Anomaly Detection in Logical Sub-Views of WSNs

Robert Zakrzewski
Communication Systems and Networks
University of Bristol
Bristol, UK
robert.zakrzewski@bristol.ac.uk

Trevor Martin
Artificial Intelligence Group
University of Bristol
Bristol, UK
trevor.martin@bristol.ac.uk

George Oikonomou
Communication Systems and Networks
University of Bristol
Bristol, UK
g.oikonomou@bristol.ac.uk

***Abstract*—Wireless sensor networks are often distributed, diverse, and large making their monitoring hard. One way to tackle it is to focus on part of the system by creating logical sub-views which can be seen as proxies of the overall system operations. In this manuscript, logical sub-views consist of traffic aggregators and their topology which are monitored for anomaly. The aggregators are selected based on diversity and importance in the system and they are modelled as graphs to capture aggregation topology and data distributions. The aggregators' selection criteria, the method for comparison of partially overlapping sub-views, normal aggregation profiles acquisition, and measures of anomaly are proposed. A simulated wireless sensor network is used to acquire data at the edge and apply the method to demonstrate that focusing on system sub-views and comparing aggregation profiles facilitates anomaly detection also caused elsewhere in the system and the impact the anomaly has on aggregators.**

***Index Terms*—Traffic aggregation, Anomaly detection, Machine learning, Sensor networks, Graph, Cyber-security.**

I. Introduction

Wireless sensor networks (WSN) are vulnerable to attacks due to limited resources available for protection and deployment which by nature is ubiquitous and diverse. Ubiquity makes monitoring difficult. A "black box" approach can be used to analyze what is observable externally (e.g. at the cloud edge) to infer about something unusual occurring in the system. Data (also encrypted) and network topology are observable and they may indicate attacks, operational problems or misconfiguration. From an operational point of view, monitoring sub-views containing critical infrastructure such as data aggregators helps in deciding on upgrades, redesign or detect system misconfiguration and malicious activity. By capturing, and measuring normal aggregation topology and data profiles, it enables the system to guard the normal state and flag up any deviations. Aggregation topology and data profiles are modelled as graphs (sub-views) which are subject to change causing some graphs to have non overlapping sets of vertices making comparison difficult.

In our previous work [1], anomaly detection and change measurement method was introduced for the entire system modelled as graphs with the the same set of vertices representing sensors, and scalar similarity measures were defined. In this paper the method was extended to monitor a fraction of the system (sub-views), and perform anomaly detection in the sub-views which partly overlap (graphs with partly overlapping sets of vertices). A similarity vector as opposed to a scalar similarity value in [1] is proposed, and a hybrid classifier based on decision trees for topology matching and clustering algorithm based on local density estimation for anomaly measurement are introduced. A method allowing acquiring normal aggregation data and topology profiles (ground truth) with the metrics measuring deviation from normal is presented. The method allows scaling the sub-views to reflect computational resources available, and measures aggregation traffic which address scalability of the method in [1]. It was showed that sub-views monitoring can be used to detect system change triggered elsewhere in the system and the impact the anomaly has on aggregators.

II. Related work

Anomaly detection in wireless sensor networks may use supervised or unsupervised machine learning algorithms. The supervised machine learning methods are based on variants of support vector machine, tree based classifiers, or clustering algorithms [2], [3]. They require labelled training and test data for normal and abnormal scenarios, the latter always difficult to acquire to be robust for future threats. This is partly addressed in unsupervised methods by making implicit assumption about when data is normal e.g. based on density, distance, variance [4], [5] which still might be inadequate for the threats which fit into the assumed normal condition. Neural networks and deep learning were also used for anomaly detection [6], however their computational resource requirements and lack of generality (e.g. feature selections to address particular anomaly or attacks) makes them difficult to apply in the constrained networks with unknown threat models. Fault detection methods in [7] have many specialised techniques but they lack generality.

In this contribution a technique for traffic aggregation profiles comparison is proposed to facilitate passive and generic anomaly detection with change measurement. Aggregation topology pattern is reflected by a graph structure, whereas aggregation data pattern is expressed by graph structure and weights assigned to edges. The comparison metrics are calculated and used as the inputs for machine learning algorithms which by design require only normal profiles for training. This feature in particular enables detection of unknown threats, misconfiguration and operational problems as long as they have impact on the normal aggregation profiles. The method can be deployed at cloud edge to tackle the challenge of geographical distribution of devices.

III. System view (WSN)

Topologies representing data path connectivity are formed in WSNs based on 6LoWPAN/IPv6 which are deployed in a multi-hop mesh scenario. In this setup some routing protocol needs to be used to form paths for data collection from sensors to the edge, as well as for traffic in the reverse direction. The routing protocol used in this work is RPL (Routing Protocol for Low-Power and Lossy Networks). RPL

perceives the network as a Destination Oriented Directed Acyclic Graph (DODAG). DODAG is not always a tree, but in practice the existing RPL implementations often result in the routing topology being a tree. This therefore means that traffic flowing up the tree towards the root causes a funneling effect. The method presented in this paper can be applied to any protocol layer (e.g. a bespoke network stack), and to measure traffic volumes at a different layer in the stack. However, in this paper it is assumed that sensor data traffic is MQTT over TCP and MQTT/TCP protocols are used to push data to the cloud. In this context, MQTT traffic volumes are measured between sensors in the traffic aggregation sample.

