sensors

MDPI

Review

Green Intrusion Detection Systems: A Comprehensive Review and Directions

Swapnoneel Roy [1,*], Sriram Sankaran [2] and Mini Zeng [3]

[1] School of Computing, University of North Florida, Jacksonville, FL 32224, USA
[2] Center for Cybersecurity Systems and Networks, Amrita Vishwa Vidyapeetham, Amritapuri 690525, Kerala, India; srirams@am.amrita.edu
[3] Computing Sciences, Jacksonville University, Jacksonville, FL 32211, USA; mzeng@ju.edu
* Correspondence: s.roy@unf.edu; Tel.: +1-904-620-2985

Abstract: Intrusion detection systems have proliferated with varying capabilities for data generation and learning towards detecting abnormal behavior. The goal of green intrusion detection systems is to design intrusion detection systems for energy efficiency, taking into account the resource constraints of embedded devices and analyzing energy–performance–security trade-offs. Towards this goal, we provide a comprehensive survey of existing green intrusion detection systems and analyze their effectiveness in terms of performance, overhead, and energy consumption for a wide variety of low-power embedded systems such as the Internet of Things (IoT) and cyber physical systems. Finally, we provide future directions that can be leveraged by existing systems towards building a secure and greener environment.

Keywords: energy optimization; intrusion detection; cybersecurity; anomaly detection

Citation: Roy, S.; Sankaran, S.; Zeng, M. Green Intrusion Detection Systems: A Comprehensive Review and Directions. *Sensors* **2024**, *24*, 5516. https://doi.org/10.3390/s24175516

Academic Editors: Jose Manuel Molina López and Alessandra Rizzardi

Received: 25 May 2024
Revised: 5 August 2024
Accepted: 13 August 2024
Published: 26 August 2024

1. Introduction

In the contemporary landscape of information technology, the proliferation of interconnected systems and the exponential growth of digital data have brought about unprecedented opportunities as well as formidable challenges. One of the most pressing challenges is to ensure the security of these systems against malicious intrusions that continue to evolve in sophistication and scale. Intrusion detection systems (IDSs) play a pivotal role in fortifying the defenses of networks and computing infrastructures by identifying and thwarting unauthorized access, malicious activities, and anomalies in real-time.

However, the effectiveness of conventional IDS solutions is often hindered by their substantial resource consumption, particularly in terms of energy and computational resources. As concerns about environmental sustainability and energy efficiency continue to mount, there is a growing imperative to develop greener alternatives that mitigate the environmental impact of intrusion detection operations while maintaining high levels of security efficacy.

The sustainability of intrusion detection-based systems in critical sectors like healthcare, smart cities, and defense can indeed be a significant concern due to several key challenges and factors. There are several issues that can lead to unsustainable practices. For instance, IDS can consume significant computational resources, especially if they are inspecting large volumes of network traffic or processing numerous events. This can lead to scalability issues and increased operational costs. In addition, IDSs require regular updates to keep up with emerging threats and vulnerabilities. This involves updating signatures, rules, and policies, which can be time-consuming and resource-intensive.

In addition, IDSs can generate false positives (incorrectly identifying benign activities as malicious) or false negatives (inability to detect actual attacks). Managing these false alerts requires human intervention and can lead to alert fatigue, where security analysts overlook genuine threats amidst a sea of false alarms. In addition, IDSs must adapt

to changing threats and network environments. This requires continuous monitoring, tuning, and customization to remain effective against new attack vectors. Depending on the deployment model, IDSs may monitor and analyze sensitive network traffic, raising privacy concerns, especially in environments where user privacy is paramount.

Finally, deploying and managing IDSs can be complex, requiring specialized skills and knowledge. This complexity can deter organizations from implementing IDSs or result in misconfigurations that reduce their effectiveness. Implementing and maintaining IDSs involves significant upfront and ongoing costs, including hardware, software, training, and personnel. Overall, while IDSs are essential for detecting and mitigating cyber threats, their sustainability depends on effectively addressing these challenges through efficient resource utilization, automation, continuous improvement, and alignment with organizational goals and priorities.

Green intrusion detection systems (GIDSs) have emerged as a promising paradigm that reconcile the imperatives of security and sustainability. These systems are characterized by their emphasis on minimizing energy consumption, reducing carbon footprint, and optimizing resource utilization without compromising on the core objective of threat detection and prevention. Using innovative techniques from various domains, such as machine learning, data analytics, optimization algorithms, and hardware design, GIDS aims to achieve a harmonious balance between security requirements and environmental responsibility.

This review seeks to answer the central question: *how can green intrusion detection systems effectively integrate sustainability principles without compromising their ability to detect and prevent threats?*

Our Contributions

In this comprehensive review, we delve into the multifaceted landscape of green intrusion detection systems, exploring their underlying principles, methodologies, applications, and challenges. Through a systematic examination of existing research endeavors and technological advancements, we seek to provide insight into the state-of-the-art approaches toward the development of GIDSs, along with critical assessments of their strengths, limitations, and potential avenues for improvement.

To our knowledge, two previously published review articles exist in the literature [1,2] that specifically review IDSs based on energy optimization. Our current review article differs from theirs in the following ways: (1) our focus is primarily on energy optimization techniques that exist for intrusion detection systems, unlike [1,2] who focused primarily on intrusion detection techniques and secondarily on energy optimization; (2) our survey includes works that were added to the literature after both of the existing reviews.

This review includes the literature covering the following grounds: appropriate study designs that rigorously evaluate IDSs, specific sectors with critical infrastructure (e.g., healthcare, smart cities, and defense) impacted by IDSs, interventions aimed at enhancing sustainability, current practices in IDS deployment and maintenance, and results of interest related to both security efficacy and environmental impact. By addressing these aspects, the review aims to provide a comprehensive understanding of how to develop and implement green intrusion detection systems that align with contemporary demands for both security and sustainability.

Furthermore, we are striving to outline future directions and research opportunities in the field of GIDSs, envisioning innovative strategies and solutions that could propel the evolution of green and sustainable intrusion detection technologies. By fostering interdisciplinary collaboration and knowledge exchange, we aspire to contribute to the ongoing discourse on cybersecurity and environmental stewardship, paving the way towards a more secure, resilient, and environmentally conscious digital ecosystem.

2. Generic Energy Optimization Techniques in Cybersecurity

Figure 1 presents a comprehensive landscape of energy-aware security depicted in a pictorial manner. The problem of energy-aware security necessitates the need for a

holistic approach to integrate techniques to defend against cyber attacks while minimizing carbon footprint. In particular, novel mechanisms for attack modeling are necessary to understand the ever-increasing capabilities of attacks in diverse forms of networks such as Wi-Fi, cellular, fog and edge computing, and the Internet of Things. In addition, mitigating attacks requires the development of green intrusion detection systems that are capable of increasing detection accuracy while optimizing resources and analyzing the resulting energy–performance–security trade-offs.

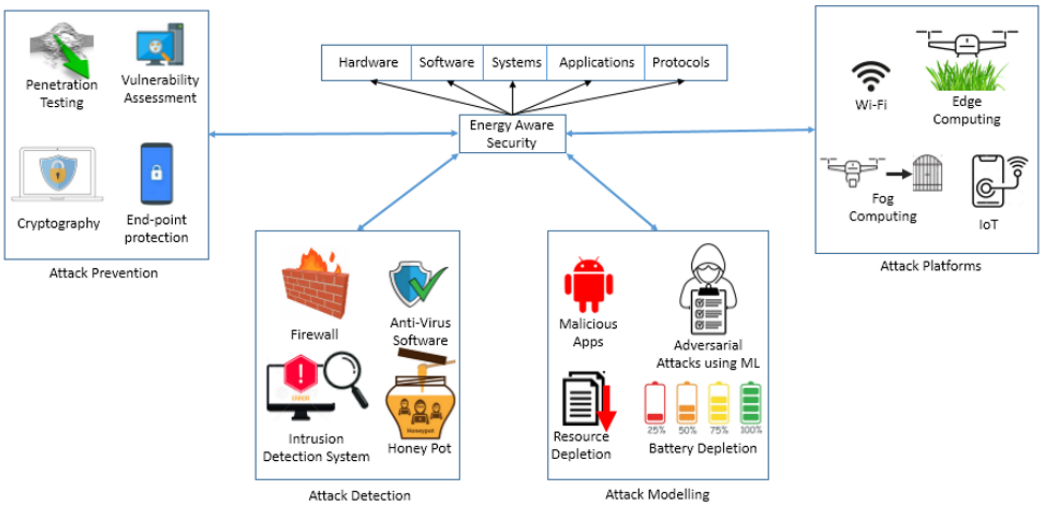

Figure 1. Energy-aware security.

In this section, we provide a summary of energy optimization techniques at different layers, such as hardware, hypervisors, operating systems, system software, and applications. In addition, we explore the relationship between energy efficiency in each of the layers and the design of green intrusion detection systems. A summary of the energy optimization techniques at different layers of the computing systems is presented in Table 1.

Table 1. Summary of existing approaches for energy optimization.

Layer	Techniques	Papers
Hardware	Hardware Acceleration	[3–7]
Hypervisors	Virtualization and Containerization	[8–16]
Operating Systems	Resource Management	[17–24]
System Software	Task Offloading	[25–30]
Applications	Adaptive Sampling and Filtering	[31–36]

2.1. Energy-Efficient Hardware Selection

A few approaches exist on energy efficient hardware towards designing green intrusion detection systems.

A study [3] examined how much energy intrusion detection software and hardware is used. With a 15-fold increase in throughput, the hardware version only required 0.03% of the energy needed by the software version of the identical algorithm. A different experiment [4] revealed that machine learning algorithms implemented on hardware utilized 46% less energy than their software counterparts, suggesting that energy-efficient hardware implementation is possible.

It was discovered that an anomaly-based intrusion detection technique for embedded systems is both hardware implementation-friendly and energy-efficient [5]. Energy-efficient intrusion detection is crucial to extend the life of wireless sensor networks. The suggested techniques for grouping nodes to carry out intrusion detection are designed to reduce average energy usage and increase network longevity.

Field-programmable gate arrays, or FPGAs, have been investigated for application in intrusion detection. A convolutional neural network based on CPU and FPGA has demonstrated a notable increase in energy efficiency over its software version [6].

