Zakrzewski, R., Martin, T. P., & Oikonomou, G. (2021). Anomaly detection of data and topology patterns in WSNs. In *17th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 535-542). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/DCOSS52077.2021.00087>

Peer reviewed version

Link to published version (if available):
[10.1109/DCOSS52077.2021.00087](https://doi.org/10.1109/DCOSS52077.2021.00087)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the accepted author manuscript (AAM). The final published version (version of record) is available online via IEE at 10.1109/DCOSS52077.2021.00087. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Anomaly detection of data and topology patterns in 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 which makes detection of cyber-attacks or misconfiguration hard. Topology and data patterns change may result from attacks leading to the compromise of data and service availability or indicate operational problems. Graphs are often used to model topology and data paths to describe and compare state of a system. For anomaly detection, the definition of normal patterns, deviation from normal, and criteria when to declare anomaly are required. In this contribution the process of acquisition of normal patterns (ground truth), and criteria when to declare anomaly based on graph comparison are proposed. The anomaly detection is suitable for deployment at the edge of a network. Finally, the inability to define all security threats is addressed by a custom tree-based classifier which only requires normal patterns for training. A simulated wireless sensor network was used to acquire data and apply the method. Our experiments show that data and topology change can be detected at the edge of a network.

Index Terms—Anomaly detection, Machine learning, Sensor network, Data pattern, Topology, Graph, Cyber-security, Fault detection.

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. This typically involves data (also encrypted) and network topology which are subject to attacks such as denial of service, and malicious topology change to facilitate data eavesdropping or may be impacted by faults and miss-configuration. Defining normal topology and data patterns enables guarding the normal state and flagging up any deviations. Motivation for this work is to enable WSN to perform anomaly detection of topology and data also caused by unknown triggers. In this contribution, topology and data patterns are modelled as graphs. To allow detection, graphs adjacency matrix factorisation and topology and data change measurements are proposed. A method allowing acquiring normal data and topology patterns (ground truth) with criteria for detection of deviations from normal are presented. The method allows scaling the graph to reflect computational resources available and can be deployed at edge devices, concentrators, and hubs.

II. RELATED WORK

Anomaly detection in wireless sensor networks may use supervised or unsupervised machine learning algorithms. Typically inputs pre-processing and feature extraction are required to produce meaningful results. The supervised machine learning methods are based on variants of support vector machine, tree based classifiers, or clustering algorithms [1]–[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]–[6] which still might be inadequate for the threats which fit into the assumed normal condition. Neural networks and deep learning was also used for anomaly detection [7], 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. Prior art for WSN fault detection techniques in [8] has many specialised techniques but they lack generality. In this contribution a graph comparison technique suited for directed, labelled, weighted graphs is used to facilitated passive and generic anomaly detection in WSNs. Topology is reflected by graph structure, whereas data pattern by graph structure and weights assigned to edges. The metrics are used as the inputs for a machine learning algorithm and in particular a classifier based on decision trees is proposed which by design requires limited training to classify topology, and data patterns as normal or abnormal. Only normal patterns are used for training, negative samples are implicitly defined as the complement of normal. This feature in particular enables detection of unknown threats. The method can be deployed on edge devices. If meaningful classification results can not be provided, scores are proposed to indicate support in the ground truth for an unknown topology to be malicious.

III. SYSTEM VIEW

In WSN deployments, it is a common approach for sensors to push their data to the cloud with help of MQTT and TCP protocols. Sensors also relay data to/from other sensors to facilitate the distribution of WSNs. This makes the system vulnerable to attacks and miss-configuration with the impact

exacerbated as the impaired devices get closer to the edge router acting as the routing tree root. The graph is built by matching current topology used for routing with the captured MQTT traffic at the border router. In this contribution edges represent data flows; vertices represent devices/sensors. 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. In this contribution the uplink MQTT traffic is of interest. 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 are also possible, e.g. bidirectional flows, other protocol data, or data aggregation schemes. The edges and vertices in the graph can be chosen based on different criteria such as device function and traffic properties, allowing flexibility and customisation.

The current topology for data transfer is acquired at a border router which 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. The topology changes to ensure the highest probability of successful data delivery.

IV. MATHEMATICAL SUMMARY

The weighted, annotated (labelled), directed graphs can be described by 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 .

