
Article:
Pasikhan, 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 (2023) Incremental hybrid intrusion detection for 6LoWPAN. Computers & Security, 135. 103447.

https://doi.org/10.1016/j.cose.2023.103447

Computers & Security 135 (2023) 103447

Contents lists available at ScienceDirect

Computers & Security

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

Incremental hybrid intrusion detection for 6LoWPAN

Aryan Mohammadi Pasikhan *, John A. Clark, Prosanta Gope

Department of Computer Science, The University of Sheffield, Sheffield, UK

ARTICLE INFO

Keywords:
6LoWPAN
RPL
Intrusion Detection System (IDS)
Increase rank attack
DIO suppression attack

ABSTRACT

IPv6 over Low-powered Wireless Personal Area Networks (6LoWPAN) has grown in importance in recent years, with the Routing Protocol for Low Power and Lossy Networks (RPL) emerging as a major enabler. However, RPL can be subject to attack, with severe consequences. Most proposed IDSs have been limited to specific RPL attacks and typically assume a stationary environment. In this article, we propose the *first* adaptive hybrid IDS to efficiently detect and identify a wide range of RPL attacks (including DIO Suppression, Increase Rank, and Worst Parent attacks, which have been overlooked in the literature) in evolving data environments. We apply our framework to networks under various levels of node mobility and maliciousness. We experiment with several incremental machine learning (ML) approaches and various 'concept-drift detection' mechanisms (e.g. ADWIN, DDM, and EDDM) to determine the best underlying settings for the proposed scheme.

1. Introduction

Internet of Things (IoT) networks are generally Low-Power and Lossy Networks (LLNs) consisting of heterogeneous devices with limited power, memory, and processing resources. LLNs have been deployed in various sectors such as agriculture, control, the built environment and rural environment monitoring (Pasikhani et al., 2021b). For efficient routing in LLNs, the Internet Engineering Task Force (IETF) introduced the Routing Protocol for Low-Power and Lossy Networks (RPL) (Alexander et al., 2012). Global connectivity, resource constraints and RPL vulnerabilities expose 6LoWPAN to various routing threats, internally (within the 6LoWPAN) and externally (through the Internet). Existing routing attacks (e.g. Blackhole, Grayhole, Wormhole, and DODAG Informational Solicitation (DIS) flooding attacks) (Pasikhani et al., 2021b) cause the RPL to generate suboptimal routing topologies, isolate legitimate nodes, and cause significant overheads over the target network and nodes.

To deal with the security threats in RPL, a variety of Intrusion Detection System (IDS) proposals have been introduced in the literature. A network-based IDS can identify threats by analysing sniffed packets. IDSs can be signature-based, anomaly-based, specification-based, or hybrid (Pasikhani et al., 2021b). Signature-based IDSs use known signatures of attacks to identify intrusions. They can classify known intrusions accurately but require huge storage space to maintain the database of reference signatures, which must be updated continually. They cannot reliably detect hitherto unseen intrusions. Anomaly detectors build a profile of normal behaviour and detect significant deviations from that normal profile. Although anomaly-based IDS requires less storage space to identify abnormal instances (Raza et al., 2013), it is prone to generate many false-positive (FP) classifications (i.e., identifying legitimate activity as anomalous) (Hassan et al., 2023; Maheswaran et al., 2023; Darabkh et al., 2022). Specification-based approaches typically detect deviational behaviour from a formalised specification, e.g. that provided by a protocol description. The hybrid detection strategy combines existing detection strategies to incorporate their strengths and minimise their downsides. Various approaches to intrusion detection are found in the literature, e.g. statistical, rule-based, and machine learning-based. There are three major categories of ML-based IDS (ML-IDS): supervised (having access to labelled normal and malicious data instances), unsupervised (without access to any labelled data), or semi-supervised (where not all data is labelled, or else access is restricted to normal instances (Bhuyan et al., 2012)).

The 6LoWPAN has a streaming data environment. An IDS does not have access to the entire data stream and cannot afford to store all incoming data instances. Existing IDSs proposed for 6LoWPAN work only in stationary environments where the number of nodes in each scenario does not change. However, 6LoWPAN has an evolving data environment where node movement, inaccessibility, changes in running applications, and unforeseen attacks alter the data stream distribution. 6LoWPAN nodes cannot store a large volume of data. Moreover, in non-stationary evolving environments, the data distribution evolves unpredictably and so the system needs to update its model incrementally

* Corresponding author.
E-mail address: aryan.pasikhani@sheffield.ac.uk (A.M. Pasikhan).

https://doi.org/10.1016/j.cose.2023.103447
Received 3 February 2023; Received in revised form 28 July 2023; Accepted 20 August 2023
Available online 24 August 2023

or retrain it using recently observed batches of data. To address the aforesaid issues, "*concept drift*" detection approaches have been introduced in different network paradigms to enable adaptivity of the IDS (Gama et al., 2014). A "*concept*" can be defined as a joint distribution $P(X|Y)$, where X denotes a vector of attribute values (features) and Y is the target value (label) (Webb et al., 2016). Concept drift is a shift in the data distribution $P(X)$, where $P_t(X,Y) \neq P_{t+1}(X,Y)$. Thus, over time, the likelihood that observed data indicates normal system operation may change, e.g. if new malware has been crafted, or has otherwise adapted, to 'look like' benign software. The rate of concept drift is unknown to the system and can be abrupt, incremental, gradual or recurring (Gama et al., 2014). Concept-drift Detection (CD) methods can enable an IDS to adapt to unforeseen intrusions and identify shifts in the network data stream (Bhuyan et al., 2012). Additionally, CD approaches use storage and memory resources efficiently and facilitate fast classification.

Developing an adaptive IDS capable of accurately classifying the 6LoWPAN evolving data stream is a challenging task. The classifier needs to update itself with each change (shift) in the environment to continue to detect novel attacks. Re-training a classifier using the entire training data is computationally expensive and generally infeasible. This article proposes the use of streaming data mining techniques and drift detection to provide a novel adaptive form of hybrid ensemble capable of enhancing system performance. The proposed scheme can identify various routing attacks. Internal attacks (sourced inside 6LoWPAN) include sinkhole, blackhole, and grayhole attacks. External (sourced over the Internet) attacks include wormhole and DIS flooding attacks.

Different ensembling techniques have been adopted and compared in this article. A passive decentralised monitoring technique (where anomaly-based IDS agents passively monitor network communications and send abnormal/suspicious observations to the central IDS for further analysis) is used to collect and monitor LLN traffic from different locations and avoid additional computational overheads over legitimate nodes for intrusion detection purposes.

1.1. Desirable properties

Our proposed IDS approach aims to achieve the following Desirable Properties (DPs).

- **DP1:** the IDS should be able to identify routing attacks in an evolving data stream environment by updating its detection model when drift is detected.
- **DP2:** the IDS should not need excessive memory and computational resources whilst being able to identify routing attacks precisely.
- **DP3:** the IDS should work over 6LoWPAN networks incorporating mobile nodes.
- **DP4:** the IDS should be able to detect a wide range of RPL attacks.
- **DP5:** the IDS should be able to detect both known and previously unseen intrusions.

1.2. Motivation and contribution

The RPL is vulnerable to various routing threats (e.g. Sinkhole, Blackhole, and Wormhole (Mayzaud et al., 2016a; Pasikhani et al., 2021b)). Further more, the 6LoWPAN data environment evolves on an unpredictable basis. Different IDSs have been proposed in the literature to detect existing RPL attacks in 6LoWPAN (as discussed in Section 2). However, none of the existing IDS satisfies all the desirable properties (as mentioned in Section 1.1). In 6LoWPAN, an IDS observes a considerable (unbounded) volume of data as a continuous flow (Darabkh et al., 2022; Hassan et al., 2023); hence, it cannot explicitly store all observations to identify anomalous activities. To maintain detection performance, it is expected that the IDS modify its detection model on a regular basis and incrementally adapt to unforeseen data distributions (Maheswaran et al., 2023). This article proposes and evaluates an adaptive heterogeneous ensemble hybrid IDS framework to detect various types of RPL attacks in 6LoWPAN. The hybrid detection strategy helps the proposed framework to balance the computational cost of the anomaly-based intrusion detection and the storage cost of the signature-based intrusion detection over legitimate nodes. Besides, various incremental ML algorithms and ensemble techniques are evaluated to determine the most suitable combinations for the proposed system. The major contributions of this article are:

- The *first* adaptive hybrid IDS to detect internal and external RPL attacks.
- An efficient concept-drift-based ML-IDS, maintaining effectiveness in the face of environmental change.
- An effective approach to identifying a wide range of RPL attacks, including less researched ones, including Sinkhole (SH), Blackhole (BH), Greyhole (GH), DIS Flooding (DA), increase Rank (IR), Wormhole (WH), DIO Suppression (DS), Worst Parent (WP), Version Number (VN), and Neighbour Attack (NA).
- An IDS which is resilient against known and previously unseen RPL intrusions.
- A comprehensive and publicly available dataset for ML-based IDSs containing an extensive range of RPL attacks.

1.3. Organisation

The rest of the article is organised as follows. In Section 2, we review the related works and declare our contributions against each of them. In Section 3, we present our proposed scheme. In Section 4, we describe our implementation and evaluation details. Section 5 concludes the paper.

2. Related work

A broad range of routing vulnerabilities in 6LoWPAN and the lack of effective built-in security mechanisms in RPL (Pasikhani et al., 2021b) have encouraged researchers to develop IDSs for detecting RPL attacks. Various monitoring and detection strategies (Pasikhani et al., 2021b) have been considered. These Kaliyar et al. (2020); Pongle and Chavan (2015); Mayzaud et al. (2016b); Shafique et al. (2018) typically use a specification-based IDS to detect Sinkhole (SH), Wormhole (WH) and DIS flooding (DA) attacks. 54% of existing IDSs employed a specification-based detection strategy for detecting routing attacks in 6LoWPAN (Pasikhani et al., 2021b). Specification-based IDSs employ a set of static rules for identifying intrusions; they cannot update their rules automatically. Only 21% of reported works have considered a hybrid detection strategy (Pasikhani et al., 2021b) but none considers mobility of nodes.