Using Xilinx's Deep Learning Processing Unit IP on a Zynq Ultrascale+ (XCZU3EG) FPGA, Khandelwal et al. [7] offered a lightweight multi-attack quantized machine learning model that was trained and verified using the public CAN Intrusion Detection dataset. With an accuracy of over 99% and a false positive rate of 0.07%, the quantized model detects denial of service and fuzzing assaults with a similar accuracy rate to state-of-the-art methods reported in the literature. With software operations operating on the ECU, intrusion detection system execution uses just 2.0W and delivers a 25% reduction in per-message processing delay over state-of-the-art implementations. The deployment is perfect for real-time IDS in in-vehicle systems because it allows the ECU function to coexist with the IDS with little modification of the tasks.

2.2. Energy-Efficient Hypervisors

The foundation of virtualization technology, the hypervisor, optimizes resource consumption by minimizing the total energy footprint and allowing many intrusion detection system instances to run on the same physical hardware. The process of virtualization involves the establishment of virtual instances of memory, storage, and CPUs that are under the supervision of hypervisors. In this section, we look at techniques that help hypervisors run more efficiently and use less energy overall, like hardware-assisted virtualization and power-aware scheduling.

Using power-conscious power consumption profiles, scheduling algorithms [8–11] seek to maximize virtual machine execution. Hypervisors can lower total power consumption by effectively scheduling workloads with complimentary resource usage patterns. For example, scheduling memory- and CPU-intensive operations concurrently can result in more equitable and effective resource use. Furthermore, these methods emphasize the importance of power-aware scheduling policies for cloud computing systems with high workload variability, with the goal of minimizing energy consumption and ensuring service level agreements (SLAs) while minimizing the system response time [12,13].

Modern CPUs and other hardware have virtualization-supporting characteristics like AMD-V and Intel VT-x. These technologies offer ways to reduce the overhead associated with virtualization, which improves the effectiveness of virtual machine management. Using these hardware-assisted features allows hypervisors to operate at high performance levels with less power consumption [14–16]. Thus, it is clear that when considering ARM and MIPS processors, hardware-assisted virtualization is essential for increasing performance predictability and lowering overhead in hypervisors for embedded systems. The advancement of hardware-accelerated hypervisors has also solved issues and greatly enhanced virtualization performance.

2.3. Energy-Efficient Operating Systems

Managing the computing system's resources, such as memory, storage, and computation, is a crucial part of developing operating systems that use less energy. Due to fixed resource budgets, this issue is made worse in resource-constrained systems such as the Internet of Things. Scaling resources up or down in response to demand is the aim of resource management. By scaling resources down during low-load times and up during high-load times, techniques like dynamic power management (DPM) and dynamic voltage and frequency scaling (DVFS) allow systems to consume less energy [17].

By putting components in low-power states while not in use, sleep modes and duty cycling techniques can help intrusion detection systems consume less energy. This method works especially well in settings where network traffic is irregular. Traditionally, energy management in Internet of Things applications has been a hardware support feature. However, with the introduction of low-power modes in current hardware designs, it has become necessary to integrate energy management at the operating system level [17].

In comparison to traditional Linux task scheduling, the energy-aware scheduler (EAS) improves energy efficiency by over 30% by introducing micro-operations performed per joule (OPJ) as a metric for run-time task energy efficiency [18,19]. In order to achieve optimum energy efficiency, a new system software architecture is presented for heterogeneous ISA platforms [37], highlighting the critical role that system software must play in energy reduction.

Reducing operating frequencies and balancing workloads on heterogeneous multicore architectures are two strategies that are the focus of research on energy-efficient mobile computing [20–23]. However, integrating energy-efficient methods into large-scale computing systems can be difficult. However, new methods such as SYnergy promise to achieve precise energy savings by integrating language, a compiler, runtime, and a job scheduler [24].

2.4. Energy-Efficient System Software

Existing research has examined fog and edge computing-based Internet of Things frameworks for energy-efficient system software. Although fog and edge computing are essential for managing the information flow of complex and massive networks, such as the Internet of Things (IoT), their usage can have an effect on electricity prices and carbon emissions [25]. One way to lessen the requirement for data transfer is to deploy IDS functions at the network edge, which is closer to the data source. Localized processing and decision-making are made possible by edge computing, which promotes faster and more energy-efficient processes.

In order to extend network lifespan, research focuses on developing energy-efficient solutions for the edge–fog environment, such as thorough frameworks for energy-efficiency analysis and intelligent energy-management techniques [26–28]. In fog computing, task offloading is essential for maximizing resource usage and enhancing system performance. Long-term device functioning and reducing environmental impact require effective energy management [29]. In order to optimize fog node selection and minimize energy usage while respecting service-level agreement (SLA) parameters, studies have presented energy-aware task offloading strategies and dynamic programming approaches [27,29].

In fog computing systems, there are trade-offs between processing and communication resources. Algorithms for figuring out the best resource provisioning to reduce overall energy consumption without compromising service latency performance have been developed based on analytical findings [30].

2.5. Energy-Efficient Applications

Energy can be saved at the application level by reducing the volume of data processed by the IDS through the use of adaptive data sampling and filtering techniques. Low energy usage and high detection accuracy are maintained by the system through clever selection and processing of only the most pertinent data.

Roy et al. [31] created a general energy complexity model (ECM) based on double-data-rate synchronous dynamic random access memory (DDR SDRAM) as its reference architecture. DDR's primary memory is separated into banks, each of which has a set number of pieces. Each bank distributes data in segments. In addition, each bank has a distinct component called the sensory amplifier. For each data access, the necessary data chunk must be brought into the relevant bank's sense amplifier. Each sense amplifier is limited to holding one chunk at a time, so the current chunk needs to be returned to its bank before a new one is brought in for subsequent access. This model has been applied in different cybersecurity protocols (e.g., [32,38–42]) to optimize their energy consumption.

For smart home contexts, Nimmy et al. [33] created a lightweight authentication protocol based on geometric secret sharing and nonuniformity of the photoresponse (PRNU). The PRNU of a smartphone camera functions as a biometric for distinct identification, eliminating the need to memorize passwords. In comparison to current methods, their

suggested system was demonstrated to be both energy-efficient and lightweight. Often, energy consumed by a protocol $E_{protocol}$ is computed using the following equation:

$$E_{protocol} = E_{comp} + E_{comm}, \quad (1)$$

where E_{comp} and E_{comm} refer to energy consumed due to computation and communication, respectively.

Protocols perform encryption/decryption along with compression to secure packets, which incur computation energy, and these packets are sent to receiver, thus impacting communication energy. Thus, the energy consumed by computation (E_{comp}) and communication (E_{comm}) can be further broken down in the following way:

$$E_{comp} = E_{encrypt} + E_{decrypt} + E_{compress}, \quad (2)$$

where $E_{encrypt}$, $E_{decrypt}$, and $E_{compress}$ refer to the energy consumed due to operations such as encryption, decryption, and compression, respectively.

$$E_{comm} = E_{transmit} + E_{receive} + E_{idle} + E_{sleep}, \quad (3)$$

where $E_{transmit}$, $E_{receive}$, E_{idle}, and E_{sleep} refer to energy consumed due to different communication modes such as transmit, receive, idle, and sleep, respectively.

Additionally, Nimmy et al. [34] took advantage of the behavioral traits of Internet of Things (IoT) devices to construct an anomaly detection system utilizing generated data that took into account the effect of malevolent actions like DoS and brute-force attacks on the amount of power that IoT devices consumed. An analysis revealed that behavioral traits such as electricity use have the potential to identify well-known attacks on smart homes.

A lightweight blockchain-based framework for the Narrowband Internet of Things was developed by Mohan et al. [35]. It has features like partitioning dynamic base station memory to improve memory utilization efficiency and scalability and linear hash chain-based storage to avoid costly Merkle tree verification. An analysis of the suggested framework revealed how lightweight it is in relation to current methods. Moreover, to address the shortcomings of the Cu-MAC protocol and enhance channel quality, Mohan et al. [36] created EP-CuMAC for the Narrowband Internet of Things. Furthermore, technologies based on deep learning and machine intelligence are being developed to reduce the effect of retransmissions. The evaluation revealed that, compared to other methods, EP-CuMAC was able to manage the trade-offs between energy, performance, and security.

2.6. Lightweight Operating Systems and Virtualization

The open-source operating system Contiki was created mainly with Internet of Things (IoT) devices in mind. It is an operating system that runs on microcontrollers with little resources, including 8- or 16-bit CPUs, and is lightweight, low-power, and memory-efficient. Swedish researcher Adam Dunkels [43] originally released Contiki in 2003 as a component of his doctoral thesis. Since then, its capacity to operate on a variety of hardware platforms and support a number of communication protocols, including IPv6, RPL, and CoAP, has helped it become more well-known in the IoT field.

The framework was built with the Contiki operating system by Arshad et al. [44], who also carried out a thorough study to find any potential performance trade-offs. In order to accomplish effective and economical intrusion detection for IoT systems, the collaborative intrusion detection for IoT (COLIDE) architecture makes use of the basic idea of collaboration between individual sensor nodes and the edge router. An edge router component and a device level make up the framework. The assessment findings show that, while providing effective collaborative intrusion detection for IIoT systems, the suggested framework can decrease energy and communication overheads.

2.7. Low-Power Networking Protocols

Nobakht et al. [45] suggested a host-based IDS for smart homes that makes use of software-defined technology (SDN). They suggested the Internet of Things (IoT-IDM) as a framework for intrusion detection and mitigation that would protect smart devices installed in residential settings at the network level. The IoT-IDM monitors the activity of the intended smart device network within the home and looks for potentially harmful or suspicious behavior. This framework is implemented using OpenFlow, an enabling communication protocol for software-defined networking technologies. Ultimately, an IoT-IDM prototype is created, and an actual IoT device, a smart lightbulb, is used to illustrate the applicability and effectiveness of the suggested framework. Various detection modules, such as signature, anomaly, or specification-based approaches, can be used to carry out the detection. The authors assert that putting the intrusion detection module inside an Internet of Things (IoT) device lowers communication cost; however, this also increases the IoT device's processor overhead, which is important for such low-powered devices.

Using two routing methods, loop-free (LF) KPS and KPS, Ghosh et al. [46] developed an energy-efficient approach for detecting intrusions through unmanned borders and other sensitive sites with prolonged network lifetimes. They demonstrated how data transfer via KPS and LF-KPS protocols will guarantee an extended lifetime for the deployed network by contrasting these two methods with LEACH and TEEN.

