Proceedings Paper:
Pasikhani, A.M. orcid.org/0000-0003-3181-4026, Clark, J.A. orcid.org/0000-0002-9230-9739 and Gope, P. orcid.org/0000-0003-2786-0273 (2022) Reinforcement-learning-based IDS for 6LoWPAN. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 20-22 Oct 2021, Shenyang, China. Institute of Electrical and Electronics Engineers (IEEE) , pp. 1049-1060. ISBN 9781665416597

https://doi.org/10.1109/trustcom53373.2021.00144

Reinforcement-Learning-based IDS for 6LoWPAN

Aryan Mohammadi Pasikhani
Department of Computer Science
The University of Sheffield
Sheffield, UK
amohammadipasikhani1@sheffield.ac.uk

John A. Clark
Department of Computer Science
The University of Sheffield
Sheffield, UK
john.clark@sheffield.ac.uk

Prosanta Gope
Department of Computer Science
The University of Sheffield
Sheffield, UK
p.gope@sheffield.ac.uk

***Abstract*—The Routing Protocol for low power Lossy networks (RPL) is a critical operational component of low power wireless personal area networks using IPv6 (6LoWPANs). In this paper we propose a Reinforcement Learning (RL) based IDS to detect various attacks on RPL in 6LoWPANs, including several unaddressed by current research. The proposed scheme can also detect previously unseen attacks and the presence of mobile intruders. The scheme is well suited to the resource constrained environments of our target networks.**

***Index Terms*—IDS, Reinforcement-Learning, RPL-attack, Machine Learning, RPL, 6LoWPAN**

I. Introduction

The IPv6 over low-power wireless personal area networks (6LoWPAN) standard enables resource-constrained devices to connect to the IPv6 network and be reachable over the Internet. Because of massive connectivity and significant computational constraints of Low power and Lossy Network (LLN) nodes, a new routing protocol called the Routing Protocol for low power Lossy networks (RPL) has been proposed to associate routes between LLN nodes and the IPv6 Border Router (6BR). Routing relies on the construction of suitable Destination-Oriented Directed Acyclic Graphs (DODAGs) using node rank values to structure the graphs. The ranking system enables various properties such as route discovery, loop prevention, and overhead management, but is vulnerable to several attacks [1], [2] that can significantly degrade resource utilisation, routing mechanisms and general network performance. Protecting against attacks on the RPL is of critical importance but computational limitations of LLN nodes present barriers to the adoption of highly promising leading-edge approaches such as those based on machine learning (ML). Here we show how an approach based on Reinforcement Learning (RL), a particular kind of ML, can be both effective against the range of RPL attacks and also resource efficient.

II. Related Works and Motivations

With the increasing number of LLN devices a significant number of internal and external threats against 6LoWPAN have emerged. Securing LLNs against routing attacks using Intrusion Detection Systems (IDSs) has become a significant research focus. Below we classify the relevant research articles into three categories: IDS for RPL, ML-IDS for RPL, and RL for IDS. No extant research uses an RL-based IDS to mitigate RPL attacks. (Some studies use RL to enhance IDS performance against threats to different network technologies.)

Researchers have investigated the detection of RPL attacks using signature-based, anomaly-based, and specification-based approaches, or a hybrid of those approaches. (For a survey of IoT-related IDS systems the reader is referred to [1].) Svelte [7] proposes a hybrid (signature-based and specification-based) IDS designed to monitor an LLN in a distributed manner, collecting traffic from nodes. As Svelte addressed only grayhole and blackhole attacks, the authors of [3] were encouraged to develop a specification-based IDS to detect Sybil and Wormhole attacks. In [6] a different approach to detect wormhole attacks was taken, considering nodes to be equipped with GPS to transfer their location information to the centralised specification-based IDS. [6] and [8] use passive monitoring techniques to analyse LLN traffic and detect RPL attacks using a specification-based detection strategy. The limitations of specification-based detection strategies encouraged researchers to propose ML-IDS for mitigating RPL attacks. In [8] the use of various ML methods (Naïve Bayes, MLP, SVM, and Random Forests) was investigated to detect version number, sinkhole, blackhole, Sybil, and decrease rank attacks targeting RPL using the MRHOF and OF0 objective functions (specific performance metrics the RPL routing algorithm seeks to optimise) [1], [18]. They evaluated their proposed hybrid IDS over a small-scaled LLN with a single malicious node. Similarly, [9] investigates different ML methods (J48 Decision Tree, Logistic, MLP, Naïve Bayes, Random Forest, and SVM) and proposes a hybrid ML-IDS with passive monitoring to detect sinkhole, wormhole, and DIS flooding. The unsupervised K-means and supervised Decision Tree (DT) algorithms are used by [10] to develop a centralised hybrid ML-IDS capable of detecting the wormhole attack. The work of [11] uses unsupervised Optimum-Path Forest Clustering (OPF) to develop specification-based anomaly-based decentralised ML-IDS to mitigate wormhole, sinkhole, and grayhole attacks.

Extant research has not proposed using RL to ensure security in the 6LoWPAN network. However, there are several studies [12]–[17] where RL is used to enhance IDS performance in detecting application-based attacks. They employ Q-learning [13] and a centralised hybrid IDS to perform the detection task over the data received through cluster heads in the WSN. The work of [14] employs Deep RL (DRL) for developing a centralised anomaly IDS. In their proposed model RL is used to enhance anomaly IDS detection performance. Similarly, [12] investigates different RL methods,

This paper has been accepted for oral presentation at 20th IEEE International Conference on Trust, Security and Privacy in Computing and Communication (TrustCom2021)

TABLE I
RELATED WORKS

Scheme	Method	Attack	Desirable Imperative Features for IDS					
			DF1	***DF2***	***DF3***	***DF4***	***DF5***	***DF6***
[3]	Specification-based IDS, Highest Rank Common Ancestor	Wormhole and Sybil (Cooja)	×	×	×	×	×	×
[4]	Specification-based IDS with a passive decentralised monitoring system Threshold-based	DODAG Inconsistency (Cooja)	×	×	×	✓	✓	×
[5]	Specification-based (Threshold-based) IDS with a passive decentralised monitoring system	DIS Flooding (Cooja)	×	×	×	×	✓	×
[6]	Specification-based IDS (requires geographical information of nodes)	Wormhole (Cooja)	×	×	×	×	×	×
[7]	Hybrid	Sinkhole and Grayhole (Cooja)	×	×	×	×	×	×
[8]	ML-IDS using voting technique (MLP, Random Forest)	Version, Rank, Sybil, Decrease Rank, Blackhole (Cooja)	×	×	×	×	×	×
[9]	Hybrid ML-IDS using passive monitoring technique J48 Decision Tree, Logistic, MLP, Naïve Bayse, Random Forest, and SVM	Sinkhole, Wormhole, and DIS Flooding (Cooja)	×	×	×	×	×	✓
[10]	Hybrid ML-IDS (Unsupervised K-means and supervised Decision Tree)	Wormhole attack (unknown C++ platform)	×	×	×	×	×	×
[11]	Anomaly ML-IDS Unsupervised Optimum-Path Forest Clustering (OPF)	Sinkhole, Grayhole, and wormhole (unknown C platform)	×	D/N	D/N	D/N	D/N	D/N
[12]	Anomaly-based IDS using RL in training phase. Experiment different RL algorithms (DQN, DDQN, Actor-Critic, and PG)	NSL-KDD and AWID	✓	D/N	D/N	D/N	D/N	D/N
[13]	Use RL Q-learning algorithm to develop centralised hybrid IDS in WSN	KDD Cup 1999	✓	D/N	D/N	D/N	D/N	D/N
[14]	Centralised anomaly-based IDS using Deep RL (DRL)	NSL-KDD and UNSW-NB15	✓	D/N	D/N	D/N	D/N	D/N
[15]	RL (Q-learning) based IDS	NSL-KDD	✓	D/N	D/N	D/N	D/N	D/N
[16]	Use DRL and Q-learning to enhance IDS performance through the adversarial training procedures	NSL-KDD and AWID	✓	D/N	D/N	D/N	D/N	D/N
[17]	Distributed DRL for IDS	NSL-KDD, UNSW-NB15 and AWID	✓	D/N	D/N	D/N	D/N	D/N
Our scheme	RL-based heterogeneous hybrid IDS	SH,BH,GH,IR,RA,DA,WH,DS	✓	✓	✓	✓	✓	✓

*D/N: Different Network-technology.
* In the ”Attack” column, the later entries refer to available datasets that contain a variety of attacks (but these exclude RPL attacks)

namely DQN, Double DQN (DDQN), Actor-Critic, and Policy Gradient (PG), to improve the performance of a supervised anomaly-based IDS over the training phase. Enhancement of IDS performance using an adversarial RL training environment has been used by [16], [17]. In [17], researchers employ distributed DRL to boost IDS performance and prepare it against adversarial attack. The authors of [15] investigate the use of model-free Q-learning in intrusion detection using the NSL-KDD dataset.

