives, which are real attacks that the IDS fails to detect. False positives can overload security teams, distract from identifying real threats, and slow down business operations. Conversely, a high FNR can allow real threats to operate undetected, which can have serious consequences [8–12].

1.2. Objectives of This Review

This paper aims to provide an in-depth review of IDS technologies for network security, examining both traditional methods and more recent developments. The study aims to cover the following main areas:

- Provide an overview of IDS architectures and types: This review will explore the main categories of IDSs, including host-based (HIDS) and network-based (NIDS), and the distinctions between signature-based and anomaly-based detection. Hybrid and behavior-based systems, which seek to combine different approaches to improve detection capability and reduce false positives, will also be explored.
- Discuss key detection techniques: Another goal is to provide a detailed discussion of the detection techniques used by IDSs, from signature-based techniques to modern AI and ML applications, including deep learning models. We will examine how these techniques enable IDSs to identify malicious activity even in highly complex and rapidly evolving network environments.
- Review datasets and test environments: To effectively evaluate an IDS, it is essential to have datasets that realistically represent common network traffic and attacks. This review will analyze the most commonly used standard datasets for IDS evaluation, discussing their advantages and limitations. Additionally, a section will address issues related to the representativeness of datasets, such as data staleness and lack of variety in the representation of recent attacks.
- Review IDS implementations in modern network environments: With the increasing use of cloud computing, virtualized networks, and the IoT, IDSs must be adapted to operate in these environments. Special attention will be given to discussing IDSs designed for cloud networks, Software-Defined Networking (SDN), IoT, and ICS, highlighting the specific technical and operational challenges for each environment.
- Identifying current challenges and limitations of IDSs: Despite technological advances, IDSs still face various challenges, such as managing large volumes of data, increasing threat complexity, and maintaining high performance in terms of scalability and accuracy. The main limitations of current IDSs will be discussed, providing the reader with a clear overview of the obstacles that need to be overcome to develop more effective solutions.
- Exploring emerging trends and innovations: Finally, this review will provide a look at emerging trends in the field of IDSs, with a particular emphasis on advanced technologies such as AI and ML, and the potential of distributed technologies such as blockchain to improve network security. Furthermore, progress towards the development of autonomous and proactive IDSs, capable of automatically responding to ongoing attacks, will be explored.

Maintaining a broad-spectrum analysis of IDS applications offers several advantages over reviews focused on specific topics, such as the most used AI algorithms:

- Comprehensive Overview: A broad analysis provides a holistic view of various technologies, methodologies, and applications of IDS, making it useful for readers seeking a general understanding of the field.
- Identification of Interconnections: It helps identify interconnections between different technologies and approaches, such as how AI algorithms can be integrated with other detection techniques or how scalability challenges affect different IDS architectures.
- Relevance to a Diverse Audience: A broad-spectrum review is relevant to a wider audience, including academics, cybersecurity professionals, software developers, and business decision-makers, serving as a starting point for further research or practical implementation.
- Identification of Research Gaps: By examining a wide range of topics, it is easier to identify research gaps and areas needing further study, guiding researchers towards new directions and innovation opportunities.
- Support for Strategic Decisions: For business decision-makers and security managers, a broad review provides valuable insights for making strategic decisions on technology adoption, integration of security solutions, and planning network protection.
- Flexibility and Adaptability: A broad analysis allows for better adaptation to rapid changes in the cybersecurity field. As threats and technologies evolve quickly, having a comprehensive understanding of various options and approaches enables more flexible and proactive responses.
- Foundation for Specific Studies: A broad-spectrum review can serve as a foundation for more specific studies. Readers can use the general information to delve deeper into particular topics of interest, such as AI algorithms, anomaly detection techniques, or cloud environment implementations.

In summary, maintaining a broad-spectrum analysis of IDS applications provides a comprehensive and integrated view of the field, supports a diverse audience, and facilitates the identification of new research and development opportunities. This approach is particularly valuable in a dynamic and rapidly evolving field like cybersecurity.

1.3. Comparison with Prior Reviews

Several reviews on IDSs have been published over the years, each contributing valuable insights into specific aspects of IDS technologies. However, our review aims to provide a more comprehensive and integrated perspective by addressing the following gaps identified in previous reviews:

1. Scope and Breadth. Many prior reviews focus on specific aspects of IDSs, such as particular detection techniques or applications in specific environments. For example, in [13,14], the authors focus on IDSs for IoT networks; while in [15–17], the authors focus only on ML, deep learning, and federated learning techniques. Our review covers a broad spectrum of IDS technologies. We examine traditional methods and recent advancements, including AI and ML applications, and explore their implementations across various modern network environments such as cloud computing, IoT, and industrial control systems.
2. Integration of Advanced Technologies: Previous reviews often provide limited discussion on the integration of advanced technologies like AI, ML [18], and blockchain [19]. Our review places special emphasis on these technologies, exploring their potential to enhance IDS capabilities and address emerging cybersecurity threats.
3. Comprehensive Dataset Analysis: While some reviews touch upon datasets used for IDS evaluation, our review provides an in-depth analysis of standard datasets, discussing their advantages, limitations, and challenges associated with testing IDSs.

This comprehensive dataset analysis is crucial for understanding the effectiveness of IDSs in real-world scenarios.

4. Current Challenges and Limitations: Our review thoroughly addresses the current challenges and limitations faced by IDSs, such as scalability, performance, and the reduction in false positives and negatives. We also discuss privacy and ethical concerns related to network traffic inspection, which are often overlooked in previous reviews.
5. Emerging Trends and Innovations: We explore emerging trends and innovations in the field of IDSs, providing insights into the development of autonomous and proactive IDSs capable of automatically responding to ongoing attacks. This forward-looking perspective is essential for guiding future research and development in IDS technologies.

By addressing these gaps, our review aims to offer a holistic view of the state of the art in IDSs, support a diverse audience, and identify future research and development directions in this critical area of cybersecurity.

1.4. Paper Structure

To pursue these goals, this paper is organized as follows: Section 1 introduces the topic, presenting the motivation and objectives of the review, and defining the scope of the paper. Section 2 describes the search methods and the inclusion and exclusion criteria used to collect relevant papers mentioned in this review. Section 3 provides an in-depth analysis of IDS architectures and types, covering NIDSs, HIDSs, hybrid systems, and detection techniques such as signature-based, anomaly-based, and behavior-based methods. Section 4 provides an overview of the intrusion detection techniques. Examined detection techniques include signature-based, anomaly-based, and behavior-based. This section also presents a comparison of these techniques, examining their efficacy in detecting unknown threats, assessing their computational requirements, and evaluating the incidence of false positives. Moreover, it highlights the advantages and disadvantages of each technique, alongside typical application scenarios in which they are employed. Section 5 presents the analysis of the most recent literature on IDSs in modern networks including cloud and virtualized networks, IoT and sensor networks, SDN (Software-Defined Networking), and industrial networking systems. This section also presents open challenges and future research directions in modern IDSs. Section 6 reviews standard datasets and testing procedures for IDS evaluation, including dataset preparation, preprocessing, and methodologies for assessing IDS effectiveness. Finally, Section 7 concludes the paper by summarizing key findings and outlining future research directions, emphasizing the need for continuous innovation to keep pace with evolving cybersecurity threats.

2. Search Methods and Inclusion/Exclusion Criteria

To ensure a comprehensive and systematic review of the current state and advancements in IDSs for network security, we employed a rigorous methodology to gather and evaluate relevant literature. This section details the search methods, databases consulted, keywords used, and the inclusion/exclusion criteria applied.

2.1. Search Methods

The literature search was conducted using the following electronic databases to ensure a wide coverage of relevant studies:

- ACM Digital Library.
- IEEE Xplore.
- Springer Link.
- MDPI.
- ScienceDirect.

The search was performed using a combination of keywords and phrases related to IDSs and network security. The primary keywords included the following:

- "Intrusion Detection Systems".
- "Network Security".
- "Anomaly Detection".
- "Signature-Based Detection".
- "ML in IDS".
- "AI in IDS".
- "Cloud Security".
- "IoT Security".
- "Industrial Control Systems Security".

2.2. Search Strategy

The search strategy involved the following steps:

1. Initial Search: An initial search was conducted using the primary keywords in each database. This step aimed to identify a broad range of potentially relevant articles.
2. Refinement of Search Terms: Based on the initial search results, the search terms were refined to include additional relevant keywords and phrases. Boolean operators (AND, OR) were used to combine search terms effectively.
3. Screening of Titles and Abstracts: The titles and abstracts of the retrieved articles were screened to assess their relevance to the review's objectives. Articles that did not meet the inclusion criteria were excluded at this stage.
4. Full-Text Review: The full texts of the remaining articles were reviewed to ensure they met the inclusion criteria. Any discrepancies or uncertainties were resolved through discussion among the authors.

2.3. Inclusion Criteria

The following inclusion criteria were applied to select articles for the review:

1. Time Period: Articles published between 2019 and 2024 were included. This time frame was chosen to capture the most recent advancements and challenges in IDS technologies.
2. Type of Publication: Only peer-reviewed journal articles, conference papers, and technical reports were considered. This criterion ensured the inclusion of high-quality and credible sources.
3. Relevance: Articles had to specifically address IDS technologies, including detection techniques, implementations in various network environments, and evaluations using standard datasets. Studies focusing on related topics such as ML, AI, and cybersecurity in cloud, IoT, and ICSwere also included.
4. Language: Only articles published in English were included to maintain consistency in language and ease of analysis.

2.4. Exclusion Criteria

The following exclusion criteria were applied to filter out irrelevant or redundant studies:

1. Non-English Publications: Articles not published in English were excluded to ensure consistency in language and ease of analysis.
2. Irrelevant Topics: Articles that did not focus on IDSs or related topics were excluded. For example, studies focusing solely on unrelated aspects of network security without addressing IDSs were not considered.
3. Duplicate Studies: Duplicate studies or articles presenting the same findings were excluded to avoid redundancy. In cases where multiple articles reported similar findings, the most comprehensive and recent study was included.

4. Incomplete Data: Articles lacking sufficient data or methodological details to support their findings were excluded.

2.5. Data Extraction and Synthesis

The selected articles were subjected to a detailed data extraction process, which involved the following steps:

1. Extraction of Key Information: Key information such as the study's objectives, methods, findings, and conclusions were extracted from each article. This information was organized into a structured format to facilitate comparison and synthesis.
2. Evaluation of Methodological Quality: The methodological quality of each study was assessed using predefined criteria. Studies with significant methodological flaws were excluded from the final synthesis.
3. Synthesis of Findings: The extracted data were synthesized to identify common themes, trends, and gaps in the literature. The synthesis process involved both qualitative and quantitative analysis, where applicable.

At the end of this process, we included 30 articles providing a comprehensive overview of the current state of IDS technologies, highlighting key detection techniques, implementations in various network environments, and emerging trends. Another 68 articles were analyzed to describe datasets and related testing procedures.

3. Analysis of IDS Architectures and Types

3.1. Computer Networks Components, Architecture and Vulnerabilities

Computer networks are the fundamental infrastructure for modern technologies, used to communicate, share resources, and run critical applications. Understanding the main components and common architectures of these networks is essential for designing efficient, secure, and scalable systems.

3.1.1. Major Components of Computer Networks

A computer network is made up of a combination of hardware and software, each with a specific role in ensuring connectivity and overall functionality. Important hardware components include devices such as routers, switches, modems, and access points, which act as the "backbone" of the network. A router, for example, is a crucial device that connects different networks and routes data traffic based on IP addresses. Operating at the network level, routers ensure communication between separate subnets and with the Internet. They often have security features, such as built-in firewalls, to protect data in transit. However, if not properly sized, they can become a bottleneck, especially in high-traffic networks. Efficient router design and capacity planning are critical for maintaining optimal performance in high-demand environments, as improperly scaled routers can lead to significant latency issues [20]. Switches, on the other hand, focus on connecting devices within a single local area network (LAN). These devices operate at the data link layer and use MAC addresses to determine the path of data packets. Compared to hubs, switches offer more intelligent traffic management, reducing congestion and increasing network efficiency. However, they are not designed to handle traffic between different networks, limiting themselves to intra-LAN operations [21]. In wireless networks, an access point (AP) is essential to allow devices to connect to the network without the use of cables. APs are indispensable in environments where physical cabling would be complicated or expensive, such as in large offices or public spaces. Although they offer considerable flexibility, their effectiveness can be compromised by interference or physical obstacles, and they are generally slower than wired connections [22]. In addition to network devices, terminals also play an essential role. Computers, smartphones, servers, and IoT devices are the "end points" of the network,

where data are consumed or generated. Every terminal is a potential point of vulnerability, highlighting the importance of implementing adequate security measures, such as antivirus and encryption [23]. Another fundamental aspect of computer networks is the means of transmission. Wired networks primarily use Ethernet cables, known for their reliability and low latency. However, cabling can be time-consuming and expensive, especially in distributed or constantly evolving environments. Alternatively, wireless networks, based on technologies such as Wi-Fi, offer greater flexibility, but can be subject to interference and security vulnerabilities. Finally, network management software is essential to the operation of these infrastructures. Network Operating Systems (NOSs) and communication protocols, such as TCP/IP, are responsible for coordinating the interaction between the various devices, ensuring reliable data transmission.

3.1.2. Network Architectures: Common Models and Their Characteristics

Network architectures define the logical and physical structure of a network, determining how devices interact and exchange data. Different architectures are used based on specific application needs, each with its own advantages and limitations. A very simple and straightforward architecture is peer-to-peer (P2P), in which all connected devices act as both clients and servers. This configuration eliminates the need for a centralized server, making it ideal for home networks or small workgroups. However, its simplicity also represents a limitation: as the number of devices increases, connection management becomes increasingly complex, and security concerns grow, as each node must independently protect shared resources [24–26]. In enterprise contexts, the client–server architecture is a more common choice. In this model, one or more central servers manage requests from client devices, providing access to resources such as files, applications, or databases. Centralization simplifies management and improves security, but it creates a significant dependency on the server: a server failure can paralyze the entire network. In addition, implementing and maintaining a client–server infrastructure can be expensive, especially for small organizations [27,28]. Among physical network topologies, the star configuration is one of the most common. In this structure, all devices are connected to a central point, typically a switch or a hub. This approach offers good fault tolerance since the failure of one device does not compromise the entire network. However, if the central node were to fail, the entire system would be unusable. The star topology is particularly suitable for corporate networks and offices, where reliability and ease of maintenance are priorities [29,30]. Another interesting configuration is the mesh configuration, in which each device is connected to multiple other devices. This architecture offers significant fault tolerance, since there are always alternative paths for the data, even if one or more links fail. However, the cost and complexity of this structure limit its application to specific environments, such as data centers or IoT networks, where redundancy is essential [31,32]. Finally, many modern networks adopt hybrid approaches, combining elements of multiple architectures to exploit the advantages of each. For example, a corporate network might use a star topology for its main offices, supplemented with mesh segments for distributed data centers.

3.1.3. Vulnerabilities

Computer communication networks are intricate structures that serve as both indispensable resources and critical points of vulnerability in the digital ecosystem. Each hardware and software component of a network, alongside the overarching architecture that interconnects them, plays a decisive role in determining the network's exposure to security threats. The relationship between cybersecurity and network design is deeply intertwined, as the technical choices made during the construction of a network directly

impact its resilience against attacks and breaches [33]. Within a network, communication devices such as routers, switches, and access points hold pivotal roles in managing and transmitting data. The router, for instance, serves as the interconnection point between different networks, directing data packets to their intended destinations through routing tables and IP addressing. However, due to its exposure to external traffic, the router often becomes a primary target for malicious actors. A compromised router may facilitate the interception of sensitive information, the redirection of traffic to malicious sites, or even the complete disruption of communications. The risk is further exacerbated when firmware updates are neglected, or when default configurations, such as factory-set credentials, remain unchanged [34]. Switches, which operate predominantly within local area networks (LANs), are critical devices for enabling communication among devices in the same network. While generally considered less exposed than routers, switches are not immune to exploitation, particularly in the case of misconfigurations. A common attack involves flooding the switch with excessive requests, forcing it to act as a hub and thereby increasing the risk of data interception. Additionally, the absence of adequate network segmentation may amplify the potential damage, exposing all devices within the same segment to security threats [35]. Wireless connections, while providing significant operational flexibility, introduce an additional layer of vulnerability. Access points (APs), for example, are susceptible to de-authentication attacks, whereby an attacker forcibly disconnects a legitimate device to assume its position. The inherent nature of wireless communication also makes it easier for adversaries to intercept network traffic, especially when outdated security protocols are employed, or when data encryption is insufficiently robust [36]. Endpoints, including computers, smartphones, and IoT devices, constitute another critical element of network security. These devices are often the most vulnerable points within a network, as many cyberattacks target end users directly through social engineering techniques such as phishing. Once compromised, an endpoint can serve as a gateway for unauthorized access to internal network resources or as a vector for distributing malware to other connected devices [37]. The transmission medium also plays a significant role in network security. Wired connections, such as Ethernet cables, are generally deemed secure but may still be subject to physical tampering or interception if not adequately protected. Conversely, wireless connections expand the attack surface, rendering the network more susceptible to eavesdropping, jamming, or spoofing attacks. The protocols governing communication within the network are another critical factor in determining security. Protocols such as TCP/IP, despite their robustness, are not impervious to vulnerabilities. For instance, IP spoofing attacks allow malicious actors to impersonate legitimate devices, thereby gaining unauthorized access to network resources. Similarly, the Domain Name System (DNS), essential for translating domain names into IP addresses, is frequently targeted by cache poisoning attacks that redirect users to malicious destinations [37]. The choice of network architecture significantly influences the level of exposure to threats. Peer-to-peer (P2P) networks, for example, are inherently vulnerable due to their decentralized nature, where each node functions as both a client and a server. This lack of central oversight complicates the monitoring and management of security. In contrast, client–server architectures, commonly employed in enterprise environments, offer enhanced control through centralized resource management. However, this centralization introduces a single point of failure: a successful attack on the central server can have catastrophic consequences for the entire network [38]. Star topology networks are often preferred for their simplicity and ease of management. In this configuration, a central node connects all other devices, facilitating traffic segmentation and monitoring. However, the compromise of this central node can incapacitate the entire network. On the other hand, mesh networks, characterized by redundancy and multiple data paths, provide higher fault tolerance. This resilience, however, introduces complex-

ity that complicates the detection of anomalous behavior and the monitoring of network activity [39]. Network segmentation, frequently implemented through Virtual Local Area Networks (VLANs) or firewalls, is a crucial practice for limiting access to critical resources. Although segmentation significantly reduces the risk of lateral movement by attackers, improper configuration may expose sensitive segments or allow unauthorized access [40]. Mitigating these vulnerabilities necessitates a multi-layered approach to network security. Hardware protection, the adoption of advanced cryptographic protocols, and continuous network monitoring are all fundamental components of an effective defense strategy. Moreover, user education plays a vital role in preventing social engineering attacks, which often exploit human error to circumvent technical defenses. Ultimately, securing a network cannot rely solely on advanced technologies; it requires a combination of intelligent design, rigorous configuration, and a sustained commitment to addressing emerging threats [41].