An energy-efficient security-aware architecture for wireless control systems intended for application in factory automation was proposed by Muradore et al. [47]. They suggested using packet-based selective encryption to cut down on energy usage and identify the beginning and end of an assault. Since the packet transmission rate affects energy usage as well, particularly during attacks, they advised tailoring the approach to the performance of instantaneous control.

The rest of the paper is organized in the following manner. In Section 3, certain important terms related to intrusion detection and energy efficiency are defined. Section 4 contains the relevant reviews that looked at intrusion detection methods for energy consumption and optimization. Section 5 presents the proposed taxonomy and a review of the literature on IDSs for energy optimization. Section 6 offers one of the work's most significant contributions: a thorough analysis of open issues and possible directions for further research on IDSs in energy optimization. Finally, in Section 8, we provide some concluding remarks.

3. Intrusion Detection Deployment and Techniques

This section reviews modern intrusion systems generically. IDSs are categorized on the basis of functionality. Table 2 summarizes different kinds of IDS techniques and their corresponding deployments in numerous environments.

3.1. Intrusion Detection Deployment

Intrusion detection systems (IDSs) play a crucial role in securing various environments by monitoring and analyzing activities to detect malicious behaviors. Figure 2 illustrates the deployment of IDSs in three layers: the application layer, the network layer, and the perception layer. An IoT IDS focuses on the specific needs of IoT ecosystems, employing lightweight protocols and algorithms to efficiently monitor IoT devices' traffic and behavior, detecting anomalies or unauthorized access attempts. A host-based IDS (HIDS) operates on individual devices, monitoring system calls, application logs, and file system modifications to detect suspicious activities, providing detailed information on potential threats targeting specific hosts. HIDSs can be installed on a single host or configured across a cluster of hosts, with the IDS installed on a centralized server or the cluster's primary node, a configuration known as hybrid placement. A network-based IDS (NIDS) monitors network traffic for suspicious patterns across the entire network, using methods like signature-based detection and anomaly detection to identify potential threats, ensuring a broad scope of network security. A cloud IDS is customized for cloud environments, using native cloud tools

and services to monitor virtualized resources, data flows, and user activities within the cloud infrastructure, ensuring scalable and efficient threat detection in dynamic cloud environments. Each type of IDS addresses unique security challenges, contributing to a comprehensive defense strategy. The following subsections will discuss the different categories of IDS deployment in detail.

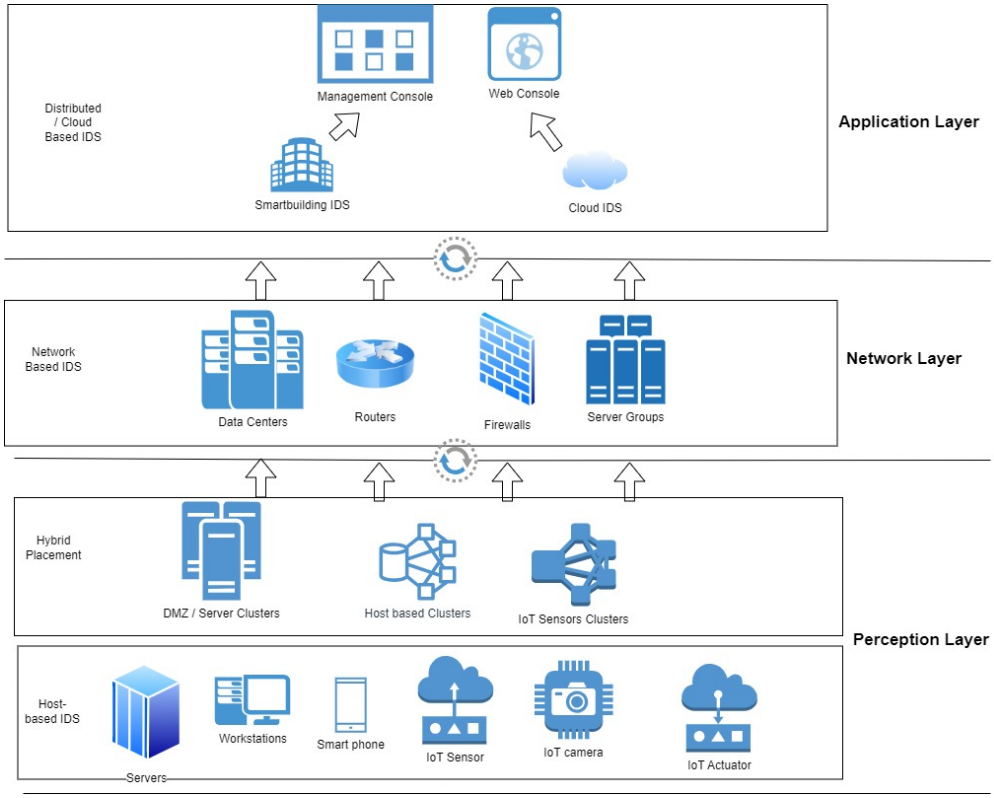

Figure 2. IDS deployment.

3.1.1. IoT Intrusion Detection

Sarika et al. [48] and Zarpelao et al. [49] offer an overview of the Internet of Things, including its security threats and different intrusion detection methods. Three layers, the application layer, network layer, and perception layer, summarize the architecture of IoT systems by Sarika et al. Placement strategies for IDSs in the Internet of Things were divided into three categories by Zarpelao et al. [49]: distributed, centralized, and hybrid. An IDS is positioned in each physical component of the LLN (low-power and lossy network) as a distributed strategy. These are self-contained nodes with optimized IDSs installed in each [50,51]. The nodes in the distributed placement may also be in charge of keeping an eye on their neighbors. A router, server, or dedicated host are examples of centralized components where the IDS is positioned in a centralized configuration. Every piece of information collected and sent to the Internet by the dispersed nodes passes through the central device. Consequently, all of the traffic exchanged may be analyzed by the IDS installed in the centralised device [52,53]. In order to capitalize on their advantages and minimize their disadvantages, hybrid IDS placement integrates the ideas of dispersed and centralized deployment. One method is to divide the network into clusters and only allow each cluster's primary node to host on an IDS [54–56].

3.1.2. Network Intrusion Detection

A network-based intrusion detection system (NIDS) is a tool used to watch and examine network traffic in order to find malicious activity or policy infractions. An NIDS examines all of the data moving via a network segment. Unauthorized access attempts, virus propagation, denial-of-service attacks, and other threats can be detected by an NIDS

by examining packets for known attack patterns, unusual behaviors, and policy violations. It frequently uses anomaly-based detection (Section 3.2) for novel or unidentified threats and signature-based detection Section 3.3.5) for known threats. To provide complete visibility and protection throughout the network architecture, an NIDS can be placed at key network locations, such as the on perimeter of the network or within particular network segments (Figure 2). Due to its real-time threat detection and response capabilities, it is a crucial part of any effective cybersecurity plan.

Various machine learning, data mining, artificial intelligence, and statistical techniques are available for anomaly-based network intrusion detection systems (NIDSs). The specifics of them will be covered in Section 3.2. One of the newest and most popular methods for identifying anomalies is deep learning. For unsupervised feature learning, Shone et al. [57] presented a deep learning classification model on an NIDS built using a stacked non-symmetric deep autoencoder. Their classifier was constructed in TensorFlow, a graphics processing unit (GPU) capable framework, and tested on the NSL-KDD and KDD-Cup'99 benchmark datasets. An effective and adaptable NIDS can be developed using the deep learning-based method suggested by Niyaz et al. [58]. They also used NSL-KDD datasets for self-taught learning (STL), a deep learning-based approach. Vianyakumar et al. [59] used millions of known good and bad network connections to train supervised learning methods like multilayer perceptron (MLP), CNNs, CNN-recurrent neural networks (CNN-RNNs), CNN-long short-term memory (CNN-LSTM), and CNN-gated recurrent units (GRUs) to model network traffic as time series, specifically transmission control protocol/internet protocol (TCP/IP) packets within a predetermined time range.

3.1.3. Host-Based Intrusion Detection

The purpose of a host-based intrusion detection systems (HIDSs) is to identify indications of malicious activity or policy violations by tracking and analyzing activities on specific devices or hosts. An HIDS offers an in-depth perspective of the security status of every host it safeguards through the analysis of system logs, file integrity, and process activities. With its concentration on the internal operations of a single device, an HIDS is more successful than a network-based IDS at spotting insider threats, illegal modifications, and local attacks. A network-based IDS, on the other hand, monitors traffic throughout the network. It can use signature-based detection to identify known attack patterns or compare the current behavior of the system to a baseline of typical activity to identify abnormalities [60]. When an HIDS detects questionable activity, including unwanted access attempts, modifications to important files, and odd system operations, it can send notifications. It is an essential line of defense for guaranteeing the integrity and security of individual systems inside an organization's overall security framework because of its comprehensive, host-specific monitoring capabilities.

Anomaly detection techniques are also widely utilized in HIDSs; an agent-based artificial immune system (ABAIS) was introduced by Ou Chung-Ming [61] into the HIDS. The risk hypothesis of the human immune system serves as the inspiration for the proposed agent-based IDS (ABIDS). An ABIDS has multiple embedded agents that work together to update the activation threshold for security responses and compute the mature context antigen value (MCAV). An anomaly detection method based on the semantic interactions of system calls was presented by Syed et al. [62]. The main idea is to model system calls as kernel module states, examine state interactions, and compare the odds of various state occurrences in anomalous and normal traces to find anomalies. By using this method, one can have a visual comprehension of the behavior of the system and make better decisions.

3.1.4. Cloud Intrusion Detection

Due to the increased use of cloud environments over the past ten years, there has been a demand for cloud intrusion detection. Strong security measures like cloud intrusion detection systems (CIDS) are essential as more and more businesses move their infrastructure and services to the cloud. Their purpose is to safeguard cloud environments from

hostile activity, illegal access, and data breaches. In order to detect any attacks in real-time, these systems track and examine user activity, system behaviors, and cloud network traffic. Cloud intrusion detection systems offer full visibility and protection in public, private, and hybrid cloud environments when coupled with the native security tools of cloud service providers and/or third-party security solutions (Figure 2). A CIDS minimizes the risk of data loss and preserves the availability and integrity of cloud resources by providing fast notifications and comprehensive forensic information to enterprises.

Palo Alto Networks powers the Google Cloud intrusion detection system (Cloud IDS), which provides network-based threat detection with signature-based detection capabilities [63]. It is intended to provide comprehensive context for security events in Google Cloud environments and to identify malware and intrusions.