The shortcomings of the statistical and rule-based detection approach (Pasikhani et al., 2021b) have encouraged researchers to apply machine learning (ML) algorithms to enhance the performance of IDS in 6LoWPAN. Among existing hybrid IDSs, only a few Shukla (2017); Foley et al. (2020); Bostani and Sheikhan (2017) are ML-based. Moreover, they Foley et al. (2020); Shukla (2017); Napiah et al. (2018); Bostani and Sheikhan (2017) use offline ML approaches, where the intrusion detection model is constructed using a stationary batch of training data. The batch-trained ML-IDS degrades as the data stream environment evolves (Bhuyan et al., 2012). Nevertheless, legitimate 6LoWPAN nodes often have limited memory and cannot store extensive records of malicious activities. This inevitably means that less critical records should be replaced with vital ones over time. To the best of our knowledge, no existing IDS for 6LoWPAN does this.

Various proposed monitoring techniques observe inter-node communication in the 6LoWPAN (Pasikhani et al., 2021b) (e.g. centralised and decentralised active or passive monitoring approaches). They Kaliyar et

Table 1
Related works.

Scheme	Method	Attacks Considered	Desirable Properties				
			DP1	*DP2*	*DP3*	*DP4*	*DP5*
Raza et al. (2013)	Active decentralised IDS	SH and GH (using Cooja simulator)	×	×	×	×	✓
Kaliyar et al. (2020)	Specification-based IDS	WH and Sybil (using Cooja simulator)	×	×	×	×	×
Shafique et al. (2018)	Specification-based active centralised IDS	SH (using Cooja simulator)	×	×	×	×	×
Shukla (2017)	Hybrid ML-based IDS	WH	×	✓	×	✓	×
Foley et al. (2020)	Ensemble Voting (MLP and RF)	SA, VN, SH, and BH	×	×	×	×	×
Bostani and Sheikhan (2017)	Unsupervised Optimum-Path Forest Clustering	SH, WH, and SF	×	×	×	×	×
Napiah et al. (2018)	Hybrid ML-IDS using passive monitoring technique	SH, WH, and DA (using Cooja simulator)	×	×	×	×	×
Shreenivas et al. (2017)	Active decentralised hybrid IDS	SH (using Cooja simulator)	×	×	×	×	×
Farzaneh et al. (2019)	Active decentralised anomaly-based IDS	DA and NA	×	×	×	×	✓
Kasinathan et al. (2013)	Passive decentralised signature-based IDS	DA (using Cooja simulator)	×	×	×	×	×
Le et al. (2016)	Active decentralised specification-based	WP, DA, SH, and DF	×	×	×	✓	×
Kareem and Tayeb (2021)	Online adaptive RF + concept drift	KDDCup99 (application layer attacks)	×	×	×	×	×
Martindale et al. (2020)	Online RF (Hoeffding Trees)	KDDCup99 (application layer attacks)	×	×	×	×	×
Örs and Levi (2023)	XGBoost and Autoencoder	DA, BH, GH, SH, and Version Number	×	×	×	✓	✓
Przybocki and Vassilakis (2023)	An Analysis into Physical and Virtual Power Draw Characteristics of Embedded LLN nodes	DA	×	×	×	×	×
Ioulianou et al. (2022)	three supervised ML algorithms	DA and BH	×	×	×	×	×
Manne and Sreekanth (2022)	counter-based (specification-based) detection algorithm	DA	×	×	×	×	×
Raghavendra et al. (2022)	Logistic Regression and KNN, RPLML-IDS	SH, GH and BH	×	×	×	×	×
Li et al. (2018)	Ensemble Weighted Voting, RF	KDDCup99 (application layer attacks)	×	D/N	D/N	D/N	D/N
Yuan et al. (2018)	Concept drift (HDDM) based ensemble incremental learning approach in IDS	KDDCup99 (application layer attacks)	✓	D/N	D/N	D/N	D/N
Singh et al. (2015)	Online Sequential-Extreme Learning Machine (OS-ELM)	NSL-KDD 2009 (application layer attacks)	✓	D/N	D/N	D/N	D/N
Our Scheme	One-Class SVM, incremental OzaBaggingADWIN using KNN, and HalfSpace-Trees	SH, BH, GH, DA, DS, IR, WH, WP, VN, and NA (Netsim v13)	✓	✓	✓	✓	✓

*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); ✓: Satisfy; ×: Not addressed; ✓*: Satisfy part of that desirable property; **SH:** Sinkhole, **BH:** Blackhole; **GH:** Grayhole; **DA:** DIS Flooding; **IR:** Increase Rank; **WH:** Wormhole; **DS:** DIO Suppression; **WP:** Worst Parent; **VN:** Version Number; **NA:** Neighbour Attack; **DP1**: Adaptive; **DP2**: Lightweight; **DP3**: Accurate in evolving data environment; **DP4**: Detect a wide range of RPL attacks; **DP5**: Detect known and unknown (a.k.a unseen) intrusions.

al. (2020); Pongle and Chavan (2015); Shafique et al. (2018); Shreenivas et al. (2017); Raza et al. (2013); Farzaneh et al. (2019); Foley et al. (2020); Shukla (2017); Bostani and Sheikhan (2017) employ an active monitoring technique to detect RPL attacks. According to Pasikhani et al. (2021b), ~77% of existing IDSs used an active monitoring technique, where legitimate nodes were required to participate in intrusion detection tasks with centralised or decentralised intrusion detectors. Active monitoring can provide more information about node configuration (e.g. geographical location, energy consumption, and CPU, RAM, and ROM usage) and result in more accurate detection of RPL attacks. However, it also causes additional computational overhead on the legitimate nodes. Consequently, some 6LoWPAN IDS papers employ passive centralised (Napiah et al., 2018; Viegas et al., 2018) and passive decentralised (Kasinathan et al., 2013; Mayzaud et al., 2016b, 2017; Pasikhani et al., 2021a) approaches. Passive monitoring does not cause any additional computation overhead for legitimate nodes Mayzaud et al. (2017). Nevertheless, it can provide IDS only with control packets that are multicasted or unicasted by monitoring nodes' neighbours.

According to Pasikhani et al. (2021b), existing IDS mainly focus on detecting sinkhole (21%), grayhole (14%), blackhole (10%) and DIS flooding (10%) attacks, while other RPL attacks are overlooked. No research in the literature examines the performance of IDS against external routing attacks (external DA and WH), and there is no research detecting DS (DIO Suppression) and IR (Increase Rank) attacks (Pasikhani et al., 2021b). Furthermore, only 13% of RPL IDS research has considered mobility (Pasikhani et al., 2021b). Table 1 shows the related works in the literature and the contributions that this article makes.

3. Proposed scheme

Our proposed scheme employs a passive decentralised monitoring approach (readers may refer to Mayzaud et al. (2017) for more details) using a cluster-based placement (Mitrokotsa and Karygiannis, 2008) strategy to analyse the data stream in 6LoWPAN. Anomaly-based detectors are spread over the 6LoWPAN to analyse their neighbours' control packets and report abnormalities to the Centralised IDS (CIDS) on the 6LoWPAN Border Router (6BR). The CIDS is an adaptive heterogeneous hybrid IDS that protects 6LoWPAN against internal and external intrusions. Fig. 1 illustrates the system architecture. The proposed scheme has three components: an anomaly-based network IDS (ANIDS) (Section 3.1), an incremental ensemble of signature-based IDSs (Section 3.2.1), and incremental ensembles of anomaly-based IDSs (Section 3.2.2) (described below). Algorithm 1 shows the proposed scheme.

3.1. Anomaly-based network IDS

Since the CIDS on the 6BR cannot observe network communications of distant nodes (since the 6BR has limited radio range and RPL may operate in storing mode (Pasikhani et al., 2021b)), the proposed scheme distributes Anomaly-based Network IDS (ANIDS) agents to passively monitor multicasted and unicasted control packets of their neighbouring nodes without requiring significant storage space. As shown in Scenario 1 (Section 4.3), a One-Class SVM (OCSVM) can provide excellent performance in detecting intrusions with negligible false-alarms and excellent recall value. The OCSVM is a novelty detection algorithm that develops a model of safe activities and classifies instances as an outlier (anomalous) if they deviate from its profile. The outcome of OCSVM is bipolar, $y_t = -1$ for $x_t \in$ outliers and $y_t = +1$ for $x_t \in$ inliers. In OCSVM, the classifier assumes that the given training dataset X contains only normal (safe) instances, $X = \{\overline{x}_1, \overline{x}_2, ..., \overline{x}_N\}$ $\overline{x}_i \in Normal$, and considers the origin of a kernel-based transformed representation as an outlier. OCSVM aims to discover a separating boundary (hyperplane) $\overline{w}.\phi(\overline{x}_i)$ that maximises the distance between normal instances ($\overline{x}$) and the origin $(0,0)$, $\overline{w}.\phi(\overline{x}_i) - \rho = 0$ (define the hyperplane) where $\overline{w}$ and $\phi(.)$ denote weight and SVM kernel (a function that projects data into a high dimensional space to increase the discriminatory capability of the classifier) respectively; ρ denotes the maximal margin (threshold), Eq. (1),

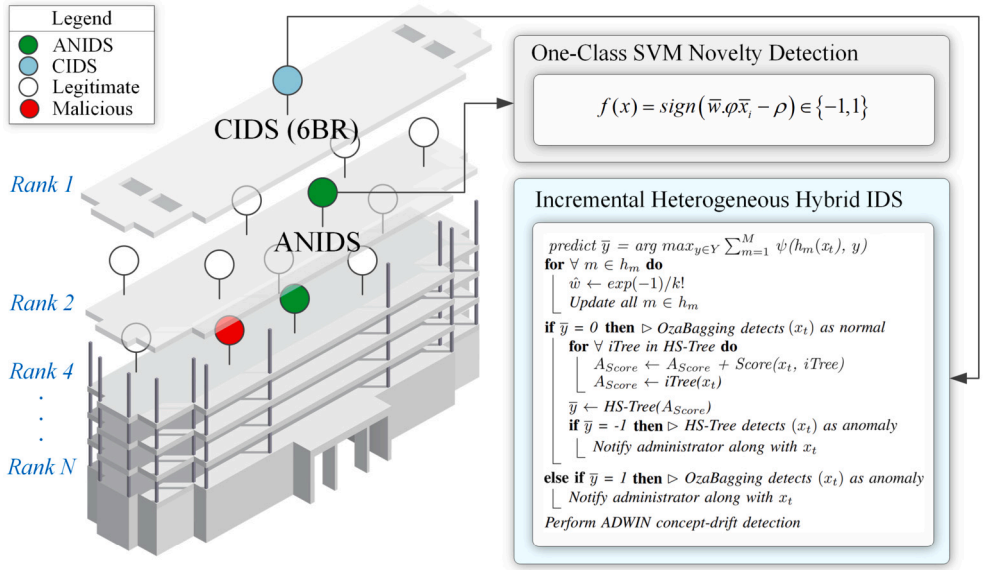


Fig. 1. System model. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

Algorithm 1: Proposed algorithm.