3.2. Types of IDSs

IDSs represent a critical layer of defense within a cybersecurity framework. By monitoring network traffic and system activities for signs of malicious behavior or policy violations, IDSs provide organizations with the ability to detect and respond to threats in real time. The classification of IDSs is a complex task, as different systems are designed to address specific operational environments, threat models, and performance requirements. Broadly speaking, IDSs can be categorized into NIDSs, HIDSs, and hybrid IDS, with additional distinctions based on detection methodology, such as signature-based, anomaly-based, and specification-based systems.

- NIDSs operate at the network level, continuously analyzing packets traversing network links to identify malicious activity. These systems are typically deployed at strategic points in the network infrastructure, such as at gateways or between subnets, to provide comprehensive visibility into traffic flows. An NIDS relies on techniques like DPI and flow analysis to detect a wide range of threats, including denial-of-service (DoS) attacks, malware propagation, and port scanning. One of the strengths of an NIDS lies in its ability to monitor multiple devices simultaneously, making it suitable for large-scale, distributed networks. However, NIDSs face challenges in environments with high traffic volumes or encrypted data. The performance of an NIDS can degrade under excessive traffic loads, leading to packet drops and missed detections. Moreover, the increasing adoption of encryption protocols such as TLS/SSL complicates packet inspection, as the payload is inaccessible without decryption, potentially creating blind spots for the system. Techniques such as decryption proxies or metadata analysis are employed to mitigate these challenges, but they introduce additional complexity and potential privacy concerns [42,43].
- HIDSs operate on individual endpoints, such as servers, desktops, or virtual machines, by monitoring system-level activities. This includes observing log files, process behavior, file system modifications, and system calls. An HIDS is well suited to detecting threats that specifically target the host, such as unauthorized access attempts, malware execution, or privilege escalation. One of the primary advantages of HIDSs is their ability to detect local threats that might evade network-level monitoring. For example, if a malicious actor gains access to a machine via a USB device or a phishing email, HIDSs can detect the subsequent unauthorized activities. Additionally, HIDSs are particularly effective in environments where network visibility is limited, such as on encrypted endpoints or in cloud-based systems. However, the reliance on individual host monitoring introduces scalability challenges in large organizations. Managing, configuring, and maintaining HIDSs across hundreds or thousands of devices requires substantial administrative effort. Moreover, HIDSs are inherently less capable

of detecting attacks that span multiple hosts or the broader network, necessitating integration with network-level systems for a holistic defense [44,45]. Figure 1 shows schematic representation of the HIDS concept.

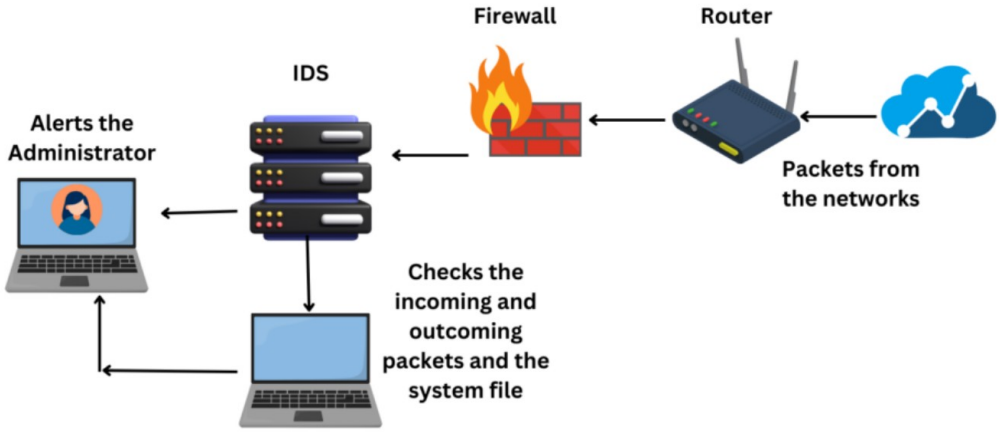

Figure 1. Schematic representation of the HIDS concept.

- Hybrid IDSs combine the capabilities of NIDSs and HIDSs to provide a more comprehensive security solution. These systems leverage the strengths of both approaches by monitoring both network traffic and host activities. By correlating data from multiple sources, hybrid IDSs can achieve greater accuracy in threat detection and reduce false positives. A key advantage of hybrid IDSs is their ability to provide contextual awareness. For instance, a network anomaly detected by the NIDS can be correlated with suspicious file access on a host, enabling more effective detection of multi-stage attacks, such as APTs. Furthermore, hybrid systems can provide defense-in-depth by applying detection mechanisms at both the network perimeter and individual endpoints. However, the integration of multiple data streams introduces complexity in system design and operation. Hybrid IDSs often require significant computational resources and sophisticated algorithms to process and correlate the vast amount of data generated. The increased complexity also raises challenges related to deployment, configuration, and maintenance, as well as potential latency in real-time detection [46,47].
- Signature-based IDSs rely on predefined patterns or "signatures" of known threats to identify malicious activity. These systems compare observed behavior or traffic against a database of signatures, such as specific byte sequences, known malware hashes, or anomalous commands. Signature-based detection is highly effective for identifying well-documented threats and offers the advantage of low false-positive rates. Despite their effectiveness against known threats, signature-based IDSs have significant limitations in detecting novel or polymorphic attacks. Cyber adversaries frequently modify their techniques to evade detection, rendering signatures obsolete. As a result, maintaining and updating the signature database is a continuous and resource-intensive process [48–50].
- Anomaly-based IDSs detect threats by identifying deviations from established baselines of normal behavior. These baselines can be defined using statistical models, ML algorithms, or heuristic approaches. Unlike signature-based systems, anomaly-based IDSs are capable of detecting previously unknown threats, making them suitable for dynamic and evolving threat landscapes. However, the reliance on behavioral baselines introduces challenges, particularly in environments with high variability. Normal behavior in such systems can be difficult to define, leading to an increased likelihood of false positives [51–53]. Furthermore, the computational overhead of

anomaly detection is often higher than signature-based approaches, necessitating robust infrastructure and optimization to ensure real-time performance. Figure 2 shows the schematic representation of both the anomaly-based and signature-based IDS concepts.

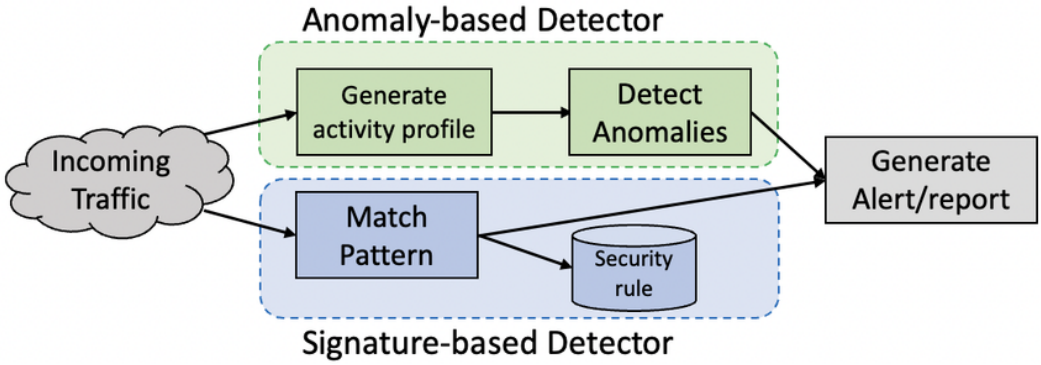

Figure 2. Schematic representation of both the anomaly-based and signature-based IDS concepts.

- Specification-based IDSs represent a hybrid approach that combines the deterministic nature of rule-based systems with the flexibility of anomaly detection. These systems rely on predefined specifications of expected behavior, often created manually by experts, to identify violations indicative of malicious activity. This approach offers higher precision compared to purely anomaly-based systems and is particularly effective in controlled environments with well-defined operational parameters, such as ICS or IoT networks. However, the manual effort required to define specifications limits scalability and adaptability to new threats or rapidly changing environments [54,55].

The choice of IDS type depends on the operational environment, threat landscape, and resource constraints of the organization. NIDSs are highly effective in perimeter defense but face challenges with encrypted traffic and high data volumes. HIDSs excel in detecting endpoint-specific threats but require significant management effort in large deployments. Hybrid IDSs offer the most comprehensive coverage but come at the cost of increased complexity. Similarly, the choice between signature-based, anomaly-based, or specification-based systems depends on the balance between detection accuracy, resource requirements, and adaptability to new threats.

3.3. Evaluation Metrics for IDSs

The evaluation of IDSs is a multifaceted process that requires careful consideration of both performance-related metrics and system-level characteristics. While traditional metrics such as detection accuracy, false-positive rates, and throughput are essential for understanding the effectiveness of an IDS, real-world applications often necessitate a broader assessment. This includes examining qualitative and quantitative attributes that influence the IDS's integration, efficiency, and applicability in specific contexts, such as embedded systems or resource-constrained environments. Traditional performance metrics focus on the detection capabilities and operational efficiency of an IDS. Detection rate, also referred to as the TPR, measures the proportion of actual attacks that the IDS correctly identifies. Achieving a high detection rate is crucial to ensure threats are promptly flagged and mitigated. FPR, in contrast, evaluates the proportion of benign activities incorrectly classified as attacks. Keeping this rate low is vital to reduce alert fatigue and maintain the trust of system administrators. Similarly, the FNR represents the proportion of actual attacks that the IDS fails to detect; minimizing this rate is essential to prevent undetected breaches. Precision, defined as the ratio of correctly identified attacks to the total number of flagged events, assesses the relevance of alerts and complements the detection rate when evaluating

system reliability. Recall, synonymous with the detection rate, emphasizes the IDS's ability to detect all relevant threats. The F1-Score provides a balanced metric that is especially useful when there are trade-offs between these two attributes. Additionally, throughput measures the volume of data the IDS can process per unit time, an important consideration in high-traffic networks where delays must be avoided. In many cases, the practicality of IDSs extends beyond detection performance and includes metrics that assess system integration. Memory footprint evaluates the amount of RAM and storage required for the IDS to function effectively, a critical consideration for embedded systems where hardware constraints often limit available memory. Systems with a minimal memory footprint are better suited for resource-constrained devices such as IoT sensors and industrial controllers. Computational overhead, which measures the processing power required by the IDS to analyze data and perform detections, directly impacts its usability in scenarios where computational resources are limited. Lightweight algorithms and optimized processing pipelines are essential to ensure that the IDS does not hinder the primary functions of the host system. Latency, or the time delay introduced by the IDS in detecting and responding to an intrusion, is another important metric. Low latency is crucial for real-time detection and response, particularly in time-sensitive environments such as ICS or critical infrastructure. Code complexity, which refers to the intricacy of the IDS's implementation, plays a significant role as well; lower complexity reduces the likelihood of software bugs, simplifies debugging, and facilitates updates or customizations. Energy efficiency, especially for mobile and battery-operated devices, is increasingly important as energy-efficient systems extend device operational times and reduce the need for frequent recharging or battery replacements. Scalability, defined as the ability of the IDS to maintain performance as the network size or traffic volume increases, is a key factor in dynamic environments where workloads can fluctuate unpredictably. The deployment context of an IDS often introduces unique requirements and constraints. In embedded systems, the IDS must prioritize minimal resource usage while maintaining effective detection capabilities. Factors such as interrupt handling efficiency, stack size utilization, and compatibility with real-time operating systems (RTOSs) are critical. Physical security considerations, such as resistance to side-channel attacks, may also influence the evaluation. In cloud-based architectures, the ability of the IDS to handle distributed workloads and virtualized environments is paramount. Support for multi-tenancy, integration with containerized services such as Kubernetes and Docker, and elasticity are essential metrics. Cloud-based IDSs must also incorporate robust logging and data aggregation mechanisms to ensure comprehensive threat visibility across dispersed resources. In enterprise environments, integration with existing security infrastructure, such as firewalls, Security Information and Event Management (SIEM) systems, and endpoint protection solutions, is crucial. Compliance with industry standards and regulations, such as GDPR or HIPAA, is often necessary for deployment in regulated industries. An IDS's ability to adapt to evolving threats and recover from failures is an increasingly important aspect of its evaluation. Adaptability assesses how easily the IDS can accommodate new attack signatures or behavioral models, with ML- or AI-based systems often demonstrating superior adaptability. Fault tolerance ensures the IDS remains operational even under adverse conditions, such as hardware malfunctions or software crashes. Interoperability measures the ability of the IDS to function seamlessly alongside other security tools and network devices, often requiring standards compliance and support for diverse protocols. Finally, resilience against adversarial attacks evaluates the system's ability to withstand targeted efforts, such as evasion tactics or adversarial ML techniques designed to exploit detection weaknesses. Evaluating an IDS often involves balancing competing priorities, such as detection accuracy versus resource consumption or scalability versus latency. For example, signature-based systems may achieve high

precision but require frequent updates to maintain effectiveness, while anomaly-based systems might detect a broader range of threats but face higher false-positive rates and computational demands. These trade-offs must be carefully analyzed to ensure the optimal selection of an IDS for its intended application. As networks and threats evolve, new evaluation criteria are emerging. These include leveraging advanced technologies, such as hardware accelerators like GPUs or TPUs, to enable real-time analysis. The ability to inspect encrypted traffic without violating privacy, compliance with ethical AI principles in ML-based systems, and the sustainability of the IDS, including its energy consumption and carbon footprint, are also gaining prominence as organizations prioritize environmentally conscious solutions.

4. Overview on Intrusion Detection Techniques

4.1. Signature-Based Detection

Signature-based detection operates on the principle of identifying malicious activities through the comparison of observed data with predefined patterns, referred to as signatures. These signatures are meticulously crafted to represent known threats, encompassing everything from specific sequences of bytes within network packets to distinct behavioral patterns exhibited by malware. This approach ensures a high degree of precision in detecting threats that match established criteria. The operational workflow of signature-based detection is a multi-step process designed to maximize efficiency and accuracy. It begins with data collection, where raw information from network traffic, system logs, and application interactions is captured. This stage often relies on tools such as Wireshark or other network sniffing technologies to monitor data streams in real time. These tools are configured to collect comprehensive information while minimizing unnecessary overhead. The collected data then undergo a preprocessing phase, which is crucial for ensuring the system's effectiveness. During this phase, irrelevant or redundant information is filtered out, and the remaining data are formatted into a consistent structure. This normalized data are then subjected to the pattern-matching phase, where they are compared against an extensive and frequently updated repository of threat signatures. Specialized tools, including Snort and Suricata, play a pivotal role at this stage, leveraging high-performance engines to conduct rapid comparisons and identify matches. If a signature match is identified, the system proceeds to the alerting and response phase, which involves notifying administrators and initiating automated mitigation measures. These measures might include blocking a specific IP address, terminating suspicious sessions, or isolating compromised endpoints. The system's response can be tailored to the organization's security policies and the criticality of the detected threat. Signature-based detection is particularly adept at identifying well-documented threats, such as malware variants, SQL injection attempts, and XSS exploits. Its strength lies in its speed and reliability, especially in environments with high traffic volumes where rapid decision-making is crucial. However, its reliance on static signatures means it cannot detect new or evolving threats, such as zero-day exploits or attacks using polymorphic techniques to alter their signatures dynamically. As such, this method is best deployed as part of a layered defense strategy, where it addresses known threats while complementary methods handle more sophisticated attack vectors. While signature-based IDSs are effective at recognizing previously known attacks, they are inadequate at detecting new, unknown, or zero-day threats. This limitation has led to the adoption of more adaptive methods, for example, ML and AI, which can dynamically identify novel attack patterns by analyzing network data [56,57].

4.2. Anomaly-Based Detection

Anomaly-based detection represents a more adaptive approach to intrusion detection, designed to identify threats by recognizing deviations from established norms of behavior. This methodology relies on constructing a comprehensive profile of what constitutes "normal" activity within a system or network and then continuously monitoring for deviations that might indicate malicious actions. The process begins with a learning phase, where historical data are collected and analyzed to establish baselines of typical behavior. This could include monitoring network traffic patterns, analyzing user access logs, or assessing system performance metrics over an extended period. The resulting baselines are then stored and used as a reference point during subsequent operations [58]. During the monitoring phase, live data are continuously compared to the established baselines. The system scrutinizes real-time activity to identify anomalies, which are deviations that exceed predefined thresholds of normalcy. For instance, an unusual spike in network traffic during off-peak hours could indicate the early stages of a distributed denial-of-service (DDoS) attack. Similarly, an unexpected access to sensitive files by a user account might signal unauthorized activity [59]. To support this phase, organizations often employ advanced log aggregation and visualization tools such as Splunk or the ELK Stack. These tools enable security teams to collect, analyze, and interpret the vast quantities of data generated by modern systems. When an anomaly is detected, the system generates alerts and may initiate automated responses depending on the severity of the detected activity. To improve detection accuracy, many anomaly-based systems incorporate feedback loops, where flagged anomalies are reviewed and classified by security personnel. The system uses this feedback to refine its baselines, reducing false positives over time and adapting to evolving patterns of legitimate behavior [60]. Anomaly-based detection is especially effective for identifying unknown threats, including zero-day attacks and APTs. However, its reliance on accurate baselines poses challenges in environments with high variability, where distinguishing between legitimate deviations and malicious anomalies is particularly difficult. The computational requirements for real-time anomaly detection are also significant, making this approach more suited to high-performance networks or systems with robust infrastructure.

