




A Review of the Progressive Odyssey of AI-Driven Intrusion Detection Within Embedded Systems

Aisha Alansari, Razan Alfaqeer, and Mohammad Hammoudeh^(✉) 

Department of Information and Computer Science, King Fahd University of
Petroleum and Minerals, Dhahran 31261, Saudi Arabia
{aisha.ansari,g202203760,m.hammoudeh}@kfupm.edu.sa

Abstract. Security of Embedded Systems (ES) has become a major concern due to their growing usage in numerous industries. Their connectivity to the internet made them vulnerable to sophisticated cyber-attacks. One of the most important strategies for strengthening their security posture is using Intrusion Detection Systems (IDS). However, the limited resources of ES make it difficult to utilize IDS. This paper reviews the primary studies that contributed to developing IDS systems applicable to ES. It examines the challenges of building such systems, reports the current trends, and proposes future recommendations to enhance the deployment of IDS in ES. The findings showed that most studies currently employ machine and deep learning algorithms to build IDS for ES. Although significant results were achieved, several gaps were reported. The proposed frameworks did not investigate the security, privacy, and interpretability concerns of employing machine and deep learning. Moreover, a feasible framework to address all the ES resource constraints is lacking. Future recommendations include solutions to enhance such models' security, privacy, and interoperability. Moreover, it includes the employment of differential privacy, explainable artificial intelligence, federated learning, and trusted executed environments.

Keywords: Embedded systems · Intrusion detection · Machine learning

1 Introduction

In recent decades, numerous technological innovations have sought to improve life. A standout among these advancements involved incorporating computing operations into a larger physical system, enabling predefined functionality, named Embedded Systems (ES). ES is defined in many ways from various angles. Vahid and Givargis [1] noted that providing a unified definition of ES is challenging due to its wide applications and the variety of technologies underlying its implementation. However, they defined an ES as a computing system integrated within a larger physical system to perform a specific, necessary function. It comprises a mix of software, hardware, and sometimes mechanical components. As

as a result, any computing system other than mainframes or general-purpose PCs is covered by the phrase [2].

Recently, ES was significantly expanded to be used in various sectors, such as power plants, vehicles, and mobile phone systems. Therefore, the security of ES is becoming a major concern due to their extensive use and exposure. ES face several security challenges due to their physical accessibility and unique operating environments [3]. They are particularly susceptible to specialized attacks, such as side-channel attacks [4]. These attacks include advanced methods, such as time and power analysis, to compromise security keys [5]. Moreover, ES have recently become widely connected to the Internet by incorporating them with networked and interconnected devices. Consequently, their vulnerability to cyber-attacks has escalated more than ever [6].

Identifying and designing precise security improvements is challenging due to numerous influencing factors, such as these systems' resource constraints [7]. Security features often impact other system aspects, such as timing. Improper handling of these aspects can lead to breaches of non-functional requirements, risking system failure [8]. For instance, selecting an encryption technique that performs complex computations can strain memory and energy, risking system functionality [10]. This arises from the nature of non-functional requirements like security, which cannot be considered in isolation. Accordingly, these challenges necessitate careful consideration in enhancing ES' cybersecurity measures [9].

In real-time ES, carefully choosing security mechanisms is crucial, especially in light of their strict temporal constraints. These systems must ensure that security measures do not impair operating efficiency since they are frequently used when delays might have serious repercussions [10]. Given the limited resources that ES have and the need for real-time threat detection and response, integrating Intrusion Detection Systems (IDS) is considered a crucial tactic for strengthening their security posture. It helps in the early detection of cyber threats, allowing for timely mitigation actions.

Much research has contributed to the development of smart IDS, achieving better network security. The attempt to develop an intelligent IDS using a Deep Neural Network (DNN) and unsupervised Machine Learning (ML) algorithms has shown encouraging results [11]. Moreover, in other studies, researchers have tried to enhance the performance of IDS by combining learning, case-based thinking, and interactive behavior. However, considering the limited processing power and energy resources of ES, those IDS solutions must be lightweight yet effective [12].