Initialisation
A stream of pair (x, y), *as* $(x_0, y_0), (x_1, y_1) \ldots (x_T, y_T)$, *arriving one-by-one over time.*
X *is an evolving data stream* $(X \rightarrow \infty)$, *where* x_t *is a set of features observed at time* t *(now).*
y *is the real class label and* $\overline{y}$ *is the classifier prediction, where* $Y = \{-1, 1\}$
C_A: $C_{OCSVM} \cup C_{HST}$ // *Anomaly Classifiers.*
C_{OCSVM}: *One-class SVM Classifiers* $\subseteq C_A$.
iTree: a HalfSpace-Tree.
ω: *Window Size.*
A_{Score}: *Anomaly Score.*
C_{HST}: *HalfSpace-Trees ensemble classifier* $\in C_A$.
M *is the number of models in the ensemble.*
h_m *is an adaptive OzaBagging ensemble model induced by learners* $m \in \{m_1 \ldots m_n\}$.
Count $\leftarrow 0$.
r: mass profile of a node in the reference window. //mass is used as a measure to rank anomalies.
l: mass of a node in the latest window.
k: Generate poisson $(\lambda = 1)$
ψ: *is the generalised Kronecker function:* $\psi(a, b)$ *is 1 if* $a == b$, *and 0 otherwise.*
for *all* (x) *in* X **do**
  $\delta \leftarrow$ *using Eq. (3)* c *classifies* (x_t), *where* $c \in C_{OCSVM}$
  if δ == -1 *(*c *has classified* (x_t) *as malicious)* **then**
    predict $\overline{y} = arg\ max_{y \in Y} \sum_{m=1}^{M} \psi(h_m(x_t), y)$
    for *all* $m \in h_m$ **do**
      $\hat{w} \leftarrow exp(-1)/k!$
      Update m *with* (x_t, y_t) *and weight* $\hat{w}$
    if $\overline{y}$ == -1 *(*h_m *detect* (x_t) *as normal)* **then**
      $A_{Score} \leftarrow 0$
      for *all iTree in* C_{HST} **do**
        $A_{Score} \leftarrow A_{Score} + Score(x_t, iTree)$ // *accumulate scores*
        UpdateMass$(x_t, iTree.root, false)$ // *update mass l in iTree*
      Report A_{Score} *as the anomaly score for* x_t
      Count++
      if *Count* == ω **then**
        Update model : Node.r $\leftarrow$ Node.l *for every node with non-zero mass r or l*
        Reset Node.l $\leftarrow 0$ *for every node with non-zero mass l*
        Count $\leftarrow 0$
    if *ADWIN detects change in error of one of the models* (h_m) **then**
      Replace the model with highest error with a new model
  Output: *Notify administrator if* x_t *is anomalous*

with N instances $\overline{x}_{i<1,N>}$. According to Maglaras and Jiang (2014), the OCSVM can be solved efficiently using the quadratic Eq. (2). The ν (Nu) is upper bounded by the fraction of outliers and lower bounded by the fraction of support vectors. The ν intends to fine-tune the trade-off between over-fitting and generalisation. The conjoint usage of ν and the slack variable ξ $(\xi \geq 0)$ enables the system to handle a dataset that contains a small fraction of outliers. In other words, ν is the probability of finding an outlier in X, where $outliers \subseteq X$. The γ (gamma) determines how much influence a single training example has. The larger the value of γ, the more closely other examples will be affected. Since it is expected that ANIDS generate some degree of false-positive alarms (wrongly classifying safe instances as intrusions), the instances that are classified as anomalies will be further analysed by the CIDS.

$$\overline{w}.\phi(\overline{x}_i) \geq \rho - \xi_i \ \forall \overline{x}_i \in X \ and \ \xi_i \geq 0, \forall\ i \in \{1, \ldots, N\} \tag{1}$$

$$Min_{\overline{w},\overline{\xi},\rho} = \left[\frac{1}{2}\|\overline{w}\|^2 + \left(\frac{1}{\nu\gamma}\sum_{i=1}^{n}\xi_i\right) - \rho\right] \tag{2}$$

$$y_i = sign(\overline{w}.\phi(\overline{x}_i) - \rho) \tag{3}$$

where the y_i in Eq. (3) is an inliner (+1) if $\overline{w}.\phi(\overline{x}_i) - \rho \geq 0$ and an outlier (-1) otherwise.

In Equations (2) and (3):

- w is the normal vector to the hyperplane.
- $\phi(x_i)$ is the mapping of an input vector x_i into the feature space.
- ρ is the bias term of the hyperplane.
- ξ_i are slack variables that measure the amount of mis-classification.
- $\nu \in (0, 1]$ is an upper bound on the fraction of outliers and a lower bound on the number of support vectors.

The goal of the OCSVM is to maximize the distance ϕ while minimizing $||w||^2$, essentially finding the maximum-margin hyperplane that separates the data from the origin in the feature space.

3.2. Central IDS

The CIDS contains an incremental heterogeneous hybrid IDS and is responsible for analysing internal and external data streams. It analyses the external network traffic coming to the 6LoWPAN and internal network communications among LLN nodes. Moreover, an observation that is classified as anomalous by any ANIDS will be reported to CIDS for more in-depth analysis. The CIDS analyses the anomalous observations through its incremental ensemble of signature-based IDS (in Section 3.2.1) and an incremental ensemble of anomaly-based IDS (in Section 3.2.2) to make more accurate classifications. Algorithm 1 shows the hybrid proposed scheme. The adaptivity through Concept-drift Detection (CD) enables the framework to enhance its intrusion

detection performance over time by adapting to unforeseen intrusions and changes in data distributions.

3.2.1. Incremental ensemble of signature-based IDSs

Incremental ensemble classifiers provide better detection performance at the cost of more computation and memory usage (Gomes et al., 2017). An ensemble classifier $f(C_1(x_t), C_2(x_t)...C_n(x_t))$ is a set of classifiers (C_i) that make predictions over a given instance of feature set (x_t). The Ozabagging classifier builds an ensemble of classifiers such that $\forall\ c_i \in C$, c_i is trained over different bootstrap instances. Since it is challenging to draw samples with replacement in an online streaming environment, the Oza bagging classifier weights the observed instances using a Poisson[1] in bootstrap replica (Bifet and Gavalda, 2009). The OZABAGADWIN (Oza and Russell, 2001; Bifet et al., 2009) is the OzaBagging with ADWIN (adaptive windowing) concept-drift detection. The OZABAGADWIN implements several ADWIN drift detectors to monitor classifier error rates. On the detection of concept drift, OZABAGWIN replaces the worst classifier $c_i \in C$ with a new classifier, described as a "replace the loser" strategy (Bifet and Gavalda, 2009). The classification of the majority of individual classifiers that make up the ensemble is taken as the classification of the instance. Where the number of classifiers is odd, there is always a majority for one class. Where the ensemble has an even number of classifiers, then a tie is possible. In such a case, the instance is judged to be malicious (Oza and Russell, 2001; Bifet et al., 2009).

Mathematically, OzaBagging works by simulating the effect of bootstrap sampling in an online fashion. For each incoming instance from the data stream:

1. It calculates the Poisson(1) random number, which corresponds to the number of times the current instance appears in the bootstrap sample.
2. It feeds the current instance to each model in the ensemble as many times as this Poisson(1) random number states. In other words, if the Poisson(1) random number is zero, the instance is not used for training a specific model; if the Poisson(1) random number is one, the instance is used once for training, and so forth.
3. Each model updates its predictions based on the incoming instance.

The ensemble's prediction is then, as with traditional bagging, the majority vote of the predictions of the individual models. By leveraging the properties of the Poisson distribution, OzaBagging manages to simulate the effect of bagging in an online setting, allowing for effective learning from data streams.

3.2.2. Incremental ensemble of anomaly-based IDSs

Although adopting adaptivity (concept-drift detection) enables a signature-based IDS to learn unforeseen intrusions (discussed in Section 3.2.3), a signature-based IDS is prone to some degree of false-negative alarms for unknown intrusions. To enable the proposed framework to identify unknown intrusions, the HalfSpace-Trees (HS-Trees) algorithm (Tan et al., 2011) analyses observations that are classified as normal so far. In HS-Trees, each tree contains nodes that capture the number of data items (known as mass) within a subspace of streaming data. In this context, the mass is used to profile the degree of anomaly. The OzaBaggingADWIN and HS-Tree form an incremental hybrid IDS on the 6BR. HS-Trees algorithm starts with the selection of a random subset from the incoming data stream. This subset called the 'mass profile', is used to initialize the trees. Each tree is built by randomly dividing the space of the mass profile into two halves, each half-space (or hyperplane) represented by a node in the tree. The process is recursively repeated for each half-space, resulting in a binary tree structure. The depth of the tree is a hyperparameter and can be adjusted based on the complexity of the data. After initialization, each data point in the mass profile is passed down to each tree. When a data point reaches a node, the mass (the count of data points) of the corresponding half-space is incremented. After all data points in the mass profile have been passed through the trees, each node in the trees will have an associated mass representing the density of data points in its half-space. As new data points arrive from the stream, they are passed down to each tree in the forest.

The anomaly score for the data point is calculated based on the masses in the nodes it encounters. A lower mass indicates a less dense region of the feature space and hence a higher anomaly score. Mathematically, the anomaly score $s(x)$ for a data point x is defined as $s(x) = 2^{-E(h(x)/c(n))}$ where $h(x)$ is the path length of data point x from the root to the leaf in a tree, $E(h(x))$ is the average path length over all trees in the forest, n is the number of instances in the mass profile, and $c(n)$ is the average path length of an unsuccessful search in a Binary Search Tree. Over time, the trees adapt to changes in the underlying data distribution. This is achieved by decrementing the mass of the nodes a data point passes through (making the region less dense) and then incrementing the mass for the new data point.