GuardDuty from Amazon Web Services (AWS) is a managed threat detection service that looks out for unusual or harmful activity. It identifies and ranks possible risks in AWS accounts, workloads, and data stored on Amazon S3 using machine learning, anomaly detection, and integrated threat intelligence [64].

Advanced threat protection for hybrid cloud workloads is another feature provided by the Microsoft Azure Security Center. It offers comprehensive security monitoring and policy management across Azure subscriptions, using analytics and machine learning to identify and address threats.

3.2. Anomaly-Based Intrusion Detection Techniques

Machine learning algorithms are used by anomaly-based intrusion detection systems (IDS) to detect changes from a system, network, or user's typical behavior. These are a few typical uses for anomaly IDS machine learning methods.

Deep Learning

Nie et al. [65] provided a deep reinforcement learning-based intrusion detection method that first used statistical aspects of historical network traffic to forecast traffic patterns. Next, they applied intrusion detection using traffic predictors. The tests confirm our algorithm's ability to identify distributed denial-of-service (DDoS) assaults. The suggested model combined machine learning and intrusion detection systems to increase the precision of green IoT intrusion detection.

Using deep learning for anomaly-based intrusion detection systems (IDSs) to secure IoT environments, a systematic literature review by Alsoufi et al. was developed [50]. It talks about supervised versus unsupervised learning, how effective deep learning approaches are, and offers insights into the examination of previous research in this field.

In Alrawashdeh et al. [66], deep belief networks and the restricted Boltzmann machine (RBM) are the main topics of discussion while using a deep learning strategy for anomaly detection. The DARPA KDD-Cup'99 dataset will be used to evaluate the architecture, performance, and potential options for expanding the approach's application to bigger datasets.

3.3. Outlier Detection

In their article [67], Jabez and Muthukumar described how they used outlier detection to find anomalies in an IDS. They explained the methodology, which includes the use of the neighborhood outlier factor (NOF), and showcased the experimental findings that demonstrate how successful the suggested strategy is.

3.3.1. KNN

A suggested anomaly-based intrusion detection system (IDS) by Chordia and Gupta [68] uses data mining approaches to lower false alarm rates and improve detection efficiency. With a focus on U2R, R2R, DoS, and probe attacks, the suggested system employs techniques like K-NN, K-means, and decision table majority rule-based methodology to monitor network traffic. The authors evaluated the effectiveness of the approach using the KDD 99

dataset, which emphasizes the lack of security event data from an IoT system to support a more thorough and balanced assessment of IDS systems for IoT.

3.3.2. Naive-Bayes

In comparison to an analogous software (SW) version, Viegas et al. [5] showed that a hardware (HW) implementation of network security algorithms can drastically reduce their energy usage. They built an anomaly-based network intrusion detection system (NIDS) using three machine learning (ML) classifiers implemented in SW and HW-decision tree (DT), Naive-Bayes (NB), and k-nearest neighbors (kNN). They suggested a new feature extraction approach with minimal processing needs and hardware implementation compatibility. The new feature extractor used a lot less memory, electricity, and processing power. Its HW implementation used only 12% and its SW implementation only 22% of the energy of a commercial device. Energy savings of up to 93% were made possible by dual-objective feature selection.

3.3.3. Statistical Model

According to Riecker et al. [69], a system that is energy-efficient and lightweight utilizes mobile agents to identify intrusions by measuring the energy usage of sensor nodes. An energy consumption prediction model based on linear regression was utilized. According to simulation studies, flooding and other denial-of-service attacks can be identified with a high degree of precision and a very low rate of false positives.

3.3.4. New Anonymous Detection Model

Since sensor networks are constructed for different intruders in different scenarios, Chen et al. [70] examined the detection probability of an arbitrary path across the barrier of sensors theoretically and took into account the maximum speed of conceivable intruders. They provided an energy-efficient scheduling problem for sensors by formulating a minimum weight ϵ-barrier problem based on the theoretical study of detection probability. In order to schedule the activation of sensors, they demonstrated that the problem is NP-hard and suggested a bounded approximation approach known as the minimum weight barrier approach (MWBA). In order to assess our design, they run comprehensive simulations to show the efficacy of our suggested algorithm in addition to conducting a theoretical analysis of MWBA performance.

The intelligent intrusion detection system Passban IDS [71] is intended for Internet of Things edge devices. It demonstrates how important it is to secure Internet of Things devices, how to use Passban on low-cost IoT gateways, and how it can identify different kinds of malicious traffic with low false-positive rates.

A low-complexity, energy-conscious approach for intrusion detection in wireless sensor networks was proposed by Misra et al. [72]. The protocol includes distributed and self-learning. When one node is compromised, the dispersed nature prevents all other nodes from being sacrificed. The protocol aims to create an intrusion detection system that is mindful of energy by contrasting the idea of stochastic learning automata with the packet sampling mechanism.

Border intrusion detection was proposed [73] by Yang et al. It provides an energy-efficient way for border patrol to increase detection accuracy while lowering the heavy human engagement. In addition, they created a brand new coverage model to identify one-way paths. According to the simulation results, the new coverage model has the ability to efficiently extend the network life and identify intrusions in border areas.

3.3.5. Signature-Based Intrusion Detection Techniques

Due to its open source nature and widespread use in the intrusion detection and prevention space, Snort is mostly utilized as a signature-based intrusion detection system. Snort warnings are also viewed using the Basic Analysis and Security Engine (BASE) [74,75].

According to Nattawat et al. [76], the Snort-IDS rules for the detection of network probe attacks can be improved. Another signature-based network intrusion prevention (NIPS) and network security (NIDS) engine is called Suricata. Its purpose is to monitor network traffic for potentially dangerous activity and suspicious activity, giving businesses the ability to strengthen their cybersecurity defenses. It can decode a wide range of protocols, examine network packets at different layers, and use both signature-based and anomaly-based detection methods to find anomalies. The processing and detection rates of Snort and Suricata were examined and contrasted by Wonhyung Park and Seongjin Ahn [77] in order to debate which is superior in environments with a single thread or many threads.

An automated machine learning architecture for Internet of Things (IoT)-enabled smart energy grids that can determine whether to develop rules for signature-based systems was proposed by Yadav et al. [78]. The framework's potential for intelligent threat mitigation in smart energy infrastructures was demonstrated by the results, which were obtained using an IoT dataset that included MITM (man in the middle) assaults.

A dynamic coding approach was presented by Amin et al. [79] to assist in the implementation of an intrusion detection system (IDS) based on distributed signatures in IP-USNs (IP-based ubiquitous sensor networks). The suggested plan is suitable for resource-constrained sensor devices as it allows the construction of lightweight IDSs in terms of messaging, storage, and energy usage.

The focus of Bostani and Sheikhan [80] was on a brand-new real-time hybrid intrusion detection system suggested for the Internet of Things. It highlights the deployment of anomaly-based and specification-based intrusion detection modules in Internet of Things situations, as well as the performance evaluation of the suggested framework.

3.4. Hybrid of Signature-Based IDS and Anomaly IDS

In their paper [81], Echateerawat et al. compared various methods for detecting intrusions in sensor networks. We examined the relationship between energy efficiency and assault detection accuracy. They proposed that the greatest features might be combined by creating a hybrid system that combines anomaly and signature IDSs.

The architecture of the hybrid intrusion detection system (eHIDS) for wireless sensor networks was proposed by Abduvaliyev et al. [82]. They used a combination of anomaly and signature-based detection techniques to create a hybrid scheme. In addition, they employed cluster-based wireless sensor networks to reduce the cost of computing and communication. They simulated the network and compared the performance of our scheme with that of other similar methods. The technique outperformed other schemes in terms of the high detection rate and energy efficiency, according to the simulation findings.

Tama et al. [83] and Rizzardi et al. [84] presented an enhanced IDS that uses two-level classifier ensembles and hybrid feature selection. The technique, dataset performance evaluation, and the importance of statistical significance tests in confirming the findings were covered.

An analysis of the suggested anomaly detection technique for supervisory control and data acquisition (SCADA) systems was provided by Bostani and Sheikhan [80]. Preprocessing methods, dimensionality reduction algorithms, dataset balance, and experimental findings demonstrating the effectiveness of the suggested strategy were all covered.

Table 2. Summary of IDS deployment and techniques

		IoT IDS	Host-Based IDS	Network-Based IDS	Cloud-Based IDS
Signature-based IDS Technique		Yadav, N. et al. [78]	Liu, M. et al. [60], Murtaza, S.S et al. [62]	Kumar, V. et al. [75], Kurundkar, G. et al. [74], Khamphakdee, N. et al. [76], Park, W. et al. [77]	
Anomaly IDS Techniques	**Deep Learning**	Oh, D. et al. [50], Nie, L. et al. [65]		Shone, N. et al. [57], Javaid, A. et al. [58], Alrawashdeh, K. et al. [66]	
	Outlier Detection KNN	Gupta, S. et al. [68]		Jabez, J. et al. [67]	
	Naive-Bayes			Viegas, E. et al. [5]	
	Supervised Statistical Model	Riecker, M. et al. [69]	Vinayakumar, R. et al. [59]		
	New Anonymous Detection	Chen, J. et al. [70], Eskandari, M. et al. [71], Misra, S. et al. [72], Yang, T. et al. [73]	Ou, C.M. [61]		
Hybrid of Signature & Anomaly IDS		Yadav, N. et al. [78], Bostani, H. et al. [80], Techateerawat, P. et al. [81], Abduvaliyev, A. et al. [82]		Tama, B.A. et al. [83]	Google Cloud IDS [63], Amazon GuardDuty [64]

4. Energy Optimization Techniques in Intrusion Detection Systems

In this section we briefly mention the techniques used in recent IDSs to optimize energy consumption. We emphasize energy optimization techniques over detection mechanisms in this section.

Migliardi et al. [85] suggested making an effort to evaluate the energy impact of security measures. Specifically, they offered a basic model for assessing the energy cost of distributed packet inspection in intrusion detection systems (IDSs) and demonstrated how to apply it to two example IDS tactics to assess energy leakage resulting from the late identification of rogue packets.

Arshad et al. [86] designed a framework called collaborative intrusion detection (COLIDE) for IoT networks. The framework specifically allows for the combined use of data from network-based detection systems and hosts. The end-host/node layer and the edge router layer make up the two tiers of the detection system. In order to correlate the warnings and carry out aggregate detection, the end-host component keeps an eye on events at the node level and communicates aberrant events to the network/edge router level system.