4.3. Behavior-Based Detection

Behavior-based detection delves deeper into the nature and intent of observed activities, focusing on whether the sequence of actions performed by users, applications, or devices aligns with known malicious objectives. This approach is particularly valuable in identifying complex, multi-stage attacks, insider threats, and sophisticated malware. The process begins with behavior profiling, where the system observes and records actions over time to construct detailed models of typical and atypical behavior. Unlike anomaly detection, which focuses on deviations from statistical norms, behavior-based detection examines the specific sequence and context of actions. For example, it might monitor file access patterns, system call sequences, or privilege escalation attempts to identify potentially malicious behavior [61]. This initial profiling is supported by the use of event logging tools and activity monitors, which capture granular details about user and system actions. The analysis phase follows, where observed behaviors are compared against pre-established models of benign and malicious activities. The system identifies behaviors that suggest malicious intent, such as attempts to bypass authentication mechanisms, access sensitive resources without proper authorization, or establish unauthorized communication channels [62]. A key feature of behavior-based detection is the correlation of actions across multiple layers of a system or network. For example, the system might link unusual login attempts with subsequent file transfers to determine whether they are part of an orchestrated attack. The ability to analyze behaviors in a broader context enables the detection of

stealthy, coordinated threats that might otherwise go unnoticed [63]. Once a potential threat is identified, the system generates alerts and may take immediate action, such as revoking access privileges, blocking specific IP ranges, or isolating affected endpoints. This response phase is critical for containing the spread of sophisticated attacks and minimizing their impact. While behavior-based detection is highly effective at identifying advanced threats, it requires significant computational resources and expertise to deploy and maintain. Its success depends on the quality of its behavior models and its ability to minimize false positives in complex environments. This method is most valuable in high-security contexts, such as financial institutions, healthcare systems, and government networks [64].

4.4. Technique Comparison

The integration and operational workflows of the signature-based, anomaly-based, and behavior-based detection methods highlight their strengths, weaknesses, and suitability for different environments and threat types.

- Detection Against Known Threats. Signature-based systems provide the highest precision for documented threats, relying on well-defined matching processes to ensure low false-positive rates. Their operational simplicity and reliance on static signature databases make them efficient but inherently limited to known attacks. In contrast, anomaly-based and behavior-based methods excel at detecting novel threats, such as zero-day attacks and polymorphic malware, by identifying deviations from baselines or malicious intent.
- Processing and Computational Requirements. Signature-based systems are computationally lightweight, as their operations primarily involve direct pattern matching. This makes them well suited for high-throughput environments, such as enterprise gateways or cloud platforms. In contrast, anomaly-based and behavior-based systems impose higher computational overhead due to the need for continuous monitoring, complex correlation, and iterative baseline refinement. These methods often require dedicated infrastructure or high-performance computing resources to operate effectively.
- False Positives and Response Automation. Anomaly-based and behavior-based detection methods are prone to higher false-positive rates, stemming from difficulties in accurately modeling normal behavior or intent. This necessitates robust alert validation workflows and automated response systems to mitigate the administrative burden. Signature-based methods, with their deterministic nature, produce far fewer false positives but lack adaptability to evolving threats.
- Integration Workflows. The integration of these techniques into a comprehensive intrusion detection framework often involves multi-layered workflows. Signature-based systems are commonly deployed at network perimeters to handle high-traffic volumes efficiently. Anomaly-based methods are integrated into deeper network layers or host systems to provide contextual awareness and detect unknown threats. Behavior-based systems complement both by focusing on specific patterns of malicious intent, often at the application or user level.
- Context-Specific Suitability:
  - Signature-based detection is ideal for environments requiring high-speed processing of predictable threats, such as enterprise networks.
  - Anomaly-based detection is particularly valuable in dynamic settings, such as cloud or IoT infrastructures, where attack surfaces evolve rapidly.
  - Behavior-based detection finds its strength in safeguarding critical systems against advanced threats, including insider attacks and stealthy malware, by focusing on intent and complex behavior correlations.

Through complementary integration of these methods, organizations can build robust intrusion detection frameworks that balance detection accuracy, computational efficiency, and adaptability to evolving threats. Table 1 reports the summarized information about techniques and methodologies comparison.

Table 1. Comparison of different IDS types and their typical applications, advantages, and disadvantages.

Type of IDS	Main Techniques	Typical Applications	Advantages	Disadvantages
NIDS [43]	DPI, flow analysis	Enterprise networks, distributed infrastructures	Monitors multiple devices simultaneously, suitable for large networks	Performance degrades with high traffic, challenges with encrypted data
HIDS [65]	System log monitoring, process behavior	Endpoints, servers, virtual machines	Detects host-specific threats, effective on encrypted endpoints	Difficult scalability in large environments, limited in detecting network-wide attacks
Hybrid IDS [66]	Combination of NIDS and HIDS	Holistic security in complex infrastructures	Enhances detection accuracy, reduces false positives	High complexity, requires significant computational resources
Signature-based [48]	Comparison with threat signature databases	Traditional networks, known malware protection	High precision for documented threats, low false positives	Ineffective against new or evolving threats, requires frequent updates
Anomaly-based [52]	Detection of deviations from normal behavior baselines	Dynamic environments like cloud, IoT	Capable of detecting unknown threats, suitable for zero-day threats	High false positives, requires robust infrastructure
Specification-based [55]	Detection based on predefined specifications of expected behavior	Industrial systems, IoT	Higher precision in controlled environments, effective in ICS or IoT	Manual effort to define specifications, limited adaptability
Behavior-based [64]	Behavior profiling, action correlation	Critical systems, financial institutions, healthcare	Detects complex attacks, insider threats	Requires significant computational resources, challenging to maintain

5. Literature Analysis on IDS Implementation in Modern Networks

Developing different IDSs for various types of networks is crucial due to the unique characteristics and security needs of each environment. In this section, we will introduce the most interesting papers addressing the development of IDSs in different network types. In particular, we will see cloud networks, virtualized networks, IoT networks, sensor networks, software-defined networks, and industrial networks. Some of the benefits of applying different IDS for different network types are as follows:

- Network Complexity: Different networks, such as enterprise, cloud, and IoT networks, have distinct architectures and traffic patterns. For instance, an NIDS is designed to monitor traffic across an entire network, making it suitable for environments with

multiple devices and varying data flows. In contrast, an HIDS focuses on individual devices, providing detailed monitoring of specific endpoints.

- Specific Threats: Each type of network faces different threats. For example, IoT networks may be more susceptible to device-specific attacks, while traditional enterprise networks might encounter more sophisticated external threats. Tailored IDSs can address these specific vulnerabilities effectively by employing different detection methodologies suited to the network type
- Specialized Monitoring: Certain IDS types specialize in monitoring specific protocols or applications (e.g., protocol-based IDSs). This specialization allows for better detection of anomalies related to particular protocols, enhancing the overall security posture of the network. For instance, a protocol IDS can detect unusual behavior in HTTP traffic that a general NIDS might overlook.
- Improved Response Strategies: Different networks require distinct incident response strategies. An HIDS can provide real-time alerts based on local activity, which is vital for quick responses to potential breaches at the device level. Meanwhile, NIDSs can aggregate data from multiple sources to identify broader attack patterns across the network.
- Efficient Resource Utilization: By implementing different types of IDSs tailored to specific networks, organizations can optimize their resource allocation. For example, deploying NIDSs in high-traffic areas can help manage bandwidth effectively while using HIDSs on critical servers ensures focused monitoring without overwhelming resources.
- Regulatory Compliance: Different industries may have varying compliance requirements regarding data protection and intrusion detection. Customized IDS solutions can help organizations meet these regulatory standards by providing the necessary logging and reporting capabilities specific to their operational context.

5.1. IDSs for Cloud and Virtualized Networks

In [67], the authors present a novel framework aimed at enhancing the security of smart city networks, particularly focusing on their cloud-based traffic. The authors propose a Cyborg Intelligence model that integrates ML techniques with biological intelligence to improve the detection and classification of cyber threats. They aim to develop a more precise and effective IDS tailored for smart city environments while reducing computational and time complexities associated with traditional intrusion detection methods. The introduced key contributions include the following:

- Cyborg Intelligence Framework: The framework combines various advanced methodologies to address the limitations of existing IDSs in smart cities, such as high computational costs and time complexity.
- Data Preprocessing: The paper introduces the Quantized Identical Data Imputation (QIDI) mechanism for effective data preprocessing and normalization, which enhances the quality of input data by filtering out irrelevant attributes.
- Feature Optimization: The Conjugate Self-Organizing Migration (CSOM) algorithm is employed to optimize feature selection, significantly improving the classifier's training process and detection accuracy.
- Intrusion Classification: The Reconciliate Multi-Agent Markov Learning (RMML) classification algorithm is utilized to categorize detected intrusions accurately, ensuring robust identification of various attack types.
- Performance Improvement: The proposed system aims to increase attack detection performance and efficiency through its unique combination of methodologies, which collectively enhance the overall security posture of smart city networks.

This article contributes significantly to the field of cybersecurity in smart cities by proposing an innovative IDS framework that leverages Cyborg Intelligence. This approach not only enhances detection capabilities but also addresses critical challenges faced by existing systems in managing the increasing complexity and volume of network traffic in urban environments. In [68], the authors present a novel framework for enhancing IDSs in cloud environments through the integration of deep learning and ML techniques. The primary goal of the research is to develop an effective IDS capable of detecting various types of cyberattacks, including user-to-root (U2R), root-to-local (R2L), probes, backdoors, worms, and denial-of-service (DoS) attacks. The proposed hybrid approach combines deep neural networks (DNNs) with ML algorithms, specifically utilizing a Group Artificial Bee Colony (G-ABC) optimization technique to improve the detection accuracy and efficiency of the IDS. To achieve this the authors employ two widely recognized datasets, NSL-KDD and UNSW-NB15, to train and evaluate the proposed model; they evaluate the effectiveness of the IDS by assessing standard performance metrics such as precision, recall, accuracy, and F-measure. To provide a comparative analysis, the results of the proposed system are compared against existing IDS solutions. The experimental results demonstrate that the proposed IDS significantly outperforms traditional methods in terms of accuracy and detection speed. By leveraging deep learning techniques alongside optimized ML algorithms, the system achieves higher precision in identifying malicious activities within cloud environments. This paper proves that integrating deep learning with advanced optimization strategies like G-ABC provides a robust solution for enhancing cybersecurity in cloud computing. This approach not only improves the detection rates of various attack types but also contributes to the overall resilience of cloud-based services against intrusions. Moreover, this research underscores the importance of evolving IDS frameworks to keep pace with the increasing complexity of cyber threats in cloud environments, advocating for further exploration into hybrid models that combine multiple ML techniques for enhanced security outcomes. In [69], the authors introduce a novel IDS that is specifically designed to address the unique challenges posed by the dynamic nature of cloud infrastructure and the evolving landscape of cyber threats. One of the standout features of the proposed IDS is its adaptive architecture. Unlike traditional systems that rely on static rules and signatures, this IDS employs DRL techniques that enable it to learn from its environment continuously allowing it to face complex and evolving cyber threats. This adaptability allows the system to recognize and respond to new attack patterns as they emerge, thus improving its overall effectiveness in detecting a wide range of intrusions. The authors report that their system achieves a classification accuracy exceeding 90% when tested on the NSL-KDD dataset, a commonly used benchmark for intrusion detection research. This level of accuracy is complemented by a significant reduction in FPRs, which is crucial for maintaining trust in automated security systems. High accuracy combined with low FPR means that the system can effectively distinguish between legitimate traffic and potential threats, minimizing unnecessary alerts and resource expenditure. Moreover, by incorporating mechanisms for hyperparameter tuning within the DRL framework, the system can optimize its performance over time. This means that as it interacts with the network environment, it refines its detection strategies based on real-time feedback, enhancing its resilience against sophisticated attacks. Finally, the authors also acknowledge areas for future research, suggesting that further exploration into novel reward mechanisms could enhance the system's learning process. Additionally, they emphasize the importance of improving the model's ability to generalize across different types of attacks and datasets, which would bolster its applicability in diverse real-world scenarios. This solution provides a significant advancement in intrusion detection for cloud infrastructures. By harnessing the power of DRL, their proposed IDS not only adapts to emerging threats but also demon-

strates high performance in terms of accuracy and efficiency. This makes it a promising solution for organizations seeking to fortify their cloud security against an increasingly complex array of cyber threats. In [70], the authors presented an autoencoder-based IDS designed for cloud and mobile environments. This system monitors network flows without analyzing packet content, allowing it to work with encrypted traffic and protect user privacy. It leverages the computational power of mobile devices for data preprocessing, while more complex operations are handled in the cloud. The intrusion detection relies on an autoencoder neural network, which detects anomalies by training only on benign samples. The system can share statistics between different detection nodes to enhance performance. Experiments show that using time-window-based features significantly improves intrusion detection. In summary, the proposed system is effective in detecting intrusions in cloud and mobile environments, though not all attack types can be detected solely with network flow data. Future research could focus on developing a more generic framework and automating preprocessing decisions. The authors of [71] present a novel approach for detecting malicious attacks in cloud computing environments using a hybrid model that combines the Gannet Optimization Algorithm (GOA) with Support Vector Machine (SVM) and Extreme Learning Machine (ELM). This hybrid model, referred to as GOA-optimized hybrid SVM-ELM, aims to enhance the security and reliability of cloud computing by accurately identifying and preventing various types of attacks. The integration of SVM and ELM leverages the strengths of both techniques, improving classification performance for detecting attacks. GOA is used to optimize the parameters of the hybrid SVM-ELM model, enhancing its accuracy and efficiency. Additionally, GOA helps in selecting the most relevant features from the input network traffic data, reducing noise and improving detection accuracy. The proposed method was tested using two datasets: the Evidence Detection in Cloud Forensics dataset and the CICIDS2017 dataset. The results demonstrated high precision, recall, and F-measure values, indicating the effectiveness of the GOA-optimized hybrid SVM-ELM model in detecting malicious attacks. Specifically, the method achieved a precision of 95%, recall of 96%, and F-measure of 98% on the Evidence Detection in Cloud Forensics dataset, and a precision of 98%, recall of 97%, and F-measure of 98% on the CICIDS2017 dataset. In summary, the GOA-optimized hybrid SVM-ELM model shows significant promise in enhancing the security of cloud computing environments by effectively detecting and preventing various types of attacks. The model's high performance metrics and reduced training time make it a viable solution for real-world applications. In [72], a hybrid intrusion detection model (HIDM) designed for cloud-based systems is shown, combining signature-based and anomaly-based detection methods to identify both known and unknown attacks. The model aims to enhance the security of cloud computing environments, which are increasingly popular due to their cost-effectiveness and efficiency but are also vulnerable to various security threats. The proposed HIDM uses a combination of ML techniques, including random forest, neural networks, and Gradient Boosting, to create a baseline for normal and intrusive activities. The model was tested on three datasets: UNSW-NB15, CICIDS2017, and NSL-KDD. The results showed high detection rates of 92.7% for UNSW-NB15, 85.1% for CICIDS2017, and 99.8% for NSL-KDD, indicating the model's effectiveness in identifying both known and unknown attacks. The HIDM operates in several phases, including data collection, preprocessing, feature selection, and classification. The model first collects and preprocesses data, eliminating duplicate records and imputing missing values. It then selects the best features using various techniques and creates models using different ML algorithms. The hybrid model combines the results of these algorithms to classify data and generate alerts for any detected intrusions. In summary, the HIDM provides a robust solution for enhancing the security of cloud-based systems by effectively detecting and preventing various types of attacks. The model's high

performance metrics and ability to handle both known and unknown threats make it a valuable tool for protecting cloud environments.

5.2. IDSs in IoT and Sensor Networks

In [73], the authors present a novel approach to enhancing cybersecurity in IoT and cloud environments through the integration of blockchain technology and advanced IDSs. The key contributions include the following:

- Deep Blockchain Framework (DBF): The authors propose a DBF designed to provide distributed intrusion detection while ensuring data privacy through blockchain and smart contracts. This framework aims to secure data migration between cloud services and protect IoT networks from cyberattacks.
- Intrusion Detection Method: The paper employs a bidirectional long short-term memory (BiLSTM) deep learning algorithm to analyze sequential network data. This approach is particularly suited for detecting complex cyber threats in real time, leveraging datasets such as UNSW-NB15 and BoT-IoT for evaluation.
- Privacy Preservation: The integration of blockchain technology facilitates immutable data exchange and enhances privacy during the migration of virtual machines (VMs) across cloud providers. Smart contracts are utilized to ensure that data handling complies with privacy standards.
- Performance Evaluation: The proposed DBF framework is compared against existing privacy-preserving intrusion detection techniques, demonstrating superior performance in terms of both detection accuracy and data security.
- System Architecture: The framework consists of four main components: cloud vendor, privacy-preservation-based blockchain with smart contracts, central coordinator unit (CCU), and collaborative intrusion detection system (CIDS).