A. Desirable Characteristics

Below we identify various desirable important characteristics that could be expected of a high performing IDS in our target domain. These are based on our own views and those of other researchers ([1], [19], [20]).

DF1: Adaptivity. The IDS should be capable of improving its performance over time as data and experience increases.

DF2: The IDS should be capable of securing LLNs with mobile normal and malicious nodes. Few studies consider mobility in RPL attacks.

DF3: The IDS should be able to detect a wide range of RPL attacks. Published IDS schemes address only subsets of known RPL attacks and do not evaluate outside the chosen subsets.

DF4: The IDS must secure 6LoWPAN against both internal and external intrusions.

DF5: The IDS should have low network traffic overheads. The 6LoWPAN is known for its lossy environment and low (250kbps) bandwidth. Many existing IDS approaches incur significant network overheads, e.g., centralised decision-making approaches such as [8], [10].

DF6: The IDS should be able to detect known and previously unseen intrusions.

B. Our Contribution

This paper introduces a new RL-based IDS (RL-IDS) that utilises heterogenous ML-based IDSs over the 6LoWPAN. A variety of internal (inside 6LoWPAN) and external (over the Internet) RPL attacks (Sinkhole, Blackhole, Grayhole, DIS flooding, Wormhole, DIO Suppression, Increase Rank, and Replay) are handled by our proposed approach. Our paper:

- proposes an RL-IDS to enhance the strength of distributed ML-IDS in detecting internal and external RPL intrusions.
- engineers a set of features and correlates its elements with the effects each RPL attack has on an LLN.
- evaluates different supervised and unsupervised ML algorithms and develops hybrid ML-IDS approach better

suited to detection of known and previously unseen malicious activities and attacks.
- proposes for the first time an IDS to detect Increase Rank (IR) , DIO Suppression (DS), and Replay attacks [1], [2].
- addresses for the first time attack scenarios with malicious mobile nodes.
- addresses for the first time both individual and combinations of RPL attacks.
- evaluates the performance of the proposed scheme in various scaled LLNs with respect to different numbers of malicious nodes.

The rest of the paper is organized as follows. In Section III, we present brief introductions to DODAGs, RPL attacks and reinforcement learning. In Section IV we indicate how informative features are developed and selected. In Section V we describe the RL-based intrusion detection scheme, ML-based detectors, and the development of a flexible system using RL algorithms. In Section V-C, the simulation setup is described and the experiments are carried out and results reported. Finally, concluding remarks and analysis of results are given in Section VI-B.

III. Preliminaries

A. DODAG

IETF has developed the RPL routing protocol [18] to enable routing among nodes in low power and lossy networks. RPL is intended to work in LLNs with a low data rate (∼250 kbps) [1], low throughput and high packet loss rate. Moreover, there is an assumption that links would be lossy and occasionally unreachable for an extended period; therefore, when the preferred path is inaccessible RPL is required to provide an alternative route. The RPL protocol constructs a network topology through the formation of a Destination-Oriented Directed Acyclic Graph (DODAG). It aims to maintain wireless communication in a large-scale wireless sensor network for various applications, including urban, industrial, residential [1]. In a DODAG nodes need to communicate to the 6LoWPAN border router to communicate with another part of the network or reach the Internet. In LLNs it is likely there is more than one path available for each node to communicate with the border router (root); however, the nodes are only permitted to have one parent (the preferred parent) with regards to the DODAG Objective Function (OF). To build and maintain a DODAG, RPL follows the neighbour discovery procedure using three ICMPv6 control messages [1], [18]: a DODAG Information Object (DIO), a Destination Advertisement Object (DAO), and a DODAG Information Solicitation (DIS). The DIO message initiates the formation of a DODAG. It contains information about the link, node metrics, and OF that each node uses to nominate the preferred parent [18]. The node metrics contain values such as the expected transmission count (ETX) and the residual energy [1], [18]. Periodically LLN nodes multicast DIOs to maintain the DODAG. The ranking system in RPL is intended to facilitate the construction of routing toward the root by determining parent and child relations between nodes.

The selection of a parent is based on the nodes' advertised ranks in their DIO messages. The rank reflects the node distance to the root; the closer they are to the root, the lower rank they obtain regarding the OF. The OF determines how the rank should be calculated in the DODAG; several OFs already have been proposed to perform rank calculation in the RPL, e.g., Objective Function Zero (OF0) and the Minimum Rank with Hysteresis Objective Function (MRHOF) [1], [18]. The node with a lower rank is more preferred by its neighbours as a parent. If the receiver of the DIO message is not connected to a parent with the same or better advertised rank, it unicasts a DAO message to the sender of the DIO message and expresses its interest to select that node as its preferred parent. In response, nodes respond to the sender of DIO unicast DAO with an acknowledgement flag enabled (DAO-Ack) to accept the DAO request. The DIS message is designed to allow new nodes to discover a DODAG in their neighbourhood.

The RPL has two routing modes of operation, namely, storing mode and non-storing mode. In storing mode parent nodes create a routing table and insert all routing entries for all descendent nodes in its sub-DODAG. While in the non-storing mode only the root (border router) collects and maintains routing information of the whole DODAG. In non-storing mode all traffic goes upward to the root, and then the root selects the routing path to transfer packets. This causes significant network overhead for nodes around the root [18].

B. RPL Attacks

The RPL is exposed to various types of routing attacks [1], [2]. In RPL, intruder alters DODAG control packets' configurations (node's rank, version number, DODAG configuration etc.) to manipulate the confidentiality, integrity and availability (CIA) of data in 6loWPAN [1], [2]. In general, the intruder may disrupt LLN by altering the DIO packet (Sinkhole, Blackhole, Grayhole, and Increase Rank attacks), replaying collected altered control packets (Wormhole and Replay attacks), or flooding control packets (DIS flooding and DIO Suppression attacks). In our previous paper [1], we provide a comprehensive analysis of existing RPL attacks. Additionally, they [2] provide a detailed overview of RPL intrusions.