They claim that by coordinating alerts from many devices, the workload at the end host will be decreased in addition to minimizing false positive rates and enhancing detection rates under spread attacks. As a result, issues like flexibility, node resource limitations, and the collaborative character of IoT networks were anticipated to be addressed by the suggested framework.

In their experiments power measurements were performed with Contiki OS's power-trace utility [87]. Their simulation results indicate that a node needs approximately 5 mW of power to process 1000 packets, which is insignificant for the ultra-low-power Tmote sky [88] in terms of energy overheads.

Wang et al. [89] provided an attack–defense game model to identify malicious nodes using a repeating game technique, with an emphasis on intrusion detection approaches. To obtain the best payoffs in the suggested game model, attackers and defenders adopt various techniques.

Machine-to-machine (M2M) mobile networks must be extremely dependable since the devices with computational capabilities in them use data that have been acquired to

compute things that are physical and then provide the results to other devices. In order to guarantee that the system can function as intended, the defense system for sensor network (SN) security in M2M mobile networks must adapt its reactions to various attack vectors. In their work [89], they present the use of game theory as a tool for designing an attack–defense game model, with the goal of determining the optimal attack and defense strategies through repeated game methods.

A repeating game model is suggested as a solution to M2M mobile network intrusion detection issues. To help M2M mobile networks analyze and determine the best tactics for attackers and defenders, a game tree model is proposed. To assess how well their model performs in comparison to two other models (all monitor (AM) and cluster head (CH)) that are currently in use, simulation is carried out.

Their attack–defense game model, which is based on game theory, almost always uses less energy than the AM and CH models. In particular, when compared to the AM model, the game theory-based attack defensive game model can save up to 50% on energy usage.

Sedjelmaci et al. [90] suggested using game theory to activate anomaly detection methods solely in anticipation of a new attack's signature; this achieves a balance between energy consumption, false positive rates, and detection rates. According to simulation results, this lightweight anomaly detection method works better than existing anomaly detection techniques because it uses less energy in scaling mode (i.e., when there are a lot of IoT devices and attackers) to detect attacks with high detection and low false-positive rates. The energy efficiency is achieved by a need-based invoking of more energy consuming anomaly detection based on a game theoretic approach (over the lower energy consuming signature detection) in intrusion detection systems.

Raza et al. designed, built, and assessed SVELTE, a novel intrusion detection system for the Internet of Things, in their article [91]. They mainly focused on routing threats such spoofing or altering information, sinkholes, and selective forwarding in their installation and assessment. Their method can be expanded to identify other attacks. They integrated SVELTE into the Contiki OS and conducted a comprehensive assessment. According to their assessment, SVELTE is able to identify any malicious node that initiates its implemented sinkhole and/or selective forwarding assaults in simulated scenarios. Nevertheless, there are some false alarms generated when malicious nodes are detected, meaning that the genuine positive rate is not 100%. Furthermore, SVELTE may be deployed on constrained nodes with low energy and memory capacities due to its negligible overhead.

Earlier techniques for energy optimization in IDSs have involved cutting down on communication overheads [92–95], cutting down on computational overheads [96–100], or dividing up in-network tasks [101,102].

5. Taxonomy of Green IDS Techniques

We used a comprehensive set of criteria and metrics to carry out a complete and methodical review of the current literature, which is described in Sections 3 and 4. These metrics are important for efficient energy optimization in intrusion detection systems. The criteria's individual components are listed below, each with a brief explanation. Table 3 presents a comparative examination of current techniques for these criteria.

- **Architecture.** An intrusion detection system's architecture details how the detection system performs detection tasks. User privacy is impacted by the system's design in addition to performance and detection accuracy. Because the standalone detection system primarily functions on a local machine or device, it is vulnerable to longer detection times due to insufficient data availability and the stealthy character of the attacker. In contrast, a collaborative architecture makes use of data from several sources, such network devices or Internet of Things devices, whether they are part of the same company or not. It can increase the accuracy of detection, but it also raises concerns about the privacy of data shared between entities. In addition, an edge router that controls communication between the local network and the Internet and

a number of IoT devices arranged into a local network, e.g., 6LoWPAN, make up a typical IoT system in terms of energy optimization and detection accuracy.

- **Detection technique.** As mentioned in Sections 3 and 4, an IDS can make use of a range of detection methods, including anomaly, signature, and game-based techniques. The selection of a detection engine affects an intrusion detection system's (IDS) capacity to identify attacks as well as the energy consumption of the system. For example, while signature-based intrusion detection systems (IDS) have been found to be energy-efficient, they are unable to identify zero-day assaults. In order to achieve effective intrusion detection, an increasing amount of artificial intelligence and deep learning is being used according to an analysis of the literature that is presented in Sections 3 and 4 and summarized in Table 3.
- **Energy optimization technique.** The technique(s) used for energy optimization in the illustrated IDS are highlighted in this review. As mentioned before, the primary objective of our paper is to review the existing energy optimization techniques for IDSs and present our ideas on the potential application of other existing energy optimization techniques onto modern IDSs for further energy optimization.

Table 3. Energy optimization techniques for different IDSs (green column lists energy-optimization techniques).

IDS	Architecture	Energy Optimization Technique	Detection Technique	Other Features
Migliardi et al. [85]	Generic NIDS	Energy Leakage Reduction	Distributed Intrusion Detection Scheme	Novel energy modeling of DIDS
Arshad et. al [44]	6LoWPAN	Energy Optimized via Coordinating Alerts from many Devices	Collaborative Intrusion Detection	Can be applied to other routing protocols
Wang et al. [89]	Machine-to-Machine (M2M) mobile Networks	Game Tree Model for finding Optimal Strategies	Attack-defense Game model	Can reduce energy consumption by up to 50% in comparison to similar IDS
Sedjelmaci et al. [90]	Wireless Sensor Networks	Need-based invoking of Anomaly Detection	Game Theoretic Approach	Low false positive rates
Raza et al. [91]	6LoWPAN	Dependent over implementation on low-power networks	IDS integrated with Mini Firewall	True positive rate is not 100% (some false alarms present)
Viegas et al. [4]	Generic NIDS	Hardware (HW) Implementation of Network Security Algorithms	Anomaly based NIDS with Machine Learning Classifiers implemented in SW and HW-Decision Tree, Naive-Bayes and k-Nearest Neighbors.	SW implementation consumes only 22% of the energy used by a commercial product and its HW implementation only 12%.
Riecker et al. [69]	Generic NIDS	Linear Regression model is Applied to Predict the Energy Consumption	Mobile Agents to Detect Intrusions based on Energy Consumption of Sensor Nodes	Denial-of-Service attacks detected with high accuracy, and low false positives

6. Future Research Directions

Table 3 reveals the following: (1) not many works exist in the literature that specifically treat energy as a first-class parameter to optimize IDSs; (2) most of energy-optimization techniques have been achieved as a byproduct of making detection more efficient in IDSs.

Specifically, to our knowledge, no work has been carried out to *engineer existing IDS algorithms* to optimize energy consumption in them (e.g., applying the ECM of [31]). In Table 4, we provide a mapping of *potential* energy optimization techniques from Section 2 on the IDS systems of Section 3.

Table 4. Potential energy optimization techniques for existing IDS (green column lists energy-optimization techniques).

IDS	Architecture	Potential Energy Optimization Technique	Detection Technique
Shone et al. [57]	GPU-enabled Tensorflow	Energy Complexity Model (ECM) [31]	Deep Learning Classification Model
Niyaz et al. [58]	GPU-enabled Tensorflow	Energy Complexity Model (ECM) [31]	Self-taught Learning (Deep Learning based Approach)
R Vinayakumar et al. [59]	Generic NIDS	Reducing Operating Frequencies and Balancing Workloads [15–19]	Supervised Learning Methods (e.g. CNN)
Ou Chung-Ming [61]	Host Based	Software Defined Network (SDN) [27]	Agent-based Artificial Immune System
Syed et al. [62]	Host Based	Software Defined Network (SDN) [27]	Semantic Interactions of System Calls
Google Cloud Intrusion Detection System [63]	Cloud IDS	Power-Aware Scheduling Algorithms [8–13]	Signature-based Detection
Amazon Web Services (AWS) GuardDuty [64]	Cloud IDS	Power-Aware Scheduling Algorithms [8–13]	Machine Learning, Anomaly Detection
Chen et al. [70]	Network Sensors	Lightweight Blockchain Based Framework for Networks [35]	Minimum Weight Barrier Algorithm (MWBA)
Chordia and Gupta [68]	Generic NIDS	Energy Efficient Wireless Control Systems [89]	Anomaly-Based IDS
Jabez and Muthukumar [67]	Generic NIDS	EP-CuMAC for Narrowband Internet of Things [36]	Neighborhood Outlier Factor

As illustrated in Table 4, potential energy-optimization techniques have been identified that can be applied to the modern IDSs. This table was created to serve as a thinking point for future researchers. In the remainder of the section, we provide our thoughts on how existing energy-optimization techniques potentially apply on the modern IDSs.

6.1. Energy Complexity Model (ECM) [31]

As mentioned in Section 2, the ECM proposed by [31] optimizes energy consumption in algorithms by engineering them to ensure parallel memory bank accesses. In theory, with a P bank DDR3 architecture with B bytes per chunk, the energy consumed by an algorithm $\mathcal{A}$ with the execution time τ is given by a [31]-derived formula:

$$E(\mathcal{A}) = \tau + (P \times B)/I. \quad (4)$$

For each P block access made overall, the so-called *parallelization index*, denoted by I, is essentially the number of parallel block accesses performed by $\mathcal{A}$ across various memory banks. According to ECM, an algorithm's ability to reduce energy consumption is inversely correlated with how well it can be designed to parallelize memory access. Furthermore, algorithms that process data in *blocks* have more potential to be engineered using ECM for energy optimization. These algorithms are called *block structured* in [31].

Shone et al. [57] and Niyaz et al. [58] used deep learning algorithms for intrusion detection, which are block structured by design. One of the central algorithms, the auto-encoder, accepts inputs in a natural block-structured form on which the ECM [31] can be potentially applied.

6.2. Reducing Operating Frequencies and Balancing Workloads

Research on energy-efficient mobile computing focuses on two strategies: lowering operating frequencies and balancing workloads on heterogeneous multicore architectures [15,16,18,19]. The techniques based on supervised learning methods (e.g., [59]), that in order to train supervised learning techniques such as multilayer perceptron (MLP), CNN, CNN-recurrent neural network (CNN-RNN), CNN-long short-term memory (CNN-LSTM),

and CNN-gated recurrent unit (GRU) to model network traffic as time series, specifically transmission control protocol/internet protocol (TCP/IP) packets within a predetermined time range, use millions of known good and bad network connections.