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

The systems represented by graphs A (baseline, normal) 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 $(\mathbf{A}-\mathbf{B})$. In (2) metric $d1$ is defined for graph comparison whereas metric $d1_{nn}$ is for n^{th} peer vertices comparisons. $\|\cdot\|$ denotes Frobenius norm.

$$\begin{aligned} d1 &= \|\mathbf{V}_A - \mathbf{V}_B\| = \|\mathbf{AT} - \mathbf{BT}\| = \|\mathbf{A}_T - \mathbf{B}_T\| \\ d1_{nn} &= \|\mathbf{a}_{Tn} - \mathbf{b}_{Tn}\| \\ a_{Tn}, b_{Tn} &- n^{\text{th}} \text{ row vector of } \mathbf{A}_T, \text{ and } \mathbf{B}_T \end{aligned} \quad (2)$$

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, normal). The matrix $\mathbf{V}_A$ and $\mathbf{V}_B$ are approximated in the least square sense by using Moore-Penrose pseudo inverse. Measure $d1$ produces values bounded

by the interval $[0..U1]$ with 0 indicating perfect alignment (similarity). $U1$ is the upper bound defined in (3) and its value depends on graph A.

$$\begin{aligned} u1_n &= \max(\{elem | elem = \|\overrightarrow{a_{Tn}} - \overrightarrow{v_k}T\|, \text{ for all } k\}) \\ \overrightarrow{v_k} &- \text{zero vector with 1 at component } k \\ \overrightarrow{a_{Tn}} &- n^{\text{th}} \text{ row vector of } \mathbf{A}_T \\ U1 &= \sum_n u1_n = \sum_n \max(\{u1_{nk}, \text{ for all } k\}) \end{aligned} \quad (3)$$

$d2$ is defined in (4) as cosine similarity measure, where $\mathbf{a}_{Tk}$, $\mathbf{b}_{Tn}$ are $k^{\text{th}}/n^{\text{th}}$ row vectors of matrices $\mathbf{A}_T$, $\mathbf{B}_T$ respectively.

$$\begin{aligned} d2_{kn}(\overrightarrow{a_{Tk}}, \overrightarrow{b_{Tn}}) &= \frac{\overrightarrow{a_{Tk}} \cdot \overrightarrow{b_{Tn}}}{\|\overrightarrow{a_{Tk}}\| \cdot \|\overrightarrow{b_{Tn}}\|} \\ d2 &= \sum_n d2_{nn} \end{aligned} \quad (4)$$

$d2$ is bounded by the interval $[U2..N]$ with N indicating perfect alignment (similarity) where N is the vertex count. $U2$ is the lower bound defined in (5) and its value depends on graph A.

$$\begin{aligned} u2_n &= \min(\{elem | elem = \frac{\overrightarrow{a_{Tn}} \cdot \overrightarrow{v_k}T}{\|\overrightarrow{a_{Tn}}\| \cdot \|\overrightarrow{v_k}T\|}, \text{ for all } k\}) \\ \overrightarrow{v_k} &- \text{zero vector with 1 at component } k \\ \overrightarrow{a_{Tn}} &- n^{\text{th}} \text{ row vector of } \mathbf{A}_T \\ U2 &= \sum_n u2_n = \sum_n \min(\{u2_{nk}, \text{ for all } k\}) \end{aligned} \quad (5)$$

For data pattern also metric $d3$ is defined in (6) bounded by the interval $[0..U3]$

$$\begin{aligned} d3_{kn} &= \arccos(d2_{kn}) \\ d3 &= \sum_n d3_{nn} \\ U3 &= \sum_n \arccos(u2_n) \end{aligned} \quad (6)$$

Given the base graph, metrics $d1$, and $d3$ have the lower bound equal to zero if the base and assessed graphs are the same and increase with the anomaly increase. Local similarity information can be obtained when the corresponding metrics for peer row vectors are compared individually and not included as part of the aggregated measure.