C. Reinforcement learning

Reinforcement learning is an important area of machine learning that enables an agent to interact with its environment and learn through a trial and error process by receiving feedback from the actions it takes. Specifically, it helps an agent/decision-maker learn the system's dynamic through observations and interactions with the environment. The environment is everything outside the agent. The agent receives the observation (current state s_t) and the reward (r_t) from the environment at each iteration and follows its action value-function (Q) to take the action that increases the long-term reward. The action is the thing that agent can do in the environment given it is in the current state. The action value-function $q_\pi(s_t, a_t)$ informs the agent how taking the action

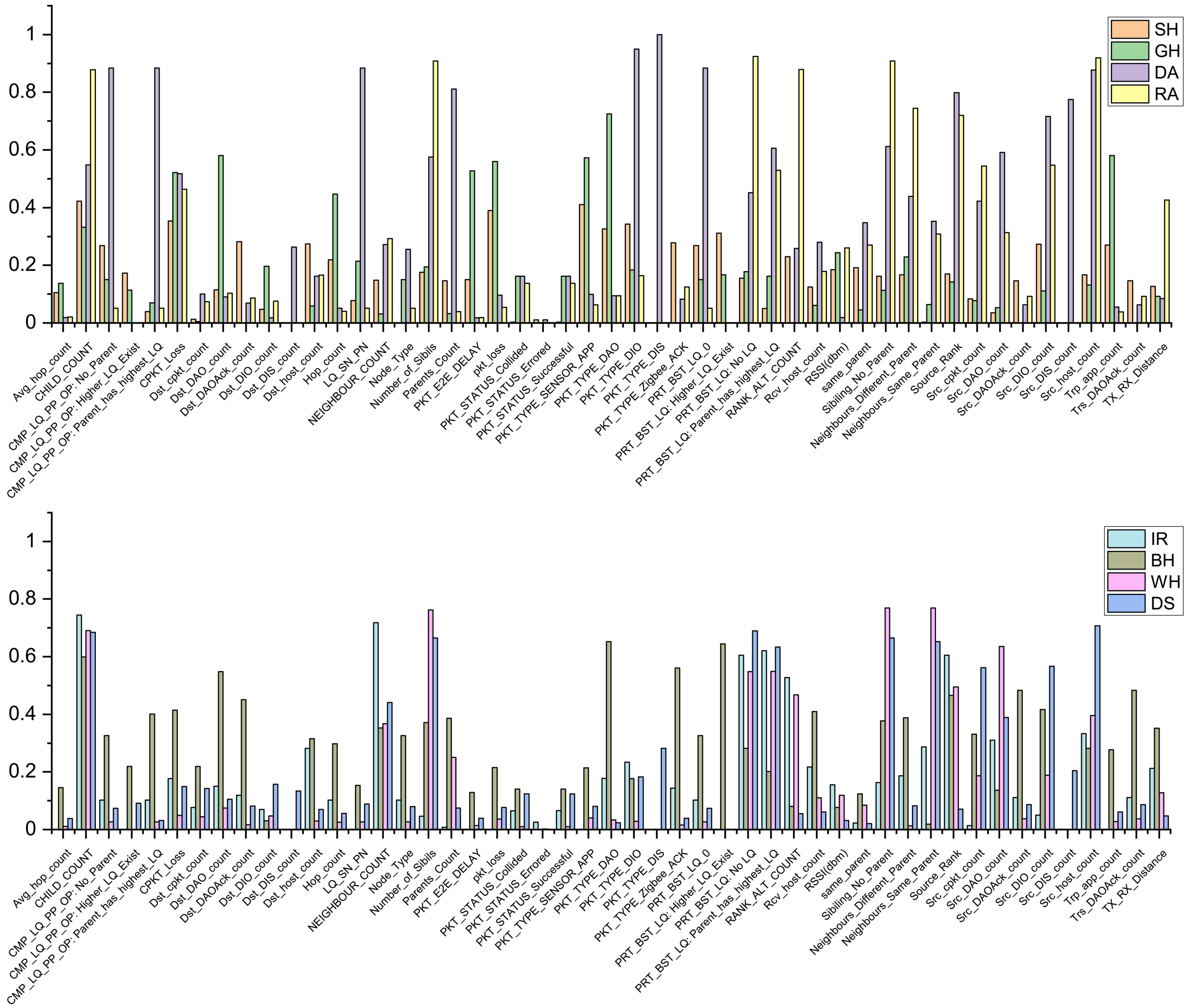


Fig. 1. Features' Correlations.

a_t is good (in terms of expected return) at the given state s_t while following policy π. The reward (r_t) can be positive or negative (penalty) and indicates to the agent how well the agent has behaved.

In RL a transition function can be formulated as a Markov Decision Process (MDP), a mathematical framework for modelling sequential decision-making. MDP characterises the agent interaction with its environment in a sequential decision-making process; the environment computes transition and rewards, and the agent generates the policy. The policy π is probability distribution that forms the behaviour of the agent. Formally, π is defined as $\pi(a|s) = P[A_t = a|S_t = s]$.This is Markovian because the actions depend only on the current state, not how the system got into that state. (Markovian means memoryless.)

There are different approaches for computing policies and value-functions, namely look-up tables and approximation methods [12]. Since 6LoWPAN has a continuous environment, using a look-up table would be a highly resource-intensive task. Therefore this paper uses the DQN and DDQN approximation method.

IV. Feature Engineering

The data elements that feed into our decision making algorithms are generally referred to as 'features'. Obtaining sets of high performing informative features is generally referred to as Feature Engineering (FE). We have identified a variety of potential features and determined how correlated they are with the effects of the various RPL attacks considered. This is illustrated in Fig. 1, using the Pearson Correlation Coefficient's absolute value.

Enhancing algorithm accuracy and interpretability is the main aim of feature selection methods [21]. Feature selection may improve accuracy and efficiency. Feature selection reduces the memory footprint necessary for storing and executing the models and storing the raw data to a lesser degree.

Similarly, it can reduce run-time, both during training and prediction. This study employs feature selection methods for constructing and selecting subsets of features to generate a good predictor.

In roughly normally distributed and categorical data, the predominant advice is to use Chi-Square. Mutual information and Gini Impurity are also reasonable options to consider. The Analysis of Variance (ANOVA) works well for categorical features (independent variables) and a continuous target (dependant variable); Pearson's R2 works well for continuous features and a continuous target.

Since the RPL traffic dataset contains both continuous and categorical features and a categorical target, we use filter method feature selection Chi-square, Gini impurity to reduce the feature set's size and make it less costly in terms of time and computational resources. The Wrapper feature selection methods are computationally expensive [21]; therefore, this study avoids implementing such methods. Based on our experiments, chi-square is fast and can avoid over-fitting while it is computationally inexpensive compared to other feature selection methods.

The Chi-square (X^2) [22] is a statistical filter method that measures the deviation from the expected distribution considering the feature event is independent of the target value. X^2 measures how expected count (E) and observed count (O) deviate from each other Eq. 1. The intuition is that if the feature is independent of the target, it is uninformative for classifying observations.

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{1}$$

V. Proposed Scheme

In this section, we present the proposed IDS methodology for 6LoWPAN networks. Since LLN nodes have limitations in terms of the computational resources, hence they cannot afford the computational requirements of extensive ML algorithms. This paper seeks to address the above issue by proposing an RL-based intrusion detection scheme that uses several lightweight ML-based detectors for analysing 6LoWPAN traffics. Each ML detector trains over a subset of the training data that includes different proportions of attacks. Therefore each detector may have various strengths and weaknesses in detecting the various RPL attacks. The proposed method uses an RL algorithm to identify the appropriate detector for analysing current network terrific. Fig. 2 illustrates the proposed scheme design.

A. ML-based Intrusion Detection

Machine learning (ML) is an intelligent method that optimises system performance using sample data. More precisely, ML algorithms build models of a problem by applying mathematical techniques on sample data sets. The sheer amount of data generated in LLN can make ML bring intelligence to the system for various purpose, including security. ML algorithms are mainly supervised and supervised methods. (In supervised approaches data is labelled with its actual class. In unsupervised approaches it isn't.)

The number of features, training samples, and parameters of ML algorithms play vital roles in defining classifiers' complexity over training and prediction phases. The higher number of features and training data increase algorithm complexity significantly and cause an adverse effect on model generalisation. Although increasing the ML algorithms' sensitivity (assigning higher depth in the decision tree, C value in SVM, smaller k in KNN etc.) may enhance model detection performance, it increases the model's complexity dramatically and leads to over-fitting [19]. Table II shows the complexity (O) of different ML classification algorithms [19].

This research employs both signature-based and anomaly-based IDS (hybrid IDS) [1] to detect known and unknown intrusions efficiently. The RPL attack detection ability of various supervised and unsupervised ML algorithms is investigated, Fig. 6. Some of these ML algorithms provide a slightly better performance, but this comes with the cost of more computational complexity and exhaustion that many LLN nodes cannot afford [19]. Since IoT has a heterogeneous node with different computational resources, this research picks various ML algorithms over the LLN to analyse RPL's communications.