6.3. Software-Defined Network (SDN)

By relocating fog nodes and creating fewer fog servers, the fog layer seeks to minimize the number of active fog servers [27]. Given a fog node placement matrix A, let it be described by

$$A_{ij} = \begin{cases} 1 & \texttt{if fog node j is placed on fog server i,} \\ 0 & \texttt{otherwise.} \end{cases}$$

The demand of fog node j' for the accessible resource type r is represented by $N_{\mathrm{j}}^{\mathrm{r}}$, while $F_{\mathrm{i}}^{\mathrm{r}}$ represents the resource type r currently available on fog server i. Let FS_{i} be a binary variable that takes on the value 1 while the fog server i is active and 0 otherwise.

$$\texttt{Minimize} \sum_{i} FS_{\mathrm{i}}, \tag{5}$$

$$\texttt{Subject to} \sum_{j=1} N_{\mathrm{j}}^{\mathrm{r}} \times A_{\mathrm{ij}} \leq F_{\mathrm{i}}^{\mathrm{r}} \ \forall i, r \tag{6}$$

$$\sum_{i} A_{ij} = 1 \ \forall j, \tag{7}$$

$$FS_{\mathrm{i}} \geq A_{ij} = 1 \ \forall i, j, \tag{8}$$

where $1 \leq i \leq |F|$, $1 \leq j \leq |N|$, and $r \in \Re$.

The number of fog servers that are turned on is minimized by the objective function (5). The second limitation (6) states that the total resource demands installed by the fog nodes on a particular fog server cannot exceed the fog server's capability. Every fog node must be installed on precisely one fog server according to Constraint (7). Equation (8) uses the variables FS_i and A_{ij} to track whether a fog server is in operation and turns it on or off.

Ou Chung-Ming [61] designed a host IDS (HIDS) based on an agent-based artificial immune system (ABAIS), having multiple embedded agents working together to update the activation threshold for security responses and compute the mature context antigen value. The ABAIS HIDS algorithm works based on an *antigen–signal* pair, solving a similar objective function described below.

If the distance between an antigen–signal pair (A, S) and its threshold vector $(T_h(I)$ of I is equal to ε, the computer host I is said to have experienced an ε-intrusion, $d(TP(S), T_h(I)) = \varepsilon$. The following is the definition of *distance* (d):

$$d(x, y) = \sum_{i=1}^{3} max(x_i - y_i, 0), \tag{9}$$

where $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3)$.

It is critical to define a valid value ε in order for ε-intrusion to qualify as a dangerous attack type. Reducing false-positives and false-negatives of IDSs is also indicated by the correct value for ε.

The similarities in objective functions in Equations (5) and (9) lead us to believe that the SDN-based energy optimization techniques in [27] can be applied to the ABAIS-based HIDS in [61]. This has been listed in Table 4. Syed et al.'s [62] kernel states modeling (KSM) approach exhibits a similar potential application of [27] to reduce energy consumption.

6.4. Power-Aware Scheduling Algorithms

Power-aware scheduling methods use power consumption profiles to optimize virtual machine execution. Hypervisors can reduce overall power consumption by efficiently allo-

cating workloads to complementary patterns of resource utilization. Concurrent scheduling of memory- and CPU-intensive tasks, for example, can lead to more equitable and efficient resource consumption. The aforementioned techniques underscore the significance of power-aware scheduling algorithms in cloud computing systems that exhibit significant workload unpredictability [8–13]. The objective is to minimize energy usage and maintain service-level agreements (SLAs) while simultaneously reducing system response times. Therefore, it will be interesting to see how these techniques apply to reduce energy consumption in cloud-based IDSs (e.g., Google [63] and Amazon [64]).

6.5. A Lightweight Blockchain-Based Framework for Networks

Since the sensor networks are built for various intruders in various conditions, Chen et al. [70] considered the maximum speed of potential intruders and theoretically analyzed the detection probability of an arbitrary path across the sensor barrier.

Mohan et al. [35] created a thin, blockchain-based framework for the Narrowband Internet of Things. Its characteristics include linear hash chain-based storage to prevent expensive Merkle tree verification and dynamic base station memory partitioning to improve memory use efficiency and scalability. The lightweight nature of the suggested framework in comparison to existing techniques was discovered through a study.

Mohan et al.'s [35] work was specifically targeted towards energy optimization in sensor networks. Therefore, Chen et al.'s [70] IDS presents the potential for the application of Mohan et al.'s techniques for further energy optimization.

7. Limitations of This Review

While this review provides a comprehensive analysis of green intrusion detection systems (GIDS) and their application in various critical sectors, several limitations should be acknowledged.

Firstly, we did not rank the quality of the literature reviewed nor discuss the funding sources used in the studies. Although we provided four taxonomy tables summarizing the relevant research in different categories, we plan to include a ranking and discussion of the ranking criteria in future studies to enhance the robustness and reliability of our findings.

Secondly, due to paper length restrictions, we were unable to include discussions, tables, or charts detailing the studies included in the review regarding sample sizes, populations studied, time frames of studies, missing data, limitations, outcome measures, and final results. Instead, our tables focus on author information, interventions, IDS techniques used, algorithms, and architectures of IDS systems. The absence of these detailed summaries may limit the ability to fully assess the comparative strengths and weaknesses of the studies reviewed.

Furthermore, while we have thoroughly discussed the sustainability aspects and effectiveness of GIDSs, the exclusion of detailed rankings and funding source discussions could limit the contextual understanding of the impact and potential biases of the reviewed studies.

In summary, despite these limitations, this review offers significant insights into the development and implementation of sustainable intrusion detection systems. We are confident that future research will address these gaps by incorporating detailed rankings, funding source analyses, and comprehensive study summaries to provide a more thorough evaluation of the GIDS literature.

8. Conclusions

In this paper, we conducted a comprehensive review of energy-optimization techniques applied to modern intrusion detection systems (IDS). The proliferation of Internet of Things (IoT) systems has significantly increased the volume and variety of security risks, highlighting the necessity of effective IDSs. Given the high energy consumption associated with most IDSs, optimizing energy usage has become a critical consideration. Our review examined state-of-the-art energy optimization strategies for contemporary IDSs, identifying potential applications of general energy-optimization approaches. Con-

sequently, this study outlines future research directions for developing energy-efficient IDSs, particularly in the context of deep learning-based systems. Therefore, we believe that the work has laid out future research routes to build energy-efficient intrusion detection systems. Of special interest will be performing a similar energy-optimization analysis on generic deep learning-based systems (e.g., [103]).

Author Contributions: Conceptualization, S.R.; Methodology, S.R., S.S. and M.Z.; Formal analysis, S.S. and M.Z.; Writing—original draft, S.R., S.S. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research (and APC) was funded by Fidelity National Financial Distinguished Professorship Grant in CIS of S. Roy. (Grant #: 0583-5504-51).

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

References

1. 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]
2. Ghosal, A.; Halder, S. A survey on energy efficient intrusion detection in wireless sensor networks. *J. Ambient Intell. Smart Environ.* **2017**, *9*, 239–261. [CrossRef]
3. França, A.L.P.d.; Jasinski, R.P.; Pedroni, V.A.; Santin, A.O. Moving Network Protection from Software to Hardware: An Energy Efficiency Analysis. In Proceedings of the 2014 IEEE Computer Society Annual Symposium on VLSI, Tampa, FL, USA, 9–11 July 2014; pp. 456–461. [CrossRef]
4. Viegas, E.; Santin, A.; Oliveira, L.; Franca, A.; Jasinski, R.; Pedroni, V. A reliable and energy-efficient classifier combination scheme for intrusion detection in embedded systems. *Comput. Secur.* **2018**, *78*, 16–32. [CrossRef]
5. Viegas, E.; Santin, A.O.; Franca, A.; Jasinski, R.; Pedroni, V.A.; Oliveira, L.S. Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems. *IEEE Trans. Comput.* **2016**, *66*, 163–177. [CrossRef]
6. Maciel, L.A.; Souza, M.A.; Freitas, H.C. Energy-Efficient CPU+ FPGA-based CNN Architecture for Intrusion Detection Systems. *IEEE Consum. Electron. Mag.* **2023**, *13*, 65–72. [CrossRef]
7. Khandelwal, S.; Shreejith, S. A lightweight multi-attack CAN intrusion detection system on hybrid FPGAs. In Proceedings of the 2022 32nd International Conference on Field-Programmable Logic and Applications (FPL), Belfast, UK, 29 August–2 September 2022; IEEE: New York, NY, USA, 2022; pp. 425–429.
8. ho Seo, J.; Tchamgoue, G.M.; Kim, K.H. Power-Aware Real-Time Virtual Machine Schedulers in Discrete DVFS Systems. In Proceedings of the 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, Dalian, China, 24–27 August 2014; IEEE: New York, NY, USA, 2014; pp. 459–463.
9. Seo, J.; Kim, K.H. A Prototype of Online Dynamic Scaling Scheduler for Real-Time Task based on Virtual Machine. *Int. J. Electr. Comput. Eng.* **2016**, *6*, 205.
10. Lin, C.C.; Chang, C.J.; Syu, Y.C.; Wu, J.J.; Liu, P.; Cheng, P.W.; Hsu, W.T. An energy-efficient hypervisor scheduler for asymmetric multi-core. In Proceedings of the 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE), Tokyo, Japan, 7–10 October 2014; IEEE: New York, NY, USA, 2014; pp. 507–509.
11. Seol, Y.I.; Kim, J.U.; Kim, Y.K. A power-aware scheduler exploiting all slacks under EDF scheduling. In *Advances in Computer Science and Its Applications (CSA) 2013*; Jeong, H., S. Obaidat, M., Yen, N., Park, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 51–57.
12. Poovizhi, J.M.R.; Devi, R. Performance Analysis of Cloud Hypervisor using Different Workloads in Virtualization. In Proceedings of the 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 16–17 December 2022; IEEE: New York, NY, USA, 2022; pp. 414–418.
13. Vilaplana, J.; Solsona, F.; Teixido, I.; Mateo, J.; Rius, J.; Abella, F. A Green Scheduling Policy for Cloud Computing. In *Adaptive Resource Management and Scheduling for Cloud Computing: Proceedings of the First International Workshop, ARMS-CC 2014, held in Conjunction with ACM Symposium on Principles of Distributed Computing, PODC 2014, Paris, France, 15 July 2014, Revised Selected Papers 1*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 26–35.
14. Ko, W.; Yoo, J.; Kang, I.; Jun, J.; Lim, S.S. Lightweight, predictable hypervisor for ARM-Based embedded systems. In Proceedings of the 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Daegu, Republic of Korea, 17–19 August 2016; IEEE: New York, NY, USA, 2016; p. 109.
15. Moratelli, C.; Filho, S.; Hessel, F. Hardware-assisted interrupt delivery optimization for virtualized embedded platforms. In Proceedings of the 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Cairo, Egypt, 6–9 December 2015; IEEE: New York, NY, USA, 2015; pp. 304–307.
16. Jiang, Z.; Audsley, N.C.; Dong, P. Bluevisor: A scalable real-time hardware hypervisor for many-core embedded systems. In Proceedings of the 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Porto, Portugal, 11–13 April 2018; IEEE: New York, NY, USA, 2018; pp. 75–84.