V. NORMAL PATTERN ACQUISITION

Normal topology patterns are modelled as graphs for which metrics ($d1_{nn}, d2_{nn}$) are calculated for n^{th} peer nodes and aggregated to produce a graph level topology measure. Vertex n hash value H_{Vn} is calculated using SHA256 algorithm, $H_{Vn} = \text{SHA256}(d1_{nn}, d2_{nn})$, and aggregated at the graph level in $H_G = \text{SHA256}(H_{V1}, \dots, H_{Vn})$. Normal data patterns are modelled as graphs with edge weights representing normalised traffic volumes acquired while topology pattern remains unchanged. Metrics ($d1$, $d3$) are calculated to produce a monotonically increasing measure at the graph level. $d1$, $d3$ have the lower bound 0 for perfect similarity, likewise $d2$ has the upper bound N for perfect similarity. The opposite bounds

of d1, d2, d3 are dependant on the selected normal graph. If the metrics are used for single peer vertices, it allows anomaly localization. To calculate metrics, weighted adjacency matrices **A1**, **A2** were built:

- To emphasize topology change, pseudo random values were generated (seed=1234, re-initialised for each vertex) from (0..1], normalised to satisfy (1), and used as weights assigned to edges following vertex ordering in **A1**
- To emphasize data change, the volumes of egress data were used to calculate normalized edge weights in **A2**

The steps taken for acquiring the normal patterns is described in Algorithm 1. Some parameters such as “traffic of interest” in this contribution is configured to be MQTT traffic. Normal

Algorithm 1: Normal pattern acquisition

Result: Set of normal data/topology values
 SET data acquisition timeout value;
 SET minimal traffic recording time for valid data;
 SET traffic of interest;
while not Timeout(data acquisition) **do**
 Start traffic recording;
 while Same topology AND Not Timeout(data acquisition) **do**
 Keep recording traffic;
 end
 Stop traffic recording;
 if recorded time > minimal traffic recording **then**
 Calculate matrix A1,A2 for time interval t;
 Save time interval t and matrices A1,A2;
 end
end
 Choose interval k for unchanged topology;
 SET A1:=A1k for the interval k;
 SET A2:=A2k for the interval k;
 Use matrix A1, A2 as the selected normal;
 Calculate and save metrics d1(A1,A1t), d2(A1,A1t),
 d1(A2,A2t), d2(A2,A2t), d3(d2(A2,A2t)), hash H_G
 for all intervals t;
 Use H_G as normal topology pattern;
 Use (d1,d3) for normal data pattern;

pattern acquisition requires two calculations of Moore-Penrose inverse for matrices A1, and A2 ($A1_{inv}$, $A2_{inv}$), which can be done offline and stored locally. Note that Moore-Penrose inverse always exists also for singular matrices. Orthogonalized adjacency matrix A1, A2 can also be stored efficiently as the matrix is sparse having 1s on the diagonal (except for non linearly independent row vectors). Calculation of normal patterns as sets of numbers (H_G , (d1, d3)) can be offloaded and stored locally. However, if the calculations need to be done at the edge, matrix multiplication and singular value decomposition used to calculate Moore-Penrose inverse can be computed efficiently and scaled depending on computational resources available [9] [10] [11]. To allow further analysis, the

time the topology remains unchanged is measured (in samples) and saved.

The acquired normal pattern is for bespoke networks as the pattern captures normality for the current deployment and current operational conditions therefore in principle it should offer better performance than generic solutions.

A. Topology change time analysis

Given a topology pattern, the time t a topology remains unchanged is saved. A weighted average $\bar{t}$, and weighted standard deviation s are calculated for t with the weights being the fraction of the longest time the topology remained unchanged. For samples t the central limit theorem is applied with $N(\bar{t}, s^2)$ describing the distribution of the normal topology change time for the current topology.

B. Ground truth acquisition

Ground truth acquisition requires:

- metrics (d1_{nn}, d2_{nn}) for hash H_G for each topology pattern
- metrics (d1,d3) for each data pattern
- measured $\bar{t}$, s for $N_{Top}(\bar{t}, s^2)$ for each topology pattern

If sufficient number of data pattern points $p(d1,d3)$ are available for a given topology the parameters $\bar{p}$, cov_p for normal distribution $N_{Data|Top}(\bar{p}, \text{cov}_p)$ can be calculated and saved. As d1, d3 are highly correlated with large Pearson correlation coefficient, simpler distribution can also be calculated $N_{Data|Top}(\bar{d1}, s_{d1}^2)$ and saved instead.

To reduce data acquisition uncertainty, samples of poor quality are discarded. This is implemented in Algorithm 1 by imposing minimum data acquisition time.

The criteria to stop data acquisition is based on low new topology discovery rate per batch and low relative change of sample mean and standard deviation.