3.2.3. Adaptivity

Adaptive learning updates the predictor model to respond to concept drift through the predictor operations. The 6LoWPAN traffic routing evolves as nodes move or become unavailable (e.g. their energy resource may deplete), which results in the reconstruction of the DODAG routing graph. Data forms a stream into the IDS with a distribution that varies over time. To reduce memory use, concept-drift-based IDS trains over a small number of training data at any point in time and does not load the entire dataset into memory (Bhuyan et al., 2012). The fundamental function of any concept drift detection approach is the mechanism to detect the drift occurrence timestamp. Accurate identification of the time that drifts happen plays a vital role in enhancing the system's adaptivity performance. Since the model never has full access to the entire data in a continuous environment, this article employs the adaptive sliding window (ADWIN) concept (Bifet and Gavalda, 2007) to perform concept drift detection. A window w is a snapshot of data; it gives more importance to the recently observed data and periodically discards the older data. ADWIN slides a window w on the prediction results as they become available in order to detect drifts. The method examines two sub-windows of sufficient length, i.e., w_0 of size n_0 and w_1 of size n_1 where $w_0 \bullet w_1 = w$. The symbol $\bullet$ represents the concatenation of two windows. A significant difference between the means of two sub-windows indicates a concept drift, i.e., when $|\hat{\mu}_{w_0} - \hat{\mu}_{w_1}| \geq \varepsilon$ where $\varepsilon = \sqrt{\frac{1}{2m} \ln \frac{4}{\delta'}}$, m represents the harmonic mean of n_0 and n_1, and $\delta' = \delta/n$. Here δ is the confidence level while n is the size of window w. Once a drift is detected, elements are removed from the tail of the window until no significant difference is observed.

4. Implementation and evaluation

In this article, we use the Netsim simulator to evaluate the performance of the proposed scheme against different RPL attacks. In this context, we consider different network configurations (e.g. a number of malicious and legitimate nodes and objective function), as described in Table 2. The simulated 6LoWPAN scenarios include 16 to 128 LLN nodes (excluding 6BR and external computers), with 10% to 30% of nodes assigned as malicious. In all scenarios, we consider 20% of the nodes, including half of the malicious nodes, to be mobile and randomly move around the terrain with a velocity of 5 m/s. Nodes distribute over terrain covering 250 $m^2 \sim 800\ m^2$ and are 25 ~ 45 m apart, *with 50 m transmission range*. Each scenario is simulated for ~360 minutes for performance benchmarking. This article uses the interleaved test-then-train

[1] As N (number of samples) $\rightarrow \infty$ the distribution of K (number of copies of each n) tends to a Poisson(1) distribution: $K \sim \frac{\exp(-1)}{k!}$ (Wang and Pineau, 2016).

Table 2
Simulation parameters.

Parameters	Values
Simulator	Tetcos Netsim V13
Number of nodes	16, 32, 64, 128
Number of Malicious nodes	~10%, ~20%, ~30%
Number of Workstations	4, 8
Transmission Range	50 m
Number of ML detectors	~10%
Number of Mobile nodes	~20%
Scenario Dimension (Terrain)	(250 × 250) to (850 × 850) s.meters
Traffic Rate	250 kbps
Simulation time	~ 21,600 seconds
Application Protocols	COAP, CBR
RPL mode	Storing mode
Mobility Modes	Random Walk, Group Walk
Path Loss Model	Log Distance, Exponent(n): 2
Distance between nodes	25 ~ 45 m
Objective Function (OF)	OF0, LQ
Receiver Sensitivity	-85 dBm

approach to evaluate the proposed scheme (Bifet et al., 2009). It is assumed that the packets in the streaming data D sequentially appear in the target network, where x_t is an unlabeled instance vector observed at time t, containing different attributes about the node configurations and the DODAG. The actual label y_t of instance x_t will be available to the system at different points in time (Bifet et al., 2009).

4.1. Data-set and feature construction

The simulations generate a dataset D, representing malicious and normal (safe) network communications. Each observation x in D denotes a set of n features $x = \{f_1, f_2 ... f_n\}$, where f_i contains specific information about the sender and receiver. The header of each RPL control packet (e.g. DIO, DIS, DAO) contains different information about the sender of the packet (Alexander et al., 2012; Barthel et al., 2012) that can facilitate the identification of anomalous network activities. Engineering a set of informative features is essential to develop an IDS to accurately classify all types of RPL attacks in the streaming data environment. Therefore, we perform feature engineering to facilitate the classification of data streams for IDS. The extracted features can enable the anomaly-based classifiers to correctly identify all the anomalies through training over normal instances and make signature-based classifiers to accurately classify each type of RPL attack. The raw instances of 6LoWPAN simulations contain a set of features that are not applicable for conducting intrusion detection tasks. For instance, features that represent node identities (e.g. IP address, MAC address, and node id) can inhibit scheme generalisation. Since this article employs a passive decentralised monitoring approach (Mayzaud et al., 2017), any feature that requires the internal configuration of legitimate nodes (e.g. power consumption, geographical location, CPU/RAM/ROM usages) are excluded. We simulated several pairs of networks $(\mathcal{A}, \mathcal{B})$ where $\mathcal{A}$ contains only the normal nodes and $\mathcal{B}$ contains both the normal and malicious nodes. Observing the statistical difference of control and application packets in $\mathcal{A}$ and $\mathcal{B}$ enable us to identify the adverse impact that each RPL attack has in the networks in $\mathcal{B}$. A simulated 6LoWPAN includes legitimate (safe) network communications (control and application packets) and malicious traffic. In each RPL attack scenario, malicious nodes cause adverse impacts inside the network by either generating malicious network traffic (e.g. DIS flooding, DIO suppression, and sinkhole attacks) or modifying legitimate network communication of their neighbouring nodes (blackhole and grayhole attacks). The abnormalities that each RPL attack causes inside 6LoWPAN constitute malicious observations.

We extract three types of features: basic, history-based, and connection-based features. Basic features contain general node information derived from ICMP v6 control packet headers (node rank, source and destination addresses, flags etc.). In contrast, the time-based features provide information about the number of times the current node sends or receives a specific application or control packet. Connection-based features carry salient information about the sender's routing configuration (RSSI, link quality etc.) and the number of collided control and application packets perceived by an IDS detector. Table 3 depicts the set of features engineered in this article. Here we apply the Mean Decrease in Impurity (MDI) importance metric to illustrate the importance of engineered features in identifying RPL attacks, as shown in Fig. 2. The connection- and history-based features play vital roles in detecting the routing attacks in 6LoWPAN.

Table 3
Engineered features.

	Feature	Description
Basic	pkt_type	Type of packet (DIO, DAO, DIS, App etc)
	pkt_status	Packet status (Collided, Successful)
	src_rank	Sender rank in DODAG
	adv_vn	Advertised version number
History-based	snd_dis_count	No. of DIS unicasted/multicasted by sender
	snd_dio_count	No. of DIO advertised by sender
	snd_dao_count	No. of DAO unicasted by sender
	snd_daoack_count	No. of DAO-Ack unicasted by sender
	snd_cpkt_count	No. control packet issued by sender
	rcvd_dis_count	No. of DIS rcvd by current node in the past
	rcvd_dio_count	No. of DIO rcvd by current node
	rcvd_dao_count	No. of DAO rcvd by current node
	rcvd_daoack_count	No. of DAO-Ack rcvd by receiver
	rcvd_cpkt_count	No. of control packets rcvd by receiver
	avg_intpkt_time	Average delay between pkts issued by snd
	rnk_alt_count	No. rank alteration by sender
	vn_alt_count	No. version number alteration by sender
	trans_app_count	No. of application trans by sender
	pkt_e2e_delay	Packet end-to-end delay
Connection-based	cpkt_loss	Control packet loss ratio
	pkt_loss	Application packet loss ratio
	avg_hopcount	Average No. of hopcount (global view)
	neighbour_count	No. of neighbouring node
	child_count	No. of children
	same_parent	Sender and the detector have same parent
	rx_sen	Average receiver sensitivity
	tx_pwr	Average transmission power
	rssi	Received signal strength indicator of sender
	cmp_snd_prt_lq	LQ of sender > LQ of parent
	prt_bst_lq	Current parent provide best link quality

4.2. Evaluating using emulators vs real test-bed

Emulators like the Tetcos NetSim offer an exceptional environment for evaluating IDS in 6LoWPAN, particularly when considering variable network sizes and diverse mobility patterns. This is because this emulator can seamlessly scale from small to large networks, effectively mimicking different network dynamics and node mobilities, something that real-world implementations find challenging to replicate due to time and logistical constraints. Furthermore, real-world testing requires substantial resources and requires rigorous ethical approvals, particularly when user data and potentially sensitive network operations are involved. These practical and ethical complexities can make the setup exceedingly time-consuming and costly.

