One-Class Anomaly Detection for Iot Sensor Telemetry in the TON_Iot Dataset

Reginald Ekene Ogu

Department of Computer Engineering, Federal University of Technology Owerri, Owerri, Imo State, Nigeria

DOI: https://doi.org/10.51244/IJRSI.2026.1311000054

Received: 02 November 2026; Accepted: 07 November 2026; Published: 22 November 2026

ABSTRACT

Deployed IoT gateways rarely have labelled attack data, so a detector trained only on normal behaviour is the realistic option for device-level monitoring. This study tests that option on the seven TON_IoT telemetry devices. Five one-class detectors were fitted per device: Mahalanobis distance, isolation forest, one-class SVM, local outlier factor and an autoencoder. Each was trained on the first 70% of normal readings in time order and evaluated on the remaining normal readings and all attack readings. Using sensor values alone, every detector scored an AUROC between 0.48 and 0.51 on five devices, meaning attack-period readings were indistinguishable from normal ones. The GPS tracker reached a detection rate of 0.96, but only at a false-alarm rate of 0.81 on held-out normal data, because its normal readings drift. The weather station gave the only usable signal (AUROC 0.75–0.82), and that signal reflects the different statistical character of its four-minute normal capture. Adding reading-interval features raised the AUROC to 0.97–1.00 on Modbus and weather data, but through a sampling artefact rather than attack behaviour. The results define which sensors are worth monitoring and what a credible telemetry benchmark would need.

Keywords: Anomaly detection, One-class classification, Internet of Things, Intrusion detection, Sensor telemetry

INTRODUCTION

A supervised intrusion detector needs examples of the attacks it is meant to catch. In a laboratory that is easy, since the attacks are launched on purpose and labelled as each attack happens. On a real IoT deployment it is a much harder requirement. The operator of a smart building or a small industrial plant typically has months of normal device data and no labelled attacks at all. Anomaly detection trained on normal data only, also called one-class or novelty detection, is the natural fit for that situation, and it has a long history in security (Pang et al., 2021; Ruff et al., 2021; Chatterjee and Ahmed, 2022).

Most work on one-class IoT detection uses network traffic (Bovenzi et al., 2023; Verkerken et al., 2022; Shi et al., 2024; Russell et al., 2023). Device telemetry, meaning the readings a sensor or controller actually reports, has received less attention, even though it is the data closest to the physical process and the data most likely to remain visible when traffic is encrypted. Surveys of anomaly detection in IoT and sensor networks identify the area as promising but under-evaluated (DeMedeiros et al., 2023; Chatterjee and Ahmed, 2022).

The TON_IoT telemetry files are an obvious place to test the idea (Alsaedi et al., 2020). The files hold readings from seven devices under normal operation and under eight attack types, and the files have been used extensively for supervised detection (Naz et al., 2022, 2023; Poonkuzhal et al., 2023). A supervised score on these files, however, says little about whether a normal-only detector would work. Supervised models can separate classes using any difference between classes, including differences created by how the data were recorded. A one-class detector has to find attacks by noticing that something is unusual relative to normal operation. That is both a harder test and a more honest one.

Pang et al. (2021) and Ruff et al. (2021) surveyed deep and shallow anomaly detection and stressed that the "normal" class must be modelled carefully if detectors are to be trusted. Classical methods remain competitive: isolation forest (Liu et al., 2008), the one-class support vector machine (Schölkopf et al., 2001) and the local outlier factor (Breunig et al., 2000) are still used as baselines in recent IoT studies. Russell et al. (2023) compared these three on IoT network data and found the local outlier factor strongest when trained on benign data only. Shi et al. (2024) used isolation forest and a deep autoencoder trained on benign NetFlow records from IoT-23. Bovenzi et al. (2023) compared deep anomaly detectors for IoT traffic and examined the robustness of those detectors, and Verkerken et al. (2022) studied how unsupervised detectors generalise across datasets. Deep anomaly-based detectors have also been proposed for IoT traffic, usually evaluated with labelled data from the same capture (Saba et al., 2022).

