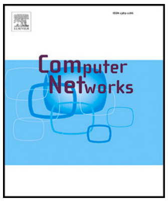
Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Survey paper

Deep Reinforcement Learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges

Jesús F. Cevallos M., Alessandra Rizzardi, Sabrina Sicari [*], Alberto Coen Porisini

Dipartimento di Scienze Teoriche e Applicate, Universita‘ degli Studi dell’Insubria, via O. Rossi 9, Varese, 21100, VA, Italy

ARTICLE INFO

Keywords:
Deep Reinforcement Learning
Intrusion Detection Systems
Internet of Things

ABSTRACT

The Internet of Things (IoT) scenario places important challenges even for deep learning-based intrusion detection systems. IoTs are highly heterogeneous networks in which multiple types of nodes and connections between them proliferate at a fast pace. From a deep learning perspective, such complexity translates into dynamic feature spaces where the extraction of semantic patterns and correlations among features may require sophisticated inductive biases to be learnt by gradient-based techniques. The research community has recently suggested using Deep Reinforcement Learning (DRL) as a potent approach to effectively identify cyber-threat attempts in IoTs.

DRL consists of a Markov Decision Process-based meta-model that permits solving high-dimensional combinatorial optimization problems where differentiable supervisory signals may be absent. For this reason, multiple intelligent intrusion detection systems have been proposed for the IoT environment where high-level requirements are been pursued alongside the detection accuracy. These goals are related to optimizing the computational overhead, reducing power consumption at the edge, and preserving the privacy of sensitive information, among others.

This survey offers a clear bird’s eye view of the most recent design choices for DRL-based intrusion detection systems with a focus on the specific context of IoT. Our aim is not only to offer an exhaustive taxonomy of design alternatives made by DRL practitioners in the field of Intrusion detection, but also to discuss the advantages and the effective deployment of each setting concerning real IoT environments. We hope this work would guide the researchers interested in Intrusion Detection for IoTs to establish solid criteria for the most effective usage of Deep Reinforcement Learning in their future work.

1. Introduction

Three years ago, Internet connections with a provenance related to the Internet of Things (IoT) overcame those from non-IoT. Connected devices are ubiquitous nowadays, and may reach 27 billion by 2025 [1]. IoT permits collecting and orchestrating fine-grain heterogeneous data to leverage high-valuable information services, support decision-making processes at any scale, and verify coarse-grain hypotheses, among other important functions. Unfortunately, all these benefits come with the inevitable extension of both the *attack surface* and the potential harm that can be made by attackers. There are numerous cyber threats that target IoT infrastructures, and *Tactics, Techniques, and Procedures* that target IoT environments are constantly evolving [2].

As a consequence, Accademia and Industry are constantly putting forward more intelligent Intrusion Detection Systems (IDS) for IoT environments as a first step towards preventing and mitigating cyber-attacks on such infrastructures. Recent advancements in IDS for IoT include the usage of Machine Learning (ML) and, more specifically, Deep Learning (DL) tech-niques to optimize the feature extraction process and the real-time detection of modern IDSes both at the core and the edge of the network [3].

IoTs are characterized by a high heterogeneity of communication protocols and standards, computing and power resource constraints, a stringent privacy preservation criterion, and highly dynamic topologies [4]. These inherent features of IoTs pose critical challenges even to powerful DL-based feature extractors when the goal is detecting intrusions. Moreover, Training DL models is a highly resource-consuming task, and many DL pipelines require vast amounts of data to learn to abstract the characteristics of normal and abnormal traffic.

For this reason, the research community has delivered new *inductive biases* that sophisticate the DL-based IDSes and help them adapt to the IoT realm. Recurrent neural networks, Generative models, Continuous Learning, and Deep Reinforcement Learning are among the recent

[*] Corresponding author.
E-mail address: sabrina.sicari@uninsubria.it (S. Sicari).

https://doi.org/10.1016/j.comnet.2023.110016
Received 15 June 2023; Received in revised form 5 August 2023; Accepted 5 September 2023
Available online 9 September 2023

research trends that try to create effective IDSes for the IoT [5,6]. This survey concentrates on Deep Reinforcement Learning (DRL), as it represents a powerful technique to learn complex tasks – such as those associated with multi-objective network management – through gradient descent [7]. Many researchers have recently used DRL to implement Intrusion Detection in the IoT. However, the specific modelization paradigms adopted in these works might be highly heterogeneous, and it might not be trivial to distill a well-formed know-how about effective and problematic settings of DRL in this field. For this reason, this work aims to organize an extensive set of DRL-associated design choices for the IoT. For each of the studied works, our research seeks to identify advantages and disadvantages, and shed light on how to take full advantage of the power of DRL in the field of IoT Intrusion Detection.

More specifically, this manuscript investigates how DRL has been recently used to learn expert intrusion detection policies in a supervised fashion, how experience-driven learning is made possible by DRL, and how other high-level goals like optimal adversarial, distributed, and federated learning schemes are proposed for IDS in the IoT using DRL. By analyzing the recent related literature, the aim is to shed light on the good practices related to DRL that augment the generalization power and accuracy of IDSes for the IoT. Moreover, this paper also puts in evidence the design choices that may hinder the practical and effective usage of DRL in this context, and finally, this work highlights future research opportunities for those who aim to exploit the full capabilities of DRL in the context of IoT-related intrusion detection.

1.1. Survey outline

This survey proposes to comprehensively and critically review the current applications of DRL in the field of IDS for IoT infrastructures. This paper is subdivided as follows:

- Section 2 briefly summarizes Deep Reinforcement Learning, the particularities of intrusion detection in IoTs, and the main reasons why DRL might empower them.
- Section 3 gives an overview of related surveys and identifies the main original contributions of our work.
- Section 4 offers a multi-dimensional analysis of the current state of the art of DRL-based solutions for IDS in IoT.
- Finally, Section 5 discusses common trends in the literature and highlights the best practices, lessons learnt, and open challenges related to DRL-based intrusion detection for the IoT.

Intrusion Detection Systems for IoT are abbreviated as IoT-IDS in the rest of this paper. A complete list of the abbreviations used in this research can be found in Table 1.

2. Background and motivation

In this section, a brief overview of Deep Reinforcement Learning, its main components, and its properties is presented and the match between these properties and the requirements posed by IoT-IDS is highlighted.

2.1. Deep reinforcement learning

Reinforcement Learning (RL) is a framework that combines Temporal Difference learning with trial-and-error selective and associative mechanisms to learn to maximize a reward signal from a dynamic system. RL can be seen as a methodology to solve optimal control problems where the objective is to control the behavior of a dynamical system over time [8]. *Deep* Reinforcement Learning instead, scales the validity of Reinforcement Learning-based solutions to high-dimensional and complex environments through the usage of Deep Artificial Neural Networks (Deep ANN) [7,9,10]. A scheme of a DRL-based Intrusion Detection System for the IoT is depicted in Fig. 1. The main components of a common DRL pipeline are now introduced following the outline of this figure.

Table 1
Abbreviations used in this research.

Abbreviation	Meaning
AC	Actor-Critic
AI	Artificial Intelligence
AE	Auto-Encoder
A3C	Asynchronous Advantage Actor-Critic
ANN	Artificial Neural Network
C-GAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
CPS	Cyber Physical System
Deep-AE	Deep Auto-Encoder
DDQN	Double Deep Q-Network
DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning
DQN	Deep Q-Network
DS2OS	Distributed Smart Space Orchestration System Dataset
DRL	Deep Reinforcement Learning
GAN	Generative Adversarial Network
GNN	Graph Neural Network
HPO	Hyper-Parameter Optimization
i-Forest	Isolation Forest
FL	Federated Learning
IAM	Identity and Access Management
IDS	Intrusion Detection System
IoT	Internet of Things
IoT-IDS	IDS for Internet of Things
I-IoT	Industrial Internet of Things
IoMT	Internet of Medical Things
LSTM	Long-Short Memory Network
MAB	Multi-Armed Bandit
ML	Machine Learning
MLP	Multi-Layer Perceptron
P2P	Peer-to-Peer
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAE	Stacked Auto-Encoders
SD-IoT	Software-Defined Internet of Things
SMO	Spider Monkey Optimization
SOA	Social Optimization Algorithm
SOTA	State-of-the-art
SVN	Smart Vehicular Network
TD3	Twin-Delayed Deep Deterministic Policy Gradient
UAV	Unmanned Aerial Vehicles
WSN	Wireless Sensor Network

2.1.1. Markov decision process

RL modelization involves an agent that learns to act inside an environment to maximize a reward signal in a long-term fashion. In the case of IoT-IDS, the environment may consist of a real IoT infrastructure or a simulated environment. In Fig. 1, the environment is represented by rectangle **(D)**. The interaction between the agent and the environment is modeled as a Markov Decision Process (MDP) [11]. In this setting, the goodness of the agent's decisions concerning the optimization objectives is commonly evaluated using a set of established scoring functions. In DRL, these functions are approximated by Deep ANNs and are indicated in the **(A)** rectangle in Fig. 1. In our specific intrusion detection case, the DRL agent's task is commonly related to evaluating network security. Moreover, security related-functionalities like alerts and mitigation countermeasures could be associated to other non-functional requirements like reliability, efficiency, timeliness, among others.

An MDP is normally represented by the tuple $(\mathscr{S},\mathscr{A},\mathscr{P},\mathscr{R})$, where:

- The state space $\mathscr{S}$ represents all the possible states of the environment in which an agent can be at any given time. In a typical DRL pipeline, the state space is a vector space that encodes the value of a set of features probed from the environment. The state space vectors are typically denoted by $\mathbf{s} \in \mathscr{S}$. In the case of IoT-IDS, the state space might contain data related to the current network traffic, system logs, node resource states, among others. The encoding of such features can be seen as a feature engineering

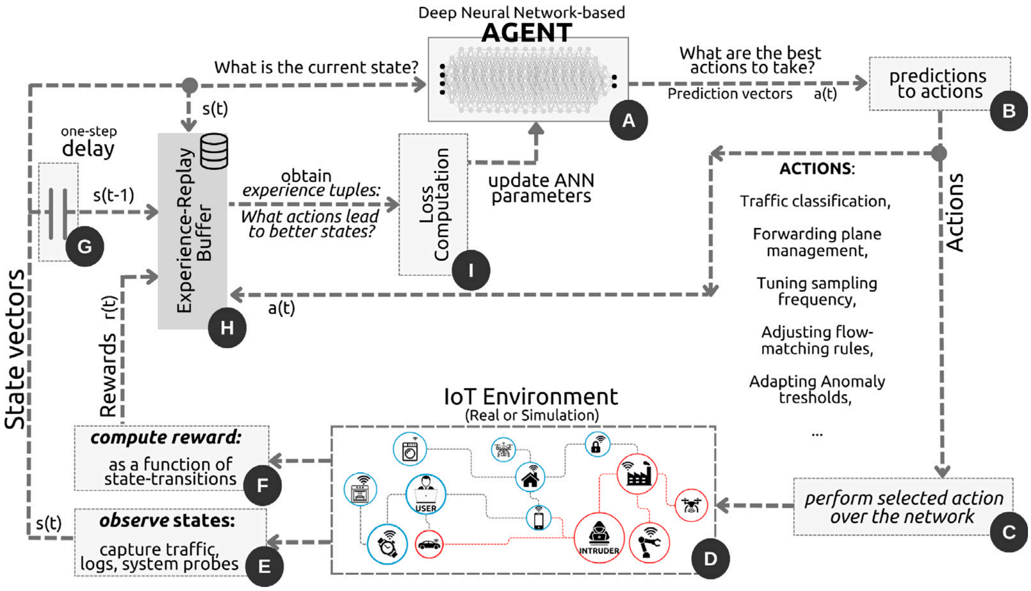

Deep Neural Network-based
AGENT
A
What is the current state?
What are the best actions to take?
Prediction vectors
a(t)
predictions to actions
B
s(t)
one-step delay
s(t-1)
G
Experience-Replay Buffer
H
obtain experience tuples:
What actions lead to better states?
Loss Computation
I
update ANN parameters
ACTIONS:
Traffic classification,
Forwarding plane management,
Tuning sampling frequency,
Adjusting flow-matching rules,
Adapting Anomaly tresholds,
...
Actions
a(t)
State vectors
Rewards r(t)
IoT Environment
(Real or Simulation)
USER
INTRUDER
D
compute reward:
as a function of state-transitions
F
observe states:
capture traffic, logs, system probes
E
s(t)
perform selected action over the network
C

Fig. 1. Schematic representation of a Deep Reinforcement Learning-based Intrusion Detection System for the Internet of Things. Deep Neural Networks are trained to approximate functions like (3) of a Markov Decision Process (MDP) meta-model. These functions drive the action policy of an adaptive agent that maximizes the long-term reward in (2).

process and is represented in rectangle **(E)** of Fig. 1. By looking at the state-space tuples, i.e., by observing a snapshot of the IoT network's state, our agent can assess the *health* of the network – from a cyber-security point of view – and infer what are the best actions to perform over such a network to improve the overall health of the system.

- The action space $\mathscr{A}$ is the collection of all the possible actions that an agent can do when interacting with the environment. In a typical DRL scenario, the agent is said to interact with the environment through an *action policy*, typically denoted as π. This policy indicates what actions to take as a function of the current state $\mathbf{s}$:

$$\pi : \mathbf{s} \rightarrow \pi(\mathbf{s}) \tag{1}$$

Some DRL configurations like *policy-based* directly approximate π with an agent's Deep ANN. Instead, in *value-based* DRL, a mapping process could be necessary to obtain the actions from the outputs of the agent. Such a process is indicated by the rectangle **(B)** in Fig. 1. In the IoT-IDS example, actions might be directly related to assessing the network state $\mathbf{s}$ from the security point of view. However, indirect association between such an assessment and the actions could also be done. For example, the actions could control the characteristics of the monitoring process or the parameters of the security assessment function. in Fig. 1, rectangle **(C)** denotes the action-taking procedure.

- MDP is strictly related to optimal control: the actions of the agent typically have an effect on the state of the environment. In DRL, state *transitions* are said to occur between one action and the other. Such transitions are governed by a transition probability distribution denoted by $\mathscr{P}$. In IoT-IDS and in IDS in general, such a probability distribution is rarely known upfront and might need to be learned from the DRL agent. To provide an example, one can think about the difficulties about predicting the whole network state changes relying exclusively in simple forwarding plane management actions. In such a case, it is not difficult to notice that state transitions are predictable *up to a certain point*. Indeed, many other hidden factors may influence the changes in the environment, and the agent should need to refine its predictions progressively through learning.
- Finally, every RL agent receives a reward for each action taken. The environment delivers rewards following a reward policy $\mathscr{R}$ which is a function of the current state $\mathbf{s}$ and the action taken a. Note the reward computation process is denoted by rectangle **(F)** in Fig. 1. In our specific use case, the rewards of the DRL agent could be associated with many non-functional requirements in addition to intrusion detection accuracy. Notice that one candidate strategy to consider various goals like accuracy, efficiency, timeliness and computation overhead minimization, could be modeling the reward function as a weighted sum of a set of goal-specific scores.

RL agents seek to maximize the *discounted future reward*, which is defined as:

$$G_\tau = R_{\tau+1} + \gamma R_{\tau+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{\tau+k+1} \tag{2}$$

where R_τ is the reward the agent receives at time τ after executing an action and γ is a fixed parameter that reduces the value of future action rewards with respect to the immediate one and is known in the literature as the *discount factor*. Notice that, as the dynamical systems' conditions change over time, the functions that describe such systems depend on time. Such a dependency is indicated with the index τ in (2) and in the rest of this work.

2.1.2. Value functions

Given a specific action policy π, the *Action-value Function*, also called *Q-value function* indicates how valuable it is to take a specific action a being at state $\mathbf{s}$ and following the policy π from the next state on:

$$Q_\pi(\mathbf{s}, a) = E_\pi[G_\tau \parallel \mathbf{s}_\tau = \mathbf{s}, a_\tau = a] \tag{3}$$

where E_π indicates the expected value when acting with the policy π, $\mathbf{s}_\tau$ indicates a state vector at time τ, and a_τ is the specific action taken at time τ.

The *state value function* instead, is commonly denoted by $V_\pi(\mathbf{s})$ and quantifies the expected return when being at a certain state $\mathbf{s}$ given that the policy $\pi(\mathbf{s})$ is being followed.

$$V_\pi(\mathbf{s}) = E_\pi[G_\tau \parallel \mathbf{s}_\tau = \mathbf{s}] \tag{4}$$

Notice that from (2) one can derive the *recursive Bellman equations*:

$$Q_\pi(\mathbf{s}_\tau, a_\tau) = R_{\tau+1} + \gamma Q_\pi(\mathbf{s}_{\tau+1}, a_{\tau+1}) \tag{5}$$

$$V_\pi(\mathbf{s}) = R_{\tau+1} + \gamma V_\pi(\mathbf{s_{\tau+1}}) \tag{6}$$

Recall that R_τ is the reward the agent receives at time τ after executing action a_τ, and γ is the discount factor.

These recursive expressions permit the iterative approximation of (3) and (4) by acting on the environment and observing the received rewards. In other words, (5) and (6) are the main building blocks of multiple differentiable loss functions that drive the agent's learning process.

2.1.3. DRL frameworks

DRL configurations may be classified concerning the specific optimization goal:

- The objective of *Policy-Based* DRL agents is to use an ANN to approximate the *optimal action policy*, denoted as $\pi^*(s)$, that maximizes (2). Policy learning schemes have provided feasible solutions for high-dimensional action spaces. A prominent example of a policy learning DRL framework is Deep Deterministic Policy Gradient (DDPG) [12].
- In *Value-Based* DRL frameworks instead, the agent's ANN is thought to approximate $Q^*(s,a)$ or other functions related to (3) or (4). One of the most famous value-based DRL algorithms is Deep Q-Learning [13], which performance has been enhanced by the research community with the introduction of various mechanisms such as target-networks [14], attention mechanisms based on action-advantage criterion [15], among others.
- Finally, *Actor-Critic* methods combine policy learning and value-based DRL [16]. This combination has enhanced the learning performance of DRL agents because the value functions act as a baseline that reduces the variance of the estimation of the gradients of the policy function. Despite Actor-Critic methods are not the only abstraction that merges policy-based and value-based reinforcement learning [17], they have received major attention from the research community.