17. Randhawa, R.H.; Ahmed, A.; Siddiqui, M.I. Power Management Techniques in Popular Operating Systems for IoT Devices. In Proceedings of the 2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 17–19 December 2018; pp. 309–314. [CrossRef]
18. Singh, D.; Kaiser, W.J. Energy efficient task scheduling on a multi-core platform using real-time energy measurements. In Proceedings of the 2014 International Symposium on Low Power Electronics and Design, La Jolla, CA, USA, 11–13 August 2014; pp. 271–274.
19. El Sayed, M.A.; Saad, E.S.M.; Aly, R.F.; Habashy, S.M. Energy-efficient task partitioning for real-time scheduling on multi-core platforms. *Computers* **2021**, *10*, 10. [CrossRef]
20. Stokke, K.R.; Stensland, H.K.; Griwodz, C.; Halvorsen, P. Load balancing of multimedia workloads for energy efficiency on the tegra k1 multicore architecture. In Proceedings of the 8th ACM on Multimedia Systems Conference, Taipei, Taiwan, 20–23 June 2017; pp. 124–135.
21. Trubaev, P.; Bulanin, A.; Shirrime, K.; Koshlich, Y.A. Particular Qualities of Execution of the Program Component for Energy Resources Management of the Belgorod Region. *J. Phys. Conf. Ser.* **2018**, *1066*, 012020. [CrossRef]
22. Nogues, E.; Pelcat, M.; Menard, D.; Mercat, A. Energy efficient scheduling of real time signal processing applications through combined DVFS and DPM. In Proceedings of the 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), Heraklion, Greece, 17–19 February 2016; IEEE: New York, NY, USA, 2016; pp. 622–626.
23. Eibel, C.; Hönig, T.; Schröder-Preikschat, W. Energy Claims at Scale: Decreasing the Energy Demand of HPC Workloads at OS Level. In Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Chicago, IL, USA, 23–27 May 2016; IEEE: New York, NY, USA, 2016; pp. 1114–1117.
24. Fan, K.; D'Antonio, M.; Carpentieri, L.; Cosenza, B.; Ficarelli, F.; Cesarini, D. SYnergy: Fine-grained Energy-Efficient Heterogeneous Computing for Scalable Energy Saving. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 12–17 November 2023; pp. 1–13.
25. Kumar, N.; Rodrigues, J.J.; Guizani, M.; Choo, K.K.R.; Lu, R.; Verikoukis, C.; Zhong, Z. Achieving energy efficiency and sustainability in edge/fog deployment. *IEEE Commun. Mag.* **2018**, *56*, 20–21. [CrossRef]
26. Alharbi, H.A.; Elgorashi, T.E.; Elmirghani, J.M. Energy efficient virtual machines placement over cloud-fog network architecture. *IEEE Access* **2020**, *8*, 94697–94718. [CrossRef]
27. Ahmad, M.A.; Patra, S.S.; Barik, R.K. Energy-efficient resource scheduling in fog computing using SDN framework. In *Progress in Computing, Analytics and Networking: Proceedings of the ICCAN 2019, Bhubaneswar, Odisha, India, 14–15 December 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 567–578.
28. Bozorgchenani, A.; Disabato, S.; Tarchi, D.; Roveri, M. An energy harvesting solution for computation offloading in Fog Computing networks. *Comput. Commun.* **2020**, *160*, 577–587. [CrossRef]
29. Tripathy, N.; Sahoo, S. Energy Aware Effective Task Offloading Mechanism in Fog Computing. In Proceedings of the International Conference on Computing, Communication and Learning, Warangal, India, 29–31 August 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 272–284.
30. Chang, P.; Miao, G. Resource provision for energy-efficient mobile edge computing systems. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; IEEE: New York, NY, USA, 2018; pp. 1–6.
31. Roy, S.; Rudra, A.; Verma, A. An energy complexity model for algorithms. In Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, Berkeley, CA, USA, 9–12 January 2013; pp. 283–304.
32. Castellon, C.E.; Khatib, T.; Roy, S.; Dutta, A.; Kreidl, O.P.; Bölöni, L. Energy-Efficient Blockchain-Enabled Multi-Robot Coordination for Information Gathering: Theory and Experiments. *Electronics* **2023**, *12*, 4239. [CrossRef]
33. Nimmy, K.; Sankaran, S.; Achuthan, K.; Calyam, P. Lightweight and privacy-preserving remote user authentication for smart homes. *IEEE Access* **2021**, *10*, 176–190. [CrossRef]
34. Nimmy, K.; Dilraj, M.; Sankaran, S.; Achuthan, K. Leveraging power consumption for anomaly detection on IoT devices in smart homes. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 14045–14056. [CrossRef]
35. Mohan, V.S.; Sankaran, S.; Nanda, P.; Achuthan, K. Enabling secure lightweight mobile Narrowband Internet of Things (NB-IoT) applications using blockchain. *J. Netw. Comput. Appl.* **2023**, *219*, 103723. [CrossRef]
36. Mohan, V.S.; Sankaran, S.; Kumar, V.; Achuthan, K. EP-CuMAC: Energy and performance-efficient integrity protection for narrow-band IoT. *Internet Things* **2024**, *25*, 101004. [CrossRef]
37. Bhat, S.K.; Saya, A.; Rawat, H.K.; Barbalace, A.; Ravindran, B. Harnessing energy efficiency of heterogeneous-isa platforms. *ACM SIGOPS Oper. Syst. Rev.* **2016**, *49*, 65–69. [CrossRef]
38. Escobar, C.C.; Roy, S.; Kreidl, O.P.; Dutta, A.; Bölöni, L. Toward a green blockchain: Engineering merkle tree and proof of work for energy optimization. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 3847–3857. [CrossRef]
39. Castellon, C.E.; Roy, S.; Kreidl, O.P.; Dutta, A.; Bölöni, L. Towards an energy-efficient hash-based message authentication code (HMAC). In Proceedings of the 2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC), Pittsburgh, PA, USA, 24–25 October 2022; IEEE: New York, NY, USA, 2022; pp. 1–7.
40. Castellon, C.; Roy, S.; Kreidl, P.; Dutta, A.; Bölöni, L. Energy efficient merkle trees for blockchains. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 20–22 October 2021; IEEE: New York, NY, USA, 2021; pp. 1093–1099.

41. Harish, P.D.; Roy, S. Towards Designing Greener Secured Hash Functions. In Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), Taipei, Taiwan, 1–3 September 2014; IEEE: New York, NY, USA; 2014, pp. 618–621.
42. Roy, S.; Ahuja, S.P.; Harish, P.D.; Talluri, S.R. Energy optimization in cryptographic protocols for the cloud. In *Applications of Security, Mobile, Analytic, and Cloud (SMAC) Technologies for Effective Information Processing and Management*; IGI Global: Hershey, PA, USA, 2018; pp. 24–48.
43. Dunkels, A.; Gronvall, B.; Voigt, T. Contiki-a lightweight and flexible operating system for tiny networked sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 16–18 November 2004; IEEE: New York, NY, USA, 2004; pp. 455–462.
44. Arshad, J.; Azad, M.A.; Abdeltaif, M.M.; Salah, K. An intrusion detection framework for energy constrained IoT devices. *Mech. Syst. Signal Process.* **2020**, *136*, 106436. [CrossRef]
45. Nobakht, M.; Sivaraman, V.; Boreli, R. A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 31 August–2 September 2016; IEEE: New York, NY, USA, 2016; pp. 147–156.
46. Ghosh, K.; Neogy, S.; Das, P.K.; Mehta, M. Intrusion detection at international borders and large military barracks with multi-sink wireless sensor networks: An energy efficient solution. *Wirel. Pers. Commun.* **2018**, *98*, 1083–1101. [CrossRef]
47. Muradore, R.; Quaglia, D. Energy-efficient intrusion detection and mitigation for networked control systems security. *IEEE Trans. Ind. Informatics* **2015**, *11*, 830–840. [CrossRef]
48. Choudhary, S.; Kesswani, N. A survey: Intrusion detection techniques for internet of things. *Int. J. Inf. Secur. Priv. (IJISP)* **2019**, *13*, 86–105. [CrossRef]
49. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; De Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
50. Oh, D.; Kim, D.; Ro, W.W. A malicious pattern detection engine for embedded security systems in the Internet of Things. *Sensors* **2014**, *14*, 24188–24211. [CrossRef]
51. Lee, T.H.; Wen, C.H.; Chang, L.H.; Chiang, H.S.; Hsieh, M.C. A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN. In *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing: HumanCom and EMC 2013*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1205–1213.
52. Wallgren, L.; Raza, S.; Voigt, T. Routing attacks and countermeasures in the RPL-based internet of things. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 794326. [CrossRef]
53. Kasinathan, P.; Pastrone, C.; Spirito, M.A.; Vinkovits, M. Denial-of-Service detection in 6LoWPAN based Internet of Things. In Proceedings of the 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) , Lyon, France, 7–9 October 2013; IEEE: New York, NY, USA, 2013; pp. 600–607.
54. Le, A.; Loo, J.; Chai, K.K.; Aiash, M. A specification-based IDS for detecting attacks on RPL-based network topology. *Information* **2016**, *7*, 25. [CrossRef]
55. Thanigaivelan, N.K.; Nigussie, E.; Kanth, R.K.; Virtanen, S.; Isoaho, J. Distributed internal anomaly detection system for Internet-of-Things. In Proceedings of the 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2016; IEEE: New York, NY, USA, 2016; pp. 319–320.
56. Cervantes, C.; Poplade, D.; Nogueira, M.; Santos, A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; IEEE: New York, NY, USA, 2015; pp. 606–611.
57. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A deep learning approach to network intrusion detection. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 41–50. [CrossRef]
58. Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, 3–5 December 2016; pp. 21–26.
59. Vinayakumar, R.; Soman, K.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; IEEE: New York, NY, USA, 2017; pp. 1222–1228.
60. Liu, M.; Xue, Z.; Xu, X.; Zhong, C.; Chen, J. Host-based intrusion detection system with system calls: Review and future trends. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [CrossRef]
61. Ou, C.M. Host-based intrusion detection systems adapted from agent-based artificial immune systems. *Neurocomputing* **2012**, *88*, 78–86. [CrossRef]
62. Murtaza, S.S.; Khreich, W.; Hamou-Lhadj, A.; Couture, M. A host-based anomaly detection approach by representing system calls as states of kernel modules. In Proceedings of the 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE), Pasadena, CA, USA, 4–7 November 2013; IEEE: New York, NY, USA, 2013; pp. 431–440.