C. Relative data change estimate

For a given topology, and ground truth data pattern point, the upper and lower bounds for metric d1, d3 can be calculated similar to U1, U3 in (3),(6) except the bounds are expressed as the maximum distance from the ground truth data point p found by a classifier (or $\bar{p}$ if several points found)). Vector $\vec{v}_k$ in (3),(5) is defined if edge k exists for vertex n in the assessed graph. Given the topology, the measure showing relative change of d1 and d3 is defined in (7).

$$\begin{aligned} d1_{rel} &= (d1 - d1_{gtruth}) / u1(d1_{gtruth}) \\ d3_{rel} &= (d3 - d3_{gtruth}) / u3(d3_{gtruth}) \end{aligned} \quad (7)$$

D. Concept drift

if $N_{Data|Top}(\bar{p}, \text{cov}_p)$ or $N_{Data|Top}(\bar{d1}, s_{d1}^2)$ is available for a given topology it can be used to indicate to what extend current data patterns is consistent with the ground truth by calculating CDF(d1, d2) or CDF(d1) for the latter case. Lower value indicate larger inconsistency and if inconsistency persists over time it may indicate data concept drift. When the distribution is not available, the persistent change of relative data measure in section V-C) is used with larger value indicating larger inconsistency. Concept drift is not further addressed in this contribution.

VI. SYSTEM SUPERVISION

System supervision is similar to normal patterns acquisition i.e. the selected normal pattern is used for the assessment by calculating metrics $d1$, $d2$, $d3$ for time intervals. The only difference is that calculated metrics ($d1, d3$), and H_G are the input features to a trained classifier. To compute metrics $d1$, $d2$, $d3$, vertex vector representations are calculated by multiplying the corresponding row vectors from the assessed graph adjacency matrices $B1, B2$ (built similarly to $A1, A2$ in section V) by $A1_{inv}$, and $A2_{inv}$ and then aggregating for all vertices. This can be scaled and calculated efficiently in hardware [9]. Note that $A1_{inv}$, $A2_{inv}$, the transformed $A1_T$, $A2_T$ were stored locally. $A1_T$, $A2_T$ are sparse having small memory footprint. The time the topology remains unchanged is also measured to allow further analysis if the classifier is unable to make the decision.

VII. ANOMALY DETECTION

This contribution uses a classifier trained with positive(normal) samples for anomaly detection (i.e. the decision regions falling outside normal regions are treated as anomaly). A tree based classifier is used for topology and data patterns detection. Limited training is required as the classifier uses topology points (hash values H_G), and data patterns ranges ($d1 \pm \text{margin}_{d1}$, $d3 \pm \text{margin}_{d3}$) or $(\overline{d1} \pm k_{d1}, \overline{d3} \pm k_{d3})$, where (k_{d1}, k_{d3}) are points on ellipse separating 95% prediction confidence). The latter is applied if data distributions are available (not used in this contribution). 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 structure of the classifier is presented in Fig. 1. The proposed tree based classification is fast and efficient requiring the maximum $(m+n/m+4)$ comparisons per classification task.

The margin is used when data pattern points are scarce for a given topology. Each data point with the margin must have sufficient coverage for classification which is specified by the coverage ratio C . For a given normal topology, the margin is defined in (8) in relation to the upper bounds $U1, U3$.

$$\text{margin}_{d1} = \frac{\sqrt{C}}{2}U1, \text{margin}_{d3} = \frac{\sqrt{C}}{2}U3 \quad (8)$$

The ground truth does not need to have data patterns for all topologies (hard to obtain in practice). At least one data pattern matching the currently measured topology is sufficient for assessment.

A. Anomaly detection(unknown topology)

If the measured topology is unknown, the classifier fails to make decision for topology and data patterns. However, the time the unknown topology remains unchanged, and the current data pattern can still be measured. As a data pattern carries both topology, and data information, it can be used to find the closest known data patterns recorded in the database (ground truth). The euclidean distance is used as a distance measure. As each known data pattern is associated with the