2.1.4. On-policy vs. off-policy RL

Notice that the value functions described in Section 2.1.2 are dependent on a given policy π. Another important dimension upon which DRL frameworks can be differentiated is centered precisely in π [7,18]. *On-policy RL* agents interact with the environment sampling actions from the same policy they are learning to approximate. *Off-policy* RL agents instead learn to approximate a fixed policy π^* while acting with another policy π. The latter approach permits handling better the important trade-off between *exploration* of the action space and *exploitation* of the acquired knowledge about the optimal policy.

Another disadvantage of on-policy DRL algorithms is that the variance of the estimates they learn augment notoriously, making convergence a slower process [18]. However, asynchronous gradient descent has proven effective for robustly applying on-policy DRL approaches to complex problems. The usage of multiple agents learning in parallel has a stabilizing effect on learning the optimal policies. The Asynchronous Advantage Actor Critic (A3C) framework [19] is a commonly used on-policy DRL technique that applies precisely these ideas.

Depending on the DRL framework and whether the learning is done on-policy or off-policy, the recursive Eqs. (5) and (6) are used in different ways to compute the loss signal. One basic loss function is that of the Deep Q-learning algorithm [20]. This algorithm approximates the Q-value function in (3) with a deep ANN and the loss function is proportional to the *temporal difference error* term:

$$[R_\tau + \gamma \max_a Q(\mathbf{s_{\tau+1}}, a, \theta) - Q(\mathbf{s}_\tau, a_\tau; \theta)]^2 \tag{7}$$

where R_τ denotes the reward obtained after taking action a_τ at state $\mathbf{s}_\tau$, and θ denotes the parameters of the ANN that approximates the Q-value function.

In DRL, gradient descent is used to update the parameters of the Deep ANNs that approximate $Q_\pi(\mathbf{s},a)$, $V_\pi(\mathbf{s})$, and/or $\pi(\mathbf{s})$ while seeking to minimize bootstrap-based errors like the one in (7) or other sophistications of it [12,15,21].

As can be seen from (7), in many DRL algorithms the loss is typically computed as a function of a state observation, $\mathbf{s}_\tau$, the correspondent action, a_τ, the correspondent reward obtained, R_τ, and the new state observation, $\mathbf{s_{\tau+1}}$. These elements constitute an *experience tuple*. Many off-policy DRL frameworks store experience tuples in the *experience-replay buffer* which sampling permits the usage of mini-batch gradient descent to optimize the agent's deep ANNs.

The inclusion of $\mathbf{s}_\tau$ and $\mathbf{s_{\tau+1}}$ in the experience tuples are represented with a one-step delay block in rectangle **(G)** of Fig. 1. Instead, the experience replay buffer is represented in block **(H)**. Finally, the loss computation is denoted by rectangle **(I)** in the same figure. Note that gradients of such a loss function drive the learning of (3), (4), or (1) in typical a DRL framework.

2.1.5. DRL applications to IoT

This work is focused on DRL-based intrusion detection systems for IoT scenarios. However, apart from IDS, DRL has been recently used for optimal control related applications on smart grid management [22], smart transportation systems [23], Unmanned Aerial Vehicles [24], Internet of Medical Things [25], industrial IoT [26], among others [27]. The reasons for the recent spread of the usage of DRL in IoT scenarios might be related to the trend of converging to autonomous IoT scenarios [28] where human intervention to control the system is minimized [29–31]. A few reasons why DRL modelization could be a preferable approach to handle IoT-IDS are provided in the next section.

2.2. Intrusion Detection Systems in the Internet of Things

The goal of an Intrusion Detection System (IDS) is to reveal illegitimate usage of any set of computing or network resources. In the realm of IoT, the effectiveness of these cyber-security systems may depend on the adaptation to particular context factors that might not always be present in other network-related scenarios. These context factors are: (1) detection criteria must adapt to highly-heterogeneous and dynamic environments, (2) the flow of information induced by the security system shall respect the privacy-preservation principle and (3) the overall communication and processing overhead of the system should be constrained and proportional to the resource constraints of IoT-related devices [32]. It is worth remarking, however, that (4) robustness against intelligent and adaptive attacks [33], and (5) scalability to manage high-volume, high-velocity, and high-variety of data [34] is a requirement that IoT-related IDSes may inherit from traditional IDSes. A brief overview of some of these requirements is now given to successively point to Deep Reinforcement Learning as a key enabler to meet them.

Requirement 1. Adaptiveness to dynamic environments. A standard layered structure of the Internet of Things identifies three main layers: the sensing or physical layer, the network layer, and the application layer [35]. While vertical interconnection between layers is a common requirement for any computer network, the vast heterogeneity in the network and the sensing layer is a strong feature of IoT. In other words, the number of device types, and communication protocols is bigger in IoT compared to other networks. Less standardization leads to the dispersion of defensive forces, and manufacturers' concerns are often heavily concentrated in market delivery and tend to neglect the time needed to maturate protection blueprints over inherently-vulnerable devices [2]. From these context factors, two main requirements turn up for IoT-oriented cyber-defense systems. First, attack vectors that target IoTs emerge at a faster pace, thus anomaly detection systems may need to have a higher generalization power over the concept of attack [3]. Second, the distribution of information that an intrusion recognizer system ingests may change rapidly compared to traditional network contexts, and thus the defensive system may need to reshape itself at a proportional velocity to be effective. Zero-day attack recognition and adaptive and dynamic defense strategies may be critical for IoT-IDSes [36].

Requirement 2. Privacy preservation. Machine learning and deep learning models are data-driven feature extraction and analysis techniques that learn to make inferences about high-level concepts abstracted from trends in the distribution of training data. Strategies for detecting illegitimate usage of computer networks often use ML to detect correlations between attack vectors and features of traffic traces, system logs, alert logs, and resource probe logs, among others. ML-driven IDSes are trained on a significant amount of these data before converging to effective detection functionalities [6]. However, these data may often be sensitive, thus, federated learning (FL) strategies for ML-driven network functions have been researched in the last years [37]. FL refers to a set of privacy-preserving approaches that permit both the extraction and transfer of knowledge from data without compromising the confidentiality of its contents and subjects. In IoT, sensitive information such as end-user location, health status, social interactions, behavior and status of industrial equipment and assets, and many more, are the main content of the communication [4]. As a consequence, training data for ML-driven IDSes for IoT may be related to confidential information, and even governmental regulations may impede gathering these data to train data-driven detection systems [38]. The actual implementation of IoT-IDSes may require federated learning more than anything else.

Requirement 3. Lightweightness. Smart Objects and devices at the edge of IoTs are often resource-constrained. Recently, lightweight communication protocols and specialized network architectures have been developed to reduce the power consumption and processing footprint of edge communications [39]. Not only routing and other communication-oriented functionalities may be lightweight, but, also, security-related tasks shall be less computation intensive at the network's periphery. The viability of intrusion detection systems may be dependent on low-complexity solutions and efficient task-offloading schemes, particularly in the IoT [3,40].

Requirement 4. Adversarial training. Cybersecurity is a constantly evolving art: Not only do the prevention and mitigation systems evolve and strengthen, but also attack vectors adapt correspondingly. In ML, the concept of an adversarial attack is related to the information instances that lay inside the decision boundary of an inference model [41]. Consequently, adversarial attacks may potentially bias the correct inference of the model. In the context of intrusion detection, an adversarial attack on a system is an instance of illegitimate usage whose signature may be only slightly different when compared to a legitimate one. An adversarial attack is intelligently and adaptively crafted by an attacker so that the detection system is unable to notice it by looking at its observable characteristics [42]. Consequently, the chances of an adversarial attack evading detection are higher with respect to other threats. The research community has put a lot of work into producing robust IDSes for which adversarial attacks are difficult to synthesize. This rat race between defenders and attackers remains active, and the specific features of IoT may represent advantageous conditions for the latter. IoT-IDSes may be effectively deployed in real scenarios if only trained with carefully-designed adversarial learning strategies [33].

2.3. Motivation

Our work is now briefly motivated. Our main arguments are that, first, in the context of IoTs, Intrusion Detection Systems represent an open area of research, and second, these systems are witnessing and may continue to witness unprecedented advantages from a *well-designed* application of Deep Reinforcement Learning.

Can DRL empower IDS for IoT?. From the context factors depicted in Section 2.2, it is not difficult to understand that intelligent intrusion detection modules for modern IoT environments often need to go beyond static traffic classification schemes. Heterogeneous sets of time series regarding network resource status, communication flows, packet headers, payload hashes, and others, need to be correlated at various time scales to infer high-level causalities between adverse effects on the networks and the intrusion of illegitimate users on it. Additionally, the distributions among which one identifies such correlations may change through time, and the resource-constrained scenarios may impose to put attention to resilience alongside detection accuracy, which may lead to conflicting objectives. As a consequence, additional levels of indirection may need to be modeled to handle multiple goals and learn correct goal priority criteria.

It is not hard to notice that discrete optimization algorithms may have a hard time dealing with a realistic IoT-IDS in the sense defined above because of the explosion of variables and constraints. Similarly, no supervised learning-based policies may exist capable of achieving such complex goals. In fact, even in the case that one has an ANN with enough representational power to learn the goals depicted above, the design of effective and realistic supervisory signals for these goals in a data set may be non-trivial.

As described in Section 2.1, reinforcement learning is largely based on the concept of Markov Decision Processes. RL creates a meta-model based on MDP that frames the optimization of potentially any non-differentiable reward signal within a set of differentiable *value functions* [10]. This framing permits the application of gradient-based optimization techniques to solve a vast spectrum of complex tasks for which the definition of a differentiable reward signal may be not possible. DRL has helped modern communication networks to achieve unprecedented effectiveness and accuracy in multi-objective resource and task allocation [43], task offloading [44], traffic management [45], on-the-fly deployment [46], among others [27].

Why to make a survey on DRL-driven IDS for Iot? This research pretends to walk through the argument that Deep Reinforcement Learning may be a game-changing strategy when one needs to face the requirements enlisted in Section 2.2. Many works are reviewed that exploit the capabilities of DRL-based agents to optimize intrusion detection considering heterogeneous and dynamic environments, robustness against adversarial and adaptive attacks, preservation of the privacy of information, and resource constraints. In Section 4, exemplary modelization insights as well as problematic design schemes are evidenced in this respect. In Section 5, instead, it is noted that IoT-IDS is itself an open area of research in the sense that vast room for improvement exists at the time of writing this manuscript. Precisely for this reason, eliciting the best practices, lessons learned and open challenges regarding the application of DRL for IoT-IDSes may be greatly useful for the research community. The next section offers an overview of some recent reviews related to our scope and identifies the originality of this work.

3. Related works

Many works concentrate on DL-based IDSes in the context of IoT. However, these surveys are not focused specifically on Deep Reinforcement Learning. Some recent remarkable reviews, related to ours are now mentioned, and the main contribution of our work is then highlighted.

Tsimenidis et al. [36] categorized deep learning applications for IDS in IoT based on the architecture of the Artificial Neural Networks used in the DL modules. They proposed a taxonomy of solutions, dividing them by the neural modules they are based on. Namely, they divided DL-based IDSes that use Deep Convolutional Networks (Deep CNN), Recurrent Neural Networks (RNN), Multi-layer Perceptrons (MLP), Deep Auto-Encoders (Deep-AE), Generative Adversarial Networks (GAN), among others. Remarkably, they identified common trends with respect to which communication layer the IDSes are built on. They found that most of the literature focuses on the network layer. The distribution configuration of IDSes was also studied, having found that most of the proposed solutions are centralized. Their call for future investigation included seeking real-time intrusion detection and the need for producing more up-to-date and balanced benchmarking datasets with respect to those used by the current literature. Unfortunately, this work did not mention any DRL-based solution for IoT-IDS.

Authors in [47] investigated eighty works published between 2016 and 2021 in the context of IoT-IDS based on ML. After surveying the most common attack vectors in the realm of IoT, they mentioned the ML-based methodologies used in general to detect these threats. They divided the applications by the attack type they try to detect. In other words, they proposed the most used algorithms for detecting Probe, User-to-Root (U2R), Remote-to-User (R2L), and Denial of Service (DoS) attacks. This work raises the need for more generalizable pipelines capable of detecting zero-day attacks. M. Abdullahi et al. also mentioned the problem that many of the datasets used by the proposed solutions may not permit a realistic assessment of these.

Wuhui Chen et al. [27] presented a remarkable survey on DRL-based appli-cations in IoT environments. They divided the surveyed works by the target application field like Smart Grids, Industrial IoTs (I-IoT), blockchain-empowered IoTs, etc. They touched on various goals like traffic management, secure connections, and network configuration inside every application field. Some solutions to the intrusion detection task are reviewed in this work. Authors in [27] also devoted a section to highlight critical points when it comes to implementing DRL-based solutions inside IoT environments. Interestingly, they mentioned the need for more effective and standardized evaluation metrics, the need for safe exploration strategies for DRL, and the need to put more attention to communication offloading and data efficiency. Our research is partly inspired by this work. With respect to the aforementioned survey, however, this work differs in that it concentrates an extensive literature search on the particular task of Intrusion Detection Systems, trying to identify the main motivations, strategies, and challenges related to the adoption of DRL for solving such a task. Having such a specific scope, our work not only aims to systematize the current literature, but also to identify modelization trends, proved good practices, lessons learnt, and open challenges in the particular field of DRL-driven IDS for the IoT.

In the work presented by P. Jayalaxmi et al. [48] instead, a multi-dimensional taxonomy was presented. More specifically, IDSes were divided into network-based and host-based, and also the solutions were divided between anomaly-based and signature-based. The authors considered not only intrusion detection, but also intrusion prevention systems for IoT. They also made the argument that current state-of-the-art (SOTA) solutions lack the ability to detect zero-day attacks and that the false-positive rate of the surveyed works needs to be reduced. Authors in [48] also identified reaching better adaptability to dynamic network scenarios as a requirement that needs further research effort. Application-level monitoring was identified also in this work as a field where large room for improvement may exist. Unfortunately, the authors did not mention DRL-based solutions in their review.

The recent publication in [49] focused on ML-driven IDSes for IoT. They proposed three axes for classifying the surveyed solutions. The first axis is the working principle (i.e., whether they are anomaly-based, signature-based, or hybrid); the second axis is the deploying scheme, and the third axis is the target type of attack the solution excels at detecting. K. Santhosh et al. mentioned the computational overhead and the recall as areas of improvement for current solutions. Additionally, the authors devoted themselves to offering a framework for an IDS that responds to some limitations of the analyzed solutions. This works also lacked analyzing DRL-based solutions for the problem of IoT-IDS.

The work in [50] was focused specifically on DRL-based IDSes. They first divided the literature by applications based on *traditional* or tabular RL frameworks and works using *Deep* RL. Inside each one of these categories, they differentiated between network-based and host-based IDSes. Further, they separated the SOTA solutions based on on-policy and off-policy RL frameworks. Z. Utic et al. noticed that literature focused on IDSes in dynamic environments mainly lacks proposing an effective verification of the optimality of the proposed solutions. Unfortunately, this work analyzed only a relatively low number of recent solutions, i.e., it analyzed less than 10 solutions published after 2019. Our work aims to be a more comprehensive and up-to-date survey and to target the specific field of IoT, rather than general-case IDS.

The work in [51] explored various dimensions in which DRL is used to secure network communications in general. The authors divided applications of DRL devoted to intrusion detection from those focused on intrusion prevention. A section instead focuses on applications developed for IoT scenarios. They also devoted a section for DRL-based identity and access management systems. One important trend mentioned by A.M. Adawadkar et al. is the lack of code accessibility in many of the surveyed works. Authors in [51] also noted that dynamic environments, i.e. environments in which where the distribution of its features may evolve through time, may still present a challenging scenario for many SOTA solutions. Notably, the authors also mentioned the need for standardizing the good practices in the field of RL applied to cyber-security in general. Different from this work, our review aims to specifically focus on the IoT environment when surveying DRL solutions for IDS.

Authors in [52] offered a summary of a selected group of machine-learning-based solutions for IoT-IDS. They divided their solutions into three main groups: supervised-learning-based, DL-based, and federated-learning based. They also included an additional section on solutions based on combined approaches. Authors restricted their literature scouting to those works explicitly focused on IoTs. Unfortunately, they included less than five contributions based on DRL. In our work instead, a more comprehensive review of the DRL paradigm for IoT-IDS is covered.

Perhaps the more similar work to ours is the one in [53], where the solutions for Intrusion detection and prevention systems based on DRL were surveyed. The authors divided the applications concerning the objectives or goals the DRL agents were supposed to optimize. They found multiple goals like binary traffic classification as benign or malicious, multi-class attack classification, hyper-parameter optimization, and more complex goals, like optimizing detection offloading, adversarial attack generation, and sampling, among others. M. Sewak et al. noted that the other promising research directions for DRL-based cybersecurity attain intelligent defense mechanisms, capable of mitigating the effects of intelligent advanced attacks. Inspired by this work, our survey paper aims to offer a more exhaustive and up-to-date survey on DRL applications for intrusion detection. Moreover, our manuscript focuses on the design criteria that enable these techniques to be used in the specific scenario of the Internet of Things.

Table 2 shows the main contribution of our work with respect to recent related surveys.

3.1. Main contribution of this research

The main goal of this survey is to assess the common Deep Reinforcement Learning design trends used for Intrusion Detection in IoT environments. This survey's contributions are the following:

1. An extensive summary of the recent DRL-based solutions for IDS in IoT is offered. For characterizing the state of the art, fifty papers proposed since 2020 are reviewed.

Table 2
Target topics and characteristics of recent related surveys.

Source	IoT	IDS	DRL	Extensive literature	Recent literature
[36]	✓	✓	✗	✓	✓
[47]	✓	✓	✗	✓	✓
[27]	✓	✗	✓	✓	✗
[48]	✓	✓	✗	✓	✓
[49]	✓	✓	✗	✓	✓
[50]	✗	✓	✓	✗	✗
[51]	✓	✗	✓	✓	✓
[52]	✓	✓	✓	✗	✓
[53]	✗	✓	✓	✗	✗
Ours	✓	✓	✓	✓	✓

2. The reviewed works are classified based on the main goal DRL has been used for. Namely, learning to imitate expert detection policies, leveraging adversarial training, federated learning, and high-level goal optimization considering both supervised and unsupervised learning. The best practices and the lessons learnt regarding the most effective delivery and exploitation of DRL in this area are identified.
3. This survey offers an overview of some recent Datasets available to benchmark IoT-IDSes, mentioning their characteristics, advantages, and limitations.
4. Current open areas of research are identified in the field of IoT-IDS and various candidate inductive biases for DRL are identified that could shed light on these challenges.