In conclusion, this paper highlights that integrating blockchain with collaborative intrusion detection can significantly enhance the security posture of IoT and cloud networks. The DBF not only improves the identification of cyber threats but also addresses critical issues related to data privacy, making it a promising solution for modern cybersecurity challenges in distributed environments. As IoT devices proliferate, the complexity and volume of network traffic increase, making it challenging to detect cyberattacks effectively. Traditional deep learning models often struggle due to high dimensionality and imbalanced data, leading to inadequate detection capabilities for both known and novel threats. In [74], the authors present a novel approach to improving IDSs specifically for IoT networks. In particular, the authors propose a Hybrid Weighted Deep Belief Network (HW-DBN) algorithm that combines an improved Gaussian–Bernoulli Restricted Boltzmann Machine (Deep GB-RBM) for feature learning with a Weighted Deep Neural Network (WDNN) classifier. This hybrid model aims to enhance the detection accuracy of various cyberattack scenarios while being adaptive to the dynamic nature of IoT environments. The effectiveness of the DeepIoT.IDS model is evaluated using the CICIDS2017 dataset, which includes diverse attack types and complex data patterns. The results demonstrate that the proposed model significantly outperforms three recent models, achieving detection accuracies of 99.38% for web attacks and 99.99% for bot attacks. Notably, it also successfully identifies low-frequency attacks that other models fail to detect. The study underscores the importance of developing advanced IDS solutions tailored for IoT networks. The HW-DBN algorithm not only improves detection rates but also adapts to the unique challenges posed by IoT traffic, marking a significant advancement in cybersecurity for connected devices. In [75], the authors present a novel IDS specifically designed to enhance the security of IoT devices. Passban is developed to protect IoT devices that are directly connected to it, leveraging the edge computing paradigm. This allows Passban to detect cyber threats

close to their data sources, which is crucial given the resource constraints of many IoT devices, such as limited processing power and memory. The proposed system employs an anomaly-based detection approach, which is effective in identifying unusual patterns that may indicate security breaches. This method is particularly suitable for IoT environments where traditional signature-based systems may fail due to the dynamic nature of attacks. Deployment on IoT Gateways: Passban can be deployed on inexpensive IoT gateways, making it accessible for various applications without requiring substantial computational resources. This deployment strategy aligns with the increasing need for real-time threat detection in edge computing scenarios. Technical Approach Detection Techniques: The paper discusses the use of the isolation forest (iForest) ensemble technique for detecting various types of attacks, including port scanning, brute force attacks, and SYN flooding. These techniques enable the system to effectively monitor real-time network traffic and identify anomalies indicative of potential intrusions. Performance Evaluation: The authors conducted extensive evaluations to assess the effectiveness of Passban IDS. The results demonstrated its capability to accurately detect intrusions while maintaining low FPRs, which is critical for operational efficiency in IoT networks. Conclusion Passban IDS represents a significant advancement in securing IoT devices against cyber threats by utilizing intelligent anomaly detection methods tailored for edge environments. Its deployment on low-cost gateways facilitates widespread adoption, potentially enhancing the overall security posture of IoT networks amidst growing vulnerabilities and attack vectors. In [49], the authors present a novel IDS designed specifically for IoT environments. This system, referred to as AS-IDS, integrates both anomaly-based and signature-based detection methods to effectively identify both known and unknown attacks within IoT networks. AS-IDS features an hybrid detection approach that combines two detection strategies:

- Anomaly-based Detection: This approach monitors deviations from normal behavior patterns to detect potential threats that may not match known attack signatures.
- Signature-based Detection: This method relies on a database of known attack signatures to identify malicious activities.

The model was trained and tested using several datasets, including the CIC-IDS 2018, MQTT-IoT-IDS2020, and BoTNeTIoT-L01 datasets. This diverse data input enhances the robustness of the detection capabilities. The effectiveness of the proposed IDS is evaluated using various metrics such as detection rate, false alarm rate, specificity, and computation time. The results indicate a high detection accuracy, achieving up to 99.81% in certain tests. The AS-IDS model demonstrates significant potential for improving IoT security by effectively predicting attacks through its hybrid approach. The role of AI enhances IDS functionality through real-time monitoring, precise threat identification, and automated response mechanisms. The authors also suggest that future implementations could benefit from real-time datasets to further enhance the model's applicability in dynamic IoT environments.

5.3. IDSs in SDN

In [76], the authors introduce a novel approach to enhance the security of IoT networks through the integration of SDN and deep learning techniques. This work addresses the growing security challenges in IoT environments, which are increasingly vulnerable to various cyber threats due to their expansive attack surface. The authors propose a deep learning-based IDS that leverages SDN's flexibility to improve anomaly detection capabilities in IoT networks. The authors employ various deep learning architectures, including long short-term memory (LSTM) networks, to analyze network traffic and detect anomalies. A comparative analysis with traditional ML methods, such as Support Vector Machines (SVMs), is conducted to evaluate performance. The proposed system is tested on multiple

datasets tailored for SDN and IoT environments, ensuring robustness and generalizability across different attack scenarios. Techniques like t-SNE are used for visualizing learned features from the model's hidden layers, confirming that the features are meaningful for detecting and classifying attacks. The IDS demonstrates superior performance metrics, including accuracy, precision, recall, and F1-Score, compared to existing methodologies. The results indicate that deep learning models can effectively identify a wide range of attacks, including brute force attacks and DDoS attacks. The authors conclude that integrating SDN with deep learning not only enhances the detection capabilities of IDS in IoT networks but also provides a scalable and adaptable framework for future security implementations. Finally, the authors emphasize the need for continuous monitoring and updates to maintain the effectiveness of the proposed system against evolving threats. In [77], the authors presents a comprehensive exploration of how ML techniques can enhance NIDSs specifically within the framework of SDN. The study emphasizes the unique architecture of SDN, which separates the control plane from the data plane, allowing for centralized management and a global view of network traffic. This architecture facilitates more effective monitoring and detection of malicious activities compared to traditional networks, which often have limited visibility. The authors propose integrating various ML algorithms into NIDSs to leverage the advantages of SDN's centralized control. They detail how different ML models can be employed to analyze network traffic and identify intrusions effectively. The paper also reviews existing literature on ML-based intrusion detection, highlighting the evolution of techniques and their applicability in SDN environments. The key points of this work include the following:

- Feature Selection and Data Processing: The study discusses the importance of selecting relevant features from datasets, particularly using the NSL-KDD dataset as a benchmark. It mentions that out of 41 features, 12 were selected for optimal performance in detecting intrusions.
- Classifier Performance: A range of classifiers are evaluated, including CNN, deep neural networks (DNNs), RNNs, long short-term memory (LSTM), and Gated Recurrent Units (GRUs). The results indicate high accuracy rates, with the best performing classifiers achieving over 98% accuracy in detecting attacks.
- Challenges and Future Directions: The paper identifies ongoing challenges in implementing ML-based NIDSs, such as adapting to new attack vectors that emerge as network technologies evolve. It calls for further research into refining ML models to improve their adaptability and effectiveness in real-time scenarios.

The findings suggest that integrating ML with SDN provides a robust framework for developing effective IDSs. The proposed methodologies not only enhance detection capabilities but also pave the way for future innovations in network security within SDN architectures. The authors advocate for continued exploration of ML techniques to address emerging cybersecurity threats in increasingly complex networking environments. In [78], the authors introduce an innovative approach to enhancing network management and security within OpenStack environments by integrating SDN with advanced network functions. The authors argue that traditional network management methods often struggle to keep pace with the dynamic nature of cloud infrastructures, leading to vulnerabilities and inefficiencies. To address these challenges, the paper proposes a framework that leverages SDN's programmability and flexibility, allowing for more effective monitoring and control of network resources. This integration enables real-time adjustments to network configurations and policies, which can significantly enhance security measures against various threats, including DDoS attacks. The framework is designed to be scalable, accommodating the growth of cloud services while maintaining robust security protocols. Additionally, the authors emphasize the importance of deep learning techniques in identifying and mitigat-

ing potential security threats. By employing ML algorithms, the framework can analyze traffic patterns and detect anomalies that may indicate malicious activities. This proactive approach not only improves response times but also enhances overall network resilience. In conclusion, OpenStackDP represents a significant advancement in the field of cloud security by combining SDN with intelligent monitoring solutions. This framework aims to provide a comprehensive security strategy that adapts to the evolving landscape of cyber threats while ensuring efficient resource management in OpenStack cloud infrastructures. In [79], the authors presented a comprehensive approach to enhancing the security of SDN through a multi-layered IDS. This system leverages both rule-based and ML techniques to detect and mitigate various types of network attacks. The proposed IDS architecture includes multiple layers, each designed to address specific aspects of network security. The rule-based layer focuses on detecting known attack patterns by comparing network traffic against a database of predefined signatures. This layer is effective in identifying well-known threats but may struggle with new or unknown attacks. To address this limitation, the system incorporates an ML layer that uses advanced algorithms to analyze network traffic and identify anomalies that may indicate malicious activity. This layer is capable of detecting previously unseen attacks by learning from historical data and adapting to new threat patterns. The paper highlights the effectiveness of the multi-layered IDS through extensive testing on various datasets. The results demonstrate high detection rates and low FPRs, indicating the system's ability to accurately identify and respond to network intrusions. The combination of rule-based and ML techniques provides a robust defense mechanism, ensuring comprehensive protection for SDN environments. In summary, the multi-layered IDS for SDN offers a powerful solution for enhancing network security by combining the strengths of rule-based detection with the adaptability of ML. This approach ensures effective detection and mitigation of both known and unknown threats, making it a valuable tool for protecting modern network infrastructures. The authors in [80] present a novel approach to enhancing network security in SDN environments through the application of ML techniques. The authors, Alzahrani and Alenazi, emphasize the growing need for effective IDSs due to the unique vulnerabilities introduced by the SDN architecture, which separates the control and forwarding planes of network operations. To address these security challenges, the researchers developed two distinct datasets using Mininet and the Ryu controller, incorporating various feature extraction tools. These datasets were utilized to train several supervised binary classification algorithms, including AdaBoost, decision trees, random forests, naive Bayes, multi-layer perceptron, Support Vector Machines, and XGBoost. Among these, the decision tree algorithm demonstrated exceptional performance with an F1-Score of 0.9995 for attack classes and 0.9983 for normal classes, along with a throughput of over 6.7 million samples per second while using only three features. The study highlights the importance of data preprocessing in reducing model complexity and improving overall system throughput to meet real-time operational requirements. The results indicate that the proposed ML-based IDS can effectively classify various types of attacks, including UDP flood and TCP SYN flood, thereby significantly enhancing the security posture of SDN environments. This research contributes valuable insights into developing high-performance IDS solutions tailored for the evolving landscape of network security threats.

5.4. IDSs in Industrial Networking Systems

Industrial control networks are crucial for managing production processes, but they face significant threats from intrusions and cyberattacks. Traditional IDSs struggle with the unique characteristics of ICN traffic, which include high dimensionality, irregularity, and temporal correlations. In [81], the authors present a novel approach for enhancing

the security of industrial control networks (ICNs) against malicious intrusions. Their approach is based on a multi-feature data clustering optimization model designed to improve the accuracy and efficiency of intrusion detection in ICNs. The most interesting innovations include the following:

- Associative Recurrent Network (ARN): A new recurrent neural network model that effectively manages the relationship between past and current data states, addressing the limitations of existing models like Gated Recurrent Units (GRUs).
- Single Attention Mechanism (S-ATT): This mechanism enhances the ARN by allowing it to retain relevant past information without relying on traditional gated structures, thus improving the model's ability to learn from historical data.

The proposed algorithm was tested against two datasets: SWaT (a dataset specifically for water treatment systems) and UNSW-NB15 (a conventional network traffic dataset). The results demonstrated high detection accuracies of 95.48% and 97.61%, respectively, indicating that the model is effective in identifying both known and novel intrusion attempts. The authors conclude that the integration of multi-feature clustering with advanced neural network architectures significantly enhances intrusion detection capabilities in industrial networks. This approach not only addresses the challenges posed by high-dimensional data but also improves computational efficiency, making it a promising solution for real-time industrial applications. In [82], the authors address the critical issue of cybersecurity in the Industrial Internet of Things (IIoT). They highlight the increasing vulnerabilities faced by IIoT systems due to various cyberattacks, which can lead to significant financial and reputational damage for organizations. In particular, the authors propose a novel intrusion detection paradigm that utilizes a deep learning model combined with a hybrid rule-based feature selection method. This approach aims to enhance the accuracy and efficiency of detecting intrusions within IIoT networks. Moreover, it emphasizes the importance of feature selection in developing an effective NIDS. The proposed method employs a rule-based feature selection technique that helps in managing the extensive datasets typically associated with IIoT environments. The authors tested this solution using two well-known datasets, NSL-KDD and UNSW-NB15 (see Section 6 for more detail on these two datasets). The results demonstrated impressive performance metrics, achieving an accuracy of 99.0%, a detection rate of 99.0%, and an FPR of 1.0% for the NSL-KDD dataset, while for the UNSW-NB15 dataset, it achieved 98.9% accuracy, 99.9% detection rate, and 1.1% FPR. Various evaluation metrics were utilized to validate the effectiveness of the proposed intrusion detection method, confirming its suitability for real-world IIoT applications. Another work focusing on IIoT id presented in [83]. In this work, the authors present a novel approach to network intrusion detection specifically tailored for the Industrial Internet of Things (IIoT) using a one-dimensional convolutional neural network (1D-CNN). They propose a deep learning architecture that leverages 1D-CNN for effectively detecting various types of network intrusions, including DoS attacks and port scans. This model is particularly advantageous for handling the unique characteristics of IIoT data, which often involve sequential and time-series information. The authors provide a comprehensive analysis using the CICIDS2017 dataset, which includes diverse attack scenarios. The study emphasizes the importance of preprocessing and data analysis to enhance model performance. The proposed 1D-CNN model demonstrates significant improvements in detection rates compared to traditional methods. It is noted for its reduced computational complexity, making it more efficient than two-dimensional CNNs, especially in resource-constrained environments typical of IIoT applications. The paper includes a detailed comparison with existing IDSs, highlighting the advantages of using 1D-CNNs over other ML techniques. This includes aspects such as training efficiency and accuracy in classifying different types of intrusions. Finally, the authors suggest further research avenues, including optimizing

hyperparameters and integrating hybrid models that combine various detection techniques to improve overall system robustness against both known and unknown threats. In [84], the authors explore advancements in detecting cyber threats within ICS, particularly focusing on SCADA systems. First, they begin with a comprehensive review of recent anomaly detection techniques applicable to SCADA systems, emphasizing both theoretical ML approaches and practical frameworks. Then, they highlight the critical need for effective detection methods due to the increasing reliance on ICS in essential services and their vulnerability to various cyber threats, including natural disasters. A significant contribution of the paper is the introduction of a complete framework for an Intrusion and Anomaly Detection System (IADS). This framework comprises several components:

- Detection Probes: Specific tools designed to monitor system behavior.
- Event Processing Layer: A mechanism for managing and analyzing detected events.
- Core Anomaly Detection Component: Utilizes ML algorithms to identify deviations from normal operational patterns.

The authors conducted evaluations of the proposed framework within a large-scale hybrid testbed, comparing various anomaly detection scenarios using different ML techniques. This empirical analysis aims to establish the effectiveness of the proposed IADS in real-world settings, demonstrating its capability to enhance security measures against sophisticated cyber threats. Finally, the authors discuss existing challenges in the field, such as the limitations of current detection methodologies and the need for more robust frameworks that can adapt to evolving threats. In [85], the authors present a novel approach, DeepFed, to enhance cybersecurity in industrial environments through federated deep learning. This approach proposes a federated learning framework specifically designed for detecting cyber threats in Industrial Cyber–Physical Systems (CPS). This method addresses the challenges of traditional IDSs, which often rely on centralized data processing, potentially compromising data privacy and security. The key features of DeepFeb include the following:

- Federated Learning Approach: Instead of sending sensitive data to a central server, DeepFed allows local devices to train models on their own data and share only the model updates. This significantly reduces the risk of exposing sensitive information while still improving the overall model performance through collaborative learning from multiple devices.
- Model Architecture: The framework employs deep learning techniques, utilizing CNN and RNN to enhance detection capabilities. These models are trained locally on devices that collect real-time data from industrial environments, ensuring that the system remains responsive and effective against various cyber threats.
- Security Protocol: To protect the integrity and privacy of model parameters during training, DeepFed incorporates a Paillier cryptosystem-based secure communication protocol. This cryptographic approach ensures that even if model updates are intercepted, they cannot be exploited by malicious actors.

The authors conducted extensive experiments using the Edge-IIoTset dataset, which includes real-world attack scenarios. The results demonstrated that DeepFed achieves comparable accuracy to centralized ML models while maintaining enhanced privacy and requiring less bandwidth for data transmission. As the number of federated learning rounds increased, the model's performance improved significantly, showcasing its effectiveness in anomaly detection within CPS environments. DeepFed represents a significant advancement in intrusion detection methodologies for industrial settings by leveraging federated learning principles. This approach not only enhances security but also addresses critical concerns related to data privacy and bandwidth efficiency, making it a promising

solution for future applications in industrial cybersecurity. Industrial control networks are critical for managing real-time operations in various sectors, but they face significant security threats such as malicious intrusions and cyberattacks. In [86], the authors introduce a novel IDS designed specifically for the challenges posed by industrial control networks (ICNs) operating within fog computing environments. To this aim, they address the unique characteristics of ICN traffic, which includes high dimensionality, irregular patterns, and temporal correlations, making traditional IDS approaches less effective. The two main key points of their work are as follows:

- Associative Recurrent Network (ARN): The core innovation of this paper is the introduction of the ARN model. This recurrent neural network is tailored to effectively capture both long-term and short-term dependencies in network traffic data, overcoming limitations found in traditional Gated Recurrent Units (GRUs).
- Single Attention Mechanism (S-ATT): To enhance the model's performance, the authors implement an attention mechanism that allows the ARN to retain important past information without relying on complex gated structures. This mechanism directly learns the relationships between past hidden states and current inputs, improving the model's ability to detect intrusions.