Accordingly, this paper aims to review and discuss the findings of the primary studies that built IDS for ES. It identifies the current trends in developing IDS in ES and outlines their limitations. Moreover, it proposes future directions to enhance the application of IDS in ES.

The research questions that this paper aims to answer are as follows:

- What are the most used techniques in developing IDS in ES?
- What are the approaches followed to address the resource constraints of ES while deploying IDS?

- What are the limitations of applying IDS in ES?
- What could be the possible future work directions to enhance the deployment of IDS in ES?

The rest of this paper is organized as follows: Sect. 2 presents background information on IDS used in ES. Section 3 presents the conducted literature and Sect. 4 describes the methodology for collecting relevant primary studies. Moreover, Sect. 5 analyzes the topic in terms of the key challenges, trends, and critiques, Sect. 6 outlines future recommendations, and Sect. 7 concludes the paper.

2 Background

ES are exposed to several security challenges that can compromise their integrity, availability, and confidentiality. This is due to its resource constraints. Some of the restrictions on ES that lead to security issues include low processing power. This limitation stops the system from executing sophisticated security programs. Limited power supply availability is another barrier, leaving ES with few power resources that could lead to exhaustion attacks. Additionally, the nature of ES functioning in an unregulated or hostile environment leaves them open to physical attacks. In this case, attackers can physically access the system and interfere by breaking sensors or peripherals or eavesdropping on the system bus [13].

In ES, the processors cannot efficiently implement advanced security mechanisms, such as data encryption, or incorporate complex secure platforms [14]. Furthermore, the Central Processing Unit (CPU) does not have sufficient hardware protection against logical and physical attacks [15]. Advanced CPUs can help prevent or mitigate various attacks. Still, these advanced processors are more expensive, and their use is usually limited to smart device cards or dedicated secure elements in SoCs [17]. Even if we assume that the processor performance has been optimized according to advanced encryption requirements, portable systems may face challenges. The high energy consumption required for such security measures often exceeds these systems' limited power capabilities [16].

The National Institute of Standards and Technology (NIST) defines IDS as monitoring occasions that occur within a network or computer system and examining them to look for indications of potential invasions [18]. IDS can potentially contribute to protecting ES against several attacks. IDS are categorized based on how rapidly they can deploy and identify threats. They have been divided into two distinct categories for deployment: host-based and network-based. In order to gather data on the host device's activities and identify any unusual or malicious activity in that device, a host-based IDS is installed within a local device. In contrast, a network-based IDS monitors and finds threats in traffic using specific methodologies. IDS uses two distinct detection mechanisms: anomaly-based and signature-based. The signature-based IDS performs pattern matching using some techniques to detect any abnormal activity. It matches the activities across the network system with the predefined attack signatures stored

in the IDS database [19]. In other literature, the mechanisms are divided into four categories, namely anomaly detection, heuristic analysis, signature-based, rule-based, machine, and Deep Learning (DL) models.

To protect the system from malicious activities, monitoring procedures can be carried out using hardware or software. The use of the IDS security program comes after that of the firewall and antivirus software. As soon as a system malfunctions, the IDS modifies the network [21]. Along with detecting intrusions, the IDS records network activity within the system. When a host or network intrusion happens, the IDS notifies the system.

3 Review of Recent IDSs for ES

This section reviews the primary studies that developed IDS for ES. The review is arranged chronologically in each subsection according to the publication date. Table 1 summarizes the data collected from the reviewed primary studies.

3.1 Host-Based IDS

Kadar et al. [37] aimed to secure runtime program execution mix-critically ES by proposing a host-based IDS. The authors applied partitioning by using the Multiple Independent Layers of Security (MILS) architecture. Moreover, they defined the Host-based IDS service as a component of the separation kernel inside the architecture of a MILS system to separate it from possible adversaries. They first defined the way of executing the program formally. After that, they formulated the Host-based IDS around a substantial execution state description encompassing signals from software and hardware systems. In terms of recovery strategies, the proposed methodology is chosen from among four approaches based on trust and criticality levels. The approaches include the following: kill/reboot, isolate, migrate, and signal/log.