Fig. 1. Tree based classifier

topology, for which sample mean $\bar{t}$ and sample standard deviation s are known, they can be used as approximation of the parameters for the normal distribution associated with the unknown topology. For the ground truth containing fewer samples, aggregating means and variances from k -closest data points can be used assuming independence and averaging. In this contribution two closest points were used (i.e. $k=2$). The scores defined in (9) use cumulative distribution function $\Theta(t)$ for normal distribution $\mathcal{N}(\bar{t}, s^2)$. The scores are bounded by $[0..1]$. The higher the scores the more support for the events of interest to be topology ripple, topology attack or stable topology. Topology ripples happen when e.g. radio propagation change affects a node which is to be propagated through all the affected nodes until the system reaches a stable topology state. The more affected nodes the shorter the time that the topology remains unchanged as the change propagates through affected nodes gradually. This is reflected in $score_{ripple}$. On the contrary, topology attacks such as rank decrease attack reduces topology change volatility of a stable network which is captured in $score_{attack}$. $score_{stable}$ indicates how much support there is in the ground truth for a new unseen topology be part of the stable networks.

$$\begin{aligned} score_{attack}(t) &= \begin{cases} -1 + 2\Theta(t), & \text{if } t \geq \bar{t} \\ 0, & \text{otherwise} \end{cases} \\ score_{ripple}(t) &= \begin{cases} 1 - 2\Theta(t), & \text{if } t \leq \bar{t} \\ 0, & \text{otherwise} \end{cases} \\ score_{stable}(t) &= \begin{cases} 1 - score_{attack}(t), & \text{if } t \geq \bar{t} \\ 1 - score_{ripple}(t), & \text{otherwise} \end{cases} \end{aligned} \quad (9)$$

B. Data labelling (normal and abnormal patterns)

The normal patterns were acquired for topology, data and labelled respectively. Normal topology patterns are discriminative, whereas normal data patterns are data points ($d1, d3$) with margins. As metrics $d1$ and $d3$ are bounded, the margin defines the coverage ratio each data point has which is reflected in the classifier parameter C equal 0.00005.

Main challenge when using supervised learning algorithms is class imbalance, i.e. inability to generate sufficiently representative and diverse attack vectors. This impacts classification

TABLE I
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)

results adversely making classifiers unfit for anomalies absent in the training data. This is addressed by a classifier trained only with positive(normal) samples. Negative samples also includes unknown threats and are expressed mathematically in (10) as the complement of normal patterns.

$$\begin{aligned}
 \Omega &= N \cup A \text{ - all patterns set} \\
 N &\text{ - Normal patterns set (positive samples)} \\
 A &\text{ - Abnormal patterns set (negative samples)} \\
 A &= \Omega \setminus (N \text{ +/- margin})
 \end{aligned} \tag{10}$$

The margin is only used for data patterns. It may be associated with data volatility and acquisition uncertainty. Data acquisition uncertainty can be reduced by discarding low quality samples (e.g.insufficient data aggregation time for a given topology) as it is defined in Algorithm 1. The margin can be adjusted to consider long term data drift. As positive samples regions are well defined, it allows the system to infer anomaly given the normal condition and detect unseen attacks.

C. Classifier

The classifier proposed in Fig. 1 is defined for Ω despite requiring only set N for training. The classifier performs classifications for topology and data patterns to assign the extracted graph's features to classes: abnormal/normal/undefined topology, and abnormal/normal/undefined data. The classifier for anomaly detection uses the rules presented in Table I.

VIII. SIMULATION

For demonstration, the method was applied to a wireless sensor network simulated in Cooja [12], a simulator for IEEE 802.15.4 [13] 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 paths [14]. Topologies currently used for data transfer were acquired from the routing tables at the edge (border) router. The system architecture presented in Fig. 2 consists of 15 sensors. The chosen number of sensors is suitable to demonstrate how the method can be applied. Applicability of the method for larger networks is not impaired as the system can be broken down to smaller sub-systems as required and each analyzed separately. The system was attacked by malicious devices causing topology and data pattern changes. Since typical deployment is assumed to be stationary, and the default lifespan of routes were modified to 180 sec to allow variability, the simulation time was chosen to be 4 hours for all scenarios. Each simulation is started with a different randomly generated

Fig. 2. Simulation (e2e architecture)

Fig. 3. Simulation (normal operation)

seed. Sampling time was chosen to be 100 sec which is adequate given configured routes lifespan. The current routing topology was captured at the border router for each sampling interval, with MQTT traffic aggregated over the interval (see also section III). For topology patterns visualisation, metrics d1, d2 are used in some figures instead topology hash values to improve presentation.

A. Normal operation (baseline)

The simulated WSN for normal operation is presented in Fig. 3. The nodes were labelled, and matrices A1/A2 were built as discussed in section V. The same labelling order is used for the assessed system.