The proposed method was evaluated using two datasets: SWaT (a benchmark for water treatment systems) and UNSW-NB15 (a general network traffic dataset). Results demonstrated high accuracy rates of 95.48% and 97.61%, respectively, indicating that Deep-IFS outperforms existing IDS solutions in detecting both known and unknown attacks. This approach significantly advances intrusion detection capabilities within IIoT environments by leveraging deep learning techniques that are sensitive to temporal dynamics in network traffic. Table 2 reports the summarized pros and cons about typical IDS applications.

Table 2. Comparison of various IDS types and their typical uses, benefits, and drawbacks.

Application Type	IDS Type	Pros	Cons
Cloud and Virtualized Networks [67]	Cyborg Intelligence Framework (QIDI, CSOM, RMML)	High precision and reduced computational complexity; optimized feature selection and accurate classification	Requires integration of multiple advanced methodologies, increasing implementation complexity
Cloud [87]	Deep Learning + G-ABC	Better detection compared to traditional methods; high accuracy and detection speed	Depends on specific datasets (NSL-KDD, UNSW-NB15); complexity of hybrid models
Cloud [88]	Adaptive IDS with DRL	Adaptive architecture; high accuracy (>90%) and low FPR	Continuous tuning required and challenges in generalizing across different attack types
IoT and Sensor Networks [89]	Deep Blockchain Framework (DBF)	Improved data privacy and security; distributed detection through smart contracts	Complexity of blockchain integration; potential latency increase
IoT [74]	Hybrid Weighted Deep Belief Network (HW-DBN)	High detection accuracy; adaptability to dynamic IoT data	Risk of computational overload with high-dimensional data
IoT [75]	Passban (Anomaly-based on Edge Computing)	Suitable for resource-constrained IoT devices; low FPRs	Limited to attacks showing clear anomaly patterns
IoT [49]	AS-IDS (Anomaly + Signature-based)	Combines benefits of anomaly and signature-based detection; high accuracy (99.81%)	Requires constant signature updates; dependent on dataset quality

Table 2. *Cont.*

Application Type	IDS Type	Pros	Cons
SDN [90]	Deep Learning-based IDS	High flexibility and capability to adapt to various attack scenarios	Implementation complexity and need for continuous strategy updates
Industrial Networks [91]	Multi-feature Clustering with ARN and S-ATT	High accuracy (95.48% and 97.61%) on specific datasets; real-time detection optimization	Complexity in managing high-dimensional data and temporal correlations
IIoT [83]	1D-CNN for Industrial Networks	Reduced computational complexity; suitable for sequential and time-series scenarios	Potential limitation to specific attack patterns; less effective in non-sequential scenarios
IIoT [85]	DeepFed with Federated Learning	Improved data privacy; reduced bandwidth needs due to distributed model updates	Complexity in implementing security protocols; challenges in device synchronization

5.5. Open Challenges and Future Research Directions in Modern IDSs

The field of IDSs faces numerous open challenges and opportunities for future research. One of the foremost challenges is the rapid evolution of cyber threats. Attackers continuously develop sophisticated techniques, such as polymorphism, metamorphism, and adversarial attacks powered by AI, to bypass detection mechanisms. IDSs must adapt to these evolving threats by employing advanced detection methods capable of identifying novel and obfuscated attack patterns. This need for adaptability highlights a gap in current IDS solutions, which often struggle to maintain effectiveness against unknown or dynamically changing threats. Another persistent issue is the high rate of false positives, which leads to alert fatigue among system administrators and undermines trust in the system. Balancing precision and sensitivity remains a key technical challenge, particularly for anomaly-based IDSs, where the detection of rare events often comes at the cost of increased false alarms. Enhancing the reliability of these systems without compromising their ability to detect subtle malicious activities requires innovative approaches to algorithm design and evaluation. Modern computing environments, particularly cloud-based and distributed systems, introduce additional complexities. These environments require IDSs to operate across heterogeneous and decentralized infrastructures while maintaining scalability and real-time performance. Cloud environments, in particular, demand solutions that can handle elastic workloads, monitor virtualized resources, and manage multi-tenancy without degrading detection accuracy. Furthermore, encrypted communication protocols, such as TLS 1.3, have become increasingly prevalent, making it challenging for IDSs to analyze traffic content. Traditional DPI is no longer feasible in many cases, pushing researchers to explore alternative methods, such as metadata analysis and encrypted traffic pattern recognition. The integration of AI and ML has significantly enhanced the capabilities of IDSs. However, these advancements come with their own challenges, particularly in terms of interpretability. XAI is becoming a critical research area to ensure that decision-making processes in IDSs are transparent and comprehensible. Without clear explanations of why certain events are flagged as malicious, it becomes difficult for operators to trust and validate these systems, especially in mission-critical environments. Energy efficiency is another growing concern, particularly for IDSs deployed in resource-constrained environments, such as IoT devices and embedded systems. The design of lightweight and energy-efficient models is essential to ensure the practicality of IDSs in these scenarios. Similarly, robustness against adversarial attacks is an emerging area of focus. ML-based IDSs are vulnerable to crafted inputs designed to exploit detection weaknesses, necessitating the development of more resilient algorithms capable of defending against these targeted attempts. Future

research directions in IDSs are exploring advanced applications of ML and deep learning. Techniques such as CNNs, RNNs, and transformer-based architectures hold promise for detecting complex attack patterns within large datasets. Federated learning is another promising avenue, enabling collaborative model training across organizations without compromising data privacy. This approach can improve IDS capabilities while addressing concerns about sharing sensitive information. In decentralized architectures, the adoption of edge computing presents opportunities to enhance the real-time capabilities of IDSs. By processing data closer to the source, edge-based IDSs can reduce latency and alleviate the burden on central systems. Blockchain technology also shows potential for improving the security and transparency of IDSs, particularly in managing logs and correlating security events. The immutability of blockchain can ensure the integrity of recorded data, fostering trust in distributed IDS implementations. Finally, the evaluation of IDSs must evolve to address emerging concerns, such as sustainability and ethical AI principles. Incorporating metrics that assess energy consumption, carbon footprint, and compliance with ethical standards is crucial for the development of IDSs that align with broader organizational and societal goals. Advanced evaluation frameworks should integrate these dimensions alongside traditional metrics like detection accuracy and scalability to provide a holistic view of an IDS's performance. In conclusion, modern IDSs must navigate a complex and rapidly changing landscape characterized by evolving threats, diverse deployment environments, and increasing expectations for transparency, efficiency, and sustainability. By addressing these challenges and leveraging emerging technologies, future research can pave the way for IDS that are not only more effective but also more robust, interpretable, and adaptable to the demands of the modern digital ecosystem.

6. Datasets and Testing Environments

6.1. Standard Datasets for IDS Evaluation

The availability of high-quality datasets is critical for developing and evaluating IDSs. For this reasons, numerous datasets are used to detect and analyze network-based attacks. These datasets serve as benchmarks for evaluating IDSs and are crucial for developing ML models. Below is a detailed explanation of some of the most prominent network traffic-based datasets commonly employed in IDS research: KDD Cup 99 Dataset: The KDD Cup 99 dataset is an evolution of the DARPA dataset, adapted for a ML competition in 1999. It contains over 5 million records of network traffic, with attacks simulated across categories like DoS, R2L, U2R, and Probing. Despite being widely used for benchmarking IDS models, the KDD Cup 99 dataset has several issues, including a significant number of redundant records and outdated attack scenarios, which led to its reduced applicability in modern contexts. These limitations prompted the development of more refined datasets in later years [92–96]. NSL-KDD 99 Dataset: The NSL-KDD dataset was designed to overcome the limitations of the KDD Cup 99 dataset, particularly by addressing its redundancy and imbalance issues. NSL-KDD contains approximately 150,000 records, significantly fewer than the original KDD dataset, as duplicate and unnecessary entries were removed. Like KDD Cup 99, NSL-KDD includes simulations of DoS, R2L, U2R, and Probing attacks, but its smaller and more balanced size makes it more suitable for evaluating IDS models, especially in academic research [97–103]. Kyoto 2006-2009 Dataset: The Kyoto dataset was created to enhance the feature set of the KDD Cup 99 dataset. It includes 24 features, 14 of which were directly borrowed from KDD Cup 99, while 10 new features were added specifically to improve intrusion detection accuracy. The dataset was developed using honeypot systems installed at Kyoto University, and it offers a more modern and comprehensive view of network traffic. This makes it highly relevant for research into both traditional and contemporary cyberattacks [104–107]. ISCX-2012 Dataset: The ISCX-2012 dataset was developed

from seven days of network traffic and contains both normal and malicious traffic. Attack types include internal network infiltration, HTTP DoS, DDoS, and SSH brute force attacks. The dataset categorizes traffic into two main classes: normal and attacker. Its structured format and clear classification of different types of malicious traffic make it widely used in intrusion detection research [108–110]. UNSW-NB15 2015 Dataset: The UNSW-NB15 dataset was developed by the Australian Centre for Cyber Security (ACCS) to address the limitations of older datasets like KDD Cup 99 and NSL-KDD. It contains 49 features extracted using a range of modern tools, such as IXIA PerfectStorm (a traffic generator), Tcpdump, and Argus, simulating nine attack types: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. By simulating these contemporary attack types using the latest vulnerability data from the CVE website, UNSW-NB15 offers a more relevant and realistic environment for testing IDS systems, especially when compared to older datasets [111–116]. CSE-CIC-IDS 2017-2018 Datasets: The CSE-CIC-IDS datasets, developed in collaboration between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), represent some of the most comprehensive and up-to-date datasets for network intrusion detection with simulate real-world attack scenarios in a laboratory environment with networks of both attackers and victims. In the 2017 version, a network of three victim machines running Windows OS and one attacker running Kali Linux was set up. The dataset includes 14 different attack types, such as DoS, DDoS, Brute Force, Heartbleed, Botnet, Web Attack, and Infiltration, and 80 features were extracted using the CICFlowMeter tool [117–119]. The CSE-CIC-IDS 2018 dataset expanded upon the 2017 version by scaling up the test environment to better represent modern, large-scale networks. It featured 50 attacker machines and a victim network comprising 420 machines and 30 servers, divided into departments mimicking real-world enterprise setups. The dataset covers the same attack types as the 2017 version but with greater diversity in the network infrastructure, including multiple versions of Windows and Ubuntu operating systems. This dataset has become a standard in evaluating IDSs due to its scale, diversity of attack types, and realistic simulation of enterprise network environments [120–124]. TON_IoT Dataset: The TON_IoT (Trustworthy Operational Network Internet of Things) dataset, developed by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS), addresses the challenges of analyzing IoT and IIoT (IIoT) traffic. It includes data from three main sources: IoT sensors (e.g., temperature, pressure, humidity), network traffic generated by IoT devices and applications, and system logs from Linux and Windows machines. The dataset incorporates attacks such as DDoS, ransomware, backdoor, and injection, with 44 features representing network traffic and system logs. TON_IoT is ideal for testing IDS solutions targeting modern infrastructures [125–129]. CIC-DDoS2019 Dataset: The CIC-DDoS2019 dataset, developed by the Canadian Institute for Cybersecurity (CIC), focuses on modern DDoS attack scenarios. It includes traffic generated using updated tools and features attacks such as UDP, HTTP Flood, SYN flood, DNS Amplification, and TFTP Amplification. The dataset contains both normal and malicious traffic, with 88 features extracted using CICFlowMeter. This detailed and structured dataset is highly suitable for developing IDS models to mitigate modern DDoS threats [130–136]. All of this information regarding the various datasets is summarized in the following Table 3, which also includes the different attack classes represented in each dataset. This provides a structured view of the datasets, highlighting their key features, the number of attributes, and the specific attack types they simulate, offering a comprehensive comparison useful for intrusion detection research.

Table 3. Summary of datasets with their features and attack types classes.

Dataset	Year	Number of Features	Attack Type Classes
KDD CUP 99 [137]	1998–1999	43	DoS, Remote-to-Local (R2L), User-to-Root (U2R), Probing
NSL-KDD 99 [99,138]	1999	43	Normal, DoS, Remote-to-Local (R2L), User-to-Root (U2R), Probing
Kyoto 2006–2009 [104]	2006–2009	23	Oth, rej, rsto, rstos0, rstr, rstrh, s0, s1, s2, s3, sf, sh, shr
SCX-2012 [110,139]	2012	80	Normal, Attacker
UNSW-NB15 2015 [111,140–144]	2015	49	Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms
CSE-CIC-IDS 2017–2018 [121,145]	2017–2018	80	DoS Golden Eye, Bening, DoS Hulk, DoS Slow http, DoS Slowloris, DDoS-LOIC HTTP, DDoS-LOIC-UDP, DDoS-HOIC, SSH-Patator, FTP Patator, Brute Force, XSS, Botnet, Infiltration, SQL Injection
TON_IoT [125,146–152]	2020	44	DDoS, Ransomware, Backdoor, Injection
CIC-DDoS2019 [153,154]	2019	88	UDP Flood, HTTP Flood, SYN Flood, DNS Amplification, TFTP Amplification

6.2. Testing Procedures for IDSs

Evaluating IDSs effectively necessitates the implementation of meticulously structured testing procedures designed to assess their performance across diverse attack scenarios. This process ensures that IDS solutions can accurately detect and mitigate potential threats while maintaining their functionality in real-world environments. To this end, the design and validation of IDS models incorporate specific methodologies to guarantee comprehensive testing and practical applicability. Below, we elaborate on the fundamental steps and best practices involved in testing IDSs: Defining Objectives and Metrics: The initial phase of testing involves establishing well-defined objectives that are aligned with the specific goals of the IDS and the desired performance outcomes. These objectives guide the selection of performance metrics, which are essential for evaluating the effectiveness, efficiency, and reliability of the IDS under test. Commonly employed metrics include the following:

- Detection Accuracy (A): Measures the overall correctness of the IDS in classifying events as either benign or malicious. Formally, it is defined as

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

  where TP (true positive) and TN (true negative) represent correct classifications, while FP (false positive) and FN (false negative) represent incorrect classifications. While accuracy provides a high-level summary, it may be misleading in the case of imbalanced datasets.
- FPR: Indicates the proportion of benign traffic incorrectly flagged as malicious, calculated as

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

A low FPR is critical for reducing unnecessary alerts, which can overwhelm security analysts and degrade system usability.

- TPR or Sensitivity: Represents the ability of the IDS to correctly identify malicious traffic, expressed as
$$TPR = \frac{TP}{TP + FN}$$
High sensitivity ensures that the IDS effectively detects genuine threats, minimizing the risk of undetected attacks.
- Precision (P): Assesses the proportion of positively identified events that are indeed malicious, defined as
$$P = \frac{TP}{TP + FP}$$
Precision is particularly important in high-alert environments where false alarms need to be minimized.
- Recall (R): Equivalent to TPR, recall emphasizes the IDS's capability to detect all malicious events in the dataset. Precision and recall are often evaluated together to balance detection performance.
- F1-Score (F_1): Combines precision and recall into a single metric to provide a harmonic mean, useful in scenarios where both false positives and false negatives are equally critical:
$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}$$
- Processing Speed (T_p): For real-time IDS applications, the system's ability to process incoming traffic promptly is vital. This is typically measured in packets per second (PPS) or transactions per second (TPS). Testing involves simulating various traffic loads and measuring latency using tools like `tcpreplay` or `Wireshark`.
- Scalability: Refers to the IDS's capacity to maintain its performance metrics as the network traffic volume or system complexity increases. Scalability testing often involves stress tests to evaluate performance under heavy traffic conditions.
- Resource Utilization: Monitors the consumption of critical system resources such as CPU, memory, and network bandwidth. Tools like `Prometheus` and `Grafana` are widely employed to visualize and analyze these metrics over time, providing insights into the efficiency of the IDS.

In practice, these metrics are not evaluated in isolation; rather, trade-offs between them are considered depending on the operational requirements of the IDS. For instance, in environments where high availability is critical, reducing the FPR may be prioritized, even at the cost of a slightly lower TPR. To facilitate the dynamic assessment of these metrics during testing, monitoring frameworks such as `Prometheus` and `Grafana` are commonly employed. These tools allow for real-time visualization of performance data, enabling researchers and practitioners to identify bottlenecks, analyze trends, and refine system configurations accordingly. Additionally, statistical methods, such as ROC curves and PR curves, are used to comprehensively evaluate the trade-offs between detection thresholds and overall performance. The AUC for both ROC and PR curves provides a robust scalar metric summarizing the IDS's performance across varying conditions [155–158]. By rigorously defining and analyzing these objectives and metrics, researchers can ensure a thorough evaluation of the IDS, enabling it to meet the operational demands of contemporary network environments. Dataset Preparation and Preprocessing: The quality, structure, and relevance of datasets are pivotal to the integrity and effectiveness of IDS

testing. Datasets serve as the foundation for training, validating, and testing ML models used in IDS. A rigorous and methodical approach to dataset preparation ensures that the results obtained during evaluation are both accurate and generalizable [159–162]. Below are key steps and techniques commonly employed in this process:

- Partitioning of Datasets: Typically, datasets are divided into three subsets to facilitate unbiased evaluation:
  - Training Set: Used to train the ML model, this subset represents the majority of the dataset (usually 70–80%).
  - Validation Set: A smaller subset (10–15%) utilized during the training phase to fine-tune hyperparameters and prevent overfitting.
  - Testing Set: A final independent subset (10–15%) reserved for evaluating the model's performance, ensuring that it generalizes well to unseen data.
- Preprocessing Techniques: Preprocessing is crucial for optimizing datasets and enhancing the efficacy of ML algorithms. Key preprocessing methods include the following:
  - Normalization: Ensures that feature values are scaled within a specific range (e.g., [0, 1]) to avoid biases introduced by features with larger magnitudes. Formally,

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

    Normalization prevents numerical instability during model training, especially in algorithms sensitive to scale, such as Support Vector Machines (SVMs) or neural networks.
  - Feature Selection: Identifies the most relevant features for the IDS task to reduce dimensionality and improve computational efficiency. Techniques like Recursive Feature Elimination (RFE) or mutual information analysis can help rank features based on their predictive power.
  - Dimensionality Reduction: Methods like PCA or t-SNE are employed to project high-dimensional data into a lower-dimensional space while preserving essential patterns. For example, PCA aims to maximize variance along principal components:

$$Z = XW$$

    where X is the original data matrix, W is the matrix of eigenvectors, and Z is the reduced data representation.
  - Handling Imbalanced Data: Many IDS datasets suffer from an imbalance between benign and malicious samples. Techniques such as oversampling (e.g., SMOTE—Synthetic Minority Oversampling Technique) or undersampling can address this issue. SMOTE works by creating synthetic examples based on the feature-space similarity of existing minority-class samples.
- Augmentation and Synthetic Data Generation: When datasets lack specific attack patterns or are too small for robust model training, augmentation methods or synthetic data generation are employed. These approaches include the following:
  - Augmentation: Perturbing existing data to create new samples, such as adding noise or transforming features (e.g., shifting IP addresses while preserving traffic patterns).