In another study, Liu et al. [35] aimed to address the scalability and efficiency shortcomings of host-based IDS. A two-tier host-based IDS framework was designed. The first tier is responsible for data collection, where an interface is installed in the embedded device. The second tier comprises a cloud-based analytics centre. Logistic Regression (LR) was trained to detect attacks, achieving an Area Under the Roc Curve (AUC) of 0.980 using 10-fold cross-validation. The results indicated that the proposed framework can automatically choose parameters that will lower latency and increase scalability. Although significant results were attained, the proposed framework was not tested on real-time streaming data from ES.

Martinez and Vogel-Heuser [32] proposed a host-based IDS for embedded industrial devices. The architecture design considered the capabilities and features of the industrial domain, as well as the system, device, and environmental characteristics. The authors evaluated their architecture by deploying it in a Programmable Logic Controller (PLC) hosting a Real-Time Operating System (RTOS). They claimed that their architecture is the first completely functional Host IDS installed and assessed on a PLC.

3.2 Network-Based IDS

Viegas et al. [39] introduced a feature selection strategy to determine the optimal trade-off between intrusion detection accuracy and system energy usage. With only a 0.9% decrease in accuracy, the authors' method was able to reduce energy use by up to 93%. In another study by Viegas et al. [38], the authors presented an anomaly-based technique for network intrusion detection in embedded devices. The suggested approach keeps the classifier reliable even in the event that the contents of network traffic fluctuate. Combining a few different classifiers with a novel rejection mechanism yields dependability. The suggested method is suitable for hardware implementation and energy efficiency. According to the results reported in this work, the feature extraction and packet capture modules utilize 58% and 37% of the energy consumed by their respective software counterparts, while the ML algorithms' hardware equivalents use 46% of that of their software counterparts.

Florencio et al. [34] employed Multilayer Perceptron (MLP) to build a real-time IDS. Experiments were conducted on an Arduino architecture to assess the MLP's performance. The results indicated a 95% confidence interval. Additionally, it was observed that there were no memory limitations, supporting the feasibility of an inexpensive IDS. However, the authors did not measure the energy used.

Furthermore, Khan et al. [36] proposed a unique framework to identify malicious behavior on ES using electromagnetic side-channel signals. The framework initially captures electromagnetic radiation from a secure reference device to create a pattern baseline. The framework then tracks the electromagnetic emissions from the target device. It records any deviation from the reference patterns as abnormal behavior. The target gadget is kept physically apart from the framework to avoid any adversarial attacks. The authors evaluated the performance of their framework by injecting ransomware, Distributed Denial of Service (DDoS), and code alteration. The findings show that the proposed framework can accurately detect various assaults from distances up to 3 m with an AUC of $> 99.5\%$ and 100% detection with less than 1% false positives.

More recently, Reyes et al. [33] proposed a hardware introspection-based anomaly detection framework. The framework uses ML algorithms to identify anomalous device behavior. The Joint Test Action Group (JTAG) interface was used to extract memory traces. After that, the authors formed an image representation of the extracted data and trained a Convolutional Neural Network (CNN) to detect abnormal behavior. The model achieved an accuracy of 98.7%. While both publications achieved noteworthy results, none of them used it in a real-time ES to guarantee the performance of their techniques.

Although significant efforts were made in the field of employing IDS in ES, IDS still face a challenge in detecting and preventing attacks in an effective way in ES. This is mainly due to the unique characteristics of ES. As far as we are aware, none of the frameworks that have been suggested can account for every resource constraint that ES have. For example, Florencio et al. [34] neglected to take energy efficiency into account in their focus on creating a cost-effective IDS.

Furthermore, the goal of Viegas et al. was to create an energy-efficient architecture without taking application costs into account. Accordingly, efforts must be directed at developing an IDS that considers all the resource limitations of ES. Moreover, as far as we are aware, none of the proposed frameworks considered investigating the security concerns of employing ML in IDS.