TABLE II
ML Algorithms' Complexity

Algorithm	Training	Prediction
Decision Tree (DT)	$O(n^2p)$	$O(p)$
Support Vector Machine (SVM)	$O(p^2n + p^3)$	$O(n_{sv}p)$
k-Nearest Neighbours (KNN)	$O(np)$	$O(np)$
Gradient Boosting (GB)	$O(npn_{trees})$	$O(pn_{trees})$
Q-learning	-	$O(n^3)$
k-Means Clustering	Zero(negligible)	$O(n^2)$
Neural Network	-	$O((pn_l) + (n_{l1})(n_{l2}) + ..)$
Random Forest (RF)	$O(n^2pn_{trees})$	$O(pn_{trees})$
n = number of training samples; p = number of features; O = complexity;		

B. Reinforcement Learning-based IDS

Supervised and unsupervised ML algorithms mainly focus on data analysis problems, while RL is preferred for comparison and decision-making problems [12], [13], [19]. Fast convergence, finding the action-value function $Q(s, a)$ and optimal policy (π^*) are the main challenges in implementing RL algorithms in a dynamic environment like LLN. The tabular RL methods, such as Temporal Difference, SARSA, and Monte Carlo, are exhaustive and inefficient methods for continuous environments that have large state space. The 6LoWPAN has a non-stationary (continuous) environment

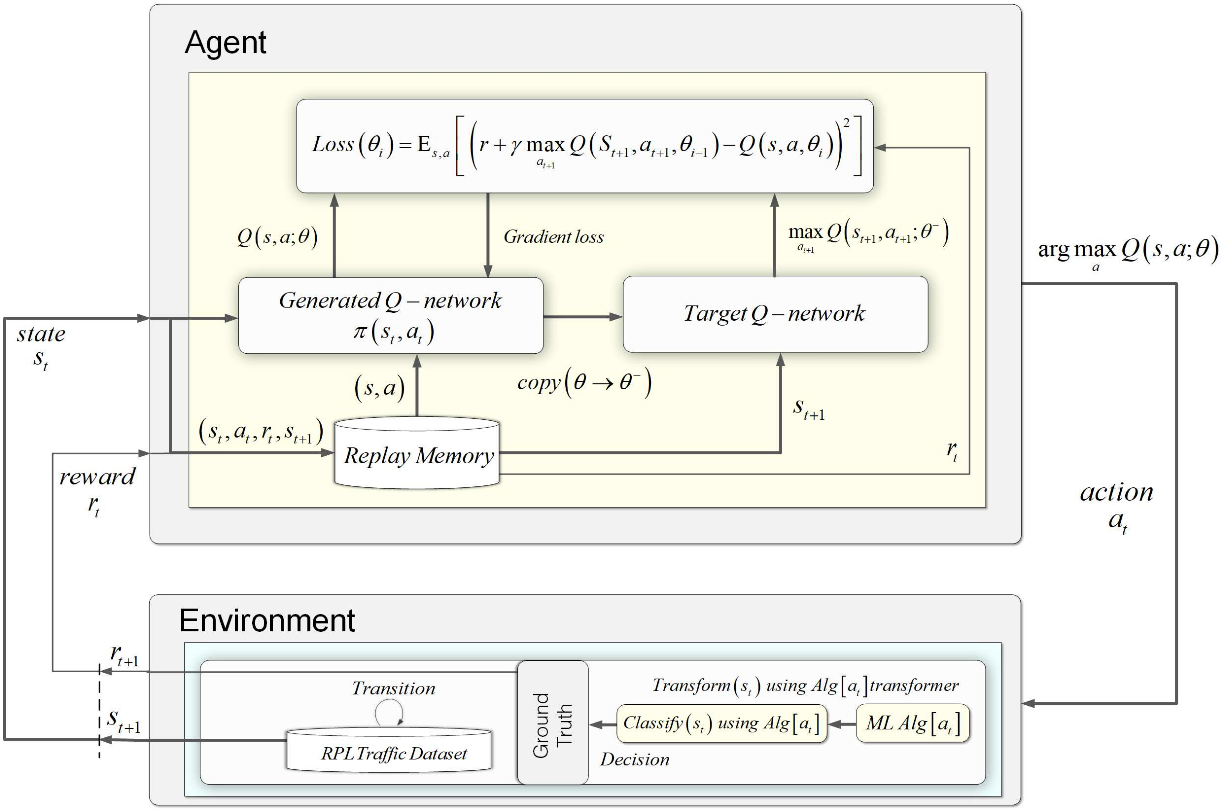

Agent
$Loss(\theta_i) = \mathrm{E}_{s,a}\left[\left(r + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}, \theta_{i-1}) - Q(s,a,\theta_i)\right)^2\right]$
$Q(s,a;\theta)$
Gradient loss
$\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-)$
$\arg\max_a Q(s,a;\theta)$
state s_t
Generated $Q-network$ $\pi(s_t, a_t)$
Target $Q-network$
(s,a)
$copy(\theta \rightarrow \theta^-)$
s_{t+1}
(s_t, a_t, r_t, s_{t+1})
Replay Memory
r_t
reward r_t
action a_t
Environment
r_{t+1}
s_{t+1}
Transition
RPL Traffic Dataset
Ground Truth
$Transform(s_t)$ using $Alg[a_t]$ transformer
$Classify(s_t)$ using $Alg[a_t]$
$ML\ Alg[a_t]$
Decision

Fig. 2. RL-IDS.

with an infinite number of states. Applying tabular methods reduces IDS efficiency and increases its computational needs since the agent will use a lookup table for taking action in each state. Therefore an RL approximation method is required to make the system generalise in the face of unforeseen states and reduce the system complexity. This paper practises DQN and DDQN algorithms to find an optimum policy (π^*) that result in the maximum long-term reward (r). The aim is to yield a policy that delivers optimal long-term returns. The policy (π) represents a probability distribution over actions given the current state (packet).

The DQN and DDQN are model-free off-policy value-based RL algorithms. The model-free algorithm does not build a model of the environment to generate policy. The model-free algorithms are suitable options for LLN since building the environment's dynamics is an expensive and unnecessary task. In off-policy learning, the agent can explore freely - its actions need not correspond to the current policy. In the DQN algorithm (Algorithm 1), the Deep Learning (DL) uses a Q-function $(Q(s,a))$, also known as the action-value function, to approximate the value of taking a specific action (a_t) in the given state (s_t) to help RL in finding the optimum policy (π^*). Since there is no relation between sequence of states in 6LoWPAN (s_{t+1} is not the result of the action the agent has taken at s_t), the discount value (γ) is assigned as 0.001 in this paper.

The Deep Q-Network (DQN) approximates the Q function. The DQN with probability ε selects a random a and with probability $1-\varepsilon$ select optimal Q-function (Q^*), (2). Executing selected action a_t the agent observes next state s_{t+1} and reward r_t and store (s, a, r, s_{t+1}) in the replay buffer D. Algorithm 1 shows how DQN functions.

Although there is a slight correlation between the incoming network terrific, the experiment replay strategy [23] is employed to guarantees the data are Independent and Identically Distributed (IID) to avoid significant oscillations or divergence. The replay buffer D is a data structure including agent experiences $e_1, e_2, \ldots, e_n$ where $e_t = (s_t, a_t, r_{t+1}, s_{t+1})$.

$$Q^*(s,a) = arg\max_{\pi}(s,a) \tag{2}$$

This paper implements a lightweight Neural Network (NN) consisting of two hidden layers using the ReLU activation function to approximate the Q-function. If the selected action a_t (ML-based IDS detectors) makes a correct classification of the current state s_t (packet), the reward is one and -1 otherwise. Since in this paper, the states (packets) are not sequential (the packet that the agent receives at s_{t+1} is not the result of the action that the agent has taken at the previous time step s_t, the γ value assigned is near to zero (0.001).