Sensors also relay data to/from other sensors to facilitate the distribution of WSNs. This makes the system vulnerable to attacks and misconfiguration with the impact exacerbated as the impaired devices get closer to the edge router due to the funneling effect. System sub-views are built by invoking the aggregators selection process on traffic flows aggregate graphs described in section III-A. Edges represent data flows; vertices represent aggregators. Vertices are annotated with labels reflecting unique IPv6 addresses allocated to sensors. Edge weights contain normalized data volumes traversing between nodes, self-looped at vertex with no traffic. Data volumes are measured as the uplink TCP payload sizes used to carry MQTT traffic. Since the method detects deviation at egress flows from vertices and the topology is a tree, the direction of flows is reversed. Other schemes for sub-views creation are also possible, e.g. to include bidirectional flows, or other protocol data.

To build a traffic flows aggregate graph, the current topology for data transfer is acquired at a border router. The border router keeps track of disseminated and received routing control information by constructing a system-wide routing table. To trace the paths taken by data, the acquired current topology and the data captured at the border router are matched. For the matching process, packet source and destination IP addresses are looked up in the routing table. Then data paths (flows) are aggregated over time in the traffic aggregation graph.

The method has $\mathcal{O}(n^2)$ space and time complexity. However, a sub-view graph with k-vertices, is a subset of the WSN modelled as an l-vertex graph. As a result, in comparison with [1], space and time complexity is reduced by the factor $(k/l)^2$ with the size of a sub-view graph adjustable by the graph pruning parameters in Alg. 1.

A. Aggregators definition and selection process

Aggregators are defined as nodes important for data transfer. The importance of nodes is measured by diversity of traffic they carry and how much they contribute to the overall system. The right selection of aggregators forming sub-views is key in order to make them good proxies of the system.

The selection process (Alg. 1) performs graph pruning. The algorithm has the following parameters:

- min flow count for an edge to survive (PAR1)
- min count of surviving edges for a vertex survival (PAR2)
- max number of surviving vertices selected (PAR3)
- data/traffic of interest (PAR4)

In this contribution, PAR4 is configured to be MQTT traffic, and flows are unique MQTT traffic endpoints.

Algorithm 1: Aggregators' selection

Inputs: Routing topologies, Traffic, PAR[1-4]
Outputs: Sub-view (graph)
SET traffic of interest (PAR4);
Build graph containing aggregated traffic flows;
Label edges with number of different flow count;
Label vertices with traffic volumes;
Rank vertices with highest traffic volume;
Rank edges with highest flow count;
Prune edges with no min flow count (PAR1);
Prune vertices with no min surviving edges (PAR2);
Select top surviving vertices up to PAR3;
Build a graph which contains surviving vertices, edges;

B. Aggregators pattern change

Aggregation profile change is measured by similarity vector composed from metrics acquired from the graphs. As sub-views are pruned graphs, they naturally focus on part of the system. The pruning helps scalability as the pruning parameters impact graph size. As sub-views contain traffic aggregates, they are impacted to a lesser extent by individual sensors' behaviour. The pruning algorithm allows customisation e.g. to take geographical location into account, or distance from the cloud edge (not done in this contribution).

C. Partly overlapping sub-views

Sub-views may partly overlap, which is modelled by graphs with partly overlapping vertices. For comparison, the graphs are matched for missing vertices. A missing vertex is modelled as a detached node with a self looped edge discussed in section IV-A.

IV. Mathematical summary

The weighted, annotated (labelled), directed graphs can be described by a weighted adjacency matrix. The weights as distribution of data volumes represent conditional probability of transition to vertex *n*, given vertex *k* and traffic type *t*.

$$w_{kn} = P(v_n|_{v_k,t}), for\ all\ k \sum_n w_{kn} = 1 \tag{1}$$

The systems represented by graphs A (baseline) and B (assessed) each described by the corresponding adjacency matrix $\mathbf{A}$ and $\mathbf{B}$ are compared. The adjacency matrices are constructed by applying the same ordering of labelled vertices. As matrix multiplication can be seen as rotations and re-scaling of each input vector, graphs change can be reflected in matrix $\mathbf{V_B}$ by applying transformation matrix $\mathbf{T}$ to matrix **(A-B)**. The transformation matrix $\mathbf{T}$ is found by calculating the Moore-Penrose pseudo inverse of $\mathbf{A}$, $\mathbf{A}^+$. $\mathbf{T}$ modifies matrices $\mathbf{A}$ and $\mathbf{B}$ to make $\mathbf{V_A}$ orthonormal with 1s on the diagonal so that matrix $\mathbf{V_B}$ reflects the graph B (assessed) in relation to graph A (baseline). The matrices $\mathbf{V_A}$ and $\mathbf{V_B}$ are approximated in the least square sense by using Moore-Penrose pseudo inverse. $\|\cdot\|$ denotes Frobenius norm. Then the measures (metrics) d_1, d_2, d_3 are calculated.