Table 1. Summary of preliminary studies

Research	Years	Data	Host based	Network based	Method	Result
[39]	2017	NIDS data set	–	✓	ML	Reduces energy use by up to 93%
[38]	2018	Intrusion dataset	–	✓	Anomaly-based with ML	Keeps the classifier reliable even in the event that the contents of network traffic fluctuate
[36]	2018	Signals	–	✓	IDEA framework	Detect various attacks from distances up to 3 m with an AUC of >99.5% and 100% detection with < 1% false positives
[34]	2018	NSL-KDD dataset	–	✓	MLP Networks	Achieves 95% confidence interval
[37]	2020	Signals	✓		MILS architecture	HIDS to secure runtime program execution mix-critically ES
[32]	2021	–	✓	–	IDS architecture	Completely functional Host-based IDS installed and assessed on a PLC
[35]	2021	ADFA-LD dataset	✓	–	ML pipeline	Automatically choose parameters to lower latency and increase scalability
[33]	2023	RAM data	–	✓	HIAD framework	Achieved an accuracy of 98.7%

4 Current State of IDS in ES

Network security is presently a major concern, according to several studies. IDS has been designed to safeguard network security. Various ML techniques, such as ensemble learning, have been utilized to improve IDS performance. There are two kinds of network attacks, namely, active attacks and passive attacks. In an active attack, the intruder sends out a command to disrupt network operations, whereas in a passive attack, the intruder intercepts data within the network. The dynamic nature of cyber attacks makes it impossible for the existing IDS to handle them, leading to several restrictions. According to Anchugam and Thangadurai [23], Ghorbani et al. [22], and Kumar et al. [20], the main causes

of intrusion in a network are as follows: bad packets, encrypted packets, weak network identification and authentication, and protocol-based attacks.

The IDS contains four main components, namely: information source, feature selection, detection engine, and response. They work to identify attacks and generate a report in a pre-defined format [19]. Figure 1 shows how components are organized in IDS.

1. **Data Collection:** The evidence of any intrusion occasion is collected from the source comprehensively. Collecting all the information is costly, and the main challenge is collecting privileged information.
2. **Feature Extraction:** A set of features is maintained to categorize attacks and eliminate irrelevant or redundant features. A feature vector is then produced from the selected subset of attributes.
3. **Detection Engine:** Data is analyzed to detect intrusion events. The strength of this component is determined by its ability to detect all categories of attacks.
4. **Response Engine:** Determines how to respond and control the reaction mechanism after an intrusion activity is detected. The engine takes two types of response, either a passive or an active response. Passive response triggers an alert without responding to the source, whereas active response blocks the source for a pre-determined period.

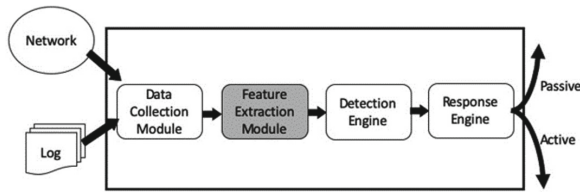


Fig. 1. General IDS architecture. Adapted from [19]

5 Key Challenges and Trends in IDS for ES

In host-based IDS, the information usually consists of data and logs generated by the host system. In contrast, network-based IDS rely on network traffic data. ES are limited by low data storage and processing power. Therefore, the amount of data collected, stored, and analyzed is restricted. Moreover, many ES require real-time or near real-time data processing for timely threat detection, posing a challenge given the limited computational resources. Operating in environments with limited network bandwidth affects data transfer for external analysis, especially when leveraging ML. Future trends focus on employing federated learning for collaborative, privacy-preserving data analysis across multiple ES without sharing raw data [24]. Moreover, other studies focus on minimizing data streams

by using energy-efficient data processing techniques, such as pruning and quantization [25].

Network traffic produces a huge amount of high-dimensional data [26]. Such data may adversely affect the detection of IDS since there is redundant information in network traffic. The same is the case in host-based IDS. The detection process may be slowed down due to the significant processing complexity needed to handle this data, making it unusable in ES. Therefore, an effective way to deal with reducing the dimensions of the data must be proposed [27]. Imbalance class distribution is another challenging issue faced in this component. It causes the classifier to be biased towards the majority class, leading to the generation of many false alarms. Solutions have been designed to combat this problem, which are divided into two main categories: the data level and the algorithmic level. The first category focuses on altering the original data distribution, whereas the second category adjusts the algorithm to fit the minority group better. In the field of ML for anomaly detection, there has been a recent focus on the significance of feature selection and class imbalance. Its goals are to find the most valuable aspects and enhance the quality of the data. Nevertheless, there is a lack of research that demonstrates how crucial feature selection is for handling a high-dimensional class imbalance issue [19].