63. Cloud, G. Google Cloud IDS. 2024. Available online: https://cloud.google.com/security/products/intrusion-detection-system (accessed on 22 May 2024).
64. Services, A.W. Amazon GuardDuty. 2024. Available online: https://aws.amazon.com/guardduty/ (accessed on 22 May 2024).
65. Nie, L.; Sun, W.; Wang, S.; Ning, Z.; Rodrigues, J.J.; Wu, Y.; Li, S. Intrusion detection in green internet of things: A deep deterministic policy gradient-based algorithm. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 778–788. [CrossRef]
66. Alrawashdeh, K.; Purdy, C. Toward an online anomaly intrusion detection system based on deep learning. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; IEEE: New York, NY, USA, 2016; pp. 195–200.
67. Jabez, J.; Muthukumar, B. Intrusion Detection System (IDS): Anomaly detection using outlier detection approach. *Procedia Comput. Sci.* **2015**, *48*, 338–346. [CrossRef]
68. Anita S., C.; Gupta, S. An effective model for anomaly IDS to improve the efficiency. In Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India, 8–10 October 2015; IEEE: New York, NY, USA, 2015; pp. 190–194.
69. Riecker, M.; Biedermann, S.; Hollick, M. Lightweight energy consumption based intrusion detection system for wireless sensor networks. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; ACM: New York, NY, USA, 2013; pp. 1784–1791.
70. Chen, J.; Li, J.; Lai, T.H. Energy-efficient intrusion detection with a barrier of probabilistic sensors: Global and local. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4742–4755. [CrossRef]
71. 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]
72. Misra, S.; Krishna, P.V.; Abraham, K.I. Energy efficient learning solution for intrusion detection in wireless sensor networks. In Proceedings of the 2010 Second International Conference on COMmunication Systems and NETworks (COMSNETS 2010), Bangalore, India, 5–9 January 2010; IEEE: New York, NY, USA, 2010; pp. 1–6.
73. Yang, T.; Mu, D.; Hu, W.; Zhang, H. Energy-efficient border intrusion detection using wireless sensors network. *EURASIP J. Wirel. Commun. Netw.* **2014**, *2014*, 46. [CrossRef]
74. Kurundkar, G.; Naik, N.; Khamitkar, S. Network intrusion detection using Snort. *Int. J. Eng. Res. Appl.* **2012**, *2*, 1288–1296.
75. Kumar, V.; Sangwan, O.P. Signature based intrusion detection system using SNORT. *Int. J. Comput. Appl. Inf. Technol.* **2012**, *1*, 35–41.
76. Khamphakdee, N.; Benjamas, N.; Saiyod, S. Improving intrusion detection system based on snort rules for network probe attack detection. In Proceedings of the 2014 2nd International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 28–30 May 2014; IEEE: New York, NY, USA, 2014; pp. 69–74.
77. Park, W.; Ahn, S. Performance comparison and detection analysis in snort and suricata environment. *Wirel. Pers. Commun.* **2017**, *94*, 241–252. [CrossRef]
78. Yadav, N.; Truong, L.; Troja, E.; Aliasgari, M. Machine learning architecture for signature-based IoT intrusion detection in smart energy grids. In Proceedings of the 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy, 14–16 June 2022; IEEE: New York, NY, USA, 2022; pp. 671–676.
79. Amin, S.O.; Siddiqui, M.S.; Hong, C.S.; Choe, J. A novel coding scheme to implement signature based IDS in IP based Sensor Networks. In Proceedings of the 2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops, New York, NY, USA, 1–5 June 2009; IEEE: New York, NY, USA, 2009; pp. 269–274.
80. Bostani, H.; Sheikhan, M. Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. *Comput. Commun.* **2017**, *98*, 52–71. [CrossRef]
81. Techateerawat, P.; Jennings, A. Energy efficiency of intrusion detection systems in wireless sensor networks. In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, Hong Kong, China, 18–22 December 2006; IEEE: New York, NY, USA, 2006; pp. 227–230.
82. Abduvaliyev, A.; Lee, S.; Lee, Y.K. Energy efficient hybrid intrusion detection system for wireless sensor networks. In Proceedings of the 2010 International Conference on Electronics and Information Engineering, Kyoto, Japan, 1–3 August 2010; IEEE: New York, NY, USA, 2010; Volume 2, pp. V2–V25.
83. Tama, B.A.; Comuzzi, M.; Rhee, K.H. TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE Access* **2019**, *7*, 94497–94507. [CrossRef]
84. Cevallos M., J.F.; Rizzardi, A.; Sicari, S.; Porisini, A.C. Deep Reinforcement Learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges. *Comput. Netw.* **2023**, *236*, 110016.
85. Migliardi, M.; Merlo, A. Modeling the energy consumption of distributed IDS: A step towards Green security. In Proceedings of the 2011 Proceedings of the 34th International Convention MIPRO, Opatija, Croatia; IEEE: New York, NY, USA, 2011; pp. 1452–1457.
86. Arshad, J.; Azad, M.A.; Mahmoud Abdellatif, M.; Ur Rehman, M.H.; Salah, K. COLIDE: A collaborative intrusion detection framework for Internet of Things. *IET Netw.* **2019**, *8*, 3–14. [CrossRef]
87. Dunkels, A.; Osterlind, F.; Tsiftes, N.; He, Z. Software-based on-line energy estimation for sensor nodes. In Proceedings of the 4th Workshop on Embedded Networked Sensors, Cork, Ireland, 25–26 June 2007; pp. 28–32.

88. Amiri, M. Measurements of Energy Consumption and Execution Time of Different Operations on Tmote Sky Sensor Nodes. Ph.D. Thesis, Masarykova Univerzita, Fakulta Informatiky, Brno, Czech Republic, 2010.
89. Wang, K.; Du, M.; Yang, D.; Zhu, C.; Sun, Y. Optimal active detection in machine-to-machine mobile networks: A repeated game approach. In Proceedings of the 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 4–8 September 2016; IEEE: New York, NY, USA, 2016; pp. 1–6.
90. Sedjelmaci, H.; Senouci, S.M.; Taleb, T. An accurate security game for low-resource IoT devices. *IEEE Trans. Veh. Technol.* **2017**, *66*, 9381–9393. [CrossRef]
91. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
92. Agah, A.; Das, S.K.; Basu, K.; Asadi, M. Intrusion detection in sensor networks: A non-cooperative game approach. In Proceedings of the Third IEEE International Symposium on Network Computing and Applications, 2004 (NCA 2004), Cambridge, MA, USA, 1 September 2004; IEEE: New York, NY, USA, 2004; pp. 343–346.
93. Da Silva, A.P.R.; Martins, M.H.; Rocha, B.P.; Loureiro, A.A.; Ruiz, L.B.; Wong, H.C. Decentralized intrusion detection in wireless sensor networks. In Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks, Montreal, QC, Canada, 13 October 2005; pp. 16–23.
94. Hassanzadeh, A.; Stoleru, R. Towards optimal monitoring in cooperative ids for resource constrained wireless networks. In Proceedings of the 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), Lahaina, HI, USA, 31 July–4 August 2011; IEEE: New York, NY, USA, 2011; pp. 1–8.
95. Rajasegarar, S.; Leckie, C.; Palaniswami, M.; Bezdek, J.C. Quarter sphere based distributed anomaly detection in wireless sensor networks. In Proceedings of the 2007 IEEE International Conference on Communications, Glasgow, UK, 24–28 June 2007; IEEE: New York, NY, USA, 2007; pp. 3864–3869.
96. Chen, R.; Wang, Y.; Wang, D.C. Reliability of wireless sensors with code attestation for intrusion detection. *Inf. Process. Lett.* **2010**, *110*, 778–786. [CrossRef]
97. Hai, T.H.; Huh, E.N.; Jo, M. A lightweight intrusion detection framework for wireless sensor networks. *Wirel. Commun. Mob. Comput.* **2010**, *10*, 559–572. [CrossRef]
98. Ioannis, K.; Dimitriou, T.; Freiling, F.C. Towards intrusion detection in wireless sensor networks. In Proceedings of the 13th European Wireless Conference, Paris, France, 1–4 April 2007; pp. 1–10.
99. Li, G.; He, J.; Fu, Y. Group-based intrusion detection system in wireless sensor networks. *Comput. Commun.* **2008**, *31*, 4324–4332. [CrossRef]
100. Su, W.T.; Chang, K.M.; Kuo, Y.H. eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks. *Comput. Netw.* **2007**, *51*, 1151–1168. [CrossRef]
101. Livani, M.A.; Abadi, M. Distributed PCA-based anomaly detection in wireless sensor networks. In Proceedings of the 2010 International Conference for Internet Technology and Secured Transactions, London, UK, 8–11 November 2010; IEEE: New York, NY, USA, 2010; pp. 1–8.
102. Shin, S.; Kwon, T.; Jo, G.Y.; Park, Y.; Rhy, H. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. *IEEE Trans. Ind. Inform.* **2010**, *6*, 744–757. [CrossRef]
103. Tam, P.; Ros, S.; Song, I.; Kang, S.; Kim, S. A Survey of Intelligent End-to-End Networking Solutions: Integrating Graph Neural Networks and Deep Reinforcement Learning Approaches. *Electronics* **2024**, *13*, 994. [CrossRef]