$$\begin{aligned} &\mathbf{d1_n} = \|\mathbf{a_{Tn}} - \mathbf{b_{Tn}}\| \\ &a_{Tn}, b_{Tn} - \text{ n}^{\text{th}} \text{ row vector of A}_\text{T}\text{, and B}_\text{T} \end{aligned} \tag{2}$$

Measure $d1_n$ produces values bounded by the interval [0..$u1_n$] with 0 indicating perfect alignment (similarity). $u1_n$ is the upper bound in (3) and its value depends on graph A.

$$u1_n = max(\{elem|elem = \|\overrightarrow{a_{Tn}} - \overrightarrow{v_k}T\|,\ for\,all\,k\})$$
$$\overrightarrow{v_k} - \text{ zero vector with 1 at component k} \quad (3)$$
$$\overrightarrow{a_{Tn}} - \text{ n}^{\text{th}} \text{ row vector of A}_\text{T}$$

$\mathbf{d2_n}$ is defined in (4) as cosine similarity measure, where $\mathbf{a_{Tn}}$, $\mathbf{b_{Tn}}$ are n^{th} row vectors of matrices $\mathbf{A_T}$, $\mathbf{B_T}$ respectively.

$$\mathbf{d2_n}(\overrightarrow{a_{Tn}}, \overrightarrow{b_{Tn}}) = \frac{\overrightarrow{a_{Tn}} \cdot \overrightarrow{b_{Tn}}}{\|\overrightarrow{a_{Tn}}\| \cdot \|\overrightarrow{b_{Tn}}\|} \quad (4)$$

$\mathbf{d2_n}$ is bounded by the interval [$u2_n$..1] with 1 indicating perfect alignment (similarity). $u2_n$ is the lower bound defined in (5) and its value depends on graph A.

$$u2_n = min(\{elem|elem = \frac{\overrightarrow{a_{Tn}} \cdot \overrightarrow{v_k}T}{\|\overrightarrow{a_{Tn}}\| \cdot \|\overrightarrow{v_k}T\|},\ for\,all\,k\})$$
$$\overrightarrow{v_k} - \text{ zero vector with 1 at component k} \quad (5)$$

For data pattern also metric $d3_n$ is defined in (6) bounded by the interval [0..$u3_n$]

$$\mathbf{d3_n} = \arccos(d2_n),\ \mathbf{u3}_\text{n} = \arccos(u2_n) \quad (6)$$

Given the base graph, metrics $d1_n$, and $d3_n$ have the lower bound equal to zero if the base and assessed vertices are the same and change reflecting anomaly increase.

Graph scope similarity information is obtained by aggregating vertex level information in the graph similarity vector $\overrightarrow{s}$ defined in (7) which is in contract to [1] which uses a scalar aggregated similarity measure.

$$\overrightarrow{s} = [d1_1, d3_1, ..., d1_n, d3_n] \quad (7)$$

Given the base graph, a L2 norm of $\overrightarrow{\mathbf{s}}$ have the lower bound equal to zero if the base and assessed graphs are the same and increase with the anomaly increase with the upper bound U_s defined in (8).

$$\overrightarrow{u_s} = [u1_1, u3_1, ..., u1_n, u3_n],\ \mathbf{U_s} = \|\overrightarrow{u_s}\| \quad (8)$$

The dimension and sparsity of the vector $\overrightarrow{s}$ is controlled by the graph pruning parameters and in particular PAR3 and PAR1 defined in Alg. 1 to avoid the phenomenon called the curse of dimensionality.

A. Partly overlapping sub-views

Applying transformation $\mathbf{T}$ can be seen as moving input data to the latent space, where the normal graph A is transformed to be represented by the orthonormal vectors (or to the least square approximation). In [1] , and in section IV it is assumed that graphs have the same labelled vertices. Graphs partially overlapping include scenarios when a set of vertices appear (case C1) or/and disappear (case C2) in the assessed and ground truth graphs as compared with the base graph (graph A).

To facilitate graph comparison, the graphs are matched for the missing vertices by the following operations:

- self-looped node is added to the base graph A (C1)
- node is inserted and detached (self-looped) in the assessed graphs (C2)

In C1, a new orthonormal vector is added to the latent space, whereas in C2 the vector is moved to the edge of the bounded hyper-plane representing the latent space.

Adding new orthonormal vector(s) to the latent space does not require recalculation of matrix $\mathbf{T}$ as an extension to matrix $\mathbf{T}$ suffice. This is advantageous as $\mathbf{T}$ can be calculated offline and $\mathbf{T_{ext}}$ requires simple matrix manipulation. The pseudo inverse formula for the block 2x2 matrices is presented in (9). $\mathbf{T}$ is already known, and $\mathbf{D}$, $\mathbf{D}^+$ are identity matrices of the size dependent on the number of new vectors added to the latent space.

$$\mathbf{A}^+ = \mathbf{T}$$
$$\mathbf{T_{ext}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}^+ = \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^+ \end{bmatrix} \quad (9)$$

This arrangement impacts metrics calculation. Despite ordering and enumeration of vertices, metrics $d1_n, d2_n, d3_n$ are not impacted by order and enumeration of vector components from the assessed adjacency matrix extending matrix $\mathbf{T}$ (as matrix $\mathbf{D}^+$ is an identity matrix, the metrics are invariant to permutations and different enumerations of the same vector components). This is addressed by the use and construction of a graph topology hash and similarity vector $\overrightarrow{s}$ in section V.