Surveys of IoT anomaly detection list many applications to sensor data, from fault detection to intrusion detection, but note a shortage of benchmarks in which normal and anomalous conditions are recorded comparably (DeMedeiros et al., 2023; Chatterjee and Ahmed, 2022). Work on industrial control systems makes a similar point about process data (Umer et al., 2022; Koay et al., 2023).

Alsaedi et al. (2020) described the TON_IoT telemetry testbed, in which IoT and IIoT services published readings through MQTT while attacks were launched from Kali Linux machines. The data have mostly been used for supervised classification, with ensemble methods per sensor (Naz et al., 2022, 2023) and transfer learning across devices (Poonkuzhal et al., 2023). Broader reviews include TON_IoT among the main IoT benchmarks (Haque et al., 2023; Heidari and Jabraeil Jamali, 2023; Gyamfi and Jurcut, 2022), together with Edge-IIoTset (Ferrag et al., 2022), X-IIoTID (Al-Hawawreh et al., 2022) and CICIoT2023 (Neto et al., 2023). Booij et al. (2022) analysed the heterogeneity of the TON_IoT features and labels.

Kapoor and Narayanan (2023), Whalen et al. (2022) and Rosenblatt et al. (2024) each showed how evaluation choices inflate results in the respective fields. Arp et al. (2024) and Apruzzese et al. (2023b) listed the security-specific versions of these problems, including sampling bias and unrealistic evaluation settings. D'hooge et al. (2022) showed that metadata features contaminate network detectors. For time-ordered data, Hewamalage et al. (2023) and Yang and Shami (2022) argued for evaluation that respects temporal order and allows for drift. One-class detection is particularly sensitive to drift, because anything that changes over time, attack or not, looks anomalous.

This paper therefore asks a deliberately narrow question: if a gateway learned each TON_IoT device's normal behaviour and nothing else, would attack-period readings stand out? The objectives of the study are (i) to evaluate five one-class detectors on each of the seven TON_IoT telemetry devices with chronological training and held-out normal data, so that false alarms caused by normal drift are measured rather than hidden; (ii) to establish whether attack-period sensor readings can be distinguished from normal ones on each device; (iii) to explain the devices on which detection appears to work; and (iv) to test whether reading-interval features add genuine attack evidence or merely reflect how the data were sampled.

MATERIALS AND METHODS

2.1 Dataset and features

The seven TON_IoT telemetry train–test files (Moustafa, 2019) were used: fridge, GPS tracker, garage door, Modbus controller, motion light, thermostat and weather station. Each holds 15,000 normal readings and between 16,106 and 24,944 attack readings (Table 1). Two feature sets were built. *Sensor values* are the reported quantities: temperature, coordinates, register values, and binary or categorical states (one-hot encoded after harmonising inconsistent spellings). *Acquisition features* add the time since the previous reading and the number of readings sharing the same timestamp. These features describe how data arrived rather than what was measured.

2.2 Evaluation protocol

Fig. 1 shows the protocol. For each device, normal readings were sorted by time. The first 70% (10,500 readings) were used to fit the detector and set its threshold. The last 30% (4,500 readings) were held out as normal test data, and all attack readings formed the anomalous test data. Chronological ordering matters because it exposes drift: if normal behaviour changes over time, held-out normal readings will look anomalous and the false-alarm rate will show it. Inputs were standardised with statistics from the training normal data only.

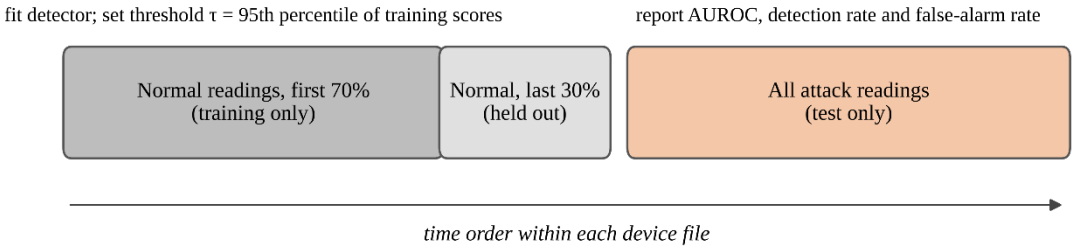

Fig. 1. Evaluation protocol. Each detector sees only the earliest 70% of a device's normal readings. Held-out normal readings measure false alarms; attack readings measure detection.