  - Synthetic Data Generation: Creating entirely new samples using techniques like GANs or probabilistic models. GANs, for instance, generate realistic data samples by training two neural networks—a generator and a discriminator—in competition:

$$\min_G \max_D \mathbb{E}[\log D(x)] + \mathbb{E}[\log(1 - D(G(z)))]$$

    where $D(x)$ is the discriminator's output for real data x, and $G(z)$ is the generator's output for random noise z.

- Validation of Dataset Authenticity: To ensure the integrity and utility of datasets, careful validation is necessary. Synthetic data, for example, must closely replicate the statistical properties of real-world data without introducing biases. Techniques like Kolmogorov–Smirnov tests or visual comparisons using density plots are commonly used for this purpose.
- Preprocessing Tools and Libraries: Several tools and libraries facilitate efficient preprocessing of IDS datasets:
  - Python Libraries:
    - `Scikit-learn`: Provides utilities for normalization, feature selection, dimensionality reduction, and cross-validation.
    - `Pandas`: Enables efficient manipulation and analysis of tabular data, including data cleaning and transformation tasks.
    - `NumPy`: Supports numerical operations such as matrix manipulations, which are fundamental in tasks like PCA.
  - Specialized Tools: Tools like CICFlowMeter can extract network flow features from raw traffic data, while frameworks like Weka offer integrated platforms for preprocessing and ML workflows.

In summary, dataset preparation and preprocessing are critical components of IDS testing. By employing rigorous methods and leveraging powerful tools, researchers can ensure that their datasets are representative, balanced, and optimized for effective evaluation. This foundational step significantly influences the overall performance and reliability of IDS systems in detecting and mitigating diverse threats. Simulated and Real-World Testing Environments: A balanced approach incorporating both simulated and real-world testing environments often delivers optimal results.

- Simulated Environments: Controlled testing conditions provided by simulation environments allow for the precise benchmarking of IDS performance. Tools such as GNS3, Cisco Packet Tracer, and Mininet are extensively utilized to model network topologies and simulate attack scenarios without endangering live production environments. Moreover, traffic generation tools like Ostinato and Scapy can create diverse attack patterns, enabling comprehensive testing of IDS capabilities [163–165].
- Real-World Testbeds: Real-world environments are critical for validating IDS solutions under realistic network conditions. Deploying IDSs in operational networks or experimental setups such as Cyber Ranges replicates authentic traffic loads, encompassing unpredictable attack patterns and legitimate user behavior. Platforms like ShadowNet and DETERLab offer robust testbeds for this purpose, facilitating practical assessments of IDS performance [166–168].

Hybrid Testing with Experimental Test-benches: A hybrid approach integrates the strengths of simulated and real-world environments to overcome their individual limitations. Simulated environments are ideal for initial validation and scalability testing, whereas experimental test-benches confirm the feasibility and effectiveness of IDS models under realistic conditions. Tools like Cuckoo Sandbox, which enables malware behavior

analysis, provide an intermediate testing framework for IDS validation [163–168]. Integration with Development Practices: The integration of automated tools and standardized development practices significantly enhances the efficiency of IDS testing. ML frameworks such as TensorFlow and PyTorch provide versatile APIs for model optimization, while platforms like Ansible and Kubernetes streamline deployment in cloud-based environments. Integrated Development Environments (IDEs) such as PyCharm and Visual Studio Code support accelerated development by offering debugging capabilities and version control. Furthermore, adhering to established coding standards, including PEP 8 for Python or MISRA for C/C++, ensures code consistency, maintainability, and compatibility [169–172]. Iterative Testing and Feedback Loops: Testing IDS solutions is inherently an iterative process. Feedback loops, which incorporate insights derived from testing outcomes, play a vital role in refining detection algorithms. Advanced techniques like adversarial training, where IDS models are tested against increasingly sophisticated simulated attacks, enhance resilience against evolving threats. Interactive tools such as Jupyter Notebooks are particularly useful in this iterative process, enabling researchers to experiment with various parameters and configurations in real time [173–175].

6.3. Challenges in Using Online Datasets

While publicly available datasets serve as a cornerstone for IDS research, their utilization is not without challenges. Addressing these issues is imperative to ensure the robustness and reliability of IDS solutions. Below, we delve into the major difficulties associated with using online datasets and propose actionable solutions: Dataset Quality and Relevance: Many publicly accessible datasets, such as KDD Cup 99 or NSL-KDD, were created several years ago and often fail to reflect modern attack patterns. Consequently, their applicability to contemporary network environments is limited. Furthermore, some datasets exhibit inherent biases, such as imbalanced attack categories or redundant records, which can distort evaluation outcomes.

- Solution: Researchers are encouraged to utilize updated datasets like UNSW-NB15 or CIC-IDS2017/2018, which encompass more recent attack scenarios. When such datasets are unavailable, synthetic data generation or adversarial data augmentation can complement existing datasets. Tools like Python's Faker library or GANs are effective in generating realistic synthetic data tailored to specific requirements.

Lack of Standardization: The absence of universal standards for dataset formats, features, and labeling conventions poses a significant barrier to the comparison of IDS models tested on disparate datasets. This lack of standardization also complicates efforts to integrate multiple datasets into a cohesive testing framework.

- Solution: The adoption of common data formats, such as NetFlow or CSV, and the use of standardized feature extraction tools like CICFlowMeter can mitigate these issues. Open-source repositories hosting preprocessed datasets in standardized formats further facilitate reproducibility and cross-comparison of research findings.

Realism vs. Privacy Concerns: Datasets derived from real-world networks often lack sufficient diversity or complexity due to stringent privacy regulations. Anonymity techniques, although essential for protecting sensitive information, may inadvertently remove critical features, diminishing the dataset's utility for IDS evaluation.

- Solution: Employing hybrid datasets that combine real-world traffic with simulated data can strike a balance between realism and privacy. Techniques like differential privacy enable the anonymity of sensitive information while preserving dataset utility for research purposes.

High Computational Effort: Many IDS datasets are large and high-dimensional, necessitating significant computational resources for preprocessing, training, and evaluation. This can pose a substantial challenge for researchers with limited access to high-performance computing infrastructure.

- Solution: Dimensionality reduction techniques, such as PCA or autoencoders, can alleviate computational demands. Additionally, cloud-based platforms like AWS and Google Cloud offer scalable resources to support the training of IDS models. Frameworks such as Apache Spark and Dask can further expedite preprocessing tasks for voluminous datasets.

Evaluation Biases: Overfitting to specific datasets remains a prevalent issue in IDS research, resulting in models that excel in controlled experiments but fail to generalize to real-world deployments.

- Solution: To enhance generalization, researchers should employ cross-validation techniques with diverse datasets and incorporate domain adaptation methods. Prioritizing datasets that include varied traffic types and attack scenarios ensures better alignment with real-world conditions. Tools like Weka and RapidMiner facilitate experimentation with multiple validation techniques, providing deeper insights into model robustness.

By proactively addressing these challenges, researchers and practitioners can unlock the full potential of public datasets, fostering the development of robust and adaptive IDSs capable of meeting the demands of an ever-evolving cybersecurity landscape.

7. Conclusions and Future Perspectives

This paper has provided a comprehensive exploration of IDSs, focusing on their architectures, detection methodologies, and application in modern networking environments. The analysis encompassed both traditional approaches and emerging advancements, particularly the integration of ML and AI to address evolving cyber threats. By examining key performance metrics, real-world challenges, and practical implementations, this work contributes to a deeper understanding of the current state of IDS technologies and highlights their critical role in securing increasingly complex and distributed networks. A central contribution of this paper lies in its detailed evaluation of IDS detection techniques, including signature-based, anomaly-based, and behavior-based methods. Each of these approaches presents unique advantages and limitations, emphasizing the necessity of a layered and context-specific deployment strategy. For instance, while signature-based systems remain highly effective against well-documented threats, their inability to detect zero-day attacks underlines the importance of integrating adaptive methods, such as anomaly-based and behavior-based systems, into modern IDS frameworks. The discussion further highlighted the increasing importance of hybrid solutions that leverage the strengths of multiple detection paradigms to achieve greater accuracy and resilience. In addition to detection methodologies, this study also examined the application of IDSs in diverse operational contexts, such as cloud environments, IoT networks, and ICS. These environments introduce unique challenges, including scalability, resource constraints, and the need for seamless integration with existing infrastructures. The findings underscore the significance of designing IDS solutions tailored to the specific requirements and constraints of these settings, with a particular emphasis on minimizing computational overhead, enhancing scalability, and ensuring compliance with regulatory standards. Despite the advancements achieved in IDS technologies, several open challenges persist, presenting opportunities for future research. One key area is the reduction of false positives, which remains a significant issue in anomaly-based and behavior-based systems. Developing advanced filtering mechanisms and incorporating XAI techniques could improve system reliability and

operator trust. Additionally, the rise of encrypted communications necessitates innovative approaches to traffic analysis that respect user privacy while maintaining robust detection capabilities. Federated learning and decentralized IDS architectures also warrant further exploration, offering promising avenues to enhance privacy and scalability in distributed environments. Future research should also prioritize energy efficiency, particularly in IoT and edge computing scenarios where resource constraints are critical. Lightweight algorithms, hardware accelerators, and energy-aware designs will be pivotal in ensuring the feasibility of IDS deployment in these contexts. Furthermore, as cyber threats become increasingly sophisticated, resilience against adversarial attacks will be essential. Robust ML models capable of detecting and mitigating adversarial inputs are expected to play a crucial role in strengthening IDS effectiveness. In conclusion, the evolution of IDS technologies reflects the dynamic and multifaceted nature of modern cybersecurity challenges. By combining traditional detection methods with advanced AI-driven approaches and addressing key implementation challenges, IDSs can continue to serve as a cornerstone of network security. As the field progresses, interdisciplinary collaboration and innovation will be vital in developing IDS solutions that are not only effective and adaptive but also ethical, sustainable, and aligned with the needs of future digital ecosystems.

Author Contributions: Conceptualization, L.D., P.D. and D.P.; methodology, L.D., P.D. and D.P.; software, L.D., P.D. and D.P.; validation, L.D., P.D. and D.P.; formal analysis, L.D., P.D. and D.P.; investigation, L.D., P.D. and D.P.; resources, L.D., P.D. and D.P.; data curation, L.D., P.D. and D.P.; writing—original draft preparation, L.D., P.D. and D.P.; writing—review and editing, L.D., P.D. and D.P.; visualization, L.D., P.D. and D.P.; supervision, L.D., P.D. and D.P.; project administration, L.D., P.D. and D.P.; funding acquisition, L.D., P.D. and D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the project PNRR CN 1 "Centro Nazionale per Simulation, Calculation and Analysis of High-Performance Data", CUP I53C22000690001 Spoke 6 "Multiscale Modeling and Engineering Applications"; partly from the project "HARdware supported Post Quantum Over-the-Air Software Update and Intrusion Detection System for NExt Generation Secure CarS" (HARDNESS), within "SEcurity and RIghts In the CyberSpace" (SERICS)–CUP D43C22003050001 Spoke 7; and partly from the Italian Ministry of Education and Research (MIUR) in the framework of the FoReLab (Future-Oriented REsearch LABoratory) "Departments of Excellence".

Data Availability Statement: No new data were generated within this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

List of the main acronyms used in this paper:

Acronym	**Extended Meaning**
IDS	Intrusion Detection System
NIDS	Network-based Intrusion Detection System
HIDS	Host-based Intrusion Detection System
AI	Artificial Intelligence
ML	Machine Learning
SDN	Software-Defined Networking
ICS	Industrial control system
IoT	Internet of Things
IIoT	Industrial Internet of Things
APT	Advanced Persistent Threat
DoS	Denial of service
DDoS	Distributed denial of service

XAI	Explainable Artificial Intelligence
CNN	Convolutional neural network
RNN	Recurrent neural network
GAN	Generative Adversarial Network
DRL	Deep Reinforcement Learning
PCA	Principal Component Analysis
PR	Precision–Recall
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
TPR	True-Positive Rate
FPR	False-Positive Rate
FNR	False-Negative Rate
F1-Score	Harmonic Mean of Precision and Recall
NSL-KDD	Network Security Laboratory—Knowledge Discovery and Data Mining
CVEs	Common Vulnerabilities and Exposures
MQTT	Message Queuing Telemetry Transport
SSH	Secure Shell
FTP	File Transfer Protocol
API	Application Programming Interface
DPI	Deep Packet Inspection
ELK Stack	Elasticsearch, Logstash, Kibana
CICFlowMeter	Canadian Institute for Cybersecurity FlowMeter
SMOTE	Synthetic Minority Oversampling Technique
XSS	Cross-Site Scripting
SCADA	Supervisory Control and Data Acquisition
VPN	Virtual Private Network

References

1. Elsayed, M.A.; Wrana, M.; Mansour, Z.; Lounis, K.; Ding, S.H.H.; Zulkernine, M. AdaptIDS: Adaptive Intrusion Detection for Mission-Critical Aerospace Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23459–23473. [CrossRef]
2. Mehedi, S.T.; Anwar, A.; Rahman, Z.; Ahmed, K.; Islam, R. Dependable Intrusion Detection System for IoT: A Deep Transfer Learning Based Approach. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1006–1017. [CrossRef]
3. Papamartzivanos, D.; Gómez Mármol, F.; Kambourakis, G. Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. *IEEE Access* **2019**, *7*, 13546–13560. [CrossRef]
4. Villegas-Ch, W.; Govea, J.; Gutierrez, R.; Maldonado Navarro, A.; Mera-Navarrete, A. Effectiveness of an Adaptive Deep Learning-Based Intrusion Detection System. *IEEE Access* **2024**, *12*, 184010–184027. [CrossRef]
5. Uhm, Y.; Pak, W. Service-Aware Two-Level Partitioning for Machine Learning-Based Network Intrusion Detection With High Performance and High Scalability. *IEEE Access* **2021**, *9*, 6608–6622. [CrossRef]
6. Khan, M.A.; Karim, M.R.; Kim, Y. A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. *Symmetry* **2019**, *11*, 583. [CrossRef]
7. Rahman, M.A.; Asyhari, A.T.; Leong, L.; Satrya, G.; Tao, M.H.; Zolkipli, M. Scalable machine learning-based intrusion detection system for IoT-enabled smart cities. *Sustain. Cities Soc.* **2020**, *61*, 102324. [CrossRef]
8. Panigrahi, R.; Borah, S.; Bhoi, A.K.; Ijaz, M.F.; Pramanik, M.; Jhaveri, R.H.; Chowdhary, C.L. Performance assessment of supervised classifiers for designing intrusion detection systems: A comprehensive review and recommendations for future research. *Mathematics* **2021**, *9*, 690. [CrossRef]
9. Arshad, J.; Azad, M.A.; Amad, R.; Salah, K.; Alazab, M.; Iqbal, R. A review of performance, energy and privacy of intrusion detection systems for IoT. *Electronics* **2020**, *9*, 629. [CrossRef]
10. Dini, P.; Elhanashi, A.; Begni, A.; Saponara, S.; Zheng, Q.; Gasmi, K. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. *Appl. Sci.* **2023**, *13*, 7507. [CrossRef]
11. Spathoulas, G.P.; Katsikas, S.K. Reducing false positives in intrusion detection systems. *Comput. Secur.* **2010**, *29*, 35–44. [CrossRef]
12. Al Jallad, K.; Aljnidi, M.; Desouki, M.S. Anomaly detection optimization using big data and deep learning to reduce false-positive. *J. Big Data* **2020**, *7*, 68. [CrossRef]
13. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]

14. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]
15. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [CrossRef]
16. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
17. Agrawal, S.; Sarkar, S.; Aouedi, O.; Yenduri, G.; Piamrat, K.; Alazab, M.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R. Federated Learning for intrusion detection system: Concepts, challenges and future directions. *Comput. Commun.* **2022**, *195*, 346–361. [CrossRef]
18. Saranya, T.; Sridevi, S.; Deisy, C.; Chung, T.D.; Khan, M. Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Comput. Sci.* **2020**, *171*, 1251–1260. [CrossRef]
19. Adele, G.; Borah, A.; Paranjothi, A.; Khan, M.S.; Poulkov, V.K. A Comprehensive Systematic Review of Blockchain-Based Intrusion Detection Systems. In Proceedings of the 2024 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA,29–31 May 2024; pp. 605–611. [CrossRef]
20. Baziana, P.A. Optical Data Center Networking: A Comprehensive Review on Traffic, Switching, Bandwidth Allocation, and Challenges. *IEEE Access* **2024**, *12*, 186413–186444. [CrossRef]
21. Chen, X.; Wu, C.; Liu, X.; Huang, Q.; Zhang, D.; Zhou, H.; Yang, Q.; Khan, M.K. Empowering Network Security With Programmable Switches: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1653–1704. [CrossRef]
22. Kanade, A.; Ranganthan, C.; Babu, A.; Ramachandran, G.; Kusuma, A.; Anand, M.; Reddy, L. Analysis of wireless network security in internet of things and its applications. *Indian J. Eng.* **2024**, *21*, e1ije1675. [CrossRef]
23. Polónio, J.; Moura, J.; Neto Marinheiro, R. On the Road to Proactive Vulnerability Analysis and Mitigation Leveraged by Software Defined Networks: A Systematic Review. *IEEE Access* **2024**, *12*, 98546–98566. [CrossRef]
24. Yahaya, A.S.; Javaid, N.; Almogren, A.; Ahmed, A.; Gulfam, S.M.; Radwan, A. A Two-Stage Privacy Preservation and Secure Peer-to-Peer Energy Trading Model Using Blockchain and Cloud-Based Aggregator. *IEEE Access* **2021**, *9*, 143121–143137. [CrossRef]
25. Jamil, F.; Iqbal, N.; Imran; Ahmad, S.; Kim, D. Peer-to-Peer Energy Trading Mechanism Based on Blockchain and Machine Learning for Sustainable Electrical Power Supply in Smart Grid. *IEEE Access* **2021**, *9*, 39193–39217. [CrossRef]
26. Mohamed, M.A.; Hajjiah, A.; Alnowibet, K.A.; Alrasheedi, A.F.; Awwad, E.M.; Muyeen, S.M. A Secured Advanced Management Architecture in Peer-to-Peer Energy Trading for Multi-Microgrid in the Stochastic Environment. *IEEE Access* **2021**, *9*, 92083–92100. [CrossRef]
27. Ray, P.P. An Introduction to Dew Computing: Definition, Concept and Implications. *IEEE Access* **2018**, *6*, 723–737. [CrossRef]
28. Lim, M. C2CFTP: Direct and Indirect File Transfer Protocols Between Clients in Client-Server Architecture. *IEEE Access* **2020**, *8*, 102833–102845. [CrossRef]
29. Azhdari, A.; Ardakan, M.A. Reliability optimization of multi-state networks in a star configuration with bi-level performance sharing mechanism and transmission losses. *Reliab. Eng. Syst. Saf.* **2022**, *226*, 108556. [CrossRef]
30. Lin, C.; Cui, L.; Coit, D.W.; Lv, M. Performance analysis for a wireless sensor network of star topology with random nodes deployment. *Wirel. Pers. Commun.* **2017**, *97*, 3993–4013. [CrossRef]
31. Jiang, X.; Zhang, H.; Barsallo Yi, E.A.; Raghunathan, N.; Mousoulis, C.; Chaterji, S.; Peroulis, D.; Shakouri, A.; Bagchi, S. Hybrid Low-Power Wide-Area Mesh Network for IoT Applications. *IEEE Internet Things J.* **2021**, *8*, 901–915. [CrossRef]
32. Ghori, M.R.; Wan, T.C.; Sodhy, G.C. Bluetooth low energy mesh networks: Survey of communication and security protocols. *Sensors* **2020**, *20*, 3590. [CrossRef] [PubMed]
33. Badea, A.; Croitoru, V.; Gheorghica, D. Computer network vulnerabilities and monitoring. In Proceedings of the 2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 7–9 May 2015; pp. 49–54. [CrossRef]
34. Aslan, Ö.; Aktuğ, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics* **2023**, *12*, 1333. [CrossRef]
35. Arogundade, O.R. Network security concepts, dangers, and defense best practical. *Comput. Eng. Intell. Syst.* **2023**, *14*.
36. Heiding, F.; Katsikeas, S.; Lagerström, R. Research communities in cyber security vulnerability assessments: A comprehensive literature review. *Comput. Sci. Rev.* **2023**, *48*, 100551. [CrossRef]
37. Hussain, K.; Rahmatyar, A.R.; Riskhan, B.; Sheikh, M.A.U.; Sindiramutty, S.R. Threats and Vulnerabilities of Wireless Networks in the Internet of Things (IoT). In Proceedings of the 2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC), Tandojam, Pakistan, 8–9 January 2024; pp. 1–8. [CrossRef]
38. Drăguşin, S.A.; Bizon, N.; Boştinaru, R.N. Comprehensive Analysis Of Cyber-Attack Techniques And Vulnerabilities In Communication Channels Of Embedded Systems. In Proceedings of the 2024 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania, 27–28 June 2024; pp. 1–12. [CrossRef]

39. Almazrouei, O.S.M.B.H.; Magalingam, P.; Hasan, M.K.; Shanmugam, M. A Review on Attack Graph Analysis for IoT Vulnerability Assessment: Challenges, Open Issues, and Future Directions. *IEEE Access* **2023**, *11*, 44350–44376. [CrossRef]
40. Khan, M.M.I.; Nencioni, G. Resource Allocation in Networking and Computing Systems: A Security and Dependability Perspective. *IEEE Access* **2023**, *11*, 89433–89454. [CrossRef]
41. Hidouri, A.; Hajlaoui, N.; Touati, H.; Hadded, M.; Muhlethaler, P. A survey on security attacks and intrusion detection mechanisms in named data networking. *Computers* **2022**, *11*, 186. [CrossRef]
42. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
43. Ullah, S.; Ahmad, J.; Khan, M.A.; Alshehri, M.S.; Boulila, W.; Koubaa, A.; Jan, S.U.; Ch, M.M.I. TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks. *Comput. Netw.* **2023**, *237*, 110072. [CrossRef]
44. Satilmiş, H.; Akleylek, S.; Tok, Z.Y. A Systematic Literature Review on Host-Based Intrusion Detection Systems. *IEEE Access* **2024**, *12*, 27237–27266. [CrossRef]
45. Nallakaruppan, M.K.; Somayaji, S.R.K.; Fuladi, S.; Benedetto, F.; Ulaganathan, S.K.; Yenduri, G. Enhancing Security of Host-Based Intrusion Detection Systems for the Internet of Things. *IEEE Access* **2024**, *12*, 31788–31797. [CrossRef]
46. Remya, S.; Pillai, M.J.; Arjun, C.; Ramasubbareddy, S.; Cho, Y. Enhancing Security in LLNs Using a Hybrid Trust-Based Intrusion Detection System for RPL. *IEEE Access* **2024**, *12*, 58836–58850. [CrossRef]
47. Bakro, M.; Kumar, R.R.; Husain, M.; Ashraf, Z.; Ali, A.; Yaqoob, S.I.; Ahmed, M.N.; Parveen, N. Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms Along with Random Forest Model. *IEEE Access* **2024**, *12*, 8846–8874. [CrossRef]
48. Kwon, H.Y.; Kim, T.; Lee, M.K. Advanced intrusion detection combining signature-based and behavior-based detection methods. *Electronics* **2022**, *11*, 867. [CrossRef]
49. Otoum, Y.; Nayak, A. As-ids: Anomaly and signature based ids for the internet of things. *J. Netw. Syst. Manag.* **2021**, *29*, 23. [CrossRef]
50. Dini, P.; Begni, A.; Ciavarella, S.; De Paoli, E.; Fiorelli, G.; Silvestro, C.; Saponara, S. Design and Testing Novel One-Class Classifier Based on Polynomial Interpolation with Application to Networking Security. *IEEE Access* **2022**, *10*, 67910–67924. [CrossRef]
51. Al-Fuhaidi, B.; Farae, Z.; Al-Fahaidy, F.; Nagi, G.; Ghallab, A.; Alameri, A. Anomaly-Based Intrusion Detection System in Wireless Sensor Networks Using Machine Learning Algorithms. *Appl. Comput. Intell. Soft Comput.* **2024**, *2024*, 2625922. [CrossRef]
52. Kumari, S.; Prabha, C.; Karim, A.; Hassan, M.M.; Azam, S. A Comprehensive Investigation of Anomaly Detection Methods in Deep Learning and Machine Learning: 2019–2023. *IET Inf. Secur.* **2024**, *2024*, 8821891. [CrossRef]
53. Dini, P.; Saponara, S. Design and Experimental Assessment of Real-Time Anomaly Detection Techniques for Automotive Cybersecurity. *Sensors* **2023**, *23*, 9231. [CrossRef]
54. Sen, Ö.; van der Velde, D.; Lühman, M.; Sprünken, F.; Hacker, I.; Ulbig, A.; Andres, M.; Henze, M. On specification-based cyber-attack detection in smart grids. *Energy Inform.* **2022**, *5*, 23. [CrossRef]
55. Hotellier, E.; Sicard, F.; Francq, J.; Mocanu, S. Standard specification-based intrusion detection for hierarchical industrial control systems. *Inf. Sci.* **2024**, *659*, 120102. [CrossRef]
56. Dini, P.; Diana, L.; Elhanashi, A.; Saponara, S. Overview of AI-Models and Tools in Embedded IIoT Applications. *Electronics* **2024**, *13*, 2322. [CrossRef]
57. Dini, P.; Saponara, S. Analysis, design, and comparison of machine-learning techniques for networking intrusion detection. *Designs* **2021**, *5*, 9. [CrossRef]
58. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl.-Based Syst.* **2020**, *189*, 105124. [CrossRef]
59. Yang, Z.; Liu, X.; Li, T.; Wu, D.; Wang, J.; Zhao, Y.; Han, H. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Comput. Secur.* **2022**, *116*, 102675. [CrossRef]
60. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
61. Wang, C.; Zhu, H. Wrongdoing Monitor: A Graph-Based Behavioral Anomaly Detection in Cyber Security. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2703–2718. [CrossRef]
62. Saathvika, S.; Accamma, B.L.; Santhosh, K.B.J. Adaptive Layered Machine Learning Approach to Detect and Mitigate Behavioral Based Intrusions in Wireless Sensor Network. In Proceedings of the 2024 Control Instrumentation System Conference (CISCON), Manipal, India, 2–3 August 2024; pp. 1–7. [CrossRef]
63. Yzzogh, H.; Kandil, H.; Benaboud, H. A comprehensive overview of AI-driven behavioral analysis for security in Internet of Things. In *The Art of Cyber Defense*; CRC Press: Boca Raton, FL, USA, 2024; pp. 40–51.
64. Shamekhi, A.; Shamsinejad Babaki, P.; Javidan, R. An intelligent behavioral-based DDOS attack detection method using adaptive time intervals. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 2185–2204. [CrossRef]
65. Soliman, K.; Sobh, M.A.; Bahaa-Eldin, A.M. Survey of Machine Learning HIDS Techniques. In Proceedings of the 2021 16th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 15–16 December 2021; pp. 1–5. [CrossRef]

66. Maseno, E.M.; Wang, Z.; Xing, H. A systematic review on hybrid intrusion detection system. *Secur. Commun. Netw.* **2022**, *2022*, 9663052. [CrossRef]
67. Onyema, E.M.; Dalal, S.; Romero, C.A.T.; Seth, B.; Young, P.; Wajid, M.A. Design of intrusion detection system based on cyborg intelligence for security of cloud network traffic of smart cities. *J. Cloud Comput.* **2022**, *11*, 26. [CrossRef]
68. Chiba, Z.; Abghour, N.; Moussaid, K.; Rida, M. Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms. *Comput. Secur.* **2019**, *86*, 291–317. [CrossRef]
69. Sethi, K.; Kumar, R.; Prajapati, N.; Bera, P. Deep reinforcement learning based intrusion detection system for cloud infrastructure. In Proceedings of the 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 7–11 January 2020; pp. 1–6.
70. Faber, K.; Faber, L.; Sniezynski, B. Autoencoder-based IDS for cloud and mobile devices. In Proceedings of the 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Melbourne, Australia, 10–13 May 2021; pp. 728–736. [CrossRef]
71. Arunkumar, M.; Kumar, K.A. GOSVM: Gannet optimization based support vector machine for malicious attack detection in cloud environment. *Int. J. Inf. Technol.* **2023**, *15*, 1653–1660. [CrossRef]
72. Vashishtha, L.K.; Singh, A.P.; Chatterjee, K. HIDM: A hybrid intrusion detection model for cloud based systems. *Wirel. Pers. Commun.* **2023**, *128*, 2637–2666. [CrossRef]
73. Alkadi, O.; Moustafa, N.; Turnbull, B.; Choo, K.K.R. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. *IEEE Internet Things J.* **2020**, *8*, 9463–9472. [CrossRef]
74. Maseer, Z.K.; Yusof, R.; Mostafa, S.A.; Bahaman, N.; Musa, O.; Al-Rimy, B.A.S. DeepIoT. IDS: Hybrid deep learning for enhancing IoT network intrusion detection. *Comput. Mater. Contin.* **2021**, *69*, 3946–3967.
75. Eskandari, M.; Janjua, Z.H.; Vecchio, M.; Antonelli, F. Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet Things J.* **2020**, *7*, 6882–6897. [CrossRef]
76. Wani, A.; Khaliq, R. SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). *CAAI Trans. Intell. Technol.* **2021**, *6*, 281–290. [CrossRef]
77. Alzahrani, A.O.; Alenazi, M.J.F. Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks. *Future Internet* **2021**, *13*. [CrossRef]
78. Krishnan, P.; Jain, K.; Aldweesh, A.; Prabu, P.; Buyya, R. OpenStackDP: A scalable network security framework for SDN-based OpenStack cloud infrastructure. *J. Cloud Comput.* **2023**, *12*, 26. [CrossRef]
79. Bour, H.; Abolhasan, M.; Jafarizadeh, S.; Lipman, J.; Makhdoom, I. A multi-layered intrusion detection system for software defined networking. *Comput. Electr. Eng.* **2022**, *101*, 108042. [CrossRef]
80. Alzahrani, A.O.; Alenazi, M.J. ML-IDSDN: Machine learning based intrusion detection system for software-defined network. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7438. [CrossRef]
81. Liang, W.; Li, K.C.; Long, J.; Kui, X.; Zomaya, A.Y. An industrial network intrusion detection algorithm based on multifeature data clustering optimization model. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2063–2071. [CrossRef]
82. Awotunde, J.B.; Chakraborty, C.; Adeniyi, A.E. Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 7154587. [CrossRef]
83. Arsalan, M.; Mubeen, M.; Bilal, M.; Abbasi, S.F. 1D-CNN-IDS: 1D CNN-based Intrusion Detection System for IIoT. In Proceedings of the 2024 29th International Conference on Automation and Computing (ICAC), Sunderland, UK, 28–30 August 2024; pp. 1–4. [CrossRef]
84. Rosa, L.; Cruz, T.; De Freitas, M.B.; Quitério, P.; Henriques, J.; Caldeira, F.; Monteiro, E.; Simões, P. Intrusion and anomaly detection for the next-generation of industrial automation and control systems. *Future Gener. Comput. Syst.* **2021**, *119*, 50–67. [CrossRef]
85. Li, B.; Wu, Y.; Song, J.; Lu, R.; Li, T.; Zhao, L. DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5615–5624. [CrossRef]
86. Abdel-Basset, M.; Chang, V.; Hawash, H.; Chakrabortty, R.K.; Ryan, M. Deep-IFS: Intrusion Detection Approach for Industrial Internet of Things Traffic in Fog Environment. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7704–7715. [CrossRef]
87. Gulia, N.; Solanki, K.; Dalal, S.; Dhankhar, A.; Dahiya, O.; Salmaan, N.U. Intrusion Detection System Using the G-ABC with Deep Neural Network in Cloud Environment. *Sci. Program.* **2023**, *2023*, 7210034. [CrossRef]
88. Sethi, K.; Kumar, R.; Mohanty, D.; Bera, P. Robust adaptive cloud intrusion detection system using advanced deep reinforcement learning. In Proceedings of the Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, 17–21 December 2020; Proceedings 10; Springer: Berlin/Heidelberg, Germany, 2020; pp. 66–85.
89. Shafay, M.; Ahmad, R.W.; Salah, K.; Yaqoob, I.; Jayaraman, R.; Omar, M. Blockchain for deep learning: Review and open challenges. *Clust. Comput.* **2023**, *26*, 197–221. [CrossRef]
90. Hussain, J.; Hnamte, V. Deep Learning Based Intrusion Detection System: Software Defined Network. In Proceedings of the 2021 Asian Conference on Innovation in Technology (ASIANCON), Pune, India, 27–29 August 2021; pp. 1–6. [CrossRef]