V. Normal aggregation profiles acquisition

The baseline aggregation graph is created by normal traffic flows aggregation, and running the aggregators' selection process (Alg.1). Aggregation profile samples (graphs) are acquired based on the criteria presented in section V-A, and compared with the baseline graph to create normal aggregation profiles. The set of graph pruning parameters for Alg.1 can be less conservative for the baseline to maximize the information contained i.e. retaining more aggregation nodes (PAR3), and surviving edges (PAR2).

Aggregation topology samples are modelled as graphs for which metrics ($d1_n$, $d2_n$) are calculated, and aggregated to produce a graph level topology measure. Vertex n hash value H_{Vn} is calculated using SHA256 algorithm, H_{Vn}=$SHA256(d1_n, d2_n)$, and aggregated at the graph level in H_G=$SHA256(H_{V1}, ..., H_{Vn})$. The ordering of vertices is kept in H_G also for vertices extending graph A inserting H_{Vk}=$SHA256(0, 1)$ for vertices absent in a graph sample but enumerated lower than vertices extending graph A.

Aggregation data pattern samples are also modelled as graphs with edge weights representing normalised traffic volumes. Similarity vector $\overrightarrow{s}$ is created using metrics $(d1_n, d3_n)$. The vertex ordering is kept, inserting vector components (0,0) for vertices absent in a graph sample but enumerated lower than vertices extending graph A.

To calculate metrics and similarity vectors, weighted adjacency matrices **A1**, **A2** are built for the baseline graph following the same vertex ordering:

- for topology change, pseudo random values are generated (re-initialised for each vertex with the same seed), normalised to satisfy (1), and used as edge weights in **A1**
- for aggregation data pattern change, the volumes of egress data are normalized in edge weights in **A2**

Similarly for aggregation sample graphs, matrices **B1**, **B2** are built following the same vertex ordering for topology and data

profiles and compared in similarity vectors. The acquisition of normal profiles is described in Alg. 2.

Algorithm 2: Normal aggregation profile acquisition

Inputs: PAR[1-7], data acquisition timeout
Outputs: Set of normal aggregation profiles
SET data acquisition timeout value;
SET aggregators selection (pruning) parameters;
SET sample acquisition parameters;
while *Not Timeout(data acquisition)* **do**
  Record traffic/topology for the sampling interval;
  Aggregate traffic flows Run aggregators selection algorithm;
  if *aggregation profile sample criteria met* **then**
    Calculate,save matrices **B**1, **B2**;
    Start new aggregation profile acquisition;
  end
end
Aggregate recorded traffic flows (baseline);
Run aggregators selection algorithm for the baseline;
Calculate matrices **A1**, **A2** and use them as baseline;
Calculate, save $U_{S(gtruth,baseline)}$ as in (8);
Calculate, save $H_G, \overrightarrow{s}$, for aggregation profile samples;
Calculate $U_{S(gtruth,top)}$ for each new aggregation topology as in section VI-A;
Calculate centroid locations for each topology $\overrightarrow{s_{c(gtruth,top)}}$ as in section V-A;
Use H_G, $\overrightarrow{s_{c(gtruth,top)}}$, $\overrightarrow{s}$, $U_{S(gtruth,top)}$ as normal profile;

A. Ground truth data acquisition

In order to capture a valid aggregation profile, an aggregation profile sample (graph) is built based on traffic flows with the requirements imposed on a profile sample. They are defined by the parameters:

- min samples a topology remains unchanged (PAR5)
- relative adjacency matrix change threshold (PAR6)
- min samples the adjacency matrix change remains below threshold (PAR7)

As part of the ground truth acquisition, centroid centers are calculated for each aggregation topology, and used for the measurements of data pattern change. For centroid centers calculation, the mean shift clustering algorithm is used. There could be more than one centroid per aggregators' topology. Ground truth acquisition requires:

- metrics ($d1_n$, $d2_n$) for hash H_G
- metrics ($d1_n$, $d3_n$) for similarity vector $\overrightarrow{s}$
- centroid locations $\overrightarrow{s_{c(gtruth,top)}}$
- max bound $U_{S(gtruth,top)}$ for each topology

The criteria to stop profiles acquisition is based on low new topology discovery rate and low centroids change per batch.

VI. System supervision and anomaly detection

System supervision is similar to normal aggregation profiles acquisition. Each assessed aggregation profile sample is compared with the baseline aggregation sample as presented in Alg. 3. The resulting topology hash and similarity vector are compared with the ground truth for anomaly detection and change measurement.