4. DRL-based IDS for IoT. a State of the Art.

This section presents the most recent works related to IDS in IoT. Our presentation is divided into five modelization categories for which DRL has been used in this context. A summary of the rationale behind such categories follows:

- **Imitation Learning**: Section 4.1 provides a review of the works that follow a *canonical* setting of DRL in which the agent interacts with the environment with actions that are directly related to intrusion detection. Namely, the agent is a mere classifier of environment observations that judges if these correspond to instances of benign network usage or malicious intrusions.
- **Adversarial Learning**: Section 4.2 offers a review of the DRL agents devoted to creating rich and challenging training regimes for ML-based IDSes. These training conditions prove useful to augment the capacity of the intrusion detector over subtle attacks and attacks that might be under-represented in the training set.
- **Federated Learning**: Deep Learning-based Intrusion Detection is a data-driven technique. A subset of the DRL applications over IDS for the IoT concentrates on creating agents whose actions are related to ensuring the privacy preservation of data for the training and operation regimes of IDSes. Recent publications related to such a goal are reviewed in Section 4.3.
- **High-level goal Optimization through supervised learning:** Sections 4.4 and 4.5 are devoted to reviewing DRL agents whose goal is related to other non-functional requirements apart of intrusion detection. Minimization of energy consumption and communication overhead are some examples of such non-functional requirements. However, Section 4.4 offers a review of agents that use pre-recorded labeled traces during training to learn patterns associated with known intrusions.
- **Experience-driven unsupervised learning:** Section 4.5 instead, reviews recent publications that do not rely on supervised learning for training multi-objective intrusion detection agents. In this section, other experience-driven rewards like online network performance or traffic fluctuations are used to train the DRL agents.

Please note that a different taxonomy could have been adopted to classify the recent works devoted to DRL-based intrusion detection in the IoT. This division tries to shed light on a set of design choices that progressively exploit the meta-level learning capabilities inherent to DRL. In other words, by following the review and discussion of the presented works from a *DRL-design* point of view, our taxonomy gives clues on how to progressively incorporate meta-level goals over intrusion detection systems with the help of DRL. Please note also that some methods that have not been designed explicitly for IoT environments may be included in our review. In such a case, though, special care is taken in highlighting key design choices that could allow the application of such models in the IoT realm.

4.1. Imitation learning

The seminal work of Lopez-Martín et al. [54] proposed a straightforward MDP-based meta-modelization for the intrusion detection task, that accommodates it to a supervised classification task. In this mapping, the agent that observes the environment and learns the best classification policy is the detection module, the state space that encodes the observations perceived by the agent is made of the preprocessed records of an available trace, and the action space of the agent corresponds to the set of labels it learns to associate to the input traces. Finally, the rewards are a function exclusively related and proportional to the similarity between the agent's actions and the labels in the annotated dataset.

Note that such a framing for the IDS problem is inherently offline: one tries to learn a policy by observing the state–action pairs of an expert supervisory signal from previously recorded data. Note also that, under some circumstances like having a low-discount factor, the agent may be induced to converge to a greedy classifier to maximize its reward. Such a design choice resembles an imitation learning task. The IDS agent is told to do nothing else than classify traffic, thus, imitating an expert policy provided in pre-collected and annotated data.

Authors in [54] validated the Deep Q-Network (DQN), Double Deep Q-Network (DDQN), Policy Gradient (PG) and Actor-Critic (AC) frameworks using both the NSL-KDD [55] and the AWID v2 [56] datasets. The authors showed the best results were obtained using the DDQN algorithm. The rest of this section reviews some works in the literature that used such a mapping or similar forms of it, and thus, implemented an offline learning strategy for an IoT-IDS DRL-based agent.

One of the most remarkable works following this modelization trend is presented in [57], where the authors used the CICDDoS2019 [58] dataset for proving the effectiveness of the proposed solution. The authors used a Conditional-GAN-based adversarial training strategy that helps to improve the overall robustness of the solution: They generated realistic malicious flow-data samples that enrich and balance the training dataset. (See Section 4.2 for more on adversarial learning strategies in IoT-IDS). Authors used CNNs to automatically extract features from flow-data samples, they then combined these latent flow representations with packet-level data to feed the IDS. The flow-level information is also subjected to a dimensionality reduction technique based on Stacked Auto-Encoders (SAE) before being fed to the IDS agent. Moreover, the authors aimed to also detect novel attacks through separate anomaly-based modules for the packet and flow level features. Bin Yang et al. proposed to use Bayesian search for the hyper-parameter optimization process.

In 2020, H. Benaddi et al. [59] proposed to use a DQN agent with two convolutional layers to optimize the multi-class classification accuracy inside an MDP model very similar to [54]. They assessed their results using the NSL-KDD dataset also. Later, in [60], a stochastic game environment is modeled between a detection-mitigation module and the attacker. The state space is different for each player, but in both cases, it is related to the current state of the network: The network is under attack, because the attack was not detected, or the network is not under attack, either because no attack has been performed or because the mitigation agent prevented the attack to enter the network.

Attacks are given a criticality score by the frequency of occurrence in the trace dataset. They also modeled various mitigation actions and rated them by their cost of actuation. The reward of the defender agent is associated with the defense/cost trade-off, and the reward of the attacker is a function of the network performance decrease caused by undetected attacks. This formulation helps to analyze from a theoretical point of view the different convergence properties of the proposed IDS. The authors implemented the defensive player as a DRL agent and realized a trace-driven experiment with the NSL-KDD dataset. Due to class imbalance, they highlighted the difficulties of effectively sampling and learning to detect under-represented classes in the dataset.

The same authors then proposed to enhance their previous agent in their successive work [61]. This newer DRL agent was trained with a GAN-based data-augmentation and oversampling technique to mitigate the effects of the imbalanced distributions. They also focused on the more recent dataset: the *Distributed Smart Space Orchestration System* (DS2OS) [62]. This dataset contains traces of a real Industrial-IoT (I-IoT) environment. The detection agent is modeled similarly to [60], but a distributional reinforcement learning [63] scheme is used in this work to accelerate convergence in the presence of a stochastic environment.

Authors in [64] also used the NSL-KDD dataset to optimize detection accuracy via optimization of the input record classification. Said Bakhshad et al. focused on the hyper-parameter optimization (HPO) task of the DRL agent even if they lacked specifying the precise DRL framework used. Also, the work in [65] offered an exhaustive discussion about the tuning process of the hyper-parameters of a DQN-alike agent trained in the NSL-KDD dataset.

Another recent work that adopted a similar imitation learning strategy using a DQN agent was presented in [66] where the offline training was done over the CSE-CIC-IDS2018 dataset [67]. Authors in [68] also focused on HPO. They used the sandpiper optimization algorithm to find the optimal learning rate for a DQN agent that seeks to detect intrusions learning from the labels in the UNSW-NB15 dataset [69].

G. Shi and G. He [70] focused also on the NSL-KDD dataset. To improve intrusion detection accuracy, they proposed two types of RL agents: one major agent and a group of minor agents. The major agent had visibility over the whole set of features in the mentioned dataset, while the minor agents learnt to classify traffic as malicious or benign by looking at different feature subsets. The overall strategy can be seen both as an ensemble learning and a regularization technique that helps the major agent to improve its accuracy based on the majoritarian output of minor agents. Both the feature segmentation scheme of minor agents and the number of them were adjusted through extensive trial and error.

Automatic feature extraction is performed by Shi Dong et al. in [71], where the feature space of the NSL-KDD and AWIDv2 datasets are transformed to a latent space via deep auto-encoders. Authors then used a DDQN agent to optimize multi-class classification accuracy following the same MDP setup in [54].

Biswajit Mondal et al. [72] presented two models for an IoT-IDS: The first one was based on a single DQN agent and the second was an ensemble of DQN agents. They also experimented with some tunings of the DQN model like DDQN, distributional RL, dueling architectures [73], and *NoisyNets* [74]. The authors obtained the state space and the supervisory rewards from the KDD-99 dataset [75]. They trained each one of the ensemble agents using small fractions of the dataset, while the single agent was trained with a bigger quantity of data. Under these conditions, they found that the ensemble strategy resulted in a higher convergence rate and recall in general, while the single agent had superior accuracy, especially in the low-represented classes.

Also, Tomás Izquierdo [76] used the KDD-99 dataset for mapping a supervised learning context to a DRL-based IDS. His work aimed to compare a vanilla DQN agent with the DQN agent powered by the *Rainbow* framework [77]. Unsurprisingly, his results showed that the rainbow framework helped to reduce convergence time and to augment overall model performance, especially in the less-represented classes. In this thesis, the author mentioned adversarial training among future work directions.

The work in [78] used a convolutional neural network module [79] inside a DQN agent that categorizes the traffic in the CSE-CIC-IDS2018 dataset. The authors assessed the performance of the agent with multiple state space designs, each one including a different set of features. Kezhou Ren et al. included the investigation of a more dynamic and online method for choosing the optimal subset of features among the future directions of their research.

Authors in [80] also used the Dueling architectures scheme over a simple DQN agent to classify traffic as benign or malicious using the NSL-KDD dataset. Their focus was to create a centralized IDS for a Smart Vehicular Network (SVN) environment where malicious traffic from onboard or roadside units is centrally detected and prevented to spread through the SVN. Interestingly, the authors implemented a training environment based on the OpenAI Gym project which is compatible with many DRL models already implemented. No adversarial or federated learning strategies are implemented in this work.

Authors in [81] concentrated instead on wireless sensor network (WSN) environments. The social optimization algorithm (SOA) [82] and the Spider Monkey Optimization [83] are combined to reduce the converging time of a DQN agent. They learn the optimal classification policy by imitating the labels in the Bot-IoT (2019) [84] and the NSL-KDD datasets.

A summary of the works presented in this section is given in Table 3.

Discussion

This paragraph highlights some significant aspects that emerged during our reflection on the way DRL was used in the works reviewed in this section.

Delivering IDSes based on imitation-learning to real IoTs. The work in [72] mentioned that an IDS pre-trained using a labeled dataset may be difficult to adapt and deploy to a real network scenario: Training a model with many of the mentioned datasets may require assuming that the agent has wide visibility over many node and network features. Such an assumption might be unrealistic under many real IoT and traditional networking scenarios. For example, some features that are visible in the traces might be encrypted in a real context, or not locally available. Moreover, delivering data gathered from various network places to a central agent for training may be ineffective, impractical, or in violation of privacy regulations, especially in the case of IoT.

In this respect, the work in [80] is worth noting: in this work, the authors proposed to deploy the IDS at the *Trusted Authority* component of a smart vehicular network. They assumed a high-performance wired connection exists between roadside units and the trusted authority component. However, the authors did not prove the effectiveness of this proposal on a real deployment.

Table 3
Recent IoT-IDS with the DRL setup in [54].

Source	Year	DRL framework	Dataset
[54]	2020	DDQN	NSL-KDD (1999) AWIDv2 (2016)
[57]	2022	DQN	CICDDoS2019 (2019)
[64]	2022	Not Specified	NSL-KDD (1999)
[71]	2021	Deep-AE + DQN	NSL-KDD (1999) and AWIDv2 (2016)
[66]	2022	DQN	CSE-CIC-IDS2018 (2018)
[59]	2020	DQN	NSL-KDD (1999)
[60] and [65]	2022	DQN	NSL-KDD (1999)
[61]	2022	Distributional DQN	DS2OS (2018)
[76]	2021	Rainbow framework	KDD99 (1999)
[72]	2022	Rainbow framework and Ensemble-DQN.	KDD99 (1999)
[68]	2021	DQN	UNSW-NB15 (2015)
[78]	2022	DQN with CNN layers.	CSE-CIC-IDS2018 (2018)
[80]	2022	Dueling-DQN.	NSL-KDD (2019)
[70]	2021	Ensemble-DDQN	NSL-KDD (1999)
[81]	2022	DQN trained with SMO and SOA	NSL-KDD (1999) and Bot-IoT (2019)

Intrusion detection as behavioral cloning. As mentioned earlier, the works reviewed in this section are mainly adapting a traffic-classification task from a supervised learning paradigm to offline DRL in a way that resembles an imitation learning problem. Indeed, many works mentioned the need of setting a low discount factor to improve detection accuracy. Note that a low discount factor approximates in a greedy *behavioral cloning* setup [85], which is a logical design choice if one assumes no contrast exists between immediate and long-term system protection. The presented works reached a good detection accuracy on many benchmark datasets. Notice, however, that detection accuracy is a metric that is agnostic of the individual risks of intrusions. Risk-aware intrusion detection and prevention may indeed imply the presence of critical states in the MDP chain, and such a condition may raise the need for DRL [86,87].

One of the main assumptions for successfully training an agent using behavioral cloning is that the traces are independent and identically distributed. Some of the reviewed works mentioned that such an assumption does not hold in many IDS datasets. For that reason, many balancing techniques have been applied. More work on this trend will be reviewed in Section 4.2.

DRL for the sake of continuous learning. Many of the reviewed works mentioned the advantage of a continuous learning strategy out of the box through the usage of DRL. However, it is easy to see that, when computing the rewards based on supervised learning, this fact may not hold anymore, or may not be effective enough for a useful deployment in a real network scenario. An interesting design choice is presented in [57] that sheds light on the problem of continuous learning of agents that learnt detection policies through annotated datasets. The authors [57] added an anomaly detection module alongside the DRL agent. Anomaly-based detection is triggered when the statistical confidence of the DRL agent is under a defined threshold.

Conclusion. A straightforward Deep Reinforcement Learning meta-modelization has been widely used by the recent literature to approximate a behavioral cloning setup where the IDS learns to classify network usage instances from a labeled dataset. However, the research community noted that it may be necessary to act balancing, adversarial learning, or other regularization strategies to think of the realistic effectiveness of these types of solutions in IoT. In the following section, additional works will be examined that may have addressed precisely these requests for additional research.

4.2. Adversarial training

As mentioned in Section 4.1, many AI-driven IDSes are trained with trace-driven simulations based on annotated datasets like the well-known NSL-KDD dataset. Many of these benchmark datasets are often unbalanced and the attack patterns of minority classes are difficult to learn by DL-based IDSes [88]. In these cases, adversarial training strategies as defined in Section 2.2 can be implemented by coupling generative strategies with oversampling and undersampling techniques: when the decision boundary of the detection module is coarse enough, the training samples that lay in this boundary might not only be generated but they could be already found in the available training dataset. In fact, these adversarial samples will correspond to under-represented attack classes.

In this respect, it is worth remarking on the seminal work of Caminero et al. [89], who proposed two DDQN agents with contrastive rewards: the detector agent learnt to classify traffic samples from the NSL-KDD dataset as malicious or benign, while the environment agent learnt to sample from the dataset the traces that tend to be misclassified by the former agent. This section reviews some recent works that followed such an adversarial training strategy for IoT-IDS.

First, authors in [90] created intelligent sampling techniques with the help of DRL. They modeled a robust IDS trained within a Double Deep Q Network (DDQN) framework. X. Ma and W. Shi took the adversarial training strategy in [89] and enhanced it by coupling this technique with the SMOTE [91] generative oversampling technique. SMOTE is the acronym of Synthetic Minority Over-sampling TEchnique, and is a framework that permits to create synthetic training samples for under-represented classes through interpolation between same-class instances in the feature space. As in [89], the rewards of the sampling agent are proportional to the correspondent miss-classification of the detection agent. Notice that by shaping the rewards with this contrastive setting, the sampling agent helps to converge to more robust detection policies.

Later, the work of E. Suwannalai and C. Polprasert [92] adapted the same adversarial training strategy in [89] but, rather than using SMOTE, they proposed to enhance the robustness of the detection agent by using a hierarchical low-level ensemble learning strategy: the first level contains an ensemble of shallow classifiers managed by a DQN agent, and the second level consists of an ensemble of DQN agents that help reducing convergence time at the edge of the network.

Also, the authors in [93] modeled the same adversarial training published in [89], but rather than using SMOTE or ensemble learning, the optimal policy achieved by the DQN agent was found with the help of the Dueling architectural inductive bias [73].

In [94] a pre-trained Generative Adversarial Network is used to create adversarial samples for training a classifier. The authors trained an agent based on the Deep Deterministic Policy Gradient framework to sample the latent-space regions of the generator that correspond to the most effective adversarial samples in terms of semantic preservation and similarity with real benign samples. Authors get inspired by the close connection between GANs and stateless Actor-Critic methods mentioned in [95]. Jun Tu et al. tested their approach with the UNSW-NB15 [69] dataset. Note that, in future works, this approach could be assessed within a more recent dataset to leverage the same data augmentation and minority class re-balancing in newer and more IoT-focused scenarios.

Juan Parras et al. [96] used Generative Adversarial Imitation Learning (GAIL) to create a defense mechanism against intelligent attackers performing back-off attacks in a wireless network that uses the CSMA/CA mechanism [97]. Both an offline and an online learning framework were proposed to learn the distribution of the data transmission policy of legitimate network nodes and learn to distinguish good stations from attacker stations by the differences in such a distribution. The authors indicated that future work directions could include improving the recall and also taking into account non-stationary environments. Remarkably, the solution in [97] is robust against multiple concurrent DRL-based attackers that exploit the Proximal Policy Optimization (PPO) framework to evade detection. A prior study in [98] presented the modeling of such an intelligent attacker.

Appruzzese et al. [99] proposed to use DRL to synthesize the most subtle adversarial samples in the context of flow-based Botnet detectors. They selected a small subset of features from traffic flow records under which low entity variations can represent malicious traffic that evades baseline classifiers based on shallow ML. A DDQN agent is leveraged to construct an augmented supervised trace dataset that augments the robustness of the detection module which is based on Random Forest and Wide&Deep classifiers. An online learning strategy could facilitate the deployment of this solution over real IoT-based scenarios.

In [100], Q.D Ngo et al. proposed an IoT-IDS that learns to recognize Botnet-related malware through static analysis of the source code. More specifically, the authors used the PSI-Graph method in [101] to encode the behavior of a program into a graph representing function calls. These graphs are encoded into a latent space using Graph2Vec [102], and a shallow ML-based detection module is trained in a supervised learning fashion to classify these graphs as benign or malicious. In this work, an RL-driven strategy was presented for adversarial training of the classifier: A Q-learning agent learned to create adversarial samples for augmenting the robustness of the IoT Botnet detector. The Q-learning agent seeked to optimize the process of crafting adversarial samples by minimizing the number and the entity of alterations needed for a malignant graph to fool the detector module.