Conversely, emulators bypass these hurdles, providing a cost-effective and efficient platform where IDS can be put to the test across multiple parameters without compromising ethical norms or spending excess time in setup. This flexibility allows researchers to thoroughly investigate IDS efficacy across numerous scenarios, thereby producing robust and comprehensive evaluations. In addition, using the emulation feature of Tetcos NetSim provides an avenue for hybrid simulation-real-world implementation approaches by enabling the attachment of actual IoT hardware to represent network nodes. This strategy combines the controlled environment of a simulator with the tangible characteristics

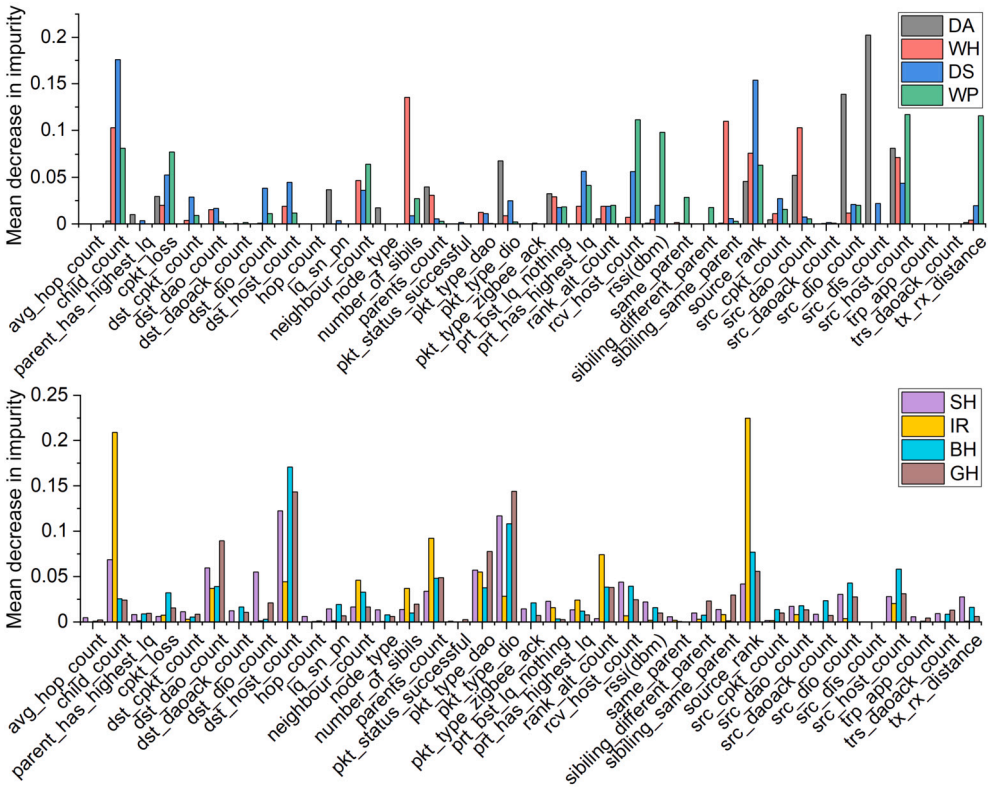


Fig. 2. Feature importance.

of real-world hardware, further enhancing the realism and applicability of the tests. Thus, it blends the best of both worlds, ensuring comprehensive IDS evaluation and validation that extends beyond purely theoretical or simulated conditions.

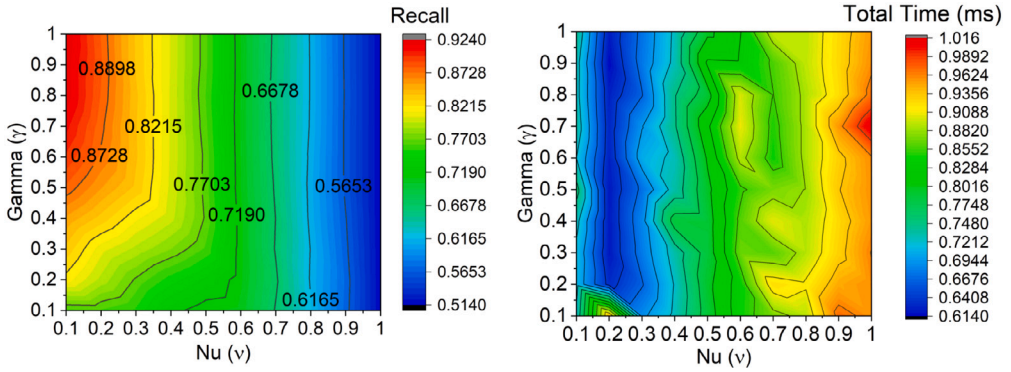
Nevertheless, it's important to acknowledge that despite the realism introduced by emulation, there are still distinct differences between this approach and a full-scale, real-world test-bed implementation. This includes the impact of hardware compatibility and performance issues, real-time network fluctuations, user behaviour, and a variety of potential external disruptions. These factors can significantly impact the performance of an Intrusion Detection System, and understanding this behaviour in a real environment provides invaluable insights that go beyond the limitations of any simulated scenario. Thus, while emulation serves as a powerful tool in the early stages of development and testing, a real-world test-bed implementation is essential to fully validate and optimise the IDS in realistic conditions.

4.3. Results and analysis

As discussed in Section 3.1, the novelty of anomaly detectors of the proposed scheme work by observing the control packets of their neighbours; if the current observation is identified as anomalous, it will be further analysed by the heterogeneous hybrid ensemble IDS on the 6BR. Below, different outlier detection, incremental ensembling, and concept drift detection algorithms are evaluated. We seek the best combination to gain the optimal F1, accuracy, recall, precision (Pasikhani et al., 2021b) and kappa (Gomes et al., 2017; Gama et al., 2014) with the least False Negative Rate (FNR) and False Positive Rate (FPR) (Pasikhani et al., 2021b). Below, we conduct *six* scenarios utilising the underlying features of the Netsim emulator to execute the proposed framework over several Raspberry Pi 4 (model B, 4GB RAM) micro-controllers to measure the execution time and the model power consumption using a UM25C digital multimeter. Table 2 depicts the network configurations that we implemented to conduct our simulations. In all of our simulations, $\sim 20\%$ of nodes are mobile and randomly move around the terrain with a velocity of 5 m/s.

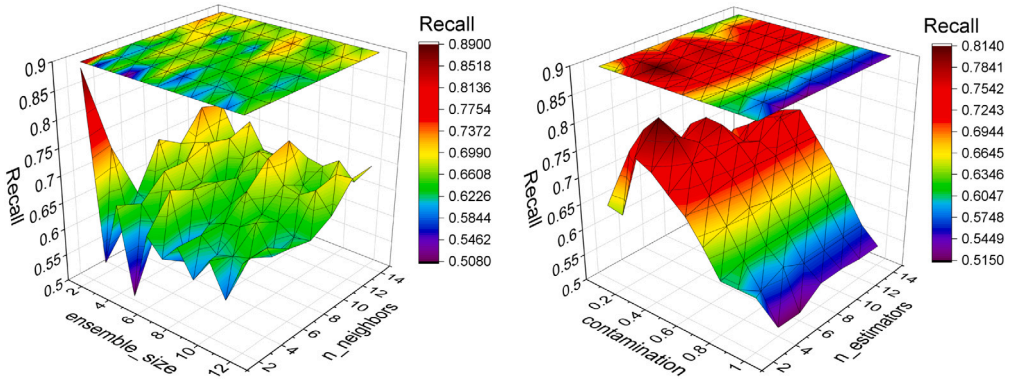
Scenario 1. The anomaly-based detector (also known as novelty detector) plays a crucial role in identifying outliers in the proposed scheme. Here we measure the performance of OCSVM in detecting RPL attacks. We have evaluated OCSVMs with different parameter values for Nu $\nu \in (0, 1]$ and Gamma $\gamma \in (0, 1]$ for finding the optimal configuration; Fig. 3(a) shows that the OCSVM with $\nu \in (0.01, 0.25)$ and $\gamma \in (0.6, 1]$ can maximise recall. However, since the aim of the ANIDS is to identify all the intrusions and maximise TPR, here we assign the OCSVM with $\nu = 0.2$ and $\gamma = 0.9$ to achieve 99.74% TPR with 89.39% recall (weighted average). Our simulations outcomes suggest that an OCSVM outperforms other existing anomaly detection algorithms, a majority-voting ensemble of Local Outlier Factor and Isolation Forest, as shown in Fig. 4.

Scenario 2. Scenario 1 showed that although the OCSVM algorithm can accurately identify outliers it also incurs 20.25% FPR. To address this issue, we conduct our second scenario to measure the performance of different incremental ensemble algorithms and rectify ANIDS misclassifications. Here, we have compared the performance of OzaBagging (Oza and Russell, 2001), LearnPPNSE (Elwell and Polikar, 2011), Online Boosting (Wang and Pineau, 2016), Online AdaC2 (Wang and Pineau, 2016), Accuracy Weighted Ensemble (Wang et al., 2003), and Online SMOTE Bagging (Wang and Pineau, 2016) algorithms in detecting RPL attacks. The outcome of our simulations (as shown in Fig. 5 and Fig. 6) shows that the combination of OzaBagging using KNNADWIN can provide the best possible outcome to identify known intrusions. In this context, as shown in Fig. 6, the OzaBaggingADWIN outperform all other incremental classifiers by ~5% in terms of F1, and ~10% in terms of accuracy and Kappa. However, OlineAdaC2 slightly (less than ~2%) outperform OzabaggingADWIN in terms of recall. OzaBagging using KNNADWIN with n_estimators (number of estimators) as 4 and n_neighbours (number of neighbours) as 6 receives 91.5% F1 and 7.8% FPR and with n_estimators as 8 and n_neighbours as 6 receives 92.2% F1 and 7.3% FPR, as depicted in Fig. 5.



(a) One-Class SVM Recall. (b) One-Class SVM Time Complexity.

Fig. 3. One-Class SVM (OCSVM).



(a) Local Outlier Factor Recall. (b) Isolation-Forest Recall.

Fig. 4. Performance of different outlier detection algorithms.

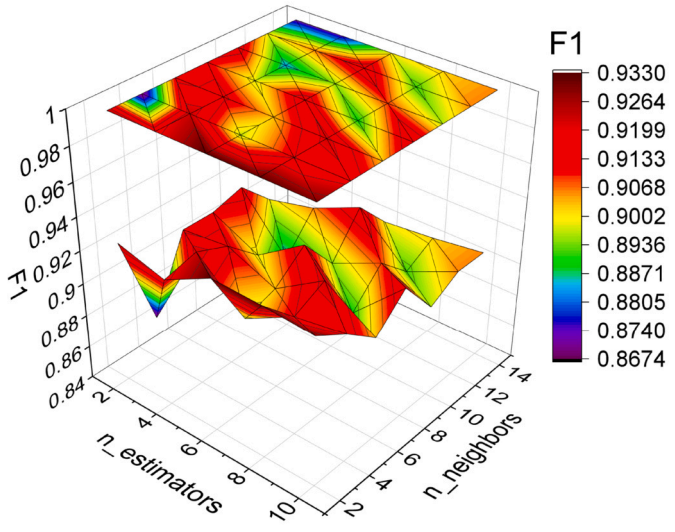


Fig. 5. OzaBagging ADWIN (KNN) F1.