Algorithm 3: Assessed aggregation profile acquisition

Inputs: PAR[1-7], timeout
Outputs: topology hash H_G, similarity vector $\overrightarrow{s}$
SET timeout value;
SET aggregators selection (pruning) parameters;
SET sample acquisition parameters;
repeat
  Record traffic/topology for the sampling interval;
  Aggregate traffic flows;
  Run aggregators selection algorithm;
  if *aggregation profile sample criteria met* **then**
    Calculate matrices **B**1, **B2**;
  end
until *Aggregation profile sample acquired or Timeout*;
Calculate, $H_G, \overrightarrow{s}$, for an aggregation profile sample;

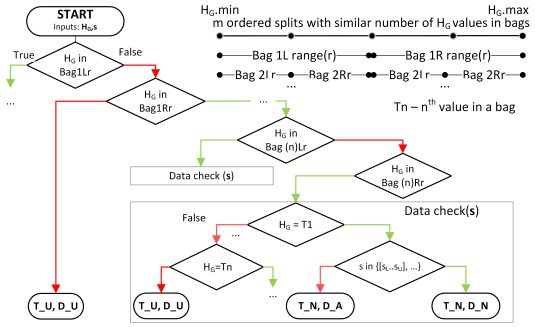

Fig. 1. Tree based classifier

For anomaly detection, two classifiers are used. The first classifier based on decision tree was proposed in [1] but in this paper metrics defined for the similarity vectors are used as the classifier inputs. The second hybrid classifier is an enhancement of the first classifier which is based on decision tree and clustering algorithm using local density estimation. The classifiers are trained with positive (normal) samples for anomaly detection. Negative samples are implicit as the complement of normal, allowing detection of unknown threats. Both classifiers allow updating the ground truth without the need for re-training.

A tree based classifier requires for training only topology hash values H_G) and aggregation data patterns ranges $\overrightarrow{s}$ +/- s_{margin}. The equal split strategy assumes comparable number of topology hashes in each bag with the optimal number of bags $m = \lceil\sqrt{2n}\rceil$ where n is the number of normal topologies. The classifier structure is presented in Fig. 1 with the truth table in Table I. The proposed tree based classification is fast and efficient requiring the maximum d*(m+n/m+2) comparisons per classification task where d is the length of the similarity vector $\overrightarrow{s}$. Each aggregation data point with the margin must have sufficient coverage for classification which is specified by the coverage ratio C. For a given baseline topology, the margin is defined in (10) in relation to the upper bounds $\mathbf{U_s}$ in (8). The margin can be adjusted to consider long term drift.

$$s_{margin} = C \cdot U_s \tag{10}$$

An enhancement in the second classifier allowing quantifying profile change, requires calculation of centroids, and the function given in Table II. The tree classifier is used for

TABLE I
TREE BASED CLASSIFIER (TRUTH TABLE)

Topology matched	Data within range	Classifier outputs classes	
		T (topology)	D (data)
False	N/A	Undefined (T_U)	Undefined (D_U)
True	False	Normal (T_N)	Abnormal (D_A)
True	True	Normal (T_N)	Normal (D_N)

TABLE II
CLASSIFIER AND OUTPUT FUNCTION

Inputs	Topology matched	Classifier		Function	
		topology	data	Output 1	Output 2
H,s	False	Abnormal	Undefined	-1	-1
	True	Normal	Cluster label	s_{dc}	d_{max}

matching topology as in Fig. 1 except the data check block returns a cluster label, which is used to calculate a sample distance s_{dc} to the closest controid. The maximum change d_{max} is retrieved from the ground truth using topology hash H_G. The mean shift clustering algorithm is used to calculate centroids, and to obtain cluster label/location for the closest controid for an assessed similarity vector. s_{dc} and d_{max} are used in (11) to calculate anomaly score.

A. Relative data change estimate

For a given topology, the relative aggregation profile change is calculated based on:

- assessed aggregation profile similarity vector $\overrightarrow{s}$
- the centroid location ($\overrightarrow{s_{c(gtruth,top)}}$)
- the upper bounds for a similarity vector change ($U_{S(gtruth,topology)}$)

$\mathbf{U_{S(gtruth,topology)}}$ is calculated in a similar way to (8) except vector $\overrightarrow{v_k}$ in (3),(5) is defined if edge k exists for vertex n in the assessed aggregation topology. Given a topology, the measure showing relative change is defined in (11).

$$\mathbf{s_{rel}} = \frac{\|(\overrightarrow{s} - \overrightarrow{s_{c(gtruth,top)}})\|}{U_{S(gtruth,top)}} = \frac{s_{dc}}{d_{max}} \qquad (11)$$

The relative data change is bounded by the interval [0..1] increasing as divergence increase, which is useful for soft decision boundary for classification and decision systems.

VII. SIMULATION

For demonstration, the method was applied to a wireless sensor network simulated in Cooja [8], a simulator for IEEE 802.15.4 [9] networks distributed as part of Contiki-NG open source operating system for constrained devices. The Routing Protocol for Low-Power and Lossy Networks (RPL) was used for routing in the non-storing mode, and symmetric routing paths [10]. Aggregation topologies and data profiles were acquired based on the data and routing tables at the edge (border) router. Traffic volumes are measured between sensors for each aggregation profile sample as TCP payload sizes in bytes used to carry MQTT traffic, and then normalised to satisfy (1). If aggregation topology cannot be constructed for a number of assessed consecutive samples (set to 30), a null aggregation profile is saved (detached and self looped nodes in the graph). The simulation consists of 15 sensors as in [1].