91. Liu, Z.; Ye, D.; Yang, C.; Ding, Y.; Liu, Y.; Tang, L.; Chen, C. Simplicity over Complexity: An ARN-Based Intrusion Detection Method for Industrial Control Network. *arXiv* **2024**, arXiv:2412.14669.
92. Pan, C.C.; Leu, Y. A Study of Imbalanced Dataset Classification on KDD99 Datasets with Reinforcement Learning Mechanism. In Proceedings of the 2023 International Conference on Machine Learning and Cybernetics (ICMLC), Adelaide, Australia, 9–11 July 2023; pp. 399–404. [CrossRef]
93. Landge, P.; Sherekar, S.S.; Chavan, S.R. An Intelligent Cyber Security System Approach using Machine Learning Techniques on KDD-99 Dataset. In Proceedings of the 2024 IEEE International Conference on Communication, Computing and Signal Processing (IICCCS), Asansol, India, 19–20 September 2024; pp. 1–6. [CrossRef]
94. Devarakonda, A.; Sharma, N.; Saha, P.; Ramya, S. Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets. *J. Phys. Conf. Ser.* **2022**, *2161*, 012043. [CrossRef]
95. Choudhary, S.; Kesswani, N. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. *Procedia Comput. Sci.* **2020**, *167*, 1561–1573. [CrossRef]
96. Al-Daweri, M.S.; Zainol Ariffin, K.A.; Abdullah, S.; Md. Senan, M.F.E. An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. *Symmetry* **2020**, *12*, 1666. [CrossRef]
97. Barach, J. Enhancing Intrusion Detection with CNN Attention Using NSL-KDD Dataset. In Proceedings of the 2024 Artificial Intelligence for Business (AIxB), Laguna Hills, CA, USA, 30 September–2 October 2024; pp. 15–20. [CrossRef]
98. Thana-Aksaneekorn, C.; Kosolsombat, S.; Luangwiriya, T. Machine Learning Classification for Intrusion Detection Systems Using the NSL-KDD Dataset. In Proceedings of the 2024 IEEE International Conference on Cybernetics and Innovations (ICCI), Chonburi, Thailand, 29–31 March 2024; pp. 1–6. [CrossRef]
99. Ngueajio, M.K.; Washington, G.; Rawat, D.B.; Ngueabou, Y. Intrusion detection systems using support vector machines on the kddcup'99 and nsl-kdd datasets: A comprehensive survey. In Proceedings of the SAI Intelligent Systems Conference, Amsterdam, The Netherlands, 1–2 September 2022; pp. 609–629.
100. Abrar, I.; Ayub, Z.; Masoodi, F.; Bamhdi, A.M. A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. In Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 10–12 September 2020; pp. 919–924. [CrossRef]
101. Kunhare, N.; Tiwari, R. Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques. In Proceedings of the 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 24–26 November 2018; pp. 127–131. [CrossRef]
102. Meena, G.; Choudhary, R.R. A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. In Proceedings of the 2017 International Conference on Computer, Communications and Electronics (Comptelix), Jaipur, India, 1–2 July 2017; pp. 553–558. [CrossRef]
103. Ravipati, R.D.; Abualkibash, M. Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets—A review paper. *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)* **2019**, *11*, 65–80. [CrossRef]
104. Song, J.; Takakura, H.; Okabe, Y.; Eto, M.; Inoue, D.; Nakao, K. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, Salzburg, Austria, 10–13 April 2011; pp. 29–36.
105. Ferriyan, A.; Thamrin, A.H.; Takeda, K.; Murai, J. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. *Appl. Sci.* **2021**, *11*, 7868. [CrossRef]
106. Sato, H.; Kobayashi, R. Koga2022 Dataset: Comprehensive Dataset with Detailed Classification for Network Intrusion Detection Systems. In Proceedings of the 2022 Tenth International Symposium on Computing and Networking Workshops (CANDARW), Himeji, Japan, 21–24 November 2022; pp. 351–357.
107. Miyamoto, K.; Iida, M.; Han, C.; Ban, T.; Takahashi, T.; Takeuchi, J. Consolidating Packet-Level Features for Effective Network Intrusion Detection: A Novel Session-Level Approach. *IEEE Access* **2023**, *11*, 132792–132810. [CrossRef]
108. Sheet, O.I.; Ibrahim, L.M. Intrusion Detection System Based on Machine Learning Techniques: A Survey. In Proceedings of the 2022 2nd International Conference on Advances in Engineering Science and Technology (AEST), Istanbul, Turkey, 28–29 March 2022; pp. 797–802. [CrossRef]
109. Lopes, I.O.; Zou, D.; Abdulqadder, I.H.; Akbar, S.; Li, Z.; Ruambo, F.; Pereira, W. Network intrusion detection based on the temporal convolutional model. *Comput. Secur.* **2023**, *135*, 103465. [CrossRef]
110. Soheily-Khah, S.; Marteau, P.F.; Béchet, N. Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset. In Proceedings of the 2018 1st International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 8–10 April 2018; pp. 219–226. [CrossRef]
111. Kasongo, S.M.; Sun, Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *J. Big Data* **2020**, *7*, 105. [CrossRef]
112. Moualla, S.; Khorzom, K.; Jafar, A. Improving the Performance of Machine Learning-Based Network Intrusion Detection Systems on the UNSW-NB15 Dataset. *Comput. Intell. Neurosci.* **2021**, *2021*, 5557577. [CrossRef]

113. Fathima, A.; Khan, A.; Uddin, M.F.; Waris, M.M.; Ahmad, S.; Sanin, C.; Szczerbicki, E. Performance evaluation and comparative analysis of machine learning models on the UNSW-NB15 dataset: A contemporary approach to cyber threat detection. *Cybern. Syst.* **2023**, 1–17. [CrossRef]
114. Kumar, A.; Guleria, K.; Chauhan, R.; Upadhyay, D. Advancing Intrusion Detection with Machine Learning: Insights from the UNSW-NB15 Dataset. In Proceedings of the 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), Bangalore, India, 28–29 June 2024; pp. 1–5.
115. Sallam, Y.F.; Abd El-Nabi, S.; El-Shafai, W.; Ahmed, H.E.d.H.; Saleeb, A.; El-Bahnasawy, N.A.; Abd El-Samie, F.E. Efficient implementation of image representation, visual geometry group with 19 layers and residual network with 152 layers for intrusion detection from UNSW-NB15 dataset. *Secur. Priv.* **2023**, *6*, e300. [CrossRef]
116. More, S.; Idrissi, M.; Mahmoud, H.; Asyhari, A.T. Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis. *Algorithms* **2024**, *17*, 64. [CrossRef]
117. Elhanashi, A.; Dini, P.; Saponara, S.; Zheng, Q. Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices. *Electronics* **2024**, *13*, 3562. [CrossRef]
118. Elhanashi, A.; Dini, P.; Saponara, S.; Zheng, Q. Integration of deep learning into the iot: A survey of techniques and challenges for real-world applications. *Electronics* **2023**, *12*, 4925. [CrossRef]
119. Korium, M.S.; Saber, M.; Beattie, A.; Narayanan, A.; Sahoo, S.; Nardelli, P.H. Intrusion detection system for cyberattacks in the Internet of Vehicles environment. *Ad Hoc Netw.* **2024**, *153*, 103330. [CrossRef]
120. Turukmane, A.V.; Devendiran, R. M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning. *Comput. Secur.* **2024**, *137*, 103587. [CrossRef]
121. Najafi Mohsenabad, H.; Tut, M.A. Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS 2018 dataset. *Appl. Sci.* **2024**, *14*, 1044. [CrossRef]
122. Verkerken, M.; D'hooge, L.; Sudyana, D.; Lin, Y.D.; Wauters, T.; Volckaert, B.; De Turck, F. A Novel Multi-Stage Approach for Hierarchical Intrusion Detection. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 3915–3929. [CrossRef]
123. Alzughaibi, S.; El Khediri, S. A cloud intrusion detection systems based on dnn using backpropagation and pso on the cse-cic-ids2018 dataset. *Appl. Sci.* **2023**, *13*, 2276. [CrossRef]
124. Elhanashi, A.; Gasmi, K.; Begni, A.; Dini, P.; Zheng, Q.; Saponara, S. Machine learning techniques for anomaly-based detection system on CSE-CIC-IDS2018 dataset. In Proceedings of the International Conference on Applications in Electronics Pervading Industry, Environment and Society, Genova, Italy, 26–27 September 2022; pp. 131–140.
125. Cao, Z.; Zhao, Z.; Shang, W.; Ai, S.; Shen, S. Using the ToN-IoT dataset to develop a new intrusion detection system for industrial IoT devices. *Multimed. Tools Appl.* **2024**, 1–29. [CrossRef]
126. Tareq, I.; Elbagoury, B.M.; El-Regaily, S.; El-Horbaty, E.S.M. Analysis of ton-iot, unw-nb15, and edge-iiot datasets using dl in cybersecurity for iot. *Appl. Sci.* **2022**, *12*, 9572. [CrossRef]
127. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
128. Li, J.; Othman, M.S.; Chen, H.; Yusuf, L.M. Cybersecurity Insights: Analyzing IoT Data Through Statistical and Visualization Techniques. In Proceedings of the 2024 International Symposium on Parallel Computing and Distributed Systems (PCDS), Singapore, 21–22 September 2024; pp. 1–10. [CrossRef]
129. Jagdish Kumar, P.; Neduncheliyan, S.; Mundher Adnan, M.; K, S.; Sudhakar, A. Anomaly-Based Intrusion Detection System Using Bidirectional Long Short-Term Memory for Internet of Things. In Proceedings of the 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 26–27 April 2024; pp. 1–4. [CrossRef]
130. Ahmim, A.; Maazouzi, F.; Ahmim, M.; Namane, S.; Dhaou, I.B. Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model. *IEEE Access* **2023**, *11*, 119862–119875. [CrossRef]
131. Zhao, J.; Liu, Y.; Zhang, Q.; Zheng, X. CNN-AttBiLSTM Mechanism: A DDoS Attack Detection Method Based on Attention Mechanism and CNN-BiLSTM. *IEEE Access* **2023**, *11*, 136308–136317. [CrossRef]
132. Yilmaz, A.A. Intrusion Detection in Computer Networks using Optimized Machine Learning Algorithms. In Proceedings of the 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey, 15–16 December 2022; pp. 1–5. [CrossRef]
133. Sayed, M.I.; Sayem, I.M.; Saha, S.; Haque, A. A Multi-Classifier for DDoS Attacks Using Stacking Ensemble Deep Neural Network. In Proceedings of the 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 30 May–3 June 2022; pp. 1125–1130. [CrossRef]
134. Wu, Z.; Zhang, H.; Wang, P.; Sun, Z. RTIDS: A Robust Transformer-Based Approach for Intrusion Detection System. *IEEE Access* **2022**, *10*, 64375–64387. [CrossRef]
135. Aktar, S.; Nur, A.Y. Towards DDoS attack detection using deep learning approach. *Comput. Secur.* **2023**, *129*, 103251. [CrossRef]

136. Gondi, L.; Sambangi, S.; Priya, P.K.; Anjum, S.S. A Machine Learning Approach for DDoS Attack Detection in CIC-DDoS2019 Dataset Using Multiple Linear Regression Algorithm. In Proceedings of the XVIII International Conference on Data Science and Intelligent Analysis of Information, Kakinada, India, 24–25 April 2023; pp. 393–403.
137. Stolfo, J.; Fan, W.; Lee, W.; Prodromidis, A.; Chan, P.K. Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection. In Proceedings of the DARPA Information Survivability Conference and Exposition. DISCEX'00, Hilton Head, SC, USA, 25–27 January 2000; pp. 130–144.
138. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
139. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
140. Sarhan, M.; Layeghy, S.; Moustafa, N.; Portmann, M. Netflow datasets for machine learning-based network intrusion detection systems. In Proceedings of the Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, 11 December 2020; Proceedings 10; Springer: Berlin/Heidelberg, Germany, 2021; pp. 117–135.
141. Moustafa, N.; Creech, G.; Slay, J. Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. In *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 127–156.
142. Moustafa, N.; Slay, J.; Creech, G. Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks. *IEEE Trans. Big Data* **2019**, *5*, 481–494. [CrossRef]
143. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. A Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
144. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
145. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the International Conference on Information Systems Security and Privacy, Madeira, Portugal, 22–24 January 2018.
146. Ashraf, J.; Keshk, M.; Moustafa, N.; Abdel-Basset, M.; Khurshid, H.; Bakhshi, A.D.; Mostafa, R.R. IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. *Sustain. Cities Soc.* **2021**, *72*, 103041. [CrossRef]
147. Moustafa, N. A systemic IoT–fog–cloud architecture for big-data analytics and cyber security systems: A review of fog computing. In *Secure Edge Computing*; CRC Press: Boca Raton, FL, USA, 2021; pp. 41–50.
148. Moustafa, N.; Ahmed, M.; Ahmed, S. Data Analytics-Enabled Intrusion Detection: Evaluations of ToN_IoT Linux Datasets. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; pp. 727–735. [CrossRef]
149. Moustafa, N.; Keshky, M.; Debiez, E.; Janicke, H. Federated TON_IoT Windows Datasets for Evaluating AI-Based Security Applications. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; pp. 848–855. [CrossRef]
150. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
151. Booij, T.M.; Chiscop, I.; Meeuwissen, E.; Moustafa, N.; Hartog, F.T.H.d. ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets. *IEEE Internet Things J.* **2022**, *9*, 485–496. [CrossRef]
152. Moustafa, N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, *72*, 102994. [CrossRef]
153. Saheb, M.C.P.; Yadav, M.S.; Babu, S.; Pujari, J.J.; Maddala, J.B. A review of DDoS evaluation dataset: CICDDoS2019 dataset. In Proceedings of the International Conference on Energy Systems, Drives and Automations, Kolkata, India, 30–31 December 2021; pp. 389–397.
154. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8. [CrossRef]
155. Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access* **2022**, *10*, 19572–19585. [CrossRef]

156. Salman, T.; Ghubaish, A.; Unal, D.; Jain, R. Safety score as an evaluation metric for machine learning models of security applications. *IEEE Netw. Lett.* **2020**, *2*, 207–211. [CrossRef]
157. Jha, S.; Kumar, R.; Hoang Son, L.; Abdel-Basset, M.; Priyadarshini, I.; Sharma, R.; Viet Long, H. Deep Learning Approach for Software Maintainability Metrics Prediction. *IEEE Access* **2019**, *7*, 61840–61855. [CrossRef]
158. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C. Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access* **2018**, *6*, 35365–35381. [CrossRef]
159. Vivone, G.; Dalla Mura, M.; Garzelli, A.; Pacifici, F. A Benchmarking Protocol for Pansharpening: Dataset, Preprocessing, and Quality Assessment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 6102–6118. [CrossRef]
160. Alghamdi, T.A.; Javaid, N. A Survey of Preprocessing Methods Used for Analysis of Big Data Originated from Smart Grids. *IEEE Access* **2022**, *10*, 29149–29171. [CrossRef]
161. Hasnain, M.; Pasha, M.F.; Ghani, I.; Mehboob, B.; Imran, M.; Ali, A. Benchmark Dataset Selection of Web Services Technologies: A Factor Analysis. *IEEE Access* **2020**, *8*, 53649–53665. [CrossRef]
162. Kahloot, K.M.; Ekler, P. Algorithmic Splitting: A Method for Dataset Preparation. *IEEE Access* **2021**, *9*, 125229–125237. [CrossRef]
163. Lansky, J.; Ali, S.; Mohammadi, M.; Majeed, M.K.; Karim, S.H.T.; Rashidi, S.; Hosseinzadeh, M.; Rahmani, A.M. Deep Learning-Based Intrusion Detection Systems: A Systematic Review. *IEEE Access* **2021**, *9*, 101574–101599. [CrossRef]
164. Khan, I.A.; Pi, D.; Khan, Z.U.; Hussain, Y.; Nawaz, A. HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems. *IEEE Access* **2019**, *7*, 89507–89521. [CrossRef]
165. Sadiq, A.S.; Alkazemi, B.; Mirjalili, S.; Ahmed, N.; Khan, S.; Ali, I.; Pathan, A.S.K.; Ghafoor, K.Z. An Efficient IDS Using Hybrid Magnetic Swarm Optimization in WANETs. *IEEE Access* **2018**, *6*, 29041–29053. [CrossRef]
166. Narayanan, A.; Sena, A.S.D.; Gutierrez-Rojas, D.; Melgarejo, D.C.; Hussain, H.M.; Ullah, M.; Bayhan, S.; Nardelli, P.H.J. Key Advances in Pervasive Edge Computing for Industrial Internet of Things in 5G and Beyond. *IEEE Access* **2020**, *8*, 206734–206754. [CrossRef]
167. Kornaros, G. Hardware-Assisted Machine Learning in Resource-Constrained IoT Environments for Security: Review and Future Prospective. *IEEE Access* **2022**, *10*, 58603–58622. [CrossRef]
168. Dasgupta, S.; Das, A.; Yogamani, S.; Das, S.; Eising, C.; Bursuc, A.; Bhattacharya, U. UnShadowNet: Illumination Critic Guided Contrastive Learning for Shadow Removal. *IEEE Access* **2023**, *11*, 87760–87774. [CrossRef]
169. Zhang, X.; Li, L.; Wang, Y.; Chen, E.; Shou, L. Zeus: Improving Resource Efficiency via Workload Colocation for Massive Kubernetes Clusters. *IEEE Access* **2021**, *9*, 105192–105204. [CrossRef]
170. Ruíz, L.M.; Pueyo, P.P.; Mateo-Fornés, J.; Mayoral, J.V.; Tehàs, F.S. Autoscaling Pods on an On-Premise Kubernetes Infrastructure QoS-Aware. *IEEE Access* **2022**, *10*, 33083–33094. [CrossRef]
171. Phuc, L.H.; Phan, L.A.; Kim, T. Traffic-Aware Horizontal Pod Autoscaler in Kubernetes-Based Edge Computing Infrastructure. *IEEE Access* **2022**, *10*, 18966–18977. [CrossRef]
172. Kaur, K.; Garg, S.; Kaddoum, G.; Ahmed, S.H.; Atiquzzaman, M. KEIDS: Kubernetes-Based Energy and Interference Driven Scheduler for Industrial IoT in Edge-Cloud Ecosystem. *IEEE Internet Things J.* **2020**, *7*, 4228–4237. [CrossRef]
173. Yang, A.; Zhuansun, Y.; Liu, C.; Li, J.; Zhang, C. Design of Intrusion Detection System for Internet of Things Based on Improved BP Neural Network. *IEEE Access* **2019**, *7*, 106043–106052. [CrossRef]
174. Mendonça, R.V.; Teodoro, A.A.M.; Rosa, R.L.; Saadi, M.; Melgarejo, D.C.; Nardelli, P.H.J.; Rodríguez, D.Z. Intrusion Detection System Based on Fast Hierarchical Deep Convolutional Neural Network. *IEEE Access* **2021**, *9*, 61024–61034. [CrossRef]
175. Sadaf, K.; Sultana, J. Intrusion Detection Based on Autoencoder and Isolation Forest in Fog Computing. *IEEE Access* **2020**, *8*, 167059–167068. [CrossRef]