Scenario 3. Above, we showed how an incremental ensemble approach could identify known intrusions efficiently. Our proposed hybrid IDS targets both known and unknown intrusions. Accordingly, we now investigate an incremental ensemble of anomaly-based classifiers which can rectify false-negative alarms of the signature-based IDS. False-negative alarms are very costly and indicate the IDS is failing in its primary task. In this scenario, we show how the inclusion of an incremental HalfSpace-Trees (HS-Trees) classifier can enhance the overall performance of the system. Fig. 6 shows that the HS-Trees algorithm forms a better hybrid IDS when it combines with the OzaBaggingADWIN compared to other incremental algorithms by around 6 to 10%. Fig. 6 gives the current and moving mean (also referred to as moving average) F1, recall, kappa, and accuracy of the incremental ML algorithms.

Scenario 4. Here, we investigate to what extent concept drift detection can provide system adaptivity. We evaluate different drift detection algorithms to select one that can ensure adaptivity in the system and also enhance the framework performance over time. We consider the following (seven) adaptive Windowing methods for concept drift detection: (ADWIN), Drift Detection Method (DDM), Early Drift Detection Method (EDDM), Kolmogorov-Smirnov Windowing (KSWIN), Page-Hinkley, Drift Detection Method based on Hoeffding's bounds (HDDM) with moving weighted average-test (HDDM-W) or moving average-test (HDDM-A) concept drift detection methods (Yuan et al., 2018; Gama et al., 2014). Results are presented in Fig. 7(a) and Fig. 7(b). From Fig. 7, we can see that ADWIN gives the best accuracy than of the concept-drift detection methods in the shortest time interval. Outcomes of Scenarios 1, 2, 3, and 4 show that our proposed scheme so far addresses *DP1* (adaptive and robust intrusion detection, which were discussed in Section 1.1) (Table 4).

Scenario 5. Here, we measure the time complexity of each component in the proposed framework. We consider 64 LLN nodes in 6LoWPAN, with 20% assumed malicious. Fig. 3(b) shows the results over 1500 network packets, where 50% of instances are assumed normal and the remaining 50% include each RPL attack type equally. We measure the time complexity for each ANIDS and CIDS separately. Fig. 3(b) shows the time complexity that the OCSVM with $\nu = 0.2$ and $\gamma = 0.8$ causes the least time complexity in the system. On the other hand, the adaptive heterogeneous hybrid IDS, developed in our Scenarios 2 and 3,

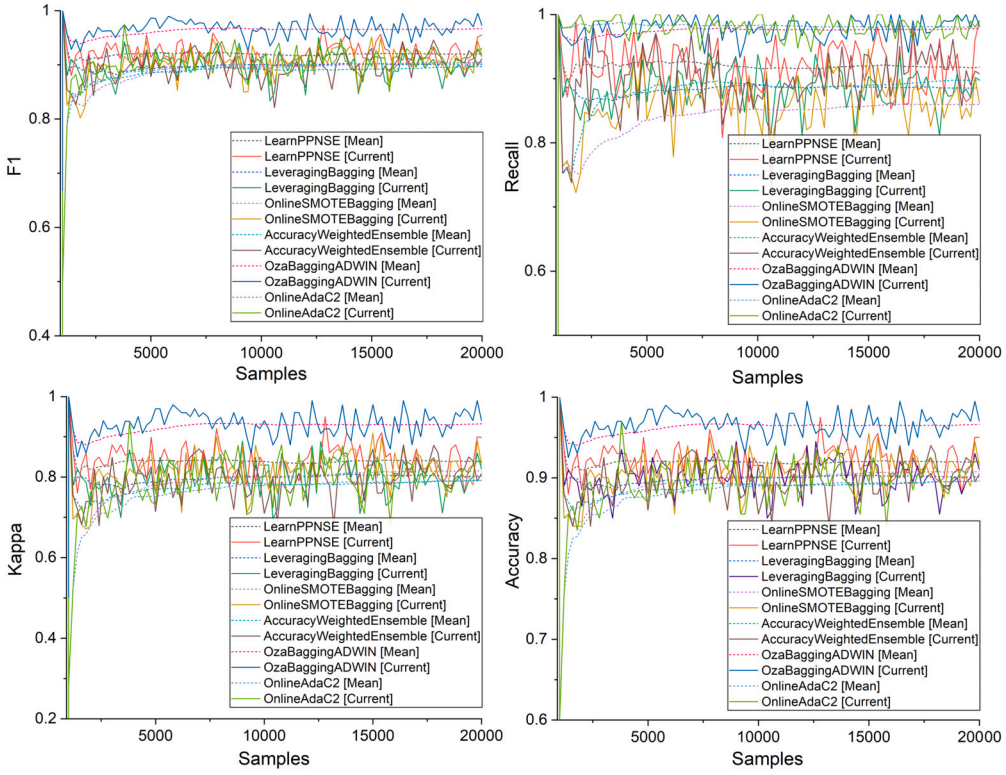


Fig. 6. Performance of the proposed scheme in detecting RPL attacks, moving mean and current.

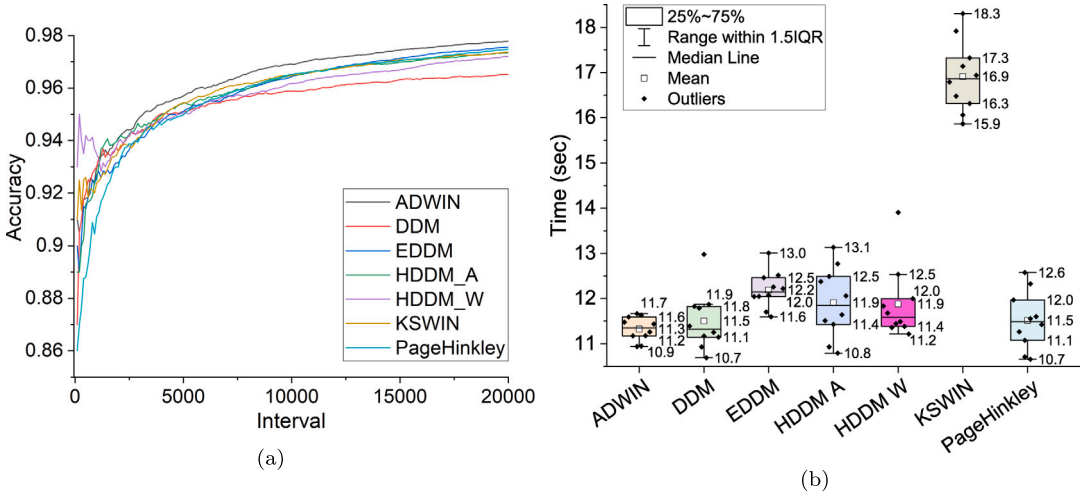


Fig. 7. Comparison of Concept-drift Detection methods.

using 4 learners and 8 neighbours (KNN) causes $O(log(n))$ time complexity in the system. Table 6 shows that ANIDS has linear and logarithmic time complexity in training and testing, while CIDS has polynomial time complexity in the proposed scheme. To measure the power consumption of each component, we use the Netsim Emulator feature to connect the physical microcontrollers with the simulation environment and connect digital ammeters to the microcontrollers. We run our simulations for 10 minutes, disabling all unnecessary background tasks and applications. The power consumption of an ANIDS and the CIDS in an LLN with 64 nodes was 3.505 J/s and 3.754 J/s, respectively, whilst a legitimate node without any ANIDS or CIDS consumed 3.17 J/s. In this way, we have satisfied *DP2* (lightweight IDS).

Scenario 6. Here, we first evaluate how well the proposed scheme detects each RPL attack in LLNs with different proportions of legitimate and adversarial nodes, while 20% of nodes, including 50% the malicious nodes, were mobile and moving, as shown in Table 7.

From Table 7, we can see that the performance of the proposed scheme is plausible in terms of the accuracy and false-negative rate (FNR) for detecting various RPL attacks. The proposed scheme can detect IR attack with high accuracy (up to ~97.9%); and the SH, BH, DS, and GH attacks with up to ~98.7%; WH with up to ~99.7%; WP with up to ~99.6%; and DA with up to ~100% accuracy. Our outcomes show that our proposed scheme satisfies *DP3* (accurate in evolving data environment) and *DP4* (detect a wide range of RPL attacks).

In Table 8, we conducted an additional experiment utilizing 128 nodes, with 1%, 2%, and 5% of these nodes assigned as malicious. Our initial assumption was that decreasing the number of malicious nodes would lead to less malicious traffic and, as a consequence, a decline in detection performance. However, our findings contradict this, showing

Table 4
Performance bench-marking with offline IDS in 6LoWPAN.

Paper	No. Nodes	No. Malicious	Duration minutes	Mobility	RPL Attacks									
					SH	BH	GH	IR	DA	DS	WH	WP	VN	NA
Bostani and Sheikhan (2017)	5~50	1~5	20	No	100%	-	85.36% ~92.68%	-	-	-	96% ~97.53%	-	-	-
Farzaneh et al. (2019)	20~40	1~30%	30	No	-	-	-	-	100%	-	-	-		
Foley et al. (2020)	11	1	30	No	93.14%	93.14%	-	-	-	-	-	-	-	-
Kasinathan et al. (2013)	10	1	-	No	-	-	-	-	-	-	-	-	-	-
Mayzaud et al. (2016b)	2~10	1	480	No	-	-	-	-	-	-	-	-		
Napiah et al. (2018)	8	1~3	~30	No	100%	-	-	-	100%	-	100%	-	-	-
Pongle and Chavan (2015)	8~24	1~2	30	No	-	-	-	-	-	-	94%	-	-	-
Raza et al. (2013)	8~64	1~4	~30	No	79%	-	81%	-	-	-	-	-	-	-
Shreenivas et al. (2017)	4~8	2	-	No	90% ~100%	-	-	-	-	-	-	-	-	-
Shukla (2017)	10~200	~2	-	No	-	-	-	-	-	-	71% ~75%	-	-	-
Proposed Scheme†	16~128	10~30%	360	Yes (20%)	91.5% ~98.7%	91.8% ~98.3%	90.6% ~98.7%	94.1% ~97.9%	99.8% ~100%	94.0% ~98.7%	90.1% ~99.7%	91.9% ~99.6%	97.5% ~98.8%	98.2% ~99.6%

*Results indicate the accuracy of the proposed IDS in detecting each type of RPL attack; † Details are shown in Table 7.