Aggregation topology and data pattern were triggered to change by sensors misconfiguration or topology attacks. The default lifespan of routes were modified to 180 sec to allow variability, and the simulation time was chosen to be 4 hours for each scenario. Parameters PAR[1-7] defined in section VII-A are the same for all scenarios. Each simulation is started with a different randomly generated seed. Sampling time was chosen to be 100 sec which is adequate given the configured routes lifespan. For visualisation of results in section VIII, PCA dimensionality reduction was used which captured 0.45, 0.26 variance in two components.

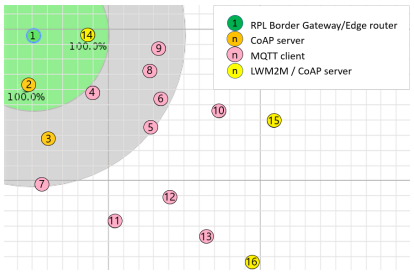

Fig. 2. Simulation

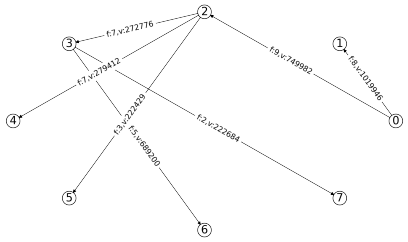

Fig. 3. Baseline aggregation graph

A. Normal operation

The simulated WSN for normal operation is presented in Fig. 2. The nodes were labelled, and matrices A1/A2/B1/B2 were built as discussed in section V.

For normal profiles acquisition, two 4-hour long batches were acquired to build the traffic flows aggregate graph. Alg. 1 was run with the parameters (PAR1=2, PAR2=2, PAR3=no limits, PAR4=MQTT) to extract the normal baseline aggregation graph presented in Fig.3 with data flows/volumes in edge labels and re-numbering nodes from the simulator according to the map P:{1:0, 2:1, 14:2, 4:3, 8:4, 9:5, 5:6, 6:7} to facilitate matrix calculations. The aggregators found are nodes 0(1), 2(14), 3(4). The baseline aggregation graph is used to find the transformations **T1**, **T2** for the baseline topology and data patterns (section IV). As the baseline aggregation graph serves as the comparison base for normal and assessed aggregation profiles, the baseline aggregation graph can also be pre-defined and not based on the normal aggregated traffic flows. For the comparison task to obtain meaningful results, consistent evaluation of assessed samples and the ground truth acquisition is needed by using the same comparison base. The pre-defined baseline aggregation graph is not used in this contribution.

1) Normal aggregation profiles (ground truth): Normal aggregation profiles were acquired (Alg.2) with profile sample parameters (PAR5=10, PAR6=0.01, PAR7=14) and plotted in Fig.5. The orange crosses represent centroid centers. Ground truth is labelled with T*x*G*y* where *x* is aggregation topology number (enumerated hash value), and *y* is the number of profile samples for that topology. The second batch did not

TABLE III
AGGREGATION PROFILE CHANGE (DATA CHANGE)

Sub-scenario (label)	Nodes changing reporting frequency
A (D)	12,13 in Fig. 2
B (D)	12,13 – repeated
C (DD)	8,9

TABLE IV
AGGREGATION PROFILE CHANGE(TOPOLOGY CHANGE)

Sub-scenario (label)	Attack node
A (T)	Node 6 in Fig. 2
B (T)	Node 6 – repeated
C (TT)	Node 9

introduce new topologies, which stopped the acquisition. However, the richer the ground truth, the better estimation of the normal aggregation profiles.

B. Assessed scenarios

1) Normal operation: A batch of data was acquired from the WSN in Fig. 2 for normal operation (scenario label *N*).

2) Misconfiguration: Two MQTT sensors change their reporting frequency from every 30 to 10 sec. Aggregation profile samples (3 batches) were collected for the scenarios in Table III.

3) Topology attack: RPL rank decrease attack is triggered [11]–[14]. Aggregation profile samples (3 batches) were collected for the scenarios in Table IV.

VIII. RESULTS

Aggregation profile data points in Fig.5 are labeled following the format T*x*[*Scenario label*]*y* where *x* is a topology number (enumerated hash value) and *y* available aggregation samples.

1) Aggregation profile change: Similarity vectors $\overrightarrow{s}$ after PCA dimensionality reduction are plotted in Fig.5. The classifier in Fig. 1 classified correctly the assessed profiles (for C=0.025 in (10)). This can be seen as discriminating if the metrics (a fingerprint) acquired from an aggregation profile sample is known in the ground truth. As binary classification result is given, no further information is available to indicate to what extent the anomaly occurred in the system. This is addressed in the relative change measure in (11) for the matched in the ground truth topologies as presented in Fig.6. The relative measure is defined on the interval [0..1], with zero indicating no change and increasing with the anomaly increase. The measure includes both aggregation topology and distribution change, as relative change depends on topology i.e. the number of egress edges in the aggregation graphs (nodes' degree). The higher node's degree, the smaller change incurred for a fix data volume change traversing an edge due to weights normalisation in (1). For these reasons to measure distribution change, the topology component is discriminated by the topology hash values.