A number of researchers have put forward strategies and plans to increase the detection accuracy of IDS using ML classifiers. Enormous amounts of network traffic may be analyzed by ML algorithms, which can also spot complex patterns that point to malicious activity. Some ML-based IDS are capable of not just detecting threats but also responding to them automatically. Although several ML-based intrusion detection techniques have been introduced, most of them cannot identify recent, unknown attacks [29]. Some recent studies have focused on employing rejection techniques with ML algorithms to counteract evolving attacks [38]. Furthermore, despite ML-based IDS showing promising results, high computational resource requirements limit the possibility of deploying them in resource-constrained ES. Accordingly, some efforts were made to build a cost-effective and energy-efficient IDS to address this limitation, such as using TinyML [25]. Scalability and latency are other challenges faced in detection engines. It is essential that detection engines can handle the growing data volume and resource constraints. Moreover, detection engines must process data in real-time or near real-time to effectively prevent or mitigate attacks. Consequently, current trends focus on proposing frameworks that can automatically choose parameters that lower latency and increase scalability.

The role of response engines is critical for threat mitigation in network-based and host-based IDS. Host-based IDS are limited by the response options compared to network IDS. Moreover, Host-based IDS finds it especially difficult to respond to attacks from legitimate users or compromised insider accounts. Network-based IDS is challenged by the ability to address threats in high-traffic environments without causing bottlenecks. Additionally, it is challenged by collaborating efficiently with other network security devices, such as routers or firewalls. As far as we are aware, not much research has been conducted on the

problems with response engines in IDS that are used in ES. Future trends to improve response engines in host-based IDS must focus on integrating the host IDS in secure cloud environments with security tools for effective response. On the other hand, threat intelligence techniques could play a vital role in enhancing the response strategies against known and emerging threats in network-based IDS.

According to the conducted literature review, it is observed that the current trend is moving towards utilizing ML and DL algorithms in IDSs used in ES. Several studies employed different ML and DL algorithms while considering the resource constraints of ES. Significant efforts have been made to improve efficiency and cost-effectiveness. However, based on our knowledge, no proposed frameworks have fully met all constraints. Moreover, as far as we are aware, none of the previous studies discussed the new security vulnerabilities that may be introduced by ML and DL models. For example, an attacker can alter the ML model and generate inaccurate predictions if they manage to have access to it. This tampering can include poisoning the model during training (data poisoning) or manipulating input data to cause misclassification (adversarial attacks). In data poisoning, attackers inject malicious data into the training set, causing the model to learn incorrect patterns. Adversarial attacks involve subtly altered input data that leads the model to make wrong predictions while appearing normal to human observers. Another example is attackers can construct malicious activities that mimic normal behavior patterns. Accordingly, the system will fail to identify anomalous behaviors.

Furthermore, interpretability is another aspect that was not covered by the previous studies. Any ML models, especially DL models, lack transparency in their decision-making processes. This can help an attacker exploit the models and make it difficult to diagnose and rectify security breaches. Consequently, interpretability could contribute to enabling informed decision-making, effective responses, system refinement, compliance, and trust in the technology. For instance, a manufacturing facility monitors its production line with an IDS based on ML. In order to identify possible cyber-attacks, the IDS is made to analyze data from sensors and equipment. If the IDS triggers an alarm pointing to a potential cyber invasion from a conveyor belt system, but this system rarely shows deviations, interpretability can play a vital role. In this case, interpretability can aid in understanding the ML model's decision. The decision can be analyzed to determine whether it was based on historical data, specific sensor readings, etc. Interpretability can also help ensure the response's appropriateness. This is critical since if the alert is a false positive, an unnecessary shutdown could be costly. Thus, efforts should be directed at introducing explainable ML and DL models in this domain.