Table 5
Unknown attack detection.

Unknown Attack	Performance Metrics				
	Accuracy	*Precision*	*F1*	*TPR*	*FPR*
SH	90.85	91.16	90.79	86.52	5.17
BH	89.75	90.30	89.74	83.62	3.55
GH	93.9	94.07	93.88	90.97	3.31
IR	91.75	92.20	91.71	86.61	3.25
DA	98.30	98.36	98.29	96.57	0
WH	98.35	98.36	98.34	97.04	0.30
DS	93.95	94.05	93.94	91.62	3.76
WP	95.10	95.18	95.09	92.93	2.71
VN	90.5	90.52	90.48	89.25	8.33
NA	94.1	94.23	94.09	96.64	8.51

Table 6
Time complexity.

Comp	Training (sec)	Testing (sec)
ANIDS	$O(N)$: 0.36 + -2.4E-08*n	$O(log(n))$: 0.22 + -0.0021*log(n)
CIDS	-	$O((log\ n)^k)$: -2.3 * $x^{0.94}$

that our proposed scheme can consistently deliver high accuracy and F1 scores, even under stealth attack scenarios. This performance is due to the effectiveness of our engineered feature, which enables our proposed scheme to accurately model the anomalies caused by each type of routing attack, regardless of the volume of malicious traffic generated within the LLN. Notably, some of these RPL attacks are particularly extreme, making them easier to detect using our scheme (e.g. DA and WH attacks).

We then consider the detection of unforeseen intrusions, where each RPL attack was excluded from the pre-training data one-by-one and exclusively covered all the adversarial activities of the evaluation data stream, as shown in Table 5 Outcomes of this scenario show our proposed scheme can address *DP5* (detect unseen/unknown intrusions).

Takeaway. As the number of nodes in an LLN expands, establishing a balance between security measures, the performance of an IDS, and computational cost becomes increasingly critical.

On the one hand, heightened security measures, such as comprehensive packet inspections, can place additional strain on the IDS. These measures require extensive processing power, which can impede the IDS's performance, leading to slower detection times and potentially lower accuracy rates. It's especially relevant in real-time environments, where rapid intrusion detection is paramount for quick mitigation actions.

On the other hand, the escalation of security measures usually leads to increased computational costs. As the network becomes larger, the demands on the IDS to process and analyse the growing traffic also rise, resulting in a higher computational load. Higher security means more sophisticated and resource-intensive algorithms are needed; thus, the cost of computational resources, storage, and energy consumption can increase significantly.

To strike an optimal balance, it is essential to implement efficient and scalable security algorithms that can maintain high detection rates without excessively burdening the computational resources. Addressing these challenges, we have developed a hybrid, incremental IDS that is designed to optimise the balance between security, performance, and computational cost. By leveraging an incremental, machine-learning-based model and a hybrid approach, our solution efficiently scales with an increasing number of nodes, ensuring robust security without imposing unnecessary computational burdens.

5. Conclusion and future works

Routing threats in 6LoWPAN and threats against RPL are highly significant. In this article, we have introduced an adaptive hybrid heterogeneous IDS scheme that is effective and efficient and can readily cope with changes to the environment and detect known and unknown routing intrusions in the 6LoWPAN. In this context, we have conducted several simulations and scenarios to evaluate to what extent our proposed scheme can satisfy desirable properties (adaptivity, lightweightness, accuracy in evolving data environments, detecting a wide range of RPL attacks, and detecting both known and unknown intrusions). Our analysis shows the development of lightweight and distributed OCSVMs can enable our scheme to detect malicious activities with a 99.74% true positive rate. It should be noted that OCSVM and HalfSpace-Trees play a vital role in detecting unforeseen intrusions in our proposed scheme. In this regard, the outcomes of our experiments show that our proposed scheme has 90.8% ~ 98.3% accuracy in detecting unforeseen attacks. Moreover, our scheme can identify a wide range of RPL attacks in different scaled evolving (containing mobile nodes) LLNs with 97.9% ~ 100% accuracy. Our experimental outcomes clearly demonstrate that our proposed scheme is lightweight both in terms of energy consumption and time complexity. Our experimental outcomes show that the incremental ensemble of OZABagging with KNNADWIN learners and HalfSpace-Trese (HS-Trees) (Tan et al., 2011) creates a hybrid IDS that provides excellent performance in detecting intrusions.

Our benchmarking results give an *indicative* comparison between our scheme and the results obtained by other authors. However, it should

Table 7
Performance bench-marking.

N	M	Accuracy										FNR									
		SH	*BH*	*GH*	*DA*	*IR*	*WH*	*DS*	*WP*	*VN*	*NA*	*SH*	*BH*	*GH*	*DA*	*IR*	*WH*	*DS*	*WP*	*VN*	*NA*
16	10%	91.5	91.8	96.2	99.8	95.8	98.3	97.4	98.6	98.2	99.1	14.1	13.8	3.4	0	7.3	2.4	2.6	2.7	1.2	0.5
	20%	98.7	95.4	98.4	100	97.9	96.5	98.7	97.5	98.8	99.4	1.8	5.4	3.0	0	4.0	4.4	2.3	2.9	0.3	0.4
	30%	97.6	97.0	96.6	100	94.1	99.6	98.2	99.5	98.6	99.3	3.4	5.4	5.8	0	11.3	0.1	2.9	0.2	0.7	0.2
32	10%	93.3	96.3	98.5	99.8	97.8	99.7	98.5	99.6	98.4	99.2	10.0	5.4	2.0	0.3	3.8	0.2	2.2	0.5	1.1	0.6
	20%	98.7	98.2	98.2	100	97.8	94.8	98.4	95.2	98.1	99.2	2.4	3.1	2.0	0	3.6	9.5	2.0	8.7	2.2	0.7
	30%	98.6	98.3	98.7	100	97.0	90.1	98.7	91.9	98.0	99.0	2.3	3.2	2.4	0	5.3	16.0	2.4	13.2	1.6	0.6
64	10%	92.5	93.1	90.6	99.9	94.9	91.6	89.5	92.7	97.7	99.1	13.8	13.2	16.8	0.1	9.1	12.0	18.9	10.6	2.9	0.2
	20%	93.0	93.4	96.2	100	94.9	91.0	97.0	96.3	97.9	99.2	11.0	11.4	6.7	0	8.4	10.7	4.9	6.7	2.8	0.4
	30%	93.7	93.8	96.2	100	96.4	94.5	98.7	96.6	98.0	99.6	11.5	9.4	7.1	0	5.0	10.1	2.4	5.9	2.3	0.1
128	10%	97.2	93.0	91.2	99.8	95.5	93.5	94.0	92.3	98.3	99.5	5.4	13.4	16.0	0.4	8.1	9.2	8.2	11.3	2.1	0.3
	20%	93.6	93.9	94.1	100	95.9	94.4	96.0	93.1	97.5	99.4	11.7	11.0	10.0	0	6.1	10.5	6.7	13.3	2.9	0.4
	30%	94.3	94.9	96.9	100	96.9	95.2	96.7	95.4	98.0	99.3	10.0	8.4	5.8	0	4.7	8.5	5.8	7.8	2.7	0.3

SH: Sinkhole; **BH:** Blackhole; **GH:** Grayhole; **DA:** DIS Flooding; **IR:** Increase Rank; **WH:** Wormhole; **DS:** DIO Suppression; **WP:** Worst Parent; **VN:** Version Number; **NA:** Neighbour Attack; **N:** Total number of nodes; **M:** No. Malicious nodes; No. Mobile nodes ~20%.

Table 8
Evaluation results against a stealthy attacker, Accuracy and F1.

N	M	Accuracy										F1									
		SH	*BH*	*GH*	*DA*	*IR*	*WH*	*DS*	*WP*	*VN*	*NA*	*SH*	*BH*	*GH*	*DA*	*IR*	*WH*	*DS*	*WP*	*VN*	*NA*
128	1%	94.5	98.4	95.8	100	99.0	100	97.6	96.7	97.6	98.6	94.5	98.3	95.8	100	99.0	100	97.6	96.7	97.6	98.6
	2%	95.2	94.2	95.5	100	98.0	100	94.6	98.6	97.5	98.8	95.2	94.2	95.5	100	97.9	100	94.6	98.6	97.4	98.8
	5%	95.8	96.4	96.5	100	95.4	99.0	95.2	100	97.9	98.2	95.7	96.4	96.5	100	95.4	99.0	95.1	100	97.9	98.2

SH: Sinkhole, **BH:** Blackhole; **GH:** Grayhole; **DA:** DIS Flooding; **IR:** Increase Rank; **WH:** Wormhole; **DS:** DIO Suppression; **WP:** Worst Parent; **VN:** Version Number; **NA:** Neighbour Attack; **N:** Total number of nodes; **M:** Number of Malicious nodes.

be noted that our results are obtained in a much more challenging environment against a wider range of RPL attacks. Although our proposed scheme is capable of satisfying all desirable properties in different scaled evolving LLNs, the placement of our ANIDS needs to be optimised in the RPL networks. Furthermore, the proposed scheme is only capable of detecting intrusions, and the prevention mechanism still remains an open research question. Hence, in our next works, we will propose an AI-enabled scheme to satisfy these needs.

CRediT authorship contribution statement

Aryan MP mainly contributed in designing the proposed model, writing paper. **Prosanta Gope** has contributed in experimental section of this paper and writing. **John Clark** has contributed in supervision and enhancing the readability.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Aryan Mohammadi Pasikhani reports was provided by The University of Sheffield. Aryan Mohammadi Pasikhani reports a relationship with The University of Sheffield that includes: employment. Aryan Mohammadi Pasikhani has patent pending to NA.

Data availability

Data will be made available on request.

References

Alexander, R., Brandt, A., Vasseur, J., Huii, J., Pister, K., Thubert, P., Levis, P., Struik, R., Kelsey, R., Winter, T., 2012. RPL: IPv6 routing protocol for low-power and lossy networks. RFC 6550. https://doi.org/10.17487/RFC6550.

Barthel, D., Vasseur, J., Pister, K., Kim, M., Dejean, N., 2012. Routing metrics used for path calculation in low-power and lossy networks. RFC 6551. https://doi.org/10.17487/RFC6551.

Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2012. Survey on incremental approaches for network anomaly detection. arXiv preprint, arXiv:1211.4493.

Bifet, A., Gavalda, R., 2007. Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM International Conference on Data Mining. SIAM, pp. 443–448.

Bifet, A., Gavalda, R., 2009. Adaptive learning from evolving data streams. In: International Symposium on Intelligent Data Analysis. Springer, pp. 249–260.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavalda, R., 2009. New ensemble methods for evolving data streams. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 139–148.

Bostani, H., Sheikhan, M., 2017. Hybrid of anomaly-based and specification-based ids for Internet of things using unsupervised opf based on mapreduce approach. Comput. Commun. 98, 52–71.

Darabkh, K.A., Al-Akhras, M., Zomot, J.N., Atiquzzaman, M., 2022. Rpl routing protocol over iot: a comprehensive survey, recent advances, insights, bibliometric analysis, recommendations, and future directions. J. Netw. Comput. Appl. 207, 103476.

Elwell, R., Polikar, R., 2011. Incremental learning of concept drift in nonstationary environments. IEEE Trans. Neural Netw. 22 (10), 1517–1531.

Farzaneh, B., Montazeri, M.A., Jamali, S., 2019. An anomaly-based ids for detecting attacks in rpl-based Internet of things. In: 2019 5th International Conference on Web Research (ICWR). IEEE, pp. 61–66.

Foley, J., Moradpoor, N., Ochen, H., 2020. Employing a machine learning approach to detect combined Internet of things attacks against two objective functions using a novel dataset. Secur. Commun. Netw. 2020.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A., 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46 (4), 1–37.

Gomes, H.M., Barddal, J.P., Enembreck, F., Bifet, A., 2017. A survey on ensemble learning for data stream classification. ACM Comput. Surv. 50 (2), 1–36.

Hassan, H.A., Hemdan, E.E., El-Shafai, W., Shokair, M., El-Samie, F.E.A., 2023. Intrusion detection systems for the Internet of thing: a survey study. Wirel. Pers. Commun. 128 (4), 2753–2778.

Ioulianou, P.P., Vassilakis, V.G., Shahandashti, S.F., 2022. Ml-based detection of rank and blackhole attacks in rpl networks. In: 2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP). IEEE, pp. 338–343.

Kaliyar, P., Jaballah, W.B., Conti, M., Lal, C., 2020. Lidl: localization with early detection of sybil and wormhole attacks in iot networks. Comput. Secur. 94, 101849.

Kareem, M.A., Tayeb, S., 2021. Ml-based nids to secure rpl from routing attacks. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, pp. 1000–1006.

Kasinathan, P., Costamagna, G., Khaleel, H., Pastrone, C., Spirito, M.A., 2013. An ids framework for Internet of things empowered by 6lowpan. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 1337–1340.

Le, A., Loo, J., Chai, K.K., Aiash, M., 2016. A specification-based ids for detecting attacks on rpl-based network topology. Information 7 (2), 25.
Li, J., Zhao, Z., Li, R., Zhang, H., 2018. Ai-based two-stage intrusion detection for software defined iot networks. IEEE Int. Things J. 6 (2), 2093–2102.
Maglaras, L.A., Jiang, J., 2014. A real time ocsvm intrusion detection module with low overhead for scada systems. Int. J. Adv. Res. Artif. Intell. 3 (10).
Maheswaran, N., Logeswari, G., Bose, S., Anitha, T., 2023. A critical review on intrusion detection systems in iot based on ml approach: a survey. In: 2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN). IEEE, pp. 1–8.
Manne, V.R.J., Sreekanth, S., 2022. Detection and mitigation of rpl routing attacks in Internet of things. In: 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, pp. 481–485.
Martindale, N., Ismail, M., Talbert, D.A., 2020. Ensemble-based online machine learning algorithms for network intrusion detection systems using streaming data. Information 11 (6), 315.
Mayzaud, A., Badonnel, R., Chrisment, I., 2016a. A taxonomy of attacks in rpl-based Internet of things. Int. J. Netw. Secur. 18 (3), 459–473.
Mayzaud, A., Sehgal, A., Badonnel, R., Chrisment, I., Schönwälder, J., 2016b. 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, pp. 366–374.
Mayzaud, A., Badonnel, R., Chrisment, I., 2017. A distributed monitoring strategy for detecting version number attacks in rpl-based networks. IEEE Trans. Netw. Serv. Manag. 14 (2), 472–486.
Mitrokotsa, A., Karygiannis, A., 2008. Intrusion detection techniques in sensor networks. Wirel. Sens. Netw. Secur. 1 (1), 251–272.
Napiah, M.N., Idris, M.Y.I.B., Ramli, R., Ahmedy, I., 2018. Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol. IEEE Access 6, 16623–16638.
Örs, F.K., Levi, A., 2023. Data driven intrusion detection for 6lowpan based iot systems. Ad Hoc Netw. 143, 103120.
Oza, N.C., Russell, S.J., 2001. Online bagging and boosting. In: International Workshop on Artificial Intelligence and Statistics. PMLR, pp. 229–236.
Pasikhani, A.M., Clark, J.A., Gope, P., 2021a. Reinforcement-learning-based ids for 6lowpan. In: 20th IEEE International Conference on Trust, Security and Privacy in Computing and Communication (TrustCom).
Pasikhani, A.M., Clark, J.A., Gope, P., Alshahrani, A., 2021b. Intrusion detection systems in rpl-based 6lowpan: a systematic literature review. IEEE Sens. J.
Pongle, P., Chavan, G., 2015. Real time intrusion and wormhole attack detection in Internet of things. Int. J. Comput. Appl. 121 (9).
Przybocki, P., Vassilakis, V.G., 2023. An analysis into physical and virtual power draw characteristics of embedded wireless sensor network devices under dos and rpl-based attacks. Sensors 23 (5), 2605.
Raghavendra, T., Anand, M., Selvi, M., Thangaramya, K., Kumar, S.S., Kannan, A., 2022. An intelligent rpl attack detection using machine learning-based intrusion detection system for Internet of things. Proc. Comput. Sci. 215, 61–70.
Raza, S., Wallgren, L., Voigt, T., 2013. Svelte: real-time intrusion detection in the Internet of things. Ad Hoc Netw. 11 (8), 2661–2674.
Shafique, U., Khan, A., Rehman, A., Bashir, F., Alam, M., 2018. Detection of rank attack in routing protocol for low power and lossy networks. Ann. Telecommun. 73 (7), 429–438.
Shreenivas, D., Raza, S., Voigt, T., 2017. Intrusion detection in the rpl-connected 6lowpan networks. In: Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, pp. 31–38.
Shukla, P., 2017. Ml-ids: a machine learning approach to detect wormhole attacks in Internet of things. In: 2017 Intelligent Systems Conference (IntelliSys). IEEE, pp. 234–240.
Singh, R., Kumar, H., Singla, R., 2015. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst. Appl. 42 (22), 8609–8624.
Tan, S.C., Ting, K.M., Liu, T.F., 2011. Fast anomaly detection for streaming data. In: Twenty-Second International Joint Conference on Artificial Intelligence.
Viegas, E., Santin, A., Oliveira, L., Franca, A., Jasinski, R., Pedroni, V., 2018. A reliable and energy-efficient classifier combination scheme for intrusion detection in embedded systems. Comput. Secur. 78, 16–32.
Wang, B., Pineau, J., 2016. Online bagging and boosting for imbalanced data streams. IEEE Trans. Knowl. Data Eng. 28 (12), 3353–3366.
Wang, H., Fan, W., Yu, P.S., Han, J., 2003. Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226–235.
Webb, G.I., Hyde, R., Cao, H., Nguyen, H.L., Petitjean, F., 2016. Characterizing concept drift. Data Min. Knowl. Discov. 30 (4), 964–994.
Yuan, X., Wang, R., Zhuang, Y., Zhu, K., Hao, J., 2018. A concept drift based ensemble incremental learning approach for intrusion detection. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 350–357.

Aryan Mohammadi Pasikhani has been a Post-Doctoral Researcher with the Security of Advanced Systems Group at the University of Sheffield. He is currently an Academic Fellow in Cybersecurity at the same institution. His focus is on publishing research that makes a high impact in the fields of Computer and Network Security. His primary research interests encompass intrusion detection and prevention systems, reinforcement learning, machine learning, quantum computing, and the security of embedded systems. Furthermore, he has been an esteemed Technical Program Committee (TPC) Member and has undertaken peer review responsibilities for several prestigious international journals and conferences, including the IEEE Internet of Things Journal, IEEE Transactions on Industrial Informatics, and the IEEE Sensors Journal.

John A. Clark is a Professor of computer and information security at The University of Sheffield and leads the Security of Advanced Systems Research Group. Previously, he was a Professor of critical systems at the University of York. His major research interests lie in cybersecurity and software engineering, most notably the use of artificial intelligence to these areas. His publications have included work on threat modeling, security policies, covert channel, analysis, cryptographic building blocks, intrusion detection, insider detection, and automated synthesis of security protocols. His current work addresses the automated discovery of classical cryptanalytic strategies, intrusion detection (particularly in the IoT systems) and its optimal configuration, and consent in healthcare IoT systems.

Prosanta Gope (Senior Member, IEEE) served as a Research Fellow with the Department of Computer Science, National University of Singapore. He is currently working as an Assistant Professor with the Department of Computer Science (Cyber Security), The University of Sheffield, U.K. He has authored more than 100 peer-reviewed articles in several reputable international journals and conferences and has four filed patents. Several of his papers have been published in high-impact journals, such as IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Industrial Electronics, and IEEE Transactions on Smart Grid. Primarily driven by tackling challenging real-world security problems, he has expertise in lightweight anonymous authentication, authenticated encryption, access control, security of mobile communications, healthcare, the Internet of Things, cloud, RFIDs, WSNs, smart-grid, and hardware security of the IoT devices. He has served as the TPC Member/Chair in several reputable international conferences, such as ESORICS, IEEE TrustCom, and ARES. He currently serves as an Associate Editor for IEEE Internet of Things Journal, IEEE Systems JOURNAL, IEEE Sensors JOURNAL, and the *Journal of Information Security and Applications* (Elsevier).