Normal aggregation topologies were acquired in the ground truth, and enumerated as T[1-4]. The relative change of normal aggregation profiles for normal topologies are plotted as green dots in Fig.6. As relative measure uses centroid locations (i.e. estimates of maximum local density of samples depicted as orange crosses in Fig.5), and data volumes acquisition in samples have variations caused by the sampling process with parameters PAR[5-7], the relative change of the aggregation profile of the ground truth is defined by ranges e.g. for topologies T1/T3 the normal range is [0..0.02] and [0..0.01] respectively. For topology T2/T4, the ground truth has one aggregation profile sample for each topology. For the batch of assessed normal data, the aggregation topology T1 was found in the ground truth, with the relative change depicted as orange stars in Fig.6 forming overlapping cluster with the ground truth data. Although aggregation topology T2 with the profile was acquired in the ground truth data, it does not occur in the assessed scenarios.

Aggregation profiles relative change for sub-scenarios A, and B in Table III, labeled as scenario D in Fig.5 represents clusters of red triangles in Fig.6 for topologies T1/T3, and one sample for T4. For sub-scenario C in Table III, a different set of sensors located in different part of WSN network changes reporting frequency causing traffic distribution change. This impacts an aggregation profile sample. The sample has topology T3 which is found in the ground truth allowing comparison with with the normal aggregation profile. Comparing sub-scenarios A/B and C, it reveals the latter causes larger relative aggregation profile change of 0.21. The change is not only caused by the change in the reporting frequency but also by the traffic distribution change in WSN. They both impact aggregators' traffic and operations. By locating aggregators impacted by the aggregation profile change, it allows targeted analysis e.g. by querying identified sensors for more information/statistics. The traffic analysis at the gateway alone does not take aggregation topology into account in data analytics, nor the impact of the WSN traffic distribution change on aggregators' nodes.

Locating aggregators impacted by the aggregation profile change requires discriminating vector components change $(d1_n, d3_n)$ in the assessed similarity vectors as compared with the matched ground truth profiles. For the matching, the topology hash is used. To demonstrate, sub-scenarios A/B in Table III for the aggregation topology T3 and T4 are used. Aggregation graphs for topology T3 and T4 are presented in Fig.4. For topology T3, the similarity vector $\overrightarrow{s}$ differs from the ground truth only in $(d1_0, d3_0)$ i.e. for node 0. This indicates the aggregator node 0 and the leaf nodes 1 and 2 are impacted by the aggregation profile change. The change is exacerbated for the same aggregation topology T3 in the sub-scenario C indicated by higher relative change measure (relative change of 0.21 in Fig.6 with assessed aggregation graph traffic volumes in edges e(0,1) = 47373, e(0,2) = 36877). The same nodes are impacted. For topology T4, the similarity vector $\overrightarrow{s}$ only differs in $(d1_0, d3_0)$, making node 0 impacted by the aggregation profile change. However, due to T4 topology structure, and the fact that the aggregator node 2 is not impacted by the change ($(d1_2, d3_2)$ comparable with the ground truth values), it is inferred that besides node 0 only the leaf node 1 is impacted. Ranking nodes with the highest vector component change in larger systems, allows grading the impact the aggregation profile change has on aggregators (not done in this contribution).

Aggregation profile topology samples unmatched in the ground truth (i.e. topologies T[5-7]), may result from insufficient data or aggregation profile change. The ground truth should be representative by sufficient acquisition time (data-set size), and low rate of new aggregation profiles discovered, the variability of which is controlled by the

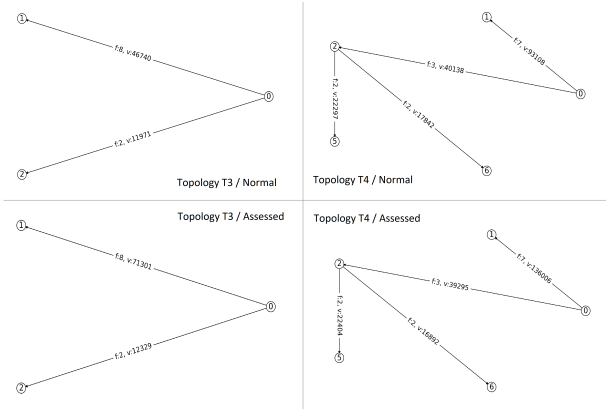

Fig. 4. Aggregation graphs change (scenario D) for topology T3/T4

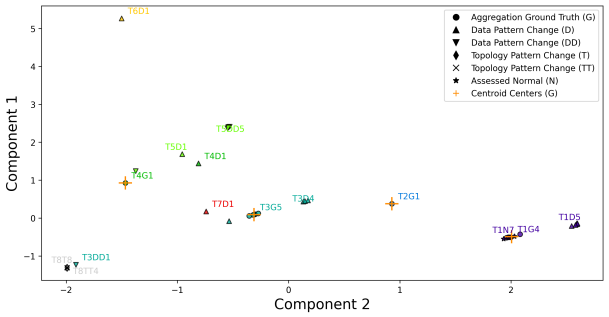

Fig. 5. Aggregation profiles

parameters in Alg. 2. If conservative pruning parameters are used, the profile is less susceptible to change caused by edge devices, discriminating the aggregated traffic better, whereas less conservative for the baseline acquisition, helps to increase the information capture by similarity vectors, and the coverage useful for the partly overlapping sub-views.