For normal pattern acquisition, three 4-hour long batches were acquired. The normal topology was selected from the first batch at the sampling instance 40 due to the long duration when topology has not changed (any topology instance from the sampling instances [1..54] can be selected – see Fig. 4). The normal data pattern was aggregated for the sampling intervals when the selected normal topology remained unchanged. To detect topology change, the graphs acquired in consecutive sampling instances s_t are compared by calculating $\|A1(s_{t_n}) - A1(s_{t_{n-1}})\|$. The value different from 0 indicates topology change. The selected normal topology graph is compared with the graphs acquired after each topology change, and a topology change measure, the metrics (d1, d2), and the hash value H_G are calculated. To produce data pattern change measure, data volumes are accumulated for the sampling instances when topology remains the same and compared with the selected normal data pattern by calculating the metrics (d1, d3). For illustration, for batch1, normal topology change,

Fig. 4. Topology change (normal)

Fig. 5. Enumerated topology hash change vs. selected base (normal)

and normal topology measure are presented in Fig. 4, and 5 respectively. The ground truth from batches 1-2, with batch 3 treated as new data (the points labelled with sample and topology number) for topology and data patterns change are presented in Fig. 6 and 7. As batch 3 did not provide any new topology nor any new data patterns, the ground truth acquisition stopped. The ground truth aggregated from batches 1-3 will be used in this contribution. However, as discussed in section V-B, the ground truth acquisition with richer data-sets provides better reflection of the normal condition and produces more accurate scores.

B. Malicious topology change (topology attack)

For the simulated WSN presented in Fig. 3, RPL rank decrease attack was triggered [15]–[18]. Graphs samples were collected for the scenarios summarized in Table II.

Fig. 8. Data patterns regions

TABLE IV
GRAPH SAMPLE SIZES

Scenario	Sub-scenario	Graph samples
Topology pattern change	A	9
	B	4
	C	6
Data pattern change	A	4
	B	9
	C	9

equals at least 10 sampling instances, data acquisition uncertainty has little impact on data pattern (e.g. data pattern point D3, D6, D8 for topology T4 in Fig. 7).

A. Topology attacks

For topology attacks, sub-scenario A will be discussed in detail with the results provided for all sub-scenarios from Table II. Topology patterns change are presented in Fig. 9. Topology numbers greater than 13 are unknown in the ground truth. Data patterns with classifier decision regions are presented in Fig. 10. The topology and data pattern for the first sample A0 is known. This can be explained by the network formation stage when malicious node is not yet part of the system. For sample A0, the classifier detects known topology and data patterns (data points D9, D12 for topology T7 in Fig. 10). Subsequent samples are unknown which for rich ground truth data-set indicates malicious topology. However for smaller data-sets it may indicate insufficient data in the ground truth. Therefore in either case further analysis is desirable, the results of which are presented in Table V. For sub-scenario A, all samples A1-A6 were classified as topology ripple and A7-A8 as topology attack with scores indicating support in the ground truth for the topology ripple, attack, or unknown stable topology (the higher the score, the more support).

B. Data pattern change (miss-configuration)

For data patterns change, sub-scenario A will be discussed in detail with the results provided for all sub-scenarios from Table III. Topology patterns change are presented in Fig. 11. Data patterns with classifier decision regions are presented in Fig. 12. Data pattern measurements are only possible for the known topology. For sub-scenario A, the classifier detected known topology for samples A0 (topology T7) and

Fig. 9. Topology attack(topology patterns change)

Fig. 10. Topology attack(data patterns with classifier regions)

A1 (topology T1) with the associated data points D9T7, D12T7, and D0T1 respectively. The known data points are used to calculate relative change of d1, d3 metrics as discussed in section V-C. The results are presented in Table VI. The relative data measures are not comparable between different topologies. However, for the same topology as for example for graph samples C2, C4, the relative measures are similar with the difference attributed in this case to data acquisition time (36 sampling instances for C2, 29 sampling instances for C4).