Data privacy is another issue that was not discussed in the literature [28]. Access to vast amounts of data, some containing private or sensitive information, is necessary for many ML models. This data may be compromised or exploited as part of the model's functioning or as a result of an exposure. For example, the models in driver-assistance systems collect vast amounts of data, including GPS

location, camera footage, braking patterns, and more. Continuous data collection could lead to privacy concerns. Moreover, Lane-keeping and parking assistance cameras record video, which may capture sensitive locations or people. This may lead to personal data exposure, unauthorized data access, data misuse, and much more. Therefore, it is essential to introduce robust mitigation measures to enhance data privacy.

6 Research Gaps and Opportunities

One of the main challenges in the field of ES is the computational complexity. The emergence of TinyML made the implementation of ML algorithms on low-powered edge devices possible. It seems that TinyML can handle the demands of implementing ML algorithms in low-power edge devices, such as microcontrollers. This technique compresses an ML model into a small size after training it on the cloud. Later, on edge devices with limited resources, the compressed model is implemented [25]. However, some security concerns must be addressed to reduce the possibility of attacking such models. Securing the developed models by quantization technique is one way to improve the security of the models. Quantization hides the real values of the weights and parameters to prevent unauthorized users from uncovering the model's architecture. Quantization can also contribute to speeding up the inference process, which is essential for real-time IDSs. Moreover, it decreases power consumption, optimizes memory usage, and improves scalability [30].

In order to address the privacy issue, robust data retention and disposal policies must be developed. Data retention could include specifying retention duration based on different data types, outlining security measures for the retention period, and auditing the retained data. Similarly, data disposal policies could include defining secure disposal methods, identifying specific triggers for data disposal, and keeping a record of data disposal. Differential privacy is another useful technique for handling the privacy issue. One way is to include methods to introduce noise into the data in every embedded device at the time of collection. This makes sure that the details of the data are hidden even while the IDS is still able to identify patterns indicative of an attack. However, to guarantee that the IDS retains its efficacy in identifying threats while safeguarding data privacy, it is crucial to remember to adjust the noise level carefully. Moreover, differential privacy can be established when training ML models. This ensures that the model, even if exposed, does not reveal sensitive information about the data on which it was trained.

Explainable Artificial Intelligence (XAI) approaches, such as Local Interpretable Model-agnostic Explanations (LIME), might also be employed at this step to guarantee the reliability of the deployed model [31]. LIME provides local interpretability, which can aid in giving insights into debugging, quality assurance, model validation, and others. With the help of XAI, users and administrators may comprehend the reasoning behind an IDS's decision to flag an action as potentially dangerous. This understanding is crucial to building trust, especially in critical environments where ES are often deployed. Moreover, it helps

administrators understand the reason behind an alert, determine the appropriate response, and enhance the model.

Another recommendation to enhance the development of IDS in ES is to use federated learning to overcome the issues of connecting with the cloud for training and storage purposes. Federated learning allows for dispersed training at the edge level and avoids sharing local data with servers. Keeping data local limits the risk of data breaches or leaks during transmission or central storage. Consequently, it could be vital in preserving privacy and reducing data transmission. Moreover, recently, embedded ML has extensively used Trusted Execution Environments (TEEs) to improve security and privacy. ML inference is carried out safely via TEEs. This is especially useful when sensitive data or models must be protected from outside attacks. Calculations are performed without revealing raw data to the public, and data is kept encrypted. Furthermore, TEEs provide safe over-the-air upgrades for embedded ML models and provide strong hardware security.