To train the NN, the loss function needs to be determined. Since the goal of NN is to predict $Q(s, a)$, this paper uses the squared difference between the actual action-value function and the prediction, (3) where θ represents the Q-function's parameter, i.e., the trainable weights of the network. The model aims to decrease the error and make current policy outcomes closer to the true Q-values. Therefore the model performs gradient (∇) descent over loss function using (4) where $Q_{target} = (r + \gamma \max_{a^{'}} Q(s^{'}, a^{'}; \theta^{-})$.

$$L(\theta) = \mathbb{E}_{\pi}[(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta) - Q(s, a; \theta))^2] \quad (3)$$

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{(s,a,r,s' \sim U(D))}[Q_{target} - Q(s, a; \theta_i))\nabla_{\theta_i} Q(s, a; \theta_i)] \quad (4)$$

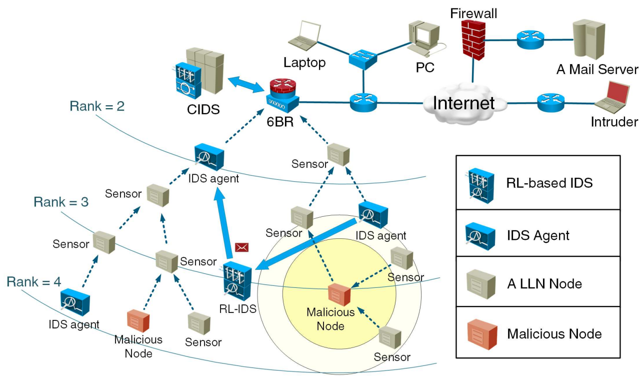


Fig. 3. System Architecture.

DDQN adds double learning to the DQN agent by using two Neural Networks (NNs). DDQN implementation and hyper-parameters are identical to DQN, and both use the off-policy Temporal Difference (TD) target [24]. However, DDQN employs two NNs, one for action prediction and another for action evaluation. Moreover, instead of MSE, DDQN uses Huber loss for loss calculation. Huber loss tunes between MSE and Mean Absolute Error (MAE) using the parameter δ as threshold value [25].

We experiment with different epsilon (ε) values in this research; a higher ε value leads to exploration and taking less selected actions (detectors). This can help the model identify undiscovered ML classifiers that are precise in analysing particular types of network traffic and RPL attacks. Exploiting enhances the system performance by selecting actions (detectors) that have proven to be good at detecting particular types of attacks. Balancing exploration and exploitation by tuning the ε $(0 < \varepsilon < 1)$ value is vital in designing an efficient system. The agent with probability epsilon (ε) explores and with $(1 - \varepsilon)$ exploits. The best strategy is to initialise epsilon as a high value for more exploration and decay it over time to select greedy actions and accumulate more rewards. This study experiments with different exploration-exploitation, ε association strategies (softmax, linearly decaying ε value, etc.) and found that the exponentially decaying ε-greedy strategy [26] provides optimal performance.

Algorithm 1 Deep Q-learning with experience replay

```
***Initialisation***
*Initialise replay memory D to capacity N*
*Initialise action-value function Q with random weights $\theta$*
*Initialise target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$*
**for** *episode=1, M* **do**
    *Initialise sequence $s_1$ =$\{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$*
    **for** *t=1, T* **do**
        *With probability $\varepsilon$ select a random action $a_t$*
        *Otherwise select $a_t$ = arg $max_a Q(\phi(s_t), \alpha; \theta)$*
        *Execute action $a_t$ in emulator and observe $r_t$ and $x_{t+1}$*
        *Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$*
        *Store transition $(\phi, a_t, r_t, \phi_{t+1})$ in D*
        *Sample random mini-batch of transitions $(\phi, a_t, r_t, \phi_{t+1})$ from D*
        **if** *episode terminates $s_{j+1}$* **then**
            $y_j = r_j$
        **else**
            $y_j$ = $r_j + \gamma max_{a'} \hat{Q}(\phi_{j+1}, a';\ \theta^{-})$
        *Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters $\theta$  Every C steps reset $\hat{Q} = Q$*
```

The computational complexity of Deep Q-Network (DQN) depends on different factors: the number of hidden layers, the number of neutrons per layer, etc. In DQN and Double DQN (DDQN), the environment has continuous state space, and computational complexity differs based on the algorithm strategy. In DQN using the experience replay method, the batch size defines the complexity [19].

TABLE III
SIMULATION PARAMETERS

Parameters	**Values**
Simulator	Tetcos Netsim V12.2
Number of nodes	16, 32, 64, 128
Number of Malicious nodes	$\sim$ 10%, $\sim$ 20%, $\sim$ 30%
Number of Workstations	4, 8
Transmission Range	50m
Number of ML detectors	$\sim$ 10%
Scenario Dimension (Terrain)	(250 × 250) to (850 × 850) s.meters
Traffic Rate	250 kbps
Simulation time	1,800 $\sim$ 21,600 seconds
Application Protocols	COAP, CBR
RPL mode	Storing and Non-storing
Mobility Modes	Random Walk, Group Walk
Path Loss Model	Log Distance, Exponent(n): 2
Distance between LLN Neighbors	25 $\sim$ 45m
Objective Function (OF)	OF0, MRHOF, LQ
Receiver Sensitivity	-85 dBm

C. Simulation Experiments

a) Exploring Datasets: In this paper, the dataset is generated through simulations of several RPL scenarios with a different number of malicious nodes. In each scenario, static and mobile nodes are randomly distributed over an LLN. The Tetcos Netsim simulator is used to simulate different RPL attack scenarios and generate raw datasets. The imbalanced dataset will be rectified during the pre-processing phase. The redundant, less informative records are removed from the dataset to make normal and malicious traffic normally distributed in the training dataset. Some ML algorithms (SVM, Logistic regression, etc.) are very sensitive about the scale of data [19] ; therefore, feature normalisation (Min-Max Scalar) and standardisation (Standard Scalar) techniques are adopted to scale features. This prevents IDS from being over-fitted to a particular type of traffic. The training dataset contains 48 features and 80,000 instances. The normal traffic constitutes 50 per cent of the dataset, while each attack equally has 5 per cent of the dataset.

Algorithm 2 RL-IDS Algorithm in action

```
Initialisation
$S_{pkt}$: Collected packet
$C_{pkt}$ {DIO, DAO, DIS, DAO-Ack, Application Packet}
CIDS: Central IDS
$RL_{alg}$: RL algorithm
$RL_{agent}$: RL agent
$IDS_{ad}$ collect $S_{pkt}$ from a LLN node
if $S_{pkt} \in C_{pkt}$ then
    $D_1 \leftarrow IDS_{ad}.analyse(S_{pkt})$
    if $D_1$ = Abnormal then
        Transfer $S_{pkt}$ → RL_Agent
        Regarding RL function-approximation algorithm (DQN or DDQN) compute $a \leftarrow argmaxQ^*(s_t, a)$ (select IDS agent) given current state $s_t$ ($S_{pkt}$)
        Take action 'a', (Transfer $S_{pkt} \rightarrow IDS_i[a]$)
        $D_2 \leftarrow IDS_i.analyse(S_{pkt})$
        Send $D_2$ → RL_agent
        if $D_2$ = Abnormal then
            Transfer the alarmed packet ($S_{pkt}$) to CIDS and notify Administrator
            if CIDS.analyse($S_{pkt}$) = intrusion then
                Send Reward (+1) → $RL_{agent}$
                Notify Administrator
            else
                Send Penalty (-1) → $RL_{agent}$
            $RL_{agent}$ receives feedback from CIDS and updates Q-function
```

b) Data Preprocessing: The data pre-processing reduces dataset complexity for ML algorithms; therefore, the ML algorithm can be trained over the pre-processed data faster and more efficiently than the raw data [27]. In this paper, the data-processing constitutes data reduction, feature engineering, normalisation, and data sampling [27].