TABLE V
MALICIOUS TOPOLOGY PATTERN CHANGE(RESULTS)

Graph sample	Topology		$Score_{Stable}$
	$Score_{Attack}$	$Score_{Ripple}$	
A0	-	-	-
A1	0	0.95	0.05
A2	0	0.99	0.01
A3	0	0.87	0.13
A4	0	0.22	0.78
A5	0	0.95	0.05
A6	0	0.88	0.12
A7	1	0	0
A8	1	0	0
B0	0	0.63	0.37
B1	0.82	0	0.18
B2	0	0.29	0.71
B3	1	0	0
C0	0	0.98	0.02
C1	1	0	0
C2	0	0.88	0.12
C3	0.65	0	0.35
C4	0	0.22	0.78
C5	0.02	0	0.98

Fig. 11. Data patterns change(topology)

Fig. 12. Data patterns change(data points)

X. CONCLUSIONS

A method was proposed to acquire normal topology and data patterns with the metrics to quantify anomaly. Topology and data patterns are expressed in relation to the selected baseline normal patterns. For new topology patterns, scores defining support in the ground truth for topology ripple, anomaly or unknown stable topology were defined. The method is termed as a “black box” approach as the anomaly cause may be triggered by various means such as routing attacks, radio jamming, faults, miss-configurations. The metrics were used as the inputs to the machine learning algorithm to assess whether anomaly occurred in the system. A common problem of class imbalance was addressed by proposing a tree based classifier trained with positive samples which has small training and execution footprint. The method is flexible as the

selection of vertices of the graphs may reflect geographical proximity, function (e.g. hubs, edge devices) or topology. Also the size of the graph can be varied depending on computational resources available, and the assessed traffic may be filtered for specific protocol or direction. The method is suitable for detecting attacks unknown at the time when normal patterns were acquired. The drift of data patterns, the analysis of threats occurring outside the graph but having impact on traffic flows and topology, and the comparison of partly overlapping systems are left for future work.

REFERENCES

- [1] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, “Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches,” *Internet of Things*, vol. 7, p. 100059, sep 2019.
- [2] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, “Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 518–533, sep 2010.
- [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] M. A. Rassam, A. Zainal, and M. A. Maarof, “An Efficient Distributed Anomaly Detection Model for Wireless Sensor Networks,” *AASRI Procedia*, vol. 5, pp. 9–14, jan 2013.
- [6] 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.
- [7] 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.
- [8] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, “A survey on fault diagnosis in wireless sensor networks,” *IEEE Access*, vol. 6, pp. 11 349–11 364, 2018.
- [9] V. Demchik, M. Bačák, and S. Bordag, “Out-of-core singular value decomposition,” jul 2019. [Online]. Available: <http://arxiv.org/abs/1907.06470>
- [10] Y. Nakatsukasa, Z. Bai, and F. Gygi, “Optimizing halley’s iteration for computing the matrix polar decomposition,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2700–2720, sep 2010.
- [11] P. Blanchard, M. Zounon, J. Dongarra, and N. Higham, “D2.9 Novel SVD Algorithms,” EU Horizon 2020 (NLAFFET), pp. 1–29, 2019.
- [12] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-level sensor network simulation with COOJA,” in *Proceedings - Conference on Local Computer Networks, LCN*, 2006, pp. 641–648.
- [13] “IEEE 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks,” 2015.
- [14] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and R. Alexander, “RPL: IPv6 Routing Protocol for LowPower and Lossy Networks, RFC 6550,” pp. 1–314, 2015.
- [15] M. A. Boudouaia, A. Ali-Pacha, A. Abouaissa, and P. Lorenz, “Security against rank attack in RPL protocol,” *IEEE Network*, vol. 34, no. 4, pp. 133–139, jul 2020.
- [16] A. R. Jadhao and S. S. Solapur, “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.
- [17] 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.
- [18] M. N. Napiyah, M. Y. I. Bin Idris, R. Ramli, and I. Ahmedy, “Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol,” *IEEE Access*, vol. 6, pp. 16 623–16 638, jan 2018.

TABLE VI
DATA PATTERN CHANGE(RESULTS)

Graph sample	Classifier		Topology	Data change [%]	
	Top	Data		D1	D3
A0	T_N	D_A	T7	8	0.27
A1	T_N	D_A	T1	19	8.24
B2	T_N	D_A	T12	24	10.51
B8	T_N	D_A	T13	24	11.16
C0	T_N	D_A	T1	5	0.58
C1	T_N	D_A	T13	5	7.79
C2	T_N	D_A	T2	3	6.59
C4	T_N	D_A	T2	2	6.24
C5	T_N	D_A	T12	6	0.46
C8	T_N	D_A	T11	12	12.71