7 Conclusion

This paper reviewed the studies that developed IDS for ES. It was carried out using data from many research articles published in various journals and conferences between the years 2018 and 2023. It investigated current trends and challenges in the development of IDSs for ES, with a specific focus on the growing use of ML and DL algorithms. The main finding of this study showed that researchers are moving toward employing ML and DL to build IDS for ES. However, it is observed that there is a lack of frameworks that address all limited resources of ES while maintaining privacy, security, and interpretability. These findings highlight the necessity of enhancing IDS for ES through several recommendations discussed in this study. Future research should explore leveraging the latest technologies, such as TinyML, federated learning, and TEEs, to facilitate secure real-time data analysis. Moreover, to maintain privacy, security, and interoperability, it is recommended to introduce strong data retention and disposal policies, develop differential privacy techniques, and utilize XAI. It is believed that the discussed recommendations can play a vital role in improving the security of IDSs in ES.

References

1. Vahid, F., Givargis, T.D.: *Embedded System Design: A Unified Hardware/Software Introduction*. Wiley, New York (2001)
2. Papp, D., Ma, Z., Buttyan, L.: Embedded systems security: threats, vulnerabilities, and attack taxonomy. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, pp. 145-152 (2015)
3. Hammoudeh, M., Newman, R.: Information extraction from sensor networks using the Watershed transform algorithm. *Inf. Fusion* **22**, 39-49 (2015)

4. Ambrose, J.A., Ragel, R.G., Jayasinghe, D., Li, T., Parameswaran, S.: Side channel attacks in embedded systems: a tale of hostilities and deterrence. In: Sixteenth International Symposium on Quality Electronic Design, Santa Clara, CA, USA, pp. 452–459 (2015)
5. Azzedin, F., Albinali, H.: Security in Internet of Things: RPL attacks taxonomy. In: The 5th International Conference on Future Networks & Distributed Systems, pp. 820–825 (2021)
6. Azzedin, F., Alhejri, I.: A layered taxonomy of internet of things attacks. In: Proceedings of the 6th International Conference on Future Networks & Distributed Systems, pp. 631–636 (2022)
7. Epiphaniou, G., Pillai, P., Bottarelli, M., Al-Khateeb, H., Hammoudeh, M., Maple, C.: Electronic regulation of data sharing and processing using smart ledger technologies for supply-chain security. *IEEE Trans. Eng. Manage.* **67**(4), 1059–1073 (2020)
8. Benoudifa, O., Wakrime, A.A., Benaini, R.: Autonomous solution for controller placement problem of software-defined networking using MuZero based intelligent agents. *J. King Saud Univ.-Comput. Inf. Sci.* **35**(10), 101842 (2023)
9. Saadatmand, M., Cicchetti, A., Sjödin, M.: On generating security implementations from models of embedded systems. In: International Conference on Software Engineering Advances, Barcelona, Spain, (2011)
10. Cysneiros, L.M., do Prado Leite, J.C.S.: Nonfunctional requirements: from elicitation to conceptual models. *IEEE Trans. Softw. Eng.* **30**(5), 328–350 (2004)
11. Gala, Y., Vanjari, N., Doshi, D., Radhanpurwala, I.: AI based techniques for network-based intrusion detection system: a review. In: 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, pp. 1544–1551 (2023)
12. Sethi, K., Kumar, R., Prajapati, N., Bera, P.: A lightweight intrusion detection system using Benford’s law and network flow size difference. In: 2020 International Conference on COMMunication Systems & NETWORKS (COMSNETS), Bengaluru, India, pp. 1–6 (2020)
13. Carlin, A., Hammoudeh, M., Aldabbas, O.: Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. *Int. J. Adv. Comput. Sci. Appl.* **6**(6) (2015)
14. Lahbib, A., Ait Wakrime, A., Laouiti, A., Toumi, K., Martin, S.: An event-B based approach for formal modelling and verification of smart contracts. In: Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020), pp. 1303–1318 (2020)
15. Aloose, A., He, H., Shaw, C., Khan, M.A.: Analytical review of cybersecurity for embedded systems. *IEEE Access* **9**, 961–982 (2021)
16. Bansod, G., Raval, N., Pisharoty, N.: Implementation of a new lightweight encryption design for embedded security. *IEEE Trans. Inf. Forensics Secur.* **10**(1), 142–151 (2015)
17. Koopman, P.: Embedded System Security. *Computer* **37**(7), 95–97 (2004)
18. National Institute of Standards and Technology. <https://www.nist.gov/publications/intrusion-detection-systems>
19. Binbusayyis, A., Vaiyapuri, T.: Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection. *Heliyon* **6**(7), e04262 (2020)
20. Kumar, S., Gupta, S., Arora, S.: Research trends in network-based intrusion detection systems: a review. *IEEE Access* **9**, 157761–157779 (2021)