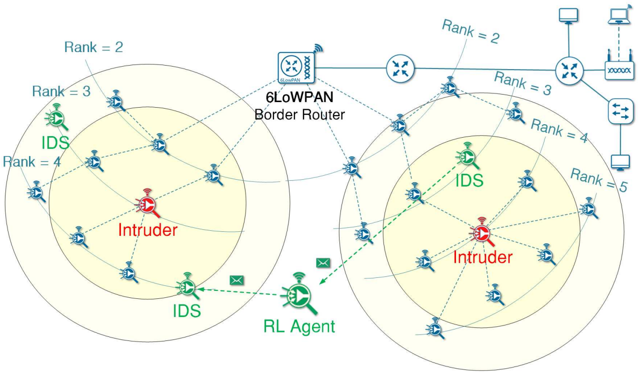
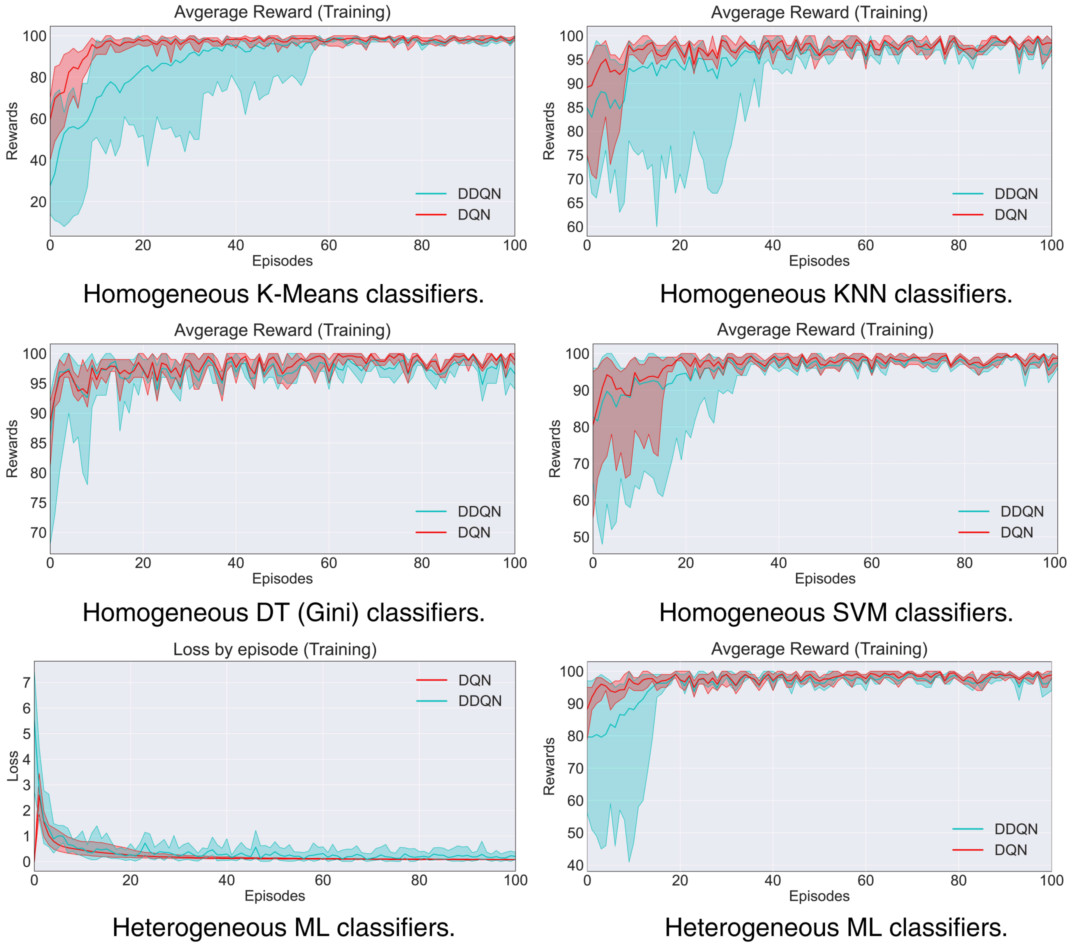


Fig. 4. Simulation Environment.

TABLE IV
ENGINEERED FEATURES

Feature	Description
pkt_type	Type of packet (DIO, DAO, DIS, App etc)
pkt_status	Packet status
dio_count	No. of DIO advertised by sender
avg_hopcount	Average No. of hopcount (global perspective)
dis_count	No. of DIS unicasted/multicasted by sender
dao_count	No. of DAO unicasted by sender
daoack_count	No. of DAO-Ack unicasted by sender
neighbour_count	No. of neighbouring node
child_count	No. of children
avg_intpkt_time	Average delay between packets
rank_alteration_count	No. rank alteration
cmp_sender_parent_lq	Compare link quality of sender with its parent
snd_ctrl_count	No. control packet transferred by sender
cmp_lq	compare if sender has lower link quality than current node but advertise better rank
rcv_dao_count	No. of DAO received by current node
rcv_dio_count	No. of DIO received by current node
rcv_dis_count	No. of DIS received by current node
rcv_daoack_count	No. of DAO-Ack received by current node
trans_app_count	No. of application packet transferred by sender
pkt_e2e_delay	packet end-to-end delay
pkt_loss	Application packet loss ratio
cpkt_loss	Control packet loss ratio
src_rank	Sender rank in DODAG
adv_vn	advertised version number
rx_sens	Average receiver sensitivity
tx_power	Average transmission power
rssi	Received signal strength indicator of sender
same_parrent	sender has same parent as detector node
rcv_cpkt_count	No. of control packets received by sender node
prt_bst_lq	Current parent provide best link quality

c) Data Generation: This paper uses Tetcos Netsim Simulator to simulate normal, and anomalous RPL traffics, Fig. 4. The Netsim is an eminent paid license software known for accurate simulation of different network technologies, including 6LoWPAN. This paper simulates several networks scenarios (using the scenario generator feature of simulator) for each type of RPL attack with different static and mobile nodes, from 8 to 128 nodes. Concerning the network's scales and the number of normal nodes, 10% to 30% of nodes associate as malicious nodes in scenarios. In Wormhole and DIS flooding attacks, half of the malicious nodes associated



Homogeneous K-Means classifiers.

Homogeneous KNN classifiers.

Homogeneous DT (Gini) classifiers.

Homogeneous SVM classifiers.

Heterogeneous ML classifiers.

Heterogeneous ML classifiers.

Fig. 5. Evaluation results of Heterogeneous and Homogeneous ML detectors.

as external intruders. In all scenarios up to 10% of nodes considered as IDS detectors in simulations. To generate a sufficient amount of malicious and normal traffics, based on the type of RPL attack each scenario is simulated for 1,800 to 21,600 seconds.

d) Feature Construction: Feature construction, also referred to as feature engineering, emphasises that engineering salient features from the observed traffic leads to enhancement in classification. Every observed network packet contains different information about node configurations and identity. Training using the identity information of nodes leads to over-specialisation (over-fitting). Therefore such features should be excluded from training datasets. Constructing features based on nodes' geographical location [6], computational resource usage (CPU, RAM, ROM usages) [8], and power consumption [8], [9] can exhaust LLN nodes' resources [1]. Moreover, this significantly increases network overhead [28] on the LLN because nodes need to transfer such logged information to the IDS.

The header of RPL control packets (DIO, DAO, DIS, and DAO-Ack packets) contains information about node configurations, version number, advertised rank [1], [18]. Extracting information from these unicasted/multicasted control packets can help in constructing several features, described in Table IV. The engineered features play a vital role in improving the proposed IDS performance in detecting each RPL attacks.

VI. Experimental Methodology

The proposed scheme employs both signature-based and anomaly-based ML algorithms to enhance the performance of IDS in detecting known and unknown intrusions. The proposed hybrid RL-IDS uses a passive decentralised monitoring technique [28] using a cluster-based placement [29] strategy to analyse 6LoWPAN traffics. The intended flow of the proposed scheme is shown in Fig.3, the algorithm itself is described in Algorithm 2. We now evaluate the performance of the proposed method over 6LoWPANs with respect to different configurations and numbers of malicious nodes to affirm the integrity of results.