The previously mentioned RL-adversarial learning framework for IoT-Botnet detection is extended by Q.D Ngo and Q.H Nguyen to dynamic source code analysis in [103]. By doing this, the authors enriched the feature engineering further to cope with malware detection when code is obfuscated. They also stated that dynamic analysis helps to catch the trigger point of IoT Botnet. In this work, the detection module is trained to classify System-Call Graphs (SCG) as benign or malignant. The goal of this refined adversarial learning strategy is to re-train the detector with adversarial samples and improve the classification of zero-day malware attacks.

M. Ibrahim and R. Elhafiz [104] adopted also graph modelization. Their work focused on Integrated Clinical Environments. The authors first constructed an attack vulnerability graph that resembles the environment. Then, Q-learning is used to determine the best route that a hypothetical attacker could take to damage the system as much as possible with the least number of actions. Numeric values are assigned to the attack graph to map vulnerability. As a future work, the authors suggested exploring dynamic attack mitigation and scaling the paradigm to bigger graphs. In later work, [105] the same authors experimented with the SARSA framework [106] for creating the same type of intelligent attacks, this time focused on wireless sensor network scenarios. These authors based both their works on a previously published framework for attack graph implementation and visualization for cyber–physical systems [107].

Discussion

This section explored DRL-based optimization of adversarial strategies for better training intrusion detection systems for the IoT. The reader can find a summary in Table 4. This paragraph mentions some interesting design-related remarks observed in these works.

DRL is not always mandatory for adversarial training. Adversarial sampling techniques and synthetic data generation can be done without DRL if one assumes to know the detector model. Visibility over the mapping between inputs and outputs of an IDS can be used to craft effective adversarial samples. Works like [108,109] use, for example, the Jacobian-based Saliency Map Attack (JSMA) [110] to generate adversarial attacks against an IDS. In Ref. [109], for example, the IDS model is not known *a-priori*, but the model is queried with many sample inputs to construct a known model substitute that permits the application JSMA.

Black-box adversarial training. However, in cases where enough knowledge of the detection model is not assumed, minimizing the divergence between the representations of malicious and benign instances may be a task particularly suited for DRL. In other words, DRL may be mandatory when one pursues online *black-box* adversarial training.

There may be also other cases in which DRL could be a simpler adversarial learning strategy compared to direct gradient-based techniques. One example is when one seeks to preserve the semantic value of the generated samples as highlighted in [94]. In such work, the authors were constrained to quantize or discretize the entity of feature perturbations to preserve semantic meaning in the input samples. With this design setup, a non-differentiable loss function may be necessary to train an adversarial sample generator.

Intelligent reward shaping. For effective adversarial strategies to be guided by DRL, special care must be placed in the design of the reward function: In many of the reviewed works, the DRL agent's actions are related to introducing slight perturbations on the input samples to restrict the width of the decision boundaries of the IDS under training. In these works, an agent's training episode ends when the action taken by the adversary agent results in the generation of a sample that fools the discriminator module. Only this final step results in a greater positive reward for the adversary agent, while in the rest of the cases, a negative reward is given to it. This mostly-negative reward shaping induces the agent to perform the minimum number of alterations to produce adversarial samples, which is in line with the definition of adversarial attacks.

Conclusion. Deep Reinforcement Learning has been used recently to improve the robustness of IDSes against adversarial attacks in various ways. In some cases, an MDP meta-model frames the efficient synthesis of adversarial samples. DRL may be necessary in cases where one does not know the input–output mapping of the detection module. If one is not under a supervised learning con text, however, the training dataset may be unbalanced as attacks are less frequent than legitimate traffic. In these other cases, DRL has been used to efficiently sample unbalanced training datasets to cope with the assumptions that permit learning intrusion detection through supervised learning. However, re-sampling may not be the unique way of addressing convergence issues in supervised learning-guided intrusion detection. Even though they may be less efficient, ensemble techniques may also address this issue. Some ensemble techniques could also be easily coupled with Federated Learning schemes. Some distributed and FL strategies in DRL-based IoT-IDSes will be reviewed in the next section.

4.3. Distributed and federated learning

The research community focused on IoT-IDS has put attention to ensemble learning techniques to enhance the optimality and convergence time of the detection policies. Also, distributed learning has been targeted by some recent works to preserve the confidentiality of sensitive information that is commonly managed by the IoTs, as explained in Section 2.2. This section now shows how recent DRL-based IoT-IDSes have been coupled with ensemble learning or FL schemes and how DRL itself has been used in this application context to create some training strategies that are based on ensemble or federated learning.

Table 4
Recent DRL-based adversarial training strategies for IoT-IDS.

Source	Year	Adversarial framework	Application context	Future work
[94]	2022	GAN based. Actor-Critic DRL controls the complexity of the GAN latent space	WSN	Validation in a higher-scale environment and with a more recent dataset.
[96]	2022	Generative Adversarial Imitation Learning for inferring the behavior of DRL-based intelligent attackers	CSMA/CD networks	Improving the recall and non-stationarity awareness.
[99]	2020	DRL-driven adversarial attacks for robust flow-based analysis	BotNet detection	Extension to other attack-types, Online learning
[90]	2020	DRL-based sampling combined with SMOTE synthetic data generation techniques	general-case traffic inspection	Explore Distributed Learning, refine the difficulty categorization of trace-samples
[92]	2020	DRL-based sampling combined with ensemble learning	general-case traffic inspection	Improve performance on R2L and Probe attacks.
[93]	2021	DRL-based sampling combined with a Dueling-DDQN IDS	general-case traffic inspection	Preserve data privacy during training
[100]	2021	Q-learning based Adversarial strategy for static-code analysis	Malware classification	Dynamic source-code analysis, detection of other attack-types
[103]	2022	As previous + dynamic-code analysis related features.	Malware classification	Extending the applicability scenario with more data.
[104]	2022	Q-learning-based optimal Attack graph exploitation	IoMT	Scaling solution, Implementing intelligent detection and defense.
[105]	2023	SARSA-based optimal Attack graph exploitation	WSN	Scaling solution, Implementing intelligent detection and defense

The work in [108] deployed various DRL agents in the subnet routers of a wireless network. These agents were designed to assess the classifications of an ensemble of shallow ML classifiers using DQN. Each DQN agent was trained by a centralized supervisory signal and applied a voting mechanism over the candidate results from all the agents under the same gateway router. The experience buffers of each DQN agent were used to online train the ensemble of shallow classifiers. A very similar training strategy exists in [92], a work already mentioned in Section 4.2. Sethi et al., however, introduced also a human-based supervisory signal that interacted with the overall system training.

In [109], a more sophisticated attention mechanism is created that learns to give different importance weights to the leaf agents' outcomes before aggregating them. This work also mentions model robustness gains thanks to this ensemble learning technique. Authors demonstrated by experimentation the improved reactive times for learning to detect previously unknown attacks when there are multiple agents learning at the same time. As a future work, the authors in [109] planned to research case-specific feature selection with the aim of reducing the communication and computation overhead in the overall IDS. Such a goal may be important for coping with the low resource availability regime of IoTs.

The work in [111] presented a continuous learning strategy in the context of IDS for a network of Unmanned Aerial Vehicles (UAV). The authors dealt with the constrained resource conditions by setting up a periodic update of the models' weights. DRL is used here to optimize the attack detection accuracy, and the DRL agents were deployed inside the UAVs. However, Omar Bouhamed et al. stated that online learning would be an energy-inefficient task and thus be infeasible in a UAV environment. They proposed a central agent with fewer resource constraints that keeps training using the data collected from the whole swarm of UAVs through cloud-computing services. The DRL model's weights will be updated each time the agent returns to a docking station to recharge its battery.

Tria Nguyen et al. implemented an FL strategy on an SDN-enabled IoT environment [37]. In this work, DDQN-based agents learned the optimal flow rule match-field control mechanism to keep the granularity of flow monitoring as fine as possible while proactively preventing the overflow of the routers' flow tables. Maximizing the detail level of flow statistics can help the IDS to detect malicious intrusions into the system. Interestingly, the FL strategy here was achieved by keeping local traffic information untouched and sending only the model weights to the SDN controller for aggregation. The aggregated model weights are then sent to the so-called DeepMonitor agents at the edge routers of the IoT.

In [112], DRL was used to optimize a more complex goal directly related to the FL strategy for an IDS focused on Industrial-IoTs: A centralized DQN agent is responsible for choosing the optimal subset of leaf classifiers that will participate in the update of the global model. The rewards of the centralized agent are related to accuracy gains on a held-out validation trace. The novelty of this work with respect to [37], consist of a GAN-based oversampling method inside each leaf agent to mitigate the effects of heterogeneity and unbalanced traces in each local environment. Nguyen et al. validated their model with the Kitsune IoT dataset (2018) [113].

A brief summary of the mentioned works is presented in Table 5.

Discussion

Ensemble learning may help to reduce communication overhead. A positive feature of the ensemble strategies presented in [108,109] is the mitigation of communication overhead: the whole parameter set of multiple models is not sent through the network back and forward to update the models. Instead, they only send the training input and their candidate outputs and receive an ensemble-aware supervisory signal to perform the learning locally. Another important feature of these works is that they leverage distributed detection to infer intrusions at the edge of IoT once the whole system is trained.

Federated learning may be more convenient with respect to ensemble learning in iot scenarios. Ensemble learning does not automatically imply federated learning. In [108,109,111] distributed learning scenarios were modeled, but authors lack to create an FL strategy. The information sharing between the leaf agents and the centralized module depicted in [108,109] may augment the risk of privacy leakage and information theft. On the other hand, research efforts like the one in [37,112] are focused on leveraging FL strategies that mitigate such risk. In these works, FL strategies for IoT-IDS were presented where DRL was used to optimize more complex functions that are directly related to the IDS accuracy while taking into account also privacy, robustness, and convergence efficiency.

By communicating the parameters of the model rather than the input data, the authors in [112] preserved privacy and shared expertise between the leaf intrusion detectors at the same time. Note also that the work in [112] results in an online distributed training setup where agents are trained in their local environments, rather than in the same environment, as vanilla ensemble methods would assume. Distributed training helps also to enrich the robustness of the system because agents learn from multiple local contexts which may be highly heterogeneous in IoT environments.

Table 5
Distributed detection & Federated learning in DRL-based IoT-IDS. Recent works.

Source	Year	Main contribution	Application context	Federated learning	Future work
[108]	2020	2-level ensemble classification. Inner level: shallow ML classifiers, Outer level: DQN.	Wireless Networks	No	Include an FL strategy, deploy shallow classifiers in edge IoT devices.
[109]	2021	As previous + attention-based mechanism for the outer DQN ensemble.	Wireless Networks	No	As previous + dynamic feature selection scheme
[111]	2021	Energy-consumption optimization through periodic model updates.	UAV	No	FL strategy
[37]	2021	DDQN agents optimize flow monitoring granularity	SDN-enabled IoT	Yes	Communication overhead could be further optimized
[112]	2022	DRL-driven FL strategy with communication overhead optimization	I-IoT	Yes	Securing and enlightening the weights communication process.

DRL-driven federated learning. The work in [112] partly imported a lightweight FL framework presented in [114]. In this remarkable framework, DRL has been used for the minimization of the communication overhead itself. The actions of the agent influence the flow of information in the network during the training phase of a set of IDS models. The reward shaping is done in such a way that the agent learns to minimize the training -and communication- rounds and maximize the model convergence speed. The authors state that each leaf model has partial visibility over the environment and thus models may have different parameter distributions after some training rounds. Note this condition may be particularly accentuated in IoT environments [115].

Authors in [114] do not send the whole learning weights to a centralized aggregator, but only the differences in the models' weights after a training episode are sent to the centralized DQN agent. The state space of the agent is the set of models' weights, which may be prohibitively high-dimensional, thus, authors use dimensionality reduction techniques to make a lighter state space. The agent is trained to choose a set of models that augment the overall performance the most without having direct visibility on the data they are trained on. Future research could lead to better exploiting the work in [114] for leveraging FL strategies to train IoT-IDSes.

Asynchronous deep reinforcement learning may help to design FL strategies. Our discussion on federated learning cannot end without mentioning the automatic advantages that the A3C framework could bring in terms of providing a federated learning strategy. Note, in this respect, the works in [116,117] that used such a DRL framework. These works did not mention the advantage of having federated learning out of the box.

Conclusion. Various works have recently coupled DRL with distributed learning and ensemble learning for IoT-IDS. Not all of them have leveraged also FL strategies that preserve information privacy at the edge of IoTs. Other works instead have used DRL to create FL strategies for IoT-IDS. Remarkably, while guaranteeing distributed detection and FL, some other works have addressed the minimization of computation overhead with the help of DRL. Communication overhead minimization and other high-level goals are problems that may only be effectively faced through DRL. Similarly, other high-level goals had been targeted by researchers while developing IoT-IDSes with DRL and the next section will explore precisely such goals.

4.4. High-level goal optimization through supervised DRL

When surveying the state of the art on DRL-based IoT-IDS, some works exist in which, even if the rewards are a function of supervisory signals obtained from the labels of an intrusion detection dataset, the authors do not use the canonical mapping proposed in [54]. In this second group of works, *reinforcement learning via supervised learning* [87] is also used, but a higher level of complexity attains the specific optimization goal that the DRL agents pursue. More in detail, rather than being related to the assessment of the state space samples as malicious or benign, the actions of the DRL agent are related to the parameters of the assessment function itself. In other words, the DRL agent is not directly related to intrusion detection, but to the *adaptive behavior* of the detection system. In this setting, the detection system changes itself to keep a satisfactory accuracy or performance in front of changes in environmental conditions. Moreover, these conditions might potentially be related not only to the incoming traffic and attacks, but they could concern the current energy, computation, and network resource availability, among other important IoT-related features. This section mentions the most recent works following this design choice and provides a correspondent summary in Table 6.

In [118] authors delivered an IoT-IDS where the actions of a Q-learning-based agent are to adjust the anomaly detection threshold that triggers alarms during a fixed period. The reward signal is a function of the gain introduced by such a threshold in terms of detected threats and reduction of false alarms. Tianbo Gu et al. noted that using the DRL permitted them to guarantee the continuous learning of the system. The authors of [118] proposed to deploy the IDS both in the IoT gateways and in the edge devices, and focused on flow-based attack detection. For these reasons, they used information theory to engineer entropy-based metrics for anomaly characterization. They argued that such metrics could lead to more robust, efficient, and lightweight anomaly detection. Their experiments used a 2019 real IoT dataset presented in [119].

A similar adaptive behavior is pursued by the work in [120], which focuses on green-IoT environments characterized by large traffic fluctuations. They leverage two DDPG-based agents. The first one learns to predict the upcoming flow characteristics of the transport layer and application layer. Its action space consists of the predicted statistical descriptors for the traffic flows and the rewards are inversely proportional to the distance with the measured traffic. The second agent instead uses the predictions of the first to detect Distributed Denial-of-Service (DDoS) attacks. The action of the detection agent is a classification threshold parameter that assesses if the difference between the predicted and observed traffic is too big to be considered benign. Laisen Nie et al. trained these agents in an offline fashion using the CICDDoS2019 dataset [58]. Augmenting the accuracy of intrusion detection over more complex environments is among the future works of the authors.

Source [121] specialized in detecting Peer-to-Peer (P2P) Botnet members. The authors proposed a three-step process to create their solution. First, DL-based classifiers were trained offline in a supervised learning fashion using the ISOT (2011) [125], ISCX (2012) [126], and Peerrush (2014) [127] datasets. M. Alauthman et al. used the classification and regression tree (CART) [128] in the feature selection process. Then, a value iteration RL strategy is leveraged to select the optimal curriculum for re-training the DL classifier.

In [122], a DRL agent based in the Proximal Policy Optimization framework is used to optimize the dimension of the latent space of a Deep ANN-based feature extractor. The latent space samples are fed to an anomaly-based IDS implemented using k-means clustering. Authors assessed the effectiveness of their approach in the CICIDS2017 [67] and the UNSW-NB15 datasets. The authors noted that reducing the convergence time of the optimal policy could be among future works.

Table 6
Recent IoT-IDS based on High-level goal optimization through Supervised DRL.

Source	Year	Description	Dataset	Future work
[118]	2020	Q-learning to optimize the anomaly detection threshold parameter in flow data	SDN-IoT (2019)[119]	Adding transport and application layer monitoring
[120]	2021	DDPG-based agents for optimizing traffic fluctuations prediction and security assessment in Green-IoTs	CICDDoS2019 (2019)	Improving detection accuracy
[121]	2020	Value-Iteration for optimal curricula Learning in P2P BotNet detection	ISOT (2011), ISCX (2012), Peerrush (2014)	Export to other network scenarios.
[122]	2022	PPO-based latent-space dimension and bening-class number optimization	CICIDS (2017), UNSW-NB (2015)	Reduce policy convergence time.
[123]	2022	Anomaly detection through GNNs in the latent-space of traffic-flows + TD3-based optimal assessment of detectors' outputs in CPS	CSECIC-IDS (2018)	Augment detection accuracy.
[124]	2021	DQN to optimize in terms of detection accuracy and power savings the subset and frequency of feature probing from WSN nodes	None	Implement the proposed framework

The remarkable work in [123] presented an IDS for Cyber-Physical Systems. Q. Lin et al. designed a topology-aware method that automatically extracted the salient features for IDS using variational graph auto-encoders [129] (VGAE). Authors used this generative method to learn representations of the local and global network contexts. These representations take into account the local topology of network exchanges, the features related to the content of communication, and the network nodes' inner features. Two evaluation modules based on graph convolutional networks [130] ingest such representations to give a confidence score of the current observation representing an intrusion or normal traffic.

A DRL agent based on a Twin-Delayed Deep Deterministic Policy Gradient (TD3) is trained to give the correct weight to the local and global scores for a weighted summation that determines the final classification. The IDS agent is supposed to ban malicious traffic upon detection. The VGAE and the GNN-based modules are pre-trained using supervised learning based on the CSECIC-IDS2018 dataset. The authors in [123] shaped the rewards of the DRL module taking into account the difference between benign and malicious traffic inside the network.

Authors of [124] proposed a framework in which the optimization goal is related to the set of attributes to probe and the probing frequency from the edge of a wireless sensor network. The optimization takes into account not only the accuracy gains in intrusion detection, but also dealing with constrained power at the leaf nodes of the network. By doing so, the authors present an automatic feature selection framework based on DRL. The reward term that assesses the efficacy of intrusion detection is an approximation based on a pre-trained shallow anomaly detection module. Unfortunately, S. Frikha et al. did not implement the proposed framework.

Discussion

The benefits introduced by the works presented in this section are now explained. More specifically, this subsection highlights important design characteristics and the main advantages of high-level goal optimization compared to the imitation learning setup presented in Section 4.1.