However, aggregation topologies are smaller subsets of the topologies in the entire WSN, and due to the traffic aggregation process, they have smaller variability making acquisition of the ground truth easier.

2) Aggregation topology change: Malicious topology change in Table IV caused aggregation profiles change resulting in no edges and vertices meeting selection criteria either because of insufficient number of edges with sufficient flow diversity or no surviving nodes with minimum number of surviving edges as discussed in section V (points T8T8, and T8TT4 in Fig.5). The aggregation topology change shall also be seen in relation to the quality of the ground truth discussed in section VIII-1, as no matching aggregation topology may result from insufficient data in the data-set and not necessarily malicious aggregation topology change.

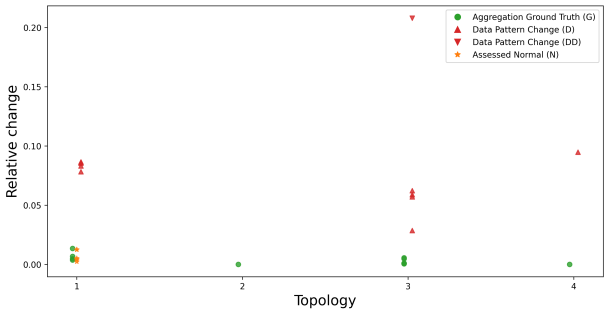

Fig. 6. Aggregation profile change (relative)

IX. Conclusions

A method was proposed to acquire aggregation profiles with the metrics quantifying anomaly. System sub-views analysis also for partly overlapping sub-views facilitated detection of changes occurring elsewhere in the system. The metrics were used in the machine learning algorithms to assess the system for anomaly and measure anomaly level for decision systems. The method is suitable for detecting unknown attacks. The trade-offs between the pruning parameters and detection sensitivity is left for further study.

References

[1] R. Zakrzewski, T. Martin, and G. Oikonomou, "Anomaly detection of data and topology patterns in WSNs," in *Proc. 17th International Conference on Distributed Computing in Sensor Systems*, jul 2021.

[2] M. Hasan, M. M. Islam, M. I. I. Zarif *et al.*, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, sep 2019.

[3] H. Sfar and A. Bouzeghoub, "Activity Recognition for Anomalous Situations Detection," *IRBM*, vol. 39, no. 6, pp. 400–406, dec 2018.

[4] A. Abid, A. Kachouri, and A. Mahfoudhi, "Outlier detection for wireless sensor networks using density-based clustering approach," *IET Wireless Sensor Systems*, vol. 7, no. 4, pp. 83–90, aug 2017.

[5] H. Saeedi Emadi and S. M. Mazinani, "A novel anomaly detection algorithm using DBSCAN and SVM in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 2, pp. 2025–2035, 2018.

[6] F. Y. Yavuz, D. Ünal, and E. Gül, "Deep learning for detection of routing attacks in the internet of things," *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 39–58, nov 2018.

[7] Z. Zhang, A. Mehmood, L. Shu *et al.*, "A survey on fault diagnosis in wireless sensor networks," *IEEE Access*, vol. 6, pp. 11 349–11 364, 2018.

[8] F. Österlind, A. Dunkels, J. Eriksson *et al.*, "Cross-level sensor network simulation with COOJA," in *Proceedings - Conference on Local Computer Networks, LCN*, 2006, pp. 641–648.

[9] "IEEE 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks," 2015.

[10] A. Brandt, J. Hui, R. Kelsey *et al.*, "RPL: IPv6 Routing Protocol for LowPower and Lossy Networks, RFC 6550," pp. 1–314, 2015.

[11] M. A. Boudouaia, A. Ali-Pacha, A. Abouaissa *et al.*, "Security against rank attack in RPL protocol," *IEEE Network*, vol. 34, no. 4, pp. 133–139, jul 2020.

[12] A. R. Jadhao and S. S. Solapure, "Analysis of routing protocol for Low Power and Lossy Networks (RPL) using Cooja simulator," in *Proc. 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, 2018, pp. 2364–2368.

[13] S. Mangelkar, S. N. Dhage, and A. V. Nimkar, "A comparative study on RPL attacks and security solutions," in *Proc. 2017 International Conference on Intelligent Computing and Control, I2C2*, 2018, pp. 1–6.

[14] M. N. Napiah, M. Y. I. Bin Idris, R. Ramli *et al.*, "Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol," *IEEE Access*, vol. 6, pp. 16 623–16 638, jan 2018.