Table 1. Evaluation sets per device.

Device	Training normal	Held-out normal	Attack readings	Distinct training states
Fridge	10,500	4,500	24,944	151
GPS tracker	10,500	4,500	23,960	10,220
Garage door	10,500	4,500	24,587	2
Modbus	10,500	4,500	16,106	2,628
Motion light	10,500	4,500	24,488	2
Thermostat	10,500	4,500	17,774	9,882
Weather	10,500	4,500	24,260	10,132

2.3 One-class detectors

Five detectors were chosen to span the main families of one-class methods.

Mahalanobis distance. With training mean μ and covariance Σ, the anomaly score of reading x is

$$s_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (1)$$

Isolation forest (Liu et al., 2008). Anomalies are isolated in fewer random splits. With $E[h(x)]$ the mean path length over 200 trees and $c(n)$ the expected path length for n samples,

$$s_{IF}(x) = 2^{-\frac{E[h(x)]}{c(n)}}, \quad c(n) = 2H_{n-1} - \frac{2(n-1)}{n} \quad (2)$$

where H_{n-1} is the harmonic number.

One-class SVM (Schölkopf et al., 2001) with a radial basis kernel and $\nu = 0.05$, scored by the negative decision function.

Local outlier factor (Breunig et al., 2000) in novelty mode with 35 neighbours. The score compares the local reachability density lrd of a point with that of its neighbours $N_k(x)$:

$$LOF_k(x) = \frac{1}{|N_k(x)|} \sum_{o \in N_k(x)} \frac{lrd_k(o)}{lrd_k(x)} \quad (3)$$

Autoencoder. A fully connected network with a narrow bottleneck, trained to reproduce its input, scored by mean squared reconstruction error over the d inputs:

$$s_{AE}(x) = \frac{1}{d} \sum_{j=1}^{d} (x_j - \hat{x}_j)^2 \quad (4)$$

2.4 Thresholds and evaluation metrics

Detection ability independent of threshold was measured by the area under the ROC curve (AUROC), where 0.5 means the scores carry no information. For an operational view, a threshold τ was set at the 95th percentile of scores on the training normal data, the kind of rule a gateway could apply without ever seeing an attack. The detection rate (DR) on attack readings A and the false-alarm rate (FAR) on held-out normal readings N are then

$$DR = \frac{|\{x \in A : s(x) > \tau\}|}{|A|}, \quad FAR = \frac{|\{x \in N : s(x) > \tau\}|}{|N|} \quad (5)$$

With no drift, FAR should sit near 0.05 by construction. A FAR well above that means the held-out normal data have moved away from the training normal data. A detector that flags readings at random would have DR ≈ FAR, so the useful quantity is the gap between DR and FAR. Per-class AUROC was also computed for each attack type against the held-out normal data.

RESULTS AND DISCUSSION

3.1 Five devices: attacks look exactly like normal operation

Fig. 2(a) gives the AUROC of every detector on sensor values. For the fridge, garage door, Modbus controller, motion light and thermostat, all five detectors scored between 0.48 and 0.51. Every one of these detectors is at chance. Fig. 3(a) shows the operational consequence: for these devices DR and FAR sit together on the diagonal, mostly between 0.03 and 0.12, so whatever is flagged is flagged at random.