Limited network visibility. When studying the works cited in this section, one realizes that a requirement of a good reward policy for IoT-IDS might be the awareness of global network features. However, significant network and node statistics for the assessment of intrusion detection systems may be difficult to compute at the learning nodes when these have limited network visibility, like in many IoT contexts. A good candidate solution for these difficulties may be the work in [123], where the authors shaped the reward function as having two components. The first component was a dense signal related to local network features, and the second was a more sparse and global supervisory signal at the upper levels of the network.

Note that hierarchical reward shaping may help the convergence while reducing communication overhead at the edge of the network. The convergence rate, lightweightness, and optimality of IoT-IDSes might all be enhanced by additional research efforts aimed at carefully designing similar reward mechanisms in this context.

DRL for pursuing adaptiveness. The optimization objective is put in a greater level of complexity in the offline-learning strategies mentioned in this section. Authors in [118] optimized the *alarm threshold adaptation* over environment observations, and thus facilitated the detection of dynamic intrusion patterns, which may heavily characterize IoT. Moreover, by using entropy-based anomaly detection, the need for adversarial training may be bypassed in this work, at least for adversarial attacks that adapt their own spatiotemporal properties.

In the same vein, by modifying the dimension of the latent space through DRL, the authors in [122] helped the overall dynamic adaptiveness of an IDS. In other words, this work may indicate that it might be crucial to design intelligent adaptive schemes to cope with changes not only in the distribution of the feature space, but in the entity of its entropy, which translated on the need of dynamic dimensionality of the correspondent latent space.

DRL for pursuing reliability. Tianbo Gu et al. [118] mentioned that a requisite that could impose the usage of DRL in their work is the need for increasing the system's reliability in the long run. Reliability is a high-level goal that may be non-trivial to learn through supervised learning. This goal may be more practically assessed by a corresponding measurable feature in the context of a simulation. Note that system reliability could be coupled with intrusion detection accuracy in a multi-objective reward shaping. Reliability-aware IoT-IDS may still be an open area of research. However, encouraging signs exist in this respect: other problems in the IoT ecosystem like routing, resource allocation, and forwarding graph embedding have been addressed in a multi-objective fashion taking into account both Quality of Service and reliability [27].

DRL for pursuing lightweightness. Even if the work in [121] is not strictly focused on IoT scenarios, the proposed DRL-based *curricula learning* approach might be worth experimenting on IoT-IDSes taking into account the high-heterogeneity of attacks in these environments. Curricula learning may be used not only to accelerate the convergence process of the detection agents, but as an HPO technique that optimizes the feature selection process to minimize the processing consumption of detection modules that may be deployed at the edge of IoTs.

Conclusion. "One can solve any problem by introducing an extra level of indirection."[1] There is a recent research trend where DRL is used for multi-objective optimization in IoT scenarios: Researchers are leveraging *X-aware* intrusion detection systems to the IoTs using labeled

[1] Sentence attributed to David J. Wheeler, later coined as the fundamental theorem of software engineering.

Table 7
Recent DRL-based IoT-IDS with Unsupervised learning.

Source	Year	Description	Application context	Future work
[116]	2021	Offline-pretrained A3C/based IDS through anomaly detection in TDP handshake-related flow data	Opportunistic IoTs	Generalize to other IoT-related scenarios and attack-types (*Microsoft Malware Prediction* DS trace-driven validations)
[117]	2022	Optimize secure routing via trustworthy forwarder node selection.	Opportunistic IoTs	Generalize to other IoT-related scenarios and attack-types (*Microsoft Malware Prediction* DS trace-driven validations)
[131]	2022	Isolate Selective Forwarding attacks by multi-objective optimal forwarder-node selection	IoMT	Reduce the exploration inductive Bias (Relax the Dijsktra constraint)
[132]	2022	Offline-pretrained DDQN with LSTM and Prioritized Experience Replay and human-based sparse rewards	Could be adapted to multiple IoT scenarios	Validate on a more realistic context (*NSL-KDD* DS trace-driven validations)
[133]	2020	DDQN-based traffic flow matching control optimization	General DDoS detection	Detection of other attack-types.
[134]	2022	Non-stationary MAB-based HPO for iForest anomaly detection	Smart Home	Extending to other IoT related Scenarios

datasets as supervisory signals in the offline learning phase. In this phase, trace-driven simulations are modeled where the input traffic obeys the distribution of a pre-recorded trace and the actions of the agent alter the intrusion system's behavior to meet functional and non-functional requirements with respect to intrusion detection. The systems' reliability, adaptiveness, and lightweightness are taken into account alongside the detection accuracy thanks to the potentialities of the RL meta-model, the approximation capabilities of Deep ANNs, and the intrusion detection expertise encoded in labeled pre-recorded traces. There might be a large room for expanding this research direction.

However, it is worth noting that it may be non-trivial to model high-level multi-objective optimization tasks alongside a supervised-learning context. It is evident how the works reviewed in this section have meticulously combined supervised and unsupervised rewards to converge to the intended multi-objective IDS in trace-driven simulations. Other works exist instead, where simpler measurable rewards are proposed for X-aware intrusion detection based on purely-unsupervised learning. Such a design trend is explored in the next section.

4.5. Unsupervised IoT-IDS based on DRL

This section presents some works where IoT-IDSes are meant to be online trained in a real deployment or in a realistic simulation using rewards provided by the environment rather than from pre-defined annotations. In other words, the following approaches measure the convenience of the intrusion detection system by solely sensing network statistics related to performance and service availability. These statistics are designed using expert domain knowledge to approximately assess the intrusion detection accuracy alongside many other high-level goals like the ones mentioned in Section 4.4.

The work in [116] focused on cognitive network security and propounded to extract features related to network activities through service discovery tools. The authors of this work optimized the classification accuracy of incoming state samples as malicious or benign. E. Muhati and D. Rawat computed the rewards of the DRL agent combining several flow-related measurements derived from the knowledge of the TCP handshake normal flow. Irregularities in this context permitted the system to detect the presence of an attack in action and shape the rewards accordingly. The DRL agent is trained using an Asynchronous Advantage Actor-critic framework (A3C) [19].

Authors in [117] used the Microsoft Malware Prediction Dataset [135] to assess a DRL agent based on the A3C framework to intelligently prevent attacks in an Opportunistic IoT scenario. N. Kandhoul and S.K. Dhurandhet studied a set of patterns in the communication of D-DoS, Hello-flood, and Synkhole attackers and designed a DRL-driven routing scheme that prevents forwarding packets to malicious nodes. They created a state space that comprises historical features related to the communication of each node like the number of packets forwarded and the time spent in sending packets. The actions of the agent are related to choosing forwarding nodes, while the rewards are modeled taking into account resulting traffic measures. However, it is not clear how authors in [117] propose to warrant the authenticity of the historical feature vector that each node that carries.

Similarly, authors in [131] used a Q-learning-based agent to optimize forwarder-node selection in a WSN environment. The goal pursued was to identify Selective Forwarding attacks. The rewards are purely associated with network performance and power metrics collected at the nodes. The agent is helped to converge by the usage of Dijkstra's algorithm and the authors assessed the effectiveness of the solution under a simulated Internet of Medical Things (IoMT) scenario. A future work direction may be experimenting with a larger and less constrained action space.

Ze Liu proposed to combine human supervisory signals with automatic rewards in a DRL-based IDS in [132]. The detection agent uses DDQN with prioritized experience replay and an LSTM module inside the Q-network to capture time-series dependencies in the state space. In this approach, the reward is a weighted sum of two functions. The first is automatic and proportional to a set of network performance indicators, while the second is a more sparse reward given by a human supervisor. The supervisor has visibility over the global network performance indicators and is in charge of correcting the misclassifications of the agent. The author of this work, however, stated that online learning may not be effective for realistic intrusion detection. Consequently, he proposed to adopt supervised learning for pre-training the DRL agent using the NSL-KDD dataset.

The work in [133] focused on SDN environments. The DDQN agent is trained to indirectly gain accuracy in IDS detection and mitigation through a multi-objective setting involving the maximization of traffic-flow control granularity while preventing the flow-control tables of SDN routers to overload. To assess the advantages of their solution, the authors simulated a real SDN network environment using the MaxiNet framework [136]. The simulated environment was used also to pre-train an MLP-based anomaly detection module that helps to approximately assess the attack detection performance. The authors demonstrated that maximizing the traffic monitoring granularity through DRL achieves better attack detection accuracy with respect to other traffic flow matching control mechanisms. Trung Phan et al. excel at detecting DDoS attacks and their future work focuses on improving detection accuracy over other attack types.

The authors in [134] used an agent based on the non-stationary multi-armed bandit (MAB) framework for optimizing the value of the *contamination hyperparameter* of an isolation Forest (i-Forest) model. The latter is an anomaly detection approach based on shallow ML that outputs soft clusterings of the input population of samples. The reward is totally unsupervised and non-differentiable as it is a function of each clustering's silhouette score. The authors experimented with the proposed solution in a simulated smart home environment. Future work directions included expanding the simulated attacks and the simulation scenarios. Tariq Z. et al. made the argument that the usage of shallow ML approaches may be efficient and an enabler of real-time intrusion detection.

Discussion

The works reviewed in this section are summarized in Table 7. This subsection mentions the advantages and disadvantages of unsupervised DRL-based IoT-IDSes and puts in evidence open research paths in this field.

Approximated assessment of detection accuracy. While examining the works in this section, it is noticed that the part of the reward signal that is related to intrusion detection accuracy is purely unsupervised: Network and node statistics are combined using expert domain knowledge to permit dynamic anomaly-based classification of the state space as benign or malign. However, anomaly-based intrusion detection is known to be inherently prone to have a low recall [36]. The work in [132] advocated for the inclusion of human-based rewards probably for this reason. However, the effective prioritization of sparse human rewards in an online DRL-based intrusion detection scenario might be an open area of research.

Online learning and safe exploration. From a theoretical point of view, the main advantage of the works presented in this section is that the correspondent rewards do not need to follow a pre-recorded trace, and neither they are based on offline traffic annotation. For this reason, online intrusion detection can be leveraged to the IoT following the reward shaping proposed in these works. However, it is easy to notice that online learning in real deployments might imply security risks. In this respect, it is noted that safe exploration strategies for reinforcement learning may help to pre-train IoT-IDSes and reduce such risks. Leveraging safe exploration for RL to IoT-IDS may be a worth-to-explore research path.

Combination with offline supervised learning. The work in [132] proposed to combine offline pre-training of a DRL-based IDS in a supervised-learning fashion with online fine-tuning of the system after deployment. Notice that such a proposal can be seen as a safe exploration strategy itself. Notice also that this setting is a continuous learning strategy. However, such a design setup might imply carefully designing the feature pre-processing phase to achieve compatibility between the state space obtained from the supervised traces and the one encountered in unsupervised environments. Standardizing the features of supervised IoT-IDS pre-recorded traces is a promising area of research that may help the research community to leverage pre-trained intrusion detectors that mitigate exploration risks and guarantee their continuous learning in unsupervised real deployments.

Software defined networking as a game-changing technology. In Section 4.4, it was mentioned that a compound reward shaping might help to improve the convergence rate of IoT-IDS taking into account the problem of low feature visibility, which is inherent to IoT scenarios. Apart from sophisticated reward policies, some recent approaches based on network function virtualization (NFV) may also help to solve the aforementioned problem. This is the case of Software-Defined IoT (SD-IoT), which is the result of using Software-Defined Networking (SDN) technologies for designing IoT infrastructures. Through NFV, SDN is known to assist create global awareness of network conditions by separating the control and forwarding plane of IoT. Many works related to the management of IoTs are basing their proposed models on this technology [137–139].

The work in [133] presented an IDS for SD-IoTs. In this work, the authors modeled a reward function combining two terms: the first one is proportional to the degree of granularity of flow-traffic monitoring, while the second one is related to the long-term maximization of forwarding performance. However, SD-IoTs may enable the design of many other multi-objective global-aware rewards. At the time of writing, there might be a lack of attention in the research community concerning the benefits of SD-IoTs for DRL-based intrusion detection systems.

Conclusion. Some literature has recently been devoted to optimizing intrusion detection accuracy through unsupervised DRL. Learning optimal control policies through experience is related to reinforcement learning by definition. Thus, the works discussed in this section could be said to employ DRL in a more natural way compared to the works presented in Section 4.1 that set up an imitation learning problem or the works in 4.4 that used supervised learning for multi-objective optimization.

However, some difficulties related to feature visibility may hamper efficient reward shaping of experience-driven IDSes. Software-Defined Networking has become a key technology that makes it possible to deploy online-trained intrusion detection systems for SD-IoT scenarios through well-designed reward functions.

Some researchers are advocating for a two-phase training design of DRL-driven intrusion detection systems: The first phase will be based on supervised learning and would pre-train the detection agent with expert pre-recorded policies, while the second phase instead would be based on anomaly-based unsupervised detection. In this respect, the next section mentions recent IoT-IDS labeled datasets that could help the research community to design the pre-training phase of modern IoT-IDSes.

4.6. Recent datasets

The data sources most commonly used in the surveyed literature for IoT-IDS benchmarking have been published before 2019, we refer the interested reader to [144] for an exhaustive study of these datasets, their main advantages and limitations. As we mentioned in Section 3, recent surveys and reviews in the field of IDS have put into evidence the need for more modern datasets and benchmarks for this particular task [36,47]. We now concentrate on mentioning some characteristics of more up-to-date data sources that can be used by new works in the field of IoT-IDS. Please note that we have found many other recent annotated datasets for IDS, but we have selected the most heterogeneous and complete among those explicitly focused on IoT. We now briefly mention the salient characteristics of the selected datasets and focus our discussion on promising research directions suitable to Deep Reinforcement Learning. An exhaustive analysis of these data sources is out of the scope of this survey (see Table 8).

The WUSTL-IOTI-2021 dataset is carefully explained and made available in [140] and is focused on emulating a real Industrial IoT (I-IoT) environment. Authors devoted to studying the four most popular SCADA communication protocols, namely, Modbus, BACnet, DNP3 and MQTT, and provided a dataset to replicate their main security vulnerabilities. Some types of attacks like Backdoor and Command Injection may be unrepresented though, and consequently, augmentation techniques may be necessary for an ML-driven IDS to learn the signatures of such attacks.

In [119] a dataset was released that contains annotated flow data including ARP spoofing, TCP SYN flooding, Fraggle (UDP flooding), Smurf, and Ping of Death attacks. Also, other flooding attacks related to the SNMP, SSDP, and TCP SYN protocols are present in this dataset. The dataset is publicly available on the internet[2] and contains 30 pcap files where each file corresponds to a trace collected over a day. There are two annotation files comprising start time, end time, flows that are influenced during the attack, and attack type, among other features. This dataset might be useful to augment the robustness of intrusion detection where the assessment criterion is connected to the potential discrepancies between the traffic and the Manufacturer Usage Description (MUD) information of each device.

[2] https://iotanalytics.unsw.edu.au/attack-data.html

Table 8
Recent remarkable datasets for IoT-IDS benchmarking.

Source	Year	Description	Advantages	Weaknesses
[140]	2021	I-IoT focused on replicating vulnerabilities of SCADA communication protocols.	Multiple types of data sources	May need to be augmented.
[119]	2019	Benign and volumetric attack IoT traffic traces. Multi-class annotations.	Contains MUD-related adversarial attacks.	Lacks containing sensor measurement data of IoT devices.
[141]	2020	Telemetry data of IoT and I-IoT services, related Operating Systems' logs and traffic	Multiple types of data-sources.	Lacks including I-IoT traffic flows.
[84]	2019	IoT related data for BotNet attack detection	Contains both real and synthetic data, contains multiple protocols.	Lacks including I-IoT traffic flows.
[142]	2021	Device-agnostic and Connectivity-agnostic for IoT and I-IoT scenarios	Multiple types of data-sources, Contains multi-stage attacks. Prone to training more generalizable IoT-IDSes	May need to be augmented.
[143]	2022	Annotated Traffic, logs, and other data related to IoT and I-IoT environments.	Multiple types of data sources, data from 10+ types of IoT devices	May need further standardization.

In [141] The TON_IoT dataset is presented. This dataset includes telemetry data of IoT and I-IoT services, as well as Operating Systems' logs and network traffic of an IoT scenario. These data were collected from a realistic representation of a medium-scale IoT network. Nine types of cyber-attacks were launched and annotated against various IoT and I-IoT sensors across the network. The generated data were stored in logs and CSV files. The dataset alongside a more detailed description can be found online.[3] This dataset can be considered an adversarial dataset since it could require to be expanded to include more benign traffic to represent a realistic network scenario.

Authors in [84] presented the Bot-IoT dataset which comprises realistically simulated IoT network traffic, along with various types of attacks. The main goal is to facilitate IoT-specific BotNet traffic pattern characterization. This dataset includes traffic from a simulated smart-home environment including multiple types of annotated attacks among which DoS, DDoS, probing, and information theft are found. When creating a new IoT-IDS, it may be a good idea to complement this dataset with traffic related to I-IoTs. Researchers can download the dataset from the Internet.[4]

Aiming to reduce the heterogeneity of IoT and I-IoT-related data, the work in [142] is devoted to creating a connectivity-agnostic and device-agnostic intrusion data set. In other words, the dataset is compatible with the I-IoT system regardless of the platforms, configurations, and deployed hardware and software of connectivity protocols. It also contains diverse types of sources like logs, alerts, traffic captures, and resource probes at the nodes of the network. This dataset resembles the behavior of multiple attack types in the context of connectivity protocols. This dataset can be accessed by the IEEE DataPort.[5]

The Edge-IoTset was released in [143] and is publicly accessible in the web.[6] This dataset was created to address the gap of including both IoT and I-IoT benign and malicious traffic flows. Apart from network traffic, the authors included logs, system resource probes, and system alerts. Fourteen types of attacks are categorized into five macro-categories, namely, DoS/DDoS attacks, Information gathering, Man in middle attacks, Injection attacks, and Malware attacks.

Discussion

Some remarks follow that might help to take more benefits from the mentioned data sources and also highlight potential lines of future research within this respect.

Robust pre-processing of unbalanced datasets. Data normalization is widely used in the pre-processing phase of ML datasets. This technique is known to reduce the convergence time of the DL models and to prevent the exploding or vanishing gradient problem. However, applying normalization to unbalanced datasets may accentuate the performance degradation of minority class predictions and the need of designing resampling techniques. In this respect, the authors in [115] suggested devoting more research to leverage preprocessing techniques that are less sensitive to class imbalance.