To evaluate the performance of the proposed scheme in detecting RPL attacks, four experiments (denoted as Exp1-

TABLE V
EVALUATION RESULTS, TRUE POSITIVE RATE AND FALSE NEGATIVE RATE

N	M	TPR								FNR							
		SH	***BH***	***GH***	***DA***	***IR***	***WH***	***DS***	***R***	***SH***	***BH***	***GH***	***DA***	***IR***	***WH***	***DS***	***R***
16	10%	99	99	100	100	93	100	92	100	1	1	0	0	7	0	8	0
	20%	95	97	98	100	100	100	98	99	5	3	2	0	0	0	2	1
	30%	95	94	97	100	94	99	99	100	5	6	3	0	6	1	1	0
32	10%	92	98	94	100	98	100	98	94	8	2	6	0	2	0	2	6
	20%	99	98	98	100	99	100	100	100	1	2	2	0	1	0	0	0
	30%	97	98	99	100	98	92	100	98	3	2	1	0	2	8	0	2
64	10%	99	98	93	100	97	95	84	100	1	2	7	0	3	5	16	0
	20%	91	90	94	100	98	100	93	97	9	10	6	0	2	0	7	3
	30%	92	90	93	100	95	99	99	100	8	10	7	0	5	1	1	0
128	10%	99	95	96	100	98	99	94	95	1	5	4	0	2	1	6	5
	20%	100	92	99	100	97	99	92	99	0	8	1	0	3	1	8	1
	30%	99	96	99	100	98	99	95	99	1	4	1	0	2	1	5	1
SH: Sinkhole, **BH:** Blackhole; **GH:** Grayhole; **DA:** DIF Flooding; **IR:** Increase Rank; **WH:** Wormhole; **DS:** DIO Suppression; **R:** Replay; **N:** Total number of nodes; **M:** Number of Malicious nodes;																	

TABLE VI
EVALUATION RESULTS, TRUE NEGATIVE RATE AND FALSE POSITIVE RATE

N	M	TNR								FPR							
		SH	***BH***	***GH***	***DA***	***IR***	***WH***	***DS***	***R***	***SH***	***BH***	***GH***	***DA***	***IR***	***WH***	***DS***	***R***
16	10%	99	99	85	100	99	95	99	93	1	1	15	0	1	5	1	7
	20%	100	96	99	100	100	100	98	100	0	4	1	0	0	0	2	0
	30%	96	99	100	100	100	99	100	100	4	1	0	0	0	1	0	0
32	10%	91	94	85	100	99	100	90	99	9	6	15	0	1	0	10	1
	20%	100	100	99	100	99	100	100	100	0	0	1	0	1	0	0	0
	30%	100	100	99	100	99	97	100	99	0	0	1	0	1	3	0	1
64	10%	88	88	97	100	99	99	88	100	12	12	3	0	1	1	12	0
	20%	98	100	98	100	97	100	100	98	2	0	2	0	3	0	0	2
	30%	98	95	96	100	98	100	100	99	2	5	4	0	2	0	0	1
128	10%	99	100	99	100	99	98	99	100	1	0	1	0	1	2	1	0
	20%	100	100	100	100	98	95	99	98	0	0	0	0	2	5	1	2
	30%	100	100	100	100	99	97	99	99	0	0	0	0	1	3	1	1
SH: Sinkhole, **BH:** Blackhole; **GH:** Grayhole; **DA:** DIF Flooding; **IR:** Increase Rank; **WH:** Wormhole; **DS:** DIO Suppression; **R:** Replay; **N:** Total number of nodes; **M:** Number of Malicious nodes;																	

Exp4) are conducted over different network configurations. In this regard, in Exp1 we evaluate the performance of the proposed scheme using different homogeneous algorithms. Exp 2 evaluates the performance of the proposed RL-IDS using various heterogeneous ML detectors for detecting RPL attacks. Different scaled LLNs have been simulated with $10\% \sim 30\%$ of malicious nodes. Exp 3 aims to evaluate the performance of RL-IDS using heterogenous detectors (hybrid detection strategy) in detecting unknown intrusions. Finally, in Exp 4, the performance of the proposed RL-IDS is evaluated against different types of RPL attacks using heterogeneous detectors, while 20% of nodes, including half of the malicious nodes, were mobile and in movement. All results are obtained from ten executions of each experiment.

This study evaluates the performance of RL-based IDS in terms of True Positive Rate (TPR), False Negative Rate (FNR), True Negative Rate (TNR), False Positive Rate (FPR), Accuracy (Acc), Precision (Pre), and F1 measure. The performance results are presented in Section VI-A. Here we use similar evaluation metrics as described in [1].

A. Experimental Setup

a) RL-IDS with homogenous detectors: In the first experiment, we aim to evaluate homogenous ML algorithms' performance in detecting RPL attacks to discover the best combination of ML-detectors for hybrid heterogeneous RL-IDS. The parameters of each ML algorithm are configured to produce lightweight detectors with low complexity in the system. Each detector uses the chi-square feature selector to obtain four features. Since each training batch includes a different proportion of each RPL attacks and normal traffic, the chi-square nominates a different set of features for each ML detector. This paper evaluates RL-based (DQN [25] and DDQN [24]) homogenous DT, KNN, K-means, SVM, and Logistic Regression (LR). The performances of different homogeneous ML algorithms using DQN and DDQN over ten runs are depicted in Fig. 5. In each run we consider 10% of nodes as IDS detectors. The performance of the proposed RL-IDS is the result of ten runs.

b) RL-IDS with heterogeneous detectors: Since each IDS detection strategy has unique strengths and weakness [1], [20], this paper develops RL-based IDS with hybrid heterogenous ML detectors to incorporate the strengths of signature-based

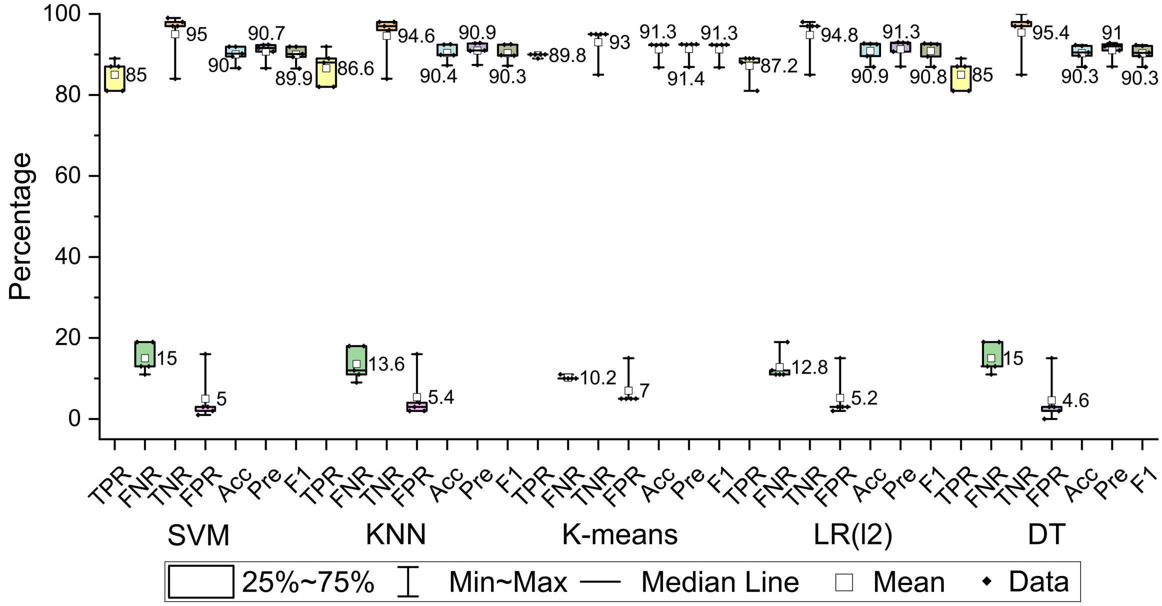


Fig. 6. Evaluation results of Homogeneous ML detectors over all RPL attacks.

and anomaly-based IDSs. The combination of SVM, One-class SVM, DT, K-means, KNN, and LR has developed to identify RPL attacks. The heterogeneous hybrid ML can provide optimum performance when we use an RL algorithm (DQN) for action-value selection, Fig. 5. To measure the performance of the proposed scheme against LLN's with different proportions of malicious nodes, we evaluate the performance of heterogeneous RL-based IDS against LLN's with different configurations, Table III, Table V and Table VI show the results of Exp 2.

c) Unknown Attack Detection: Table VII indicates how our proposed IDS approach detects RPL attacks that were not present in the training dataset. We select each attack type in turn, train our system on the remaining 7 attack types, and then evaluate how well the trained system detects the omitted attack type (i.e. the evaluation set comprises only that attack type). To the best of our knowledge, extant research does not address this issue [1].