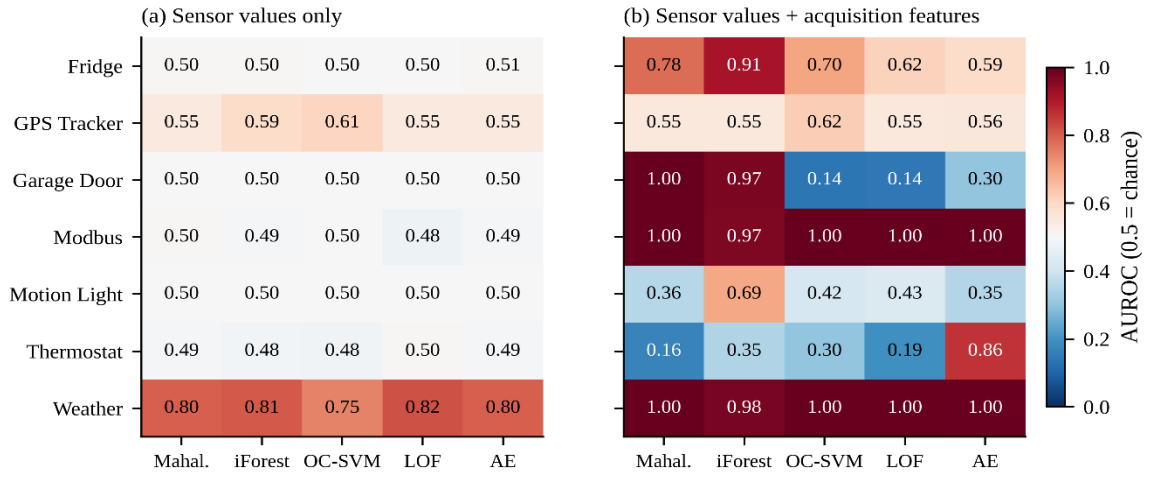

Fig. 2. AUROC of each one-class detector on each device using (a) sensor values only and (b) sensor values plus acquisition features. Values near 0.5 (white) carry no information; values below 0.5 are inverted.

For the garage door and motion light the result was fixed in advance. Each device has only two distinct training states (door closed or open; light on or off), and both states occur during normal operation and during every attack. A detector trained on normal data learns that both states are normal, and it is right. DR and FAR were both exactly zero at the 95th-percentile threshold. The fridge has 151 states, and its temperature and condition distribution is the same in every class.

The Modbus controller and thermostat are more interesting, because the readings of these two devices are rich: 2,628 and 9,882 distinct training states respectively. Even so, attack-period values fall within the range of normal values. According to the dataset description, the attacks targeted web applications, IoT gateways and computer systems around the devices (Moustafa, 2021; Alsaedi et al., 2020) rather than the values the devices reported, so it is not surprising that the readings were unaffected. The point is still worth stating: a value-based monitor on these devices would see nothing.

3.2 The GPS tracker: detection bought with false alarms

On the GPS tracker every detector caught about 96% of attack readings at the 95th-percentile threshold, which looks like success. The same detectors flagged 80–81% of the held-out normal readings, and the AUROC was only 0.55–0.61. The explanation is in Fig. 5(b). GPS readings move continuously, so the later normal readings had already left the region covered by the earlier ones by the time the attack period began. Every later reading, attack or not, therefore looks anomalous. A gateway running this detector would alarm on four normal readings in five.

It is worth noting that the coordinate values in this file extend well beyond the physical range of latitude (up to about 115 for normal readings and 549 during attacks). The data are best understood as simulated position streams rather than real geographic coordinates. That is acceptable for a testbed, but it limits what a detector could learn about realistic movement.