Architectural inductive biases for the latent space modeling. When modeling a DL pipeline in general, the inner layers of Deep ANNs tend to converge to latent vector spaces that accentuate the salient characteristics of the feature space and reduce the noise concerning the loss function minimization. The same holds to the inner layers of the agent's Deep ANNs in any DRL configuration. In this context, we note that choosing suitable architectural inductive biases could improve convergence to effective latent spaces in DRL while avoiding representational bottlenecks at the same time. For example, auto-encoders might excel at automatic feature selection, time series correlations between data records could be captured by the usage of recurrent layers instead, network-topology related patterns could be easily captured by graph convolutions, attention-mechanisms may help to dynamically weight the importance of a set of features, etc. We recommend using architectural inductive biases when feeding high-dimensional noisy and heterogeneous IoT-IDS-related data to neural networks.

Call for standardization. The high level of heterogeneity among these datasets might hinder a fair benchmarking of the state-of-art IoT-IDS solutions [145,146]. In this respect, we mention the work in [145] that proposed SDN-based techniques to standardize features taking in input from a heterogeneous set of general-case IDS data sources. This standardization led to the creation of a more uniform dataset that can be used to benchmark a wider variety of state-of-art IDS solutions. It may be worth exporting such standardization techniques with some IoT-specific IDS datasets.

Transfer learning may enrich supervised models' performance. Taking into account similar datasets instead, we recommend IoT-IDS designers also import the transfer learning idea proposed by Xia et al. [93] on a general-case IDS context. In this work, a DRL-based IDS is first trained within a trace-driven environment based on the NSL-KDD dataset, the pre-trained weights of the ANN-based agent are then *fine-tuned* based on the more recent AWID(v1) dataset [56].

The reviewed datasets may lack attention by the research community focused on DRL-driven IoT-IDS. Last but not least, we have noted a decrease in the number of publications that develop a DRL-based IoT-IDS and assess the validity of their solution using these datasets. In this respect, using DRL to indirectly optimize the detection accuracy, via the optimization of more complex and high-level optimization goals, is a research direction that may be lacking attention from the IDS research community.

[3] https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i
[4] https://ieee-dataport.org/documents/bot-iot-dataset
[5] https://ieee-dataport.org/documents/x-iiotid-connectivity-and-device-agnostic-intrusion-dataset-industrial-internet-things
[6] https://ieee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iot-and-iiot-applications

Conclusion. Recently, some new IoT-focused Intrusion Detection annotated datasets have been publicly released to the research community. Researchers should pay attention to the specific characteristics of each dataset as the heterogeneity of IoT scenarios is bigger with respect to traditional network environments. Some research efforts have been devoted to standardizing the feature set of traffic traces with the help of SDN, but this research road may need to be further explored. Recent literature that uses Deep Reinforcement Learning for IoT-IDS tends either to use outdated or non-IoT-specific datasets. Thus, exploiting the datasets mentioned in this section for DRL IoT-IDS may represent a research opportunity. In the context of DRL, effective pre-processing of these highly dimensioned and heterogeneous data sources could be obtained by carefully choosing the architectural inductive biases at the core of Deep ANN-based function approximators. Also, robust feature preprocessing techniques could mitigate the difficulties related to extracting patterns of underrepresented attacks in IDS-related data sources.

4.7. Ongoing commissioned research projects

As the interconnection of things becomes ubiquitous and new security-related challenges arise, governments are commissioning international and national research projects intending to secure the IoT ecosystem. This state-of-the-art review section ends by mentioning relevant projects that might be a good place for researchers to use and test DRL-based intrusion detection solutions focused on the IoT. At the time of writing, these projects are still in progress.

ELECTRON. The ELECTRON project is focused on securing Electrical Power and Energy Systems. More specifically, the acronym means "rEsilient and seLf-healed EleCTRical pOwer Nanogrid". The main project's goal is to enhance the resilience of energy systems against cyber, privacy, and data attacks. One of the pillars upon which this goal is based refers precisely to federated anomaly and intrusion detection. Decentralized and federated learning are listed among the methodologies for constructing a cyber-defense framework. One encouraging sign for the application of DRL in this project is the fact that a Virtual Reality-based testbed is being used to develop and test the current solutions. Researchers are using SDN technologies at the base of cybersecurity applications [147]. The project is funded by the Horizon 2020 Framework Programme (H2020) of the European Commission.

ARCADIAN-IoT. The title of this project [148] is the acronym for Autonomous trust, security, and privacy management framework for IoT. This project explicitly calls for the usage of advanced Artificial Intelligence-based frameworks and mechanisms for securing and guaranteeing privacy in the IoTs. Attention has been placed on communication overhead and federated learning by the research team [149]. This project is funded by the EU's Horizon2020 program. ARCADIAN-IoT started on May 1st, 2021, and is planned to last for 36 months.

IRIS. One of the topics of the Horizon 2020 Framework Programme is focused on Intelligent security and privacy management. In this topic, the IRIS project [150] started in September 2021 and it focuses on IoT security. More specifically, the project's mission is to create a platform for IoT threat and vulnerability detection. This project is also especially focused on creating frameworks for robustness against adversarial attacks during the training phase of AI-driven cybersecurity applications for IoT. In the context of this project, three mature data collection and test environments will be used for assessing the capabilities of the IRIS platform for securing the communication and management of smart grids, intelligent transportation systems, and the interconnected embedded systems of a smart city.

CAREER. The US National Science Foundation is supporting the CAREER project [151], whose main focus is to secure the IoT cloud services that manage trust and secure deployments of IoT devices in general. The project started in June 2022 and its estimated end date is June 2027. More specifically, the main goals of the project are understanding and systematizing cyberattacks and vulnerabilities of IoT interoperability and IoT service platforms to develop a correspondent security framework. The project includes the development of in-device channel control frameworks with innovative techniques among the building blocks of its whole solution.

5. General discussion

This section highlights some common design trends, best practices, and inconvenient design choices observed throughout this review of the recent literature devoted to DRL-based IoT-IDSes. The last contribution of this section instead stresses some important open challenges in the field.

5.1. Common trends in the state of the art

This research has examined more than fifty recent works that exploit DRL to create intrusion detection systems for the IoT. DRL-related design choices focused on imitating expert detection policies, balancing training data, improving robustness, preserving privacy, and other high-level goals related to intrusion detection were analyzed. While reviewing such papers, some common trends have been discovered that are transverse to the previous list of goals and are not necessarily connected to DRL. Such trends are now discussed, in the hope that they might help focus future research in this area.

Difficult reproducibility of work. Many of the publications reviewed do not publicly share the solutions' implementation code. This trend may represent an obstacle to the agile experimentation of the provided works in real IoT environments. In other words, a considerable percentage of the reviewed works did not specify the methodology to the point of permitting the reproduction of their experimentation. This fact may make it difficult or even impossible to fairly compare the proposed solutions. It is worth noting that the work in [51] also notes this trend in the literature about DRL-based cyber-security applications in general.

Predilection of value-based DRL approaches. A determinant predilection of researchers to use the value learning DRL approaches instead of policy learning algorithms has been observed throughout this research. One of the reasons for this preference is the discrete modelization of the action space [54]. In other words, in the majority of cases, the agent is supposed to choose between a discrete set of actions, e.g., telling if the current state-space sample corresponds or not to an infected one, or specifying which type of infection is affecting such a sample. Other works like the one in [57] were motivated to use a DRL value-learning framework because of its ease of implementation compared to some policy-learning methods. However, the benefits of efficient actor-critic and policy learning frameworks might deserve more attention from the IoT-IDS research community.

Predilection of centralized IDSes. AI-powered IoT-IDS is an active area of research due to the challenges posed by the constrained resource nature and the heterogeneity of this type of network. However, only a minority of the works explored in this research are devoted to delivering lightweight distributed IDSes to IoT infrastructures. However, centralized IDS based on ML approaches may be impractical to deliver to real-world IoT situations without incorporating proper Federated Learning, service replication, and timeliness-oriented strategies. As mentioned in 2.2, in addition to power and computation resource constraints and the high heterogeneity of IoT contexts, this drawback is caused by the need to protect data privacy. At the time this research is being written, distributed learning and distributed intrusion detection for the IoT still have plenty of space for improvement.

Simple reward shaping. In the majority of works presented in Section 4.1 a binary reward signal was given that punishes miss-classifications and encourages correct inferences. Interestingly, some of the reviewed works made experiments to assess if a continuous reward, which is proportional to the entity of the detector's confidence level, could provide better results. The majority of works in Section 4.1 stated that a discrete binary reward provides a better convergence rate compared to the weighted soft reward.

Note however that a slight complexification of the reward function permitted the authors in [109] to obtain good classification results without the need of using re-sampling techniques for dealing with unbalanced datasets. In [109], the rewards for actions that result in true-negatives and the punishments for actions that result in false-negatives had a greater entity compared to the rewards associated with true-positives and punishments for false-positives. In synthesis, in our research, it has been noticed that improvements in the detection accuracy of DRL-driven IoT-IDSes could be obtained by carefully shaping the agent's rewards.

Low focus on software defined IoTs. As mentioned in Section 4.5, software-defined IoTs is a game-changing technology that permits real-time, fine-grain, and customized feature probing in IoT scenarios [137]. Such a characteristic is being exploited for the development of modern intrusion detection systems for the IoTs based on Deep Learning [152]. However, it has been noticed that most works that use DRL for IoT-IDS do not exploit the possibilities of SD-IoT for modeling the action space, the state space, or the reward computation of the proposed IoT environment. SD-IoT could facilitate the design and implementation of efficient data aggregation, network monitoring, and service migration pipelines being at the same time aware of communication-overhead, energy-efficiency, and Quality of Service, among other non-functional requirements [153]. The optimization of these and other high-level goals while performing intrusion detection could make the difference between realistic and purely theoretical proposals of IDSes for the IoT. There might be many future work opportunities for DRL practitioners when focusing on intrusion detection on SD-IoTs.

5.2. Best practices

This section presents a list of good design choices particularly related to DRL that were encountered while reviewing the IoT-IDS literature. Successively, a response is given to the issue of when shall one use DRL for IoT-IDS.

1. **Software-Defined IoTs.** SD-IoTs permit the modeling of comprehensive state spaces regarding IoT environments. Also, global-aware and compound rewards may be possible to compute only in software-defined networking environments. Finally, high-level control actions may be realistically modeled only when one assumes an underlying decoupled control and forwarding plane in the IoT infrastructure [137–139].
2. **Optimization of goals related to non-functional requirements.** DRL action spaces may preferably be related to the optimization of high-level goals like communication-overhead reduction [111], energy-efficient detection [124], adversarial training [90], anomaly-detection criteria [118], parameter tuning [118] or hyper-parameter tuning [122], feature selection [124], curricula selection [121], monitoring granularity [37], etc.
3. **Deployment.** Intrusion detection at the edge of the network may be more efficient and less communication intensive compared to centralized detection [108,109].
4. **Inductive Biases.** When using off-policy value-based frameworks like DQN, consider exploiting the majority of the state-of-the-art inductive biases like Rainbow DQN for more efficient learning and strengthening the optimality of the policies the agent might converge to [77].
5. **DRL-based rebalancing and adversarial learning.** DRL has proven effective for creating adversarial learning through sampling [90] and semantic-aware generative adversarial strategies for IDS [94].
6. **Distributed Learning.** In distributed or ensemble learning scenarios, FL could be preserved by using an anonymized and periodically-updated labeled held-out dataset at the centralized DRL agent to assess the overall performance gains of the IDS [112].
7. **Ensemble learning.** DRL could be used to optimize the convergence properties of an ensemble of shallow ML-based intrusion detectors at the edge of IoTs. In this case, the actions of the DRL agent could be related to balancing the learning contributions of each classifier [108,109].
8. **Distributed and Federated learning.** DRL could help to design distributed and collaborative learning schemes preserving data privacy and enriching the training environment of leaf detection agents [37,112,114].
9. **Risk-aware intrusion prevention.** When modeling intrusion prevention alongside intrusion detection, a labeled pre-recorded trace could help to shape the rewards as a function of the malicious traffic in the system. Such a reward might help the agent to learn to assess the risk level of intrusion attempts [123].
10. **Offline supervised training and safe exploration.** X-aware IoT-IDSes could comprise a (periodic) offline learning phase that exploits supervised learning to shape a part of a multi-objective reward and an online learning phase where the detection-related term of the rewards are anomaly-based. Offline learning could enable safe exploration strategies for the DRL agents [116].
11. **Compound rewards.** Hierarchical rewards could reduce the communication overhead and reduce the convergence time in IDSes that deploy DRL agents at the edge of the network [132].

When shall one use deep reinforcement learning? Deep Reinforcement Learning is most helpful when one pursues multi-objective or long-term optimization. Also, DRL may be essential when one has a non-differentiable complexity layer to a vanilla classification-based intrusion detection model.

More specifically, to exploit more potentialities of DRL, higher-level optimization goals can be modeled alongside classification accuracy. In this review, it has been noticed that many of these optimization goals were indirectly related to classification accuracy while also accounting for other more essential IoT requirements like **Federated and Distributed Learning**, **Black-Box Adversarial learning**, **Optimization of network and computation resource usages**, among others. It is worth stressing that, also, other non-functional requirements that may increment the overall system's effectiveness could be achieved by the usage of DRL. For example, the optimization of **high-level feature engineering** taking into account multiple data sources [115] and the **optimization of the system's overall resilience** [154], among others. Note that focusing on these higher-level goals may be difficult to do within a supervised learning context, and thus DRL could be indispensable in this case.

Finally, note that DRL would be also worth using for directly optimizing traffic classification if one assumes some level of contrast between protecting the network in the immediate rather than preserving its security in the long-term reward, i.e. if the greedy goal and the long-term optimization goal are contrastive between them. As an example, one can think of all the environments in which it would be preferable to admit some intrusions in the present to be able to better protect the system in the future.

5.3. Lessons learnt

The drawbacks of several modelization options were apparent after carefully reading the present literature. After mentioning several problematic settings about DRL-based intrusion detection in IoT, this section emphasizes the circumstances in which one should select different DL paradigms.

1. **IDS labeled datasets are inherently imbalanced.** The available recorded labeled traces related to IDS might not be independent and identically distributed [108]. Attempts to learn intrusion detection that do not address the problem of unbalanced class distributions in data may lead to low detection accuracy for underrepresented classes.
2. **DRL-based solutions for IoT-IDS are mainly using out-of-date datasets.** When setting up a trace-driven experiment to validate an IoT-IDS, make sure to use traces related to realistic IoT environments, and, among these, choose the most recent ones [36].
3. **Annotated datasets do not contain all the possible attacks.** Cyber attacks constantly evolve, and signature-based intrusion detection may evolve at a slower pace. ML-driven intrusion detection could be anomaly-based or could be oriented to abstract the signature of attacks from raw feature space observations [48]. Adding an auxiliary anomaly-based detection module might be mandatory when deploying an IDS trained with an annotated dataset. Anomaly-based detection could be triggered whenever the confidence of the inference of the main IDS is low [57]. Additionally, periodical training cycles with up-to-date labeled traces might be necessary to guarantee the long-term efficacy of this class of IDSes.
4. **Realistic intrusion detection in the IoT needs to be adaptive to dynamic environments.** Using DRL as an imitation learning setup for learning to detect intrusions via supervised learning might preclude the overall adaptiveness of the IDS to state space distribution changes. (Refer to Section 4.1) DRL-based control of the threshold-related parameters of anomaly-based detection modules could be crucial to leverage IDSes that are adaptive to dynamic network scenarios [118].
5. **Vanilla centralized, distributed, and ensemble learning might be problematic.** Centralized IDSes may not be effectively deployed in real IoT scenarios because they might result in communication overhead and data privacy leaks [36]. On the other hand, it may be impractical to model the training of a DRL agent inside an edge IoT device both from the point of view of energy efficiency and the disponibility of training data [111]. DRL at less resource-constrained network locations could instead optimize the convergence of shallow-ML detectors [108]. Instead, when using ensemble or distributed learning, federated learning could be guaranteed by only communicating the parameters of the classifiers rather than its training data [114].
6. **Transfer learning in distributed learning scenarios may incur in communication overhead**. In the context of transfer learning, communicating the whole list of parameters of deep ANNs might incur prohibitive communication overhead. Consider sharing only incremental changes, or other compression techniques if ANN parameter sharing over the network is a requirement [114].

When shall Deep Reinforcement learning adoption be avoided for IoT-IDS? Some of the publications reviewed in this survey claim that DRL is necessary to create IDSes that have a continuous learning strategy. Note, however, that DRL is not the unique way in which continuous learning strategies might be implemented for ML-based IoT-IDSs [111,155]. Also note that, as explained in Section 4.2, DRL may neither be strictly necessary for adversarial attack crafting, especially in cases where one can get sufficient knowledge about the mapping between the model's inputs and outputs, other strategies could be adopted [156].

In many of the reviewed works, especially those in Section 4.1, an imitation learning setup is implemented using DRL where the agent is supposed to imitate an expert policy previously encoded on a labeled trace [54]. However, during the course of this review, a criterion has been developed: in cases where supervised classification data are available, using DRL for optimizing traffic classification without taking into account non-functional requirements like risk minimization, privacy preservation, or communication overhead minimization, may not only be superfluous compared to DL but also not the best option [36,157,158].

5.4. What types of attack are more likely to be picked up by an IDS based on DRL?

Throughout this review, some solid criteria may have emerged about the advantages that DRL could bring with respect to vanilla DL-based intrusion detection systems for the IoT. Based on such criteria, this section enumerates some key types of attack that a well-designed DRL intrusion detector could detect more effectively with respect to other DL an ML-based IDSes:

1. **Adversarial attacks.** Once a DL-based IDS has learned to abstract patterns of attacks from input data, some intrusions might adapt their visible signature over such data to resemble normal traffic as much as possible and fool a detector based on fixed criteria. In this case, adding a second level of abstraction like that of dynamic anomaly criteria, or a dynamic probing frequency, could lead to more adaptive detection policies that learn to defend systems even in the presence of such subtle instances of attacks.
2. **Zero-day attacks.** As the IoT environment evolves and grows, new types of attacks and penetration techniques are constantly emerging from real world scenarios. Consequently, any IDS working on a real deployment is subjected to attacks whose class was not present in the training data. Such attacks are referred to as *Zero-day* attacks and the detection of these misuses represents a challenging open area of research [159]. Deep Reinforcement Learning is suitable to model high-level optimization tasks such as meta-learning. Through the usage and combination of these techniques, DRL could help detect patterns and characterize data distributions that are different from those seen during training. In this respect, a promising strategy to deal with Zero-day attack detection and categorization may be DRL [103].
3. **Tabular-data-alike attacks.** Many of the presented works used DRL for optimizing the feature selection process, precisely because many types of attack inference over traffic captures require to deal with many uninformative features. These characteristics of traffic captures align well with typical *tabular data* scenarios, which represent an open challenge for deep learning function approximators [160]. The meta-learning modelization techniques surveyed in this work indicate that DRL is a suitable strategy to learn to detect attacks characterized by features like those typically encountered in tabular data.