21. Alooseel, A., He, H., Shaw, C., Khan, M.A.: Analytical review of cybersecurity for embedded systems. *IEEE Access* **9**, 961–982 (2020)
22. Ghorbani, A.A., Lu, W., Tavallaee, M.: *Network Intrusion Detection and Prevention: Concepts and Techniques*. Springer Science & Business Media, 47 (2009). <https://doi.org/10.1007/978-0-387-88771-5>
23. Anchugam, C.V., Thangadurai, K.: Classification of network attacks and countermeasures of different attacks. In: *Network Security Attacks and Countermeasures*, pp. 115–156. IGI Global (2016)
24. Agrawal, S., et al.: *Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions*, Computer Communications (2022)
25. Butt, M.A., Qayyum, A., Ali, H., Al-Fuqaha, A., Qadir, J.: Towards secure private and trustworthy human-centric embedded machine learning: an emotion-aware facial recognition case study. *Comput. Secur.* **125**, 103058 (2023)
26. Hammoudeh, M., Newman, R., Dennett, C., Mount, S., Aldabbas, O.: Map as a service: a framework for visualising and maximising information return from multi-modal wireless sensor networks. *Sensors* **15**(9), 22970–23003 (2015)
27. Balasaraswathi, V.R., Sugumaran, M., Hamid, Y.: Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. *J. Commun. Inf. Netw.* **2**, 107–119 (2017)
28. Walshe, M., Epiphaniou, G., Al-Khateeb, H., Hammoudeh, M., Katos, V., Dehghantanha, A.: Non-interactive zero knowledge proofs for the authentication of IoT devices in reduced connectivity environments. *Ad Hoc Netw.* **95**, 101988 (2019)
29. Liu, H., Lang, B.: Machine learning and deep learning methods for intrusion detection systems: a survey. *Appl. Sci.* **9**(20), 4396 (2019)
30. Sharmila, B.S., Nagapadma, R.: Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset. *Cybersecurity* **6**(1), 41 (2023)
31. Mahbooba, B., Timilsina, M., Sahal, R., Serrano, M.: Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. *Complexity* **2021**, 1–11 (2021)
32. Martinez, C.V., Vogel-Heuser, B.: A host intrusion detection system architecture for embedded industrial devices. *J. Franklin Inst.* **358**(1), 210–236 (2021)
33. Reyes, D.L., Perez-Pons, A., Dean, R.B.: Anomaly detection in embedded devices through hardware introspection. In: *2023 Silicon Valley Cybersecurity Conference (SVCC)*, pp. 1–7, IEEE, San Jose, CA, USA (2023)
34. de Almeida Florencio, F., Moreno, E.D., Macedo, H.T., de Britto Salgueiro, R.J.P., do Nascimento, F.B., Santos, F.A.O.: Intrusion detection via MLP neural network using an arduino embedded system. In: *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, pp. 190–195. IEEE, Salvador, Brazil (2018)
35. Liu, M., Xue, Z., He, X.: Two-tier intrusion detection framework for embedded systems. *IEEE Consum. Electron. Mag.* **10**(5), 102–108 (2020)
36. Khan, H.A., et al.: IDEA: intrusion detection through electromagnetic-signal analysis for critical embedded and cyber-physical systems. *IEEE Trans. Dependable Secure Comput.* **18**(3), 1150–1163 (2019)
37. Kadar, M., Tverdyshchev, S., Fohler, G.: Towards host intrusion detection for embedded industrial systems. In: *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pp. 5–8. IEEE, Valencia, Spain (2020)

38. 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.* **78**, 16–32 (2018)
39. Viegas, E.K., Santin, A.O., Oliveira, L.S.: Toward a reliable anomaly-based intrusion detection in real-world environments. *Comput. Netw.* **127**, 200–216 (2017)