TABLE VII
UNKNOWN ATTACK DETECTION

Unknown Attack	Performance Metrics					
	Acc	*Pre*	*TPR*	*FNR*	*TNR*	*FPR*
SH	87.3	87.7	92	8	82	18
BH	88.8	89	85	15	93	7
GH	95.8	95.9	98	2	94	6
IR	94.8	95.2	90	10	100	0
DA	100	100	100	0	100	0
WH	98.7	98.8	97	3	100	0
DS	94.9	95	92	8	97	3
RA	87.96	88.88	96	4	80	20

***Acc**: Accuracy; **Pre**: Precision.

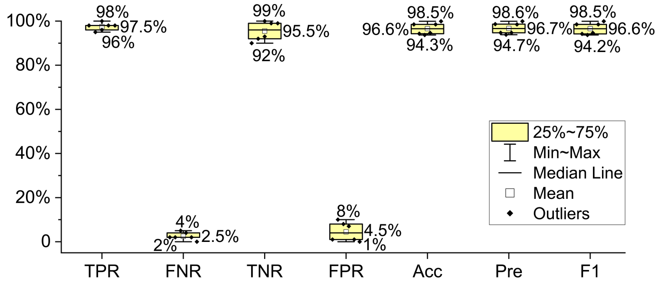


Fig. 7. Performance of hetrogenous RL-IDS in mobile scenarios.

d) LLN with mobile nodes: Only a few studies in the literature [1], [20] consider mobility among LLN nodes while mitigating some RPL attacks (SH, GH, DA, Sybil and Clone Id). To the best of our knowledge, there is no research that considers malicious mobile nodes on 6LoWPAN. In this paper we take an initial step to shed light on the rationale underlying this prominent issue. In this regard, we measure the performance of the proposed RL-based IDS with heterogeneous detectors against different RPL attack scenarios (SH, BH, GH, DA, DS, IR, WH, and RA) with 20% of nodes, and half of the malicious nodes, being mobile. Fig. 7 shows the performance of the proposed scheme.

B. Analysing results

Both the DQN and DDQN converge to optimal policies in the proposed scheme; however, DQN converges faster than DDQN with lower bias and variance, as shown in Fig. 5. The proposed scheme provides an adaptive, robust intrusion detection solution (DF1) against RPL attacks. The adaptivity and robustness of the deep reinforcement learning not only helps the IDS to become flexible against various types of known intrusions but also makes them effective in detecting unknown intrusions, as shown in Table VII (DF6). From the evaluation results (shown in Fig. 6 and Tables V and VI), we can argue that the proposed RL-IDS is effective against different RPL attacks for the networks with different configurations. Fig. 7 shows that heterogeneous RL-IDS is effective in detecting malicious nodes in mobile scenarios (DF2). Although all homogeneous detectors VI-A0a converged to the optimal policy after 20 to 40 episodes, heterogeneous detectors using RL-based IDS converge faster with better performance in the detection of known and unknown intrusions. This is because heterogeneous detectors use a combination of signature-based and anomaly-based ML detectors to develop hybrid RL-IDS. Both Table VI and Table V show that the proposed hybrid RL-IDS can provide an LLN with security against different internal (SH, BH, GH, IR, DA, WH, DS, and RA) and external (DA and WH) intrusions (DF3-4). Nevertheless, to ensure low overhead over LLNs (DF5) the proposed scheme uses the passive decentralised monitoring with ita RL-based IDS.

VII. Conclusion

We have presented a new RL-based IDS that employs hybrid heterogenous lightweight ML detectors to passively monitor 6LoWPAN traffic. Our approach has exhibited comprehensive feature engineering and has been shown to detect a much greater range of RPL attacks than extant research, including several previously unaddressed attacks. The work also addresses for the first time combinations of attacks. Also, as far as we are aware, evaluation against previously unseen RPL attacks has never been demonstrated in the literature.

References

[1] A. M. Pasikhani, J. A. Clark, P. Gope, and A. Alshahrani, "Intrusion detection systems in rpl-based 6lowpan: A systematic literature review," *IEEE Sensors Journal*, 2021.

[2] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in rpl-based internet of things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016.

[3] P. Kaliyar, W. B. Jaballah, M. Conti, and C. Lal, "Lidl: Localization with early detection of sybil and wormhole attacks in iot networks," *Computers & Security*, vol. 94, p. 101849, 2020.

[4] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "Using the rpl protocol for supporting passive monitoring in the internet of things," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 366–374.

[5] P. P. Ioulianou and V. G. Vassilakis, "Denial-of-service attacks and countermeasures in the rpl-based internet of things," in *Computer Security*. Springer, 2019, pp. 374–390.

[6] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in internet of things," *International Journal of Computer Applications*, vol. 121, no. 9, 2015.

[7] S. Raza, L. Wallgren, and T. Voigt, "Svelte: Real-time intrusion detection in the internet of things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[8] J. Foley, N. Moradpoor, and H. Ochen, "Employing a machine learning approach to detect combined internet of things attacks against two objective functions using a novel dataset," *Security and Communication Networks*, vol. 2020, 2020.

[9] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, and I. Ahmedy, "Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol," *IEEE Access*, vol. 6, pp. 16 623–16 638, 2018.

[10] P. Shukla, "Ml-ids: A machine learning approach to detect wormhole attacks in internet of things," in *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, 2017, pp. 234–240.

[11] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach," *Computer Communications*, vol. 98, pp. 52–71, 2017.

[12] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications*, vol. 141, p. 112963, 2020.

[13] S. Otoum, B. Kantarci, and H. Mouftah, "Empowering reinforcement learning on big sensed data for intrusion detection," in *Icc 2019-2019 IEEE international conference on communications (ICC)*. IEEE, 2019, pp. 1–7.

[14] Y.-F. Hsu and M. Matsuoka, "A deep reinforcement learning approach for anomaly network intrusion detection system," in *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*. IEEE, 2020, pp. 1–6.

[15] Z. S. Stefanova and K. M. Ramachandran, "Off-policy q-learning technique for intrusion response in network security," *World Academy of Science, Engineering and Technology, International Science Index*, vol. 136, pp. 262–268, 2018.

[16] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Computer Networks*, vol. 159, pp. 96–109, 2019.

[17] K. Sethi, E. S. Rupesh, R. Kumar, P. Bera, and Y. V. Madhav, "A context-aware robust intrusion detection system: a reinforcement learning-based approach," *International Journal of Information Security*, vol. 19, no. 6, pp. 657–678, 2020.

[18] O. Gaddour and A. Koubâa, "Rpl in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[19] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[20] G. Simoglou, G. Violettas, S. Petridou, and L. Mamatas, "Intrusion detection systems for rpl security: A comparative analysis," *Computers & Security*, p. 102219, 2021.

[21] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 science and information conference*. IEEE, 2014, pp. 372–378.

[22] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles," in *International Workshop on Data Mining for Biomedical Applications*. Springer, 2006, pp. 106–115.

[23] S. Adam, L. Busoniu, and R. Babuska, "Experience replay for real-time reinforcement learning control," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 201–212, 2011.

[24] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[25] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint arXiv:1708.05866*, 2017.

[26] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," *arXiv preprint arXiv:1602.04621*, 2016.

[27] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *computers & security*, vol. 30, no. 6-7, pp. 353–375, 2011.

[28] A. Mayzaud, R. Badonnel, and I. Chrisment, "A distributed monitoring strategy for detecting version number attacks in rpl-based networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 472–486, 2017.

[29] A. Mitrokotsa and A. Karygiannis, "Intrusion detection techniques in sensor networks," *Wireless Sensor Network Security*, vol. 1, no. 1, pp. 251–272, 2008.