5.5. Open challenges

While reviewing the state of the art, the future work proposal evidenced by the research community has been discussed. However, regarding the usage of DRL for IoT-IDS, some important open challenges might have received low attention from the research community. These salient open research areas are now covered.

1. **Exploiting high-level threat intelligence.** Intrusion detection in the majority of the reviewed works is based on the observation of traffic flows, packet monitoring, logs, system alerts, and other network statistics like performance and congestion. However, other works on IoT-IDS exist that, even if not based on DRL, propose to take into account a higher-level view of the system's context.

The work in [115] advocates for an expert knowledge base (EK) that is assembled with threat intelligence meta-data that includes high-level traffic behaviors, device specifications, operating conditions, and human feedback. The EK is then hashed to facilitate the deployment of this information on low-memory devices. These devices may then better detect intrusions correlating the EK information with the incoming traditional feature set like traffic, logs, etc. The high-level information in the EK may facilitate a better Root Cause Analysis which would improve the efficacy of intrusion detection and mitigation, according to authors in [115].

A long road ahead may still need to be explored concerning the exploitation of high-level threat intelligence and intrusion detection in the specific field of IoTs.

2. **Topology awareness and Graph modelization.** In terms of network theory, IoTs could be straightforwardly modeled as heterogeneous graphs, where multiple types of nodes are interconnected through multiple types of edges. However, only a few of the works in the state-of-the-art exploited the full advantages of this modelization scheme [123].

   It is well known that message-passing and other permutation-invariant inductive biases in deep learning should help in automatic feature extraction in almost any network-related scenario [161]. However, in our research, only a reduced number of works that exploit these architectural inductive biases were found. Note, however, that there exists a plethora of solutions for other complex problems related to IoTs that better exploit graph modelization and graph neural networks [162]. Future research could deepen more systematically in exploiting graph representation learning and graph reinforcement learning for leveraging powerful IoT-IDSes to the edge of the network.
3. **Energy-Efficiency and DRL.** Some of the reviewed works explicitly mentioned that the training process of data-driven IDSes may incur inconvenient power consumption at the edge of the network. Authors of AESMOTE [90] mentioned that the length of the training episodes can be adapted to the power capacity of the learning device, Omar Bouhamed et al. [111] suggested that model weights' updates should be made only when the edge devices are recharging their batteries, the authors of [121] sought to minimize the feature set to reduce the processing effort of edge-located classifiers, and both the works in [108,109] applied shallow-ML training at the edge of IoT and advocating to delegate heavier computations to the upper levels of the network where resources may be less constrained. The claims listed above are only some of the recommendations that suggest investing research efforts in energy-efficient deploying strategies for DRL-based IoT-IDSes. Perhaps the research community will take further steps in energy-efficient IoT-IDS based on DRL pipelines in the near future.
4. **Data-Efficiency and DRL.** One of the main characteristics of Deep Reinforcement Learning pipelines is the gradient-based optimization of complex non-differentiable combinatorial problems. However, it is well-known that this feature comes at the cost of low data efficiency. In other words, DRL tends to be more data-hungry compared to other DL paradigms like supervised learning or unsupervised representation learning. While many strategies are being developed to reduce the data-hunger characteristic of DRL pipelines, leveraging some of these strategies to the task of IoT-IDS might still be a vast research opportunity [53]. Promising research directions in this sense might include reward prioritization schemes for expert human feedback, and efficient and lightweight transfer learning schemes, among others.
5. **Human-reward prioritization.** Cyber threats are constantly evolving. An effective Intrusion Detection System must evolve at the same pace in its detection abilities. Zero-day attack detection pipelines are anomaly-based by definition and thus, might have a low recall. Cataloging and identifying the signatures of new attacks are still left as human-based tasks. A virtuous interaction between anomaly-based alerts and human feedback on the criteria for anomaly identification could be the base of efficient learning to detect zero-day attacks in online learning scenarios. Some works proposed adding human supervisory signals on a DRL intrusion detection setting [108]. However, future research may still need to be made to deliver prioritization schemes for sparse human-based rewards.

6. Conclusion

This survey reviewed more than fifty recent studies that use DRL to build IoT intrusion detection systems. Design decisions made in relation to DRL that were aimed at mimicking expert detection policies, balancing training data, increasing robustness, protecting privacy, and other high-level intrusion detection goals were examined. Throughout this study, a systematic literature review permitted to distill the most effective design choices and the lessons learned by DRL practitioners in the research field of DRL-based IDS for the IoT were identified and systematized.

In synthesis, complex goals for which DRL is better suited in the context of IoT-IDS are those related to the overall system conditions that derive from protecting the system in the long run (reliability, performance, reputation, etc. Instead, focusing exclusively on the optimization of detection accuracy may reduce the effectiveness of DRL, which could be instead necessary when higher-level goals are pursued alongside accuracy optimization.

Additionally, this survey pointed to active international projects in which these techniques may be applied by researchers, and provided a list of the most recent and comprehensive data sources focused on IoT for training DRL-based intrusion detectors. Modern deep learning inductive biases may still need to be exported to DRL-based pipelines devoted to IoT-IDS for achieving awareness of important aspects related to this field like high-level threat intelligence, energy consumption minimization, data efficiency, efficient learning of human feedback, and network topology, among others. We hope that also the list of future directions for the field's research and advancement that concludes this paper will be valuable to researchers.

CRediT authorship contribution statement

Jesús F. Cevallos M.: Conceptualization, Methodology, Writing – original draft. **Alessandra Rizzardi:** Conceptualization, Methodology, Writing – review & editing. **Alberto Coen Porisini:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] IEEE Coughlin Associates, Tom Coughlin, IoT trends to keep an eye on in 2023 and beyond, TechTarget, 2023, URL https://www.techtarget.com/iotagenda/opinion/IoT-trends-to-keep-an-eye-on.

[2] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, B. Stiller, Landscape of IoT security, Comp. Sci. Rev. 44 (2022) 100467.

[3] M.A. Al-Garadi, A. Mohamed, A.K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (IoT) security, IEEE Commun. Surv. Tutor. 22 (3) (2020) 1646–1685.

[4] S. Sicari, A. Rizzardi, L.A. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Comput. Netw. 76 (2015) 146–164.

[5] S.A. Salloum, M. Alshurideh, A. Elnagar, K. Shaalan, Machine learning and deep learning techniques for cybersecurity: A review, in: Proceedings of the International Conference on Artificial Intelligence and Computer Vision, AICV2020, Springer, 2020, pp. 50–57.

[6] P. Dixit, S. Silakari, Deep learning algorithms for cybersecurity applications: A technological and status review, Comp. Sci. Rev. 39 (2021) 100317.

[7] K. Arulkumaran, M.P. Deisenroth, M. Brundage, A.A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Process. Mag. 34 (6) (2017) 26–38.

[8] R. Bellman, R.E. Kalaba, Dynamic Programming and Modern Control Theory, Vol. 81, Citeseer, 1965.

[9] Y. Li, Deep reinforcement learning: An overview, 2017, arXiv preprint arXiv:1701.07274.

[10] V. Francois-Lavet, P. Henderson, R. Islam, M.G. Bellemare, J. Pineau, An introduction to deep reinforcement learning, 2018, arXiv arXiv:1811.12560.

[11] M.L. Puterman, Markov decision processes, in: Handbooks in Operations Research and Management Science, Vol. 2, Elsevier, 1990, pp. 331–434.

[12] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning., in: 4th International Conference on Learning Representations, ICLR, 2016, URL http://arxiv.org/abs/1509.02971.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[14] H.v. Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI '16, AAAI Press, 2016, pp. 2094–2100.

[15] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling network architectures for deep reinforcement learning, in: M.F. Balcan, K.Q. Weinberger (Eds.), International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 48, New York, New York, USA, 2016, pp. 1995–2003.

[16] I. Grondman, L. Busoniu, G.A.D. Lopes, R. Babuska, A survey of Actor-Critic reinforcement learning: Standard and natural policy gradients, IEEE Trans. Syst. Man Cybern. 42 (6) (2012) 1291–1307.

[17] O. Nachum, M. Norouzi, K. Xu, D. Schuurmans, Bridging the gap between value and policy based reinforcement learning, in: Advances in Neural Information Processing Systems, Vol. 30, 2017.

[18] R.S. Sutton, A.G. Barto, et al., Introduction to Reinforcement Learning, Vol. 2, MIT press Cambridge, 1998.

[19] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of the 33rd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 48, PMLR, New York, New York, USA, 2016, pp. 1928–1937.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, 2013, arXiv preprint arXiv:1312.5602.

[21] G. Dulac-Arnold, R. Evans, P. Sunehag, B. Coppin, Reinforcement learning in large discrete action spaces, 2015, arXiv abs/1512.07679.

[22] Z. Zhang, D. Zhang, R.C. Qiu, Deep reinforcement learning for power system applications: An overview, CSEE J. Power Energy Syst. 6 (1) (2019) 213–225.

[23] A. Haydari, Y. Yılmaz, Deep reinforcement learning for intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 23 (1) (2020) 11–32.

[24] F. AlMahamid, K. Grolinger, Autonomous unmanned aerial vehicle navigation using reinforcement learning: A systematic review, Eng. Appl. Artif. Intell. 115 (2022) 105321.

[25] A. Coronato, M. Naeem, G. De Pietro, G. Paragliola, Reinforcement learning for intelligent healthcare applications: A survey, Artif. Intell. Med. 109 (2020) 101964.

[26] Y. Wu, Z. Wang, Y. Ma, V.C. Leung, Deep reinforcement learning for blockchain in industrial IoT: A survey, Comput. Netw. 191 (2021) 108004.

[27] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, Y. Zhang, Deep reinforcement learning for internet of things: A comprehensive survey, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1659–1692.

[28] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, X. Shen, Deep reinforcement learning for autonomous internet of things: Model, applications and challenges, IEEE Commun. Surv. Tutor. 22 (3) (2020) 1722–1760.

[29] ETSI, Experiential Networked Intelligence (ENI); Terminology for Main Concepts in ENI, White Paper, Sophia Antipolis, France, 2021, URL https://www.etsi.org/deliver/etsi_gr/ENI/001_099/004/02.02.01_60/gr_ENI004v020201p.pdf.

[30] ETSI, Zero-Touch Network and Service Management (ZSM); Landscape, White Paper, Sophia Antipolis, France, 2022, URL https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/004/02.01.01_60/gr_ZSM004v020101p.pdf.

[31] Internet Engineering Task Force (IETF), An Autonomic Control Plane (ACP), White Paper, Santa Clara, USA, 2021, URL https://www.rfc-editor.org/rfc/rfc8994.pdf.

[32] A. Khraisat, A. Alazab, A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges, Cybersecurity 4 (2021) 1–27.

[33] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, M. Qiu, Adversarial attacks against network intrusion detection in IoT systems, IEEE Internet Things J. 8 (13) (2020) 10327–10335.

[34] M.A. Amanullah, R.A.A. Habeeb, F.H. Nasaruddin, A. Gani, E. Ahmed, A.S.M. Nainar, N.M. Akim, M. Imran, Deep learning and big data technologies for IoT security, Comput. Commun. 151 (2020) 495–517.

[35] K.K. Patel, S.M. Patel, P. Scholar, Internet of things-IOT: Definition, characteristics, architecture, enabling technologies, application & future challenges, Int. J. Eng. Sci. Comput. 6 (5) (2016).

[36] S. Tsimenidis, T. Lagkas, K. Rantos, Deep learning in IoT intrusion detection, J. Netw. Syst. Manage. 30 (2022) 1–40.

[37] T.G. Nguyen, T.V. Phan, D.T. Hoang, T.N. Nguyen, C. So-In, Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks, IEEE Trans. Cogn. Commun. Netw. 7 (4) (2021) 1048–1065.

[38] P.M. Chanal, M.S. Kakkasageri, Security and privacy in IoT: A survey, Wirel. Pers. Commun. 115 (2020) 1667–1693.

[39] C. Sobin, A survey on architecture, protocols and challenges in IoT, Wirel. Pers. Commun. 112 (3) (2020) 1383–1429.

[40] N. Kumari, A. Yadav, P.K. Jana, Task offloading in fog computing: A survey of algorithms and optimization techniques, Comput. Netw. 214 (2022) 109137.

[41] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, 2017, arXiv preprint arXiv:1706.06083.

[42] I. Corona, G. Giacinto, F. Roli, Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues, Inform. Sci. 239 (2013) 201–225.

[43] X. Xiong, K. Zheng, L. Lei, L. Hou, Resource allocation based on deep reinforcement learning in IoT edge computing, IEEE J. Sel. Areas Commun. 38 (6) (2020) 1133–1146.

[44] M. Tang, V.W. Wong, Deep reinforcement learning for task offloading in mobile edge computing systems, IEEE Trans. Mob. Comput. 21 (6) (2020) 1985–1997.

[45] A. Jarwan, M. Ibnkahla, Edge-based federated deep reinforcement learning for IoT traffic management, IEEE Internet Things J. (2022).

[46] C. Shu, Z. Zhao, G. Min, J. Hu, J. Zhang, Deploying network functions for multiaccess edge-IoT with deep reinforcement learning, IEEE Internet Things J. 7 (10) (2020) 9507–9516.

[47] M. Abdullahi, Y. Baashar, H. Alhussian, A. Alwadain, N. Aziz, L.F. Capretz, S.J. Abdulkadir, Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review, Electronics 11 (2) (2022) 198.

[48] P. Jayalaxmi, R. Saha, G. Kumar, M. Conti, T.-H. Kim, Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey, IEEE Access (2022).

[49] S. Santhosh Kumar, M. Selvi, A. Kannan, et al., A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things, Comput. Intell. Neurosci. 2023 (2023).

[50] Z. Utic, K. Ramachandran, A survey of reinforcement learning in intrusion detection, in: 2022 1st International Conference on AI in Cybersecurity, ICAIC, IEEE, 2022, pp. 1–8.

[51] A.M.K. Adawadkar, N. Kulkarni, Cyber-security and reinforcement learning—A brief survey, Eng. Appl. Artif. Intell. 114 (2022) 105116.

[52] S.U. Haq, A.M. Abbas, Advancements in intrusion detection systems for internet of things using machine learning, in: 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies, IMPACT, IEEE, 2022, pp. 1–5.

[53] M. Sewak, S.K. Sahay, H. Rathore, Deep reinforcement learning for cybersecurity threat detection and protection: A review, 2022, arXiv e-prints, arXiv–2206.

[54] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, Application of deep reinforcement learning to intrusion detection for supervised problems, Expert Syst. Appl. 141 (2020) 112963.

[55] G. Mohi-ud din, NSL-KDD, IEEE Dataport, 2018, http://dx.doi.org/10.21227/425a-3e55.

[56] C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset, IEEE Commun. Surv. Tutor. 18 (1) (2015) 184–208.

[57] B. Yang, M.H. Arshad, Q. Zhao, Packet-level and flow-level network intrusion detection based on reinforcement learning and adversarial training, Algorithms 15 (12) (2022) 453.

[58] I. Sharafaldin, A.H. Lashkari, S. Hakak, A.A. Ghorbani, Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology, ICCST, IEEE, 2019, pp. 1–8.

[59] H. Benaddi, K. Ibrahimi, A. Benslimane, J. Qadir, A deep reinforcement learning based intrusion detection system (DRL-IDS) for securing wireless sensor networks and internet of things, in: Wireless Internet: 12th EAI International Conference, WiCON 2019, TaiChung, Taiwan, November 26–27, 2019, Proceedings 12, Springer, 2020, pp. 73–87.

[60] H. Benaddi, K. Ibrahimi, A. Benslimane, M. Jouhari, J. Qadir, Robust enhancement of intrusion detection systems using deep reinforcement learning and stochastic game, IEEE Trans. Veh. Technol. 71 (10) (2022) 11089–11102.

[61] H. Benaddi, M. Jouhari, K. Ibrahimi, J. Ben Othman, E.M. Amhoud, Anomaly detection in industrial IoT using distributional reinforcement learning and generative adversarial networks, Sensors 22 (21) (2022) 8085.

[62] F. Aubet, M. Pahl, DS2os traffic traces, 2018.

[63] M.G. Bellemare, W. Dabney, R. Munos, A distributional perspective on reinforcement learning, in: International Conference on Machine Learning, PMLR, 2017, pp. 449–458.

[64] S. Bakhshad, V. Ponnusamy, R. Annur, M. Waqasyz, H. Alasmary, S. Tux, Deep reinforcement learning based intrusion detection system with feature selections method and optimal hyper-parameter in IoT environment, in: 2022 International Conference on Computer, Information and Telecommunication Systems, CITS, IEEE, 2022, pp. 1–7.

[65] H. Alavizadeh, H. Alavizadeh, J. Jang-Jaccard, Deep Q-learning based reinforcement learning approach for network intrusion detection, Computers 11 (3) (2022) 41.

[66] K. Ren, M. Wang, Y. Zeng, Y. Zhang, An unmanned network intrusion detection model based on deep reinforcement learning, in: 2022 IEEE International Conference on Unmanned Systems, ICUS, IEEE, 2022, pp. 1070–1076.

[67] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.

[68] S. Priya, K. PradeepMohankumar, Intelligent outlier detection with optimal deep reinforcement learning model for intrusion detection, in: 2021 4th International Conference on Computing and Communications Technologies, ICCCT, IEEE, 2021, pp. 336–341.

[69] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS, IEEE, 2015, pp. 1–6.

[70] G. Shi, G. He, Collaborative multi-agent reinforcement learning for intrusion detection, in: 2021 7th IEEE International Conference on Network Intelligence and Digital Content, IC-NIDC, IEEE, 2021, pp. 245–249.

[71] S. Dong, Y. Xia, T. Peng, Network abnormal traffic detection model based on semi-supervised deep reinforcement learning, IEEE Trans. Netw. Serv. Manag. 18 (4) (2021) 4197–4212.

[72] B.M.B. Mondal, A.B.D.A. Banerjee, S.G.D.S. Gupta, Network intrusion detection: A reinforcement learning approach, Res. Sq. (2022).

[73] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling network architectures for deep reinforcement learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1995–2003.

[74] M. Fortunato, M.G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, et al., Noisy networks for exploration, 2017, arXiv preprint arXiv:1706.10295.

[75] S.D. Bay, D. Kibler, M.J. Pazzani, P. Smyth, The UCI KDD archive of large data sets for data mining research and experimentation, ACM SIGKDD Explor. Newsl. 2 (2) (2000) 81–85.

[76] T. Izquierdo García-Faria, Applying the rainbow architecture to intrusion detection systems, Universitat Politècnica de Catalunya, 2021.

[77] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D. Silver, Rainbow: Combining improvements in deep reinforcement learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[78] K. Ren, Y. Zeng, Z. Cao, Y. Zhang, ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model, Sci. Rep. 12 (1) (2022) 1–18.

[79] K. O'Shea, R. Nash, An introduction to convolutional neural networks, 2015, arXiv preprint arXiv:1511.08458.

[80] Z. Wang, D. Jiang, Z. Lv, H. Song, A deep reinforcement learning based intrusion detection strategy for smart vehicular networks, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2022, pp. 1–6.

[81] G. Emil Selvan, T. Daniya, J. Ananth, K. Suresh Kumar, Network intrusion detection and mitigation using hybrid optimization integrated deep Q network, Cybern. Syst. (2022) 1–17.

[82] N. Karimi, K. Khandani, Social optimization algorithm with application to economic dispatch problem, Int. Trans. Electr. Energy Syst. 30 (11) (2020) e12593.

[83] J.C. Bansal, H. Sharma, S.S. Jadon, M. Clerc, Spider Monkey optimization algorithm for numerical optimization, Memet. Comput. 6 (2014) 31–47.

[84] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.

[85] A. Hussein, M.M. Gaber, E. Elyan, C. Jayne, Imitation learning: A survey of learning methods, ACM Comput. Surv. 50 (2) (2017) 1–35.

[86] A. Kumar, J. Hong, A. Singh, S. Levine, When should we prefer offline reinforcement learning over behavioral cloning? 2022, arXiv preprint arXiv:2204.05618.

[87] S. Emmons, B. Eysenbach, I. Kostrikov, S. Levine, RvS: What is essential for offline RL via supervised learning? 2021, arXiv preprint arXiv:2112.10751.

[88] A. Thakkar, R. Lohiya, A review of the advancement in intrusion detection datasets, Procedia Comput. Sci. 167 (2020) 636–645.

[89] G. Caminero, M. Lopez-Martin, B. Carro, Adversarial environment reinforcement learning algorithm for intrusion detection, Comput. Netw. 159 (2019) 96–109.

[90] X. Ma, W. Shi, AESMOTE: Adversarial reinforcement learning with SMOTE for anomaly detection, IEEE Trans. Netw. Sci. Eng. 8 (2) (2020) 943–956.

[91] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[92] E. Suwannalai, C. Polprasert, Network intrusion detection systems using adversarial reinforcement learning with deep Q-network, in: 2020 18th International Conference on ICT and Knowledge Engineering, ICT&KE, IEEE, 2020, pp. 1–7.

[93] Y. Xia, S. Dong, T. Peng, T. Wang, Wireless network abnormal traffic detection method based on deep transfer reinforcement learning, in: 2021 17th International Conference on Mobility, Sensing and Networking, MSN, IEEE, 2021, pp. 528–535.

[94] J. Tu, W. Ogola, D. Xu, W. Xie, Intrusion detection based on generative adversarial network of reinforcement learning strategy for wireless sensor networks, Int. J. Circuits Systems Signal Process. 16 (2022) 478–482.

[95] D. Pfau, O. Vinyals, Connecting generative adversarial networks and actor-critic methods, 2016, arXiv preprint arXiv:1610.01945.

[96] J. Parras, A. Almodóvar, P.A. Apellániz, S. Zazo, Inverse reinforcement learning: A new framework to mitigate an intelligent backoff attack, IEEE Internet Things J. 9 (24) (2022) 24790–24799.

[97] T. Lindner, D. Wyrwał, A. Kubacki, Low power wireless protocol for IoT appliances using CSMA/CA mechanism, in: Automation 2019: Progress in Automation, Robotics and Measurement Techniques, Springer, 2020, pp. 199–207.

[98] J. Parras, M. Hüttenrauch, S. Zazo, G. Neumann, Deep reinforcement learning for attacking wireless sensor networks, Sensors 21 (12) (2021) 4060.

[99] G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, M. Colajanni, Deep reinforcement adversarial learning against botnet evasion attacks, IEEE Trans. Netw. Serv. Manag. 17 (4) (2020) 1975–1987.

[100] Q.-D. Ngo, H.-T. Nguyen, V.-D. Nguyen, C.-M. Dinh, A.-T. Phung, Q.-T. Bui, Adversarial attack and defense on graph-based IoT botnet detection approach, in: 2021 International Conference on Electrical, Communication, and Computer Engineering, ICECCE, IEEE, 2021, pp. 1–6.

[101] H.-T. Nguyen, Q.-D. Ngo, V.-H. Le, A novel graph-based approach for IoT botnet detection, Int. J. Inf. Secur. 19 (5) (2020) 567–577.

[102] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, S. Jaiswal, graph2vec: Learning distributed representations of graphs, 2017, arXiv preprint arXiv:1707.05005.

[103] Q.-D. Ngo, Q.-H. Nguyen, A reinforcement learning-based approach for detection zero-day malware attacks on IoT system, in: Artificial Intelligence Trends in Systems: Proceedings of 11th Computer Science on-Line Conference 2022, Vol. 2, Springer, 2022, pp. 381–394.

[104] M. Ibrahim, R. Elhafiz, Integrated clinical environment security analysis using reinforcement learning, Bioengineering 9 (6) (2022) 253.

[105] M. Ibrahim, R. Elhafiz, Security analysis of cyber-physical systems using reinforcement learning, Sensors 23 (3) (2023) 1634.

[106] G.A. Rummery, M. Niranjan, On-Line Q-Learning Using Connectionist Systems, Technical report CUED/F-INFENG/TR, Cambridge University Engineering Department, 1994.

[107] M. Ibrahim, Q. Al-Hindawi, R. Elhafiz, A. Alsheikh, O. Alquq, Attack graph implementation and visualization for cyber physical systems, Processes 8 (1) (2019) 12.

[108] K. Sethi, E. Sai Rupesh, R. Kumar, P. Bera, Y. Venu Madhav, A context-aware robust intrusion detection system: a reinforcement learning-based approach, Int. J. Inf. Secur. 19 (2020) 657–678.

[109] K. Sethi, Y.V. Madhav, R. Kumar, P. Bera, Attention based multi-agent intrusion detection systems using reinforcement learning, J. Inf. Secur. Appl. 61 (2021) 102923.

[110] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy, EuroS&P, IEEE, 2016, pp. 372–387.

[111] O. Bouhamed, O. Bouachir, M. Aloqaily, I. Al Ridhawi, Lightweight IDS for UAV networks: A periodic deep reinforcement learning-based approach, in: 2021 IFIP/IEEE International Symposium on Integrated Network Management, IM, IEEE, 2021, pp. 1032–1037.

[112] N.H. Quyen, P.T. Duy, N.C. Vy, D.T.T. Hien, V.-H. Pham, Federated intrusion detection on non-IID data for IIoT networks using generative adversarial networks and reinforcement learning, in: Information Security Practice and Experience: 17th International Conference, ISPEC 2022, Taipei, Taiwan, November 23–25, 2022, Proceedings, Springer, 2022, pp. 364–381.

[113] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: An ensemble of autoencoders for online network intrusion detection, 2018, arXiv preprint arXiv:1802.09089.

[114] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing federated learning on non-IID data with reinforcement learning, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1698–1707.

[115] K. Krinkin, On-device context-aware misuse detection framework for heterogeneous IoT edge, Appl. Intell. (2022) 1–27.

[116] E. Muhati, D.B. Rawat, Asynchronous advantage actor-critic (A3C) learning for cognitive network security, in: 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA, IEEE, 2021, pp. 106–113.

[117] N. Kandhoul, S.K. Dhurandher, Deep q learning based secure routing approach for OppIoT networks, Internet Things 20 (2022) 100597.

[118] T. Gu, A. Abhishek, H. Fu, H. Zhang, D. Basu, P. Mohapatra, Towards learning-automation IoT attack detection through reinforcement learning, in: 2020 IEEE 21st International Symposium on" a World of Wireless, Mobile and Multimedia Networks", WoWMoM, IEEE, 2020, pp. 88–97.

[119] A. Hamza, H.H. Gharakheili, T.A. Benson, V. Sivaraman, Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity, in: Proceedings of the 2019 ACM Symposium on SDN Research, 2019, pp. 36–48.

[120] L. Nie, W. Sun, S. Wang, Z. Ning, J.J. Rodrigues, Y. Wu, S. Li, Intrusion detection in green internet of things: A deep deterministic policy gradient-based algorithm, IEEE Trans. Green Commun. Netw. 5 (2) (2021) 778–788.

[121] M. Alauthman, N. Aslam, M. Al-Kasassbeh, S. Khan, A. Al-Qerem, K.-K.R. Choo, An efficient reinforcement learning-based Botnet detection approach, J. Netw. Comput. Appl. 150 (2020) 102479.

[122] H. Han, H. Kim, Y. Kim, An efficient hyperparameter control method for a network intrusion detection system based on proximal policy optimization, Symmetry 14 (1) (2022) 161.

[123] Q. Lin, R. Ming, K. Zhang, H. Luo, et al., Privacy-enhanced intrusion detection and defense for cyber-physical systems: A deep reinforcement learning approach, Secur. Commun. Netw. 2022 (2022).

[124] M.S. Frikha, S.M. Gammar, A. Lahmadi, Multi-attribute monitoring for anomaly detection: a reinforcement learning approach based on unsupervised reward, in: 2021 10th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks, PEMWN, IEEE, 2021, pp. 1–6.

[125] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, P. Hakimian, Detecting P2P botnets through network behavior analysis and machine learning, in: 2011 Ninth Annual International Conference on Privacy, Security and Trust, IEEE, 2011, pp. 174–180.

[126] A. Shiravi, H. Shiravi, M. Tavallaee, A.A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Comput. Secur. 31 (3) (2012) 357–374.

[127] B. Rahbarinia, R. Perdisci, A. Lanzi, K. Li, PeerRush: Mining for unwanted P2P traffic, J. Inf. Secur. Appl. 19 (3) (2014) 194–208.

[128] L. Breiman, Classification and Regression Trees, Routledge, 2017.

[129] T.N. Kipf, M. Welling, Variational graph auto-encoders, 2016, arXiv preprint arXiv:1611.07308.

[130] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[131] K.S. Madhuri, J. Mungara, Adaptive reinforcement learning with Dij-Huff method to secure optimal route in smart healthcare system, Cardiometry 25 (2022) 1131–1139.

[132] Z. Liu, Reinforcement-learning based network intrusion detection with human interaction in the loop, in: Security, Privacy, and Anonymity in Computation, Communication, and Storage: 13th International Conference, SpaCCS 2020, Nanjing, China, December 18-20, 2020, Proceedings 13, Springer, 2021, pp. 131–144.

[133] T.V. Phan, T.G. Nguyen, N.-N. Dao, T.T. Huong, N.H. Thanh, T. Bauschert, DeepGuard: Efficient anomaly detection in SDN with fine-grained traffic flow monitoring, IEEE Trans. Netw. Serv. Manag. 17 (3) (2020) 1349–1362.

[134] Z.U.A. Tariq, E. Baccour, A. Erbad, M. Guizani, M. Hamdi, Network intrusion detection for smart infrastructure using multi-armed bandit based reinforcement learning in adversarial environment, in: 2022 International Conference on Cyber Warfare and Security, ICCWS, IEEE, 2022, pp. 75–82.

[135] Microsoft malware prediction, 2018, URL https://www.kaggle.com/competitions/microsoft-malware-prediction/overview/timeline.

[136] P. Wette, M. Dräxler, A. Schwabe, F. Wallaschek, M.H. Zahraee, H. Karl, Maxinet: Distributed emulation of software-defined networks, in: 2014 IFIP Networking Conference, IEEE, 2014, pp. 1–9.

[137] P. Mishra, A. Biswal, S. Garg, R. Lu, M. Tiwary, D. Puthal, Software defined internet of things security: Properties, state of the art, and future research, IEEE Wirel. Commun. 27 (3) (2020) 10–16.

[138] T.V. Phan, T. Bauschert, DeepAir: Deep reinforcement learning for adaptive intrusion response in software-defined networks, IEEE Trans. Netw. Serv. Manag. 19 (3) (2022) 2207–2218.

[139] M. Zolotukhin, S. Kumar, T. Hämäläinen, Reinforcement learning for attack mitigation in SDN-enabled networks, in: 2020 6th IEEE Conference on Network Softwarization, NetSoft, IEEE, 2020, pp. 282–286.

[140] M. Zolanvari, M.A. Teixeira, L. Gupta, K.M. Khan, R. Jain, WUSTL-IIOT-2021 dataset for IIoT cybersecurity research, 2021, URL http://www.cse.wustl.edu/~jain/iiot2/index.html.

[141] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems, IEEE Access 8 (2020) 165130–165150.

[142] M. Al-Hawawreh, E. Sitnikova, N. Aboutorab, X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things, IEEE Internet Things J. 9 (5) (2022) 3962–3977, http://dx.doi.org/10.1109/JIOT.2021.3102056.

[143] M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning, IEEE Access 10 (2022) 40281–40306.

[144] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, A. Hotho, A survey of network-based intrusion detection data sets, Comput. Secur. 86 (2019) 147–167.

[145] M. Sarhan, S. Layeghy, M. Portmann, Towards a standard feature set for network intrusion detection system datasets, Mobile Netw. Appl. (2022) 1–14.

[146] T.M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, F.T. den Hartog, ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets, IEEE Internet Things J. 9 (1) (2021) 485–496.

[147] A. Liatifis, C. Dalamagkas, P. Radoglou-Grammatikis, T. Lagkas, E. Markakis, V. Mladenov, P. Sarigiannidis, Fault-tolerant SDN solution for cybersecurity applications, in: Proceedings of the 17th International Conference on Availability, Reliability and Security, 2022, pp. 1–6.

[148] ARCADIAN-IoT, 2023, https://www.arcadian-iot.eu/. (Accessed 14 April 2023).

[149] H. Wang, L. Muñoz-González, M.Z. Hameed, D. Eklund, S. Raza, SparSFA: Towards robust and communication-efficient peer-to-peer federated learning, Comput. Secur. (2023) 103182.

[150] IRIS-H2020, 2023, https://www.iris-h2020.eu/. (Accessed 14 April 2023).

[151] CAREER: Foundations for IoT cloud security, 2023, https://www.nsf.gov/awardsearch/showAward?AWD_ID=2145675. (Accessed 14 April 2023).

[152] M. Babiker Mohamed, O. Matthew Alofe, M. Ajmal Azad, H. Singh Lallie, K. Fatema, T. Sharif, A comprehensive survey on secure software-defined network for the internet of things, Trans. Emerg. Telecommun. Technol. 33 (1) (2022) e4391.

[153] M.A. Ja'afreh, H. Adhami, A.E. Alchalabi, M. Hoda, A. El Saddik, Toward integrating software defined networks with the internet of things: a review, Cluster Comput. (2022) 1–18.

[154] A.K.C.S. Boni, Y. Hablatou, H. Hassan, K. Drira, Resilient deep reinforcement learning architecture for task offloading in autonomous IoT systems, in: The 12th International Conference on the Internet of Things, IoT 2022, 2022.

[155] L. Qi, Y. Yang, X. Zhou, W. Rafique, J. Ma, Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0, IEEE Trans. Ind. Inform. 18 (9) (2022) 6503–6511, http://dx.doi.org/10.1109/TII.2021.3139363.

[156] D. Lowd, C. Meek, Adversarial learning, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 2005, pp. 641–647.

[157] H.C. Altunay, Z. Albayrak, A hybrid cnn+ lstmbased intrusion detection system for industrial IoT networks, Eng. Sci. Technol. Int. J. 38 (2023) 101322.

[158] J. Vitorino, R. Andrade, I. Praça, O. Sousa, E. Maia, A comparative analysis of machine learning techniques for IoT intrusion detection, in: Foundations and Practice of Security: 14th International Symposium, FPS 2021, Paris, France, December 7–10, 2021, Revised Selected Papers, Springer, 2022, pp. 191–207.

[159] R. Ahmad, I. Alsmadi, W. Alhamdani, L. Tawalbeh, Zero-day attack detection: a systematic literature review, Artif. Intell. Rev. (2023) 1–79.

[160] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data? Adv. Neural Inf. Process. Syst. 35 (2022) 507–520.

[161] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, 2018, arXiv preprint arXiv:1806.01261.

[162] G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbell, L.E. Barnes, M. Boukhechba, Graph neural networks in IoT: A survey, ACM Trans. Sensor Netw. (2022).

Jesús F. Cevallos M. received a Ph.D. in Computer Science Engineering from Sapienza University (Rome) in 2022. He now covers a post-doc researcher position at University of Insubria (Varese). His main research interests are industrial applications of Deep Learning over heterogeneous networks, with a special focus on Deep Reinforcement Learning and Graph Representation Learning.

Alessandra Rizzardi is Assistant Professor at University of Insubria (Varese), where she received BS/MS degree in Computer Science 110/110 cum laude in 2011 and 2013, respectively. In 2016 she got Ph.D. in Computer Science and Computational Mathematics at the same university, under the guidance of Prof. Sabrina Sicari. Her research activity is on WSN and IoT security issues. She is member of ETT, ITL, and Sensors editorial board. She is IEEE member.

Sabrina Sicari is Associate Professor at University of Insubria (Varese). She received degree in Electronical Engineering, 110/110 cum laude, from University of Catania, in 2002, where in 2006 she got Ph.D. in Computer and Telecommunications Engineering, followed by Prof. Aurelio La Corte. She is member of COMNET, IEEE IoT, ETT, ITL editorial board. Her research activity security, privacy and trust in WSN, WMSN, IoT, and distributed systems. She is IEEE senior member.

Alberto Coen Porisini received Dr. Eng. degree and Ph.D. in Computer Engineering from Politecnico di Milano in 1987 and 1992. He is Full Professor of Software Engineering at Università degli Studi dell'Insubria since 2001, Dean of the School of Science from 2006 and Dean from 2012 to 2018. His research regards specification/design of realtime systems, privacy models and WSN.