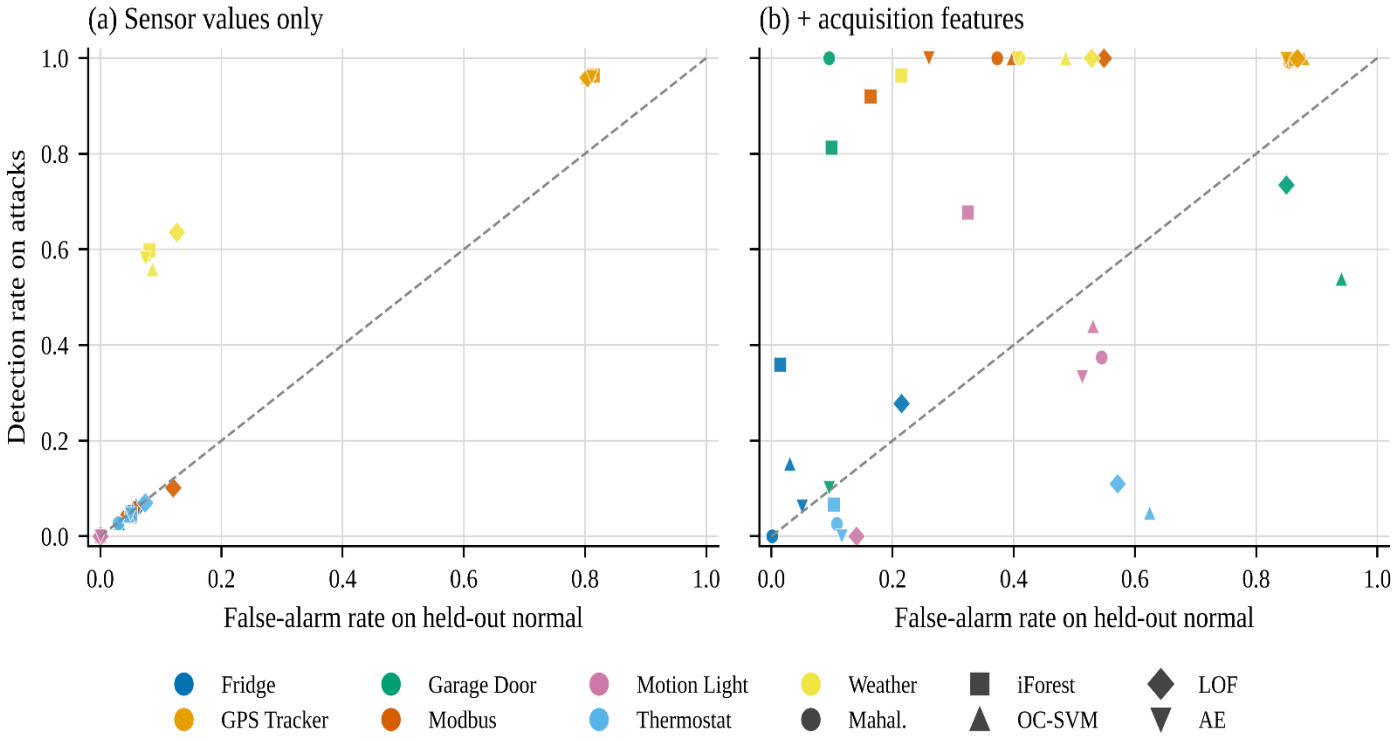

Fig. 3. Detection rate against false-alarm rate on held-out normal data for every device and detector, using (a) sensor values only and (b) sensor values plus acquisition features. Points on the dashed diagonal are no better than random flagging.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue XI November 2026

3.3 The weather station: a signal, but not from the attacks

The weather station was the only device where sensor values gave a usable separation, with AUROC between 0.75 and 0.82 and detection rates of 0.56–0.64 at false-alarm rates of 0.07–0.13. Fig. 4 shows that the separation was uneven across attack types: scanning-period readings were separated perfectly (per-class AUROC 1.00), injection well (0.93), and ransomware and XSS only weakly (0.72 and 0.75).



Fig. 4. Best per-class AUROC across the five detectors, using sensor values only. Dashes mark attack types absent from a device.

Fig. 5(a) suggests why. The weather station's 15,000 normal readings were recorded within about four minutes and are spread fairly evenly across the full temperature range of roughly 20 to 50 units. During the attack period, readings in any short stretch of time occupy a narrower band that shifts as the days pass. A detector trained on the normal capture therefore picks out readings whose joint temperature, pressure and humidity values fall in regions the four-minute capture happened not to cover. That depends on when an attack ran, not on what it did. The perfect separation of scanning, the earliest attack, is consistent with this reading. The GPS tracker shows the same perfect per-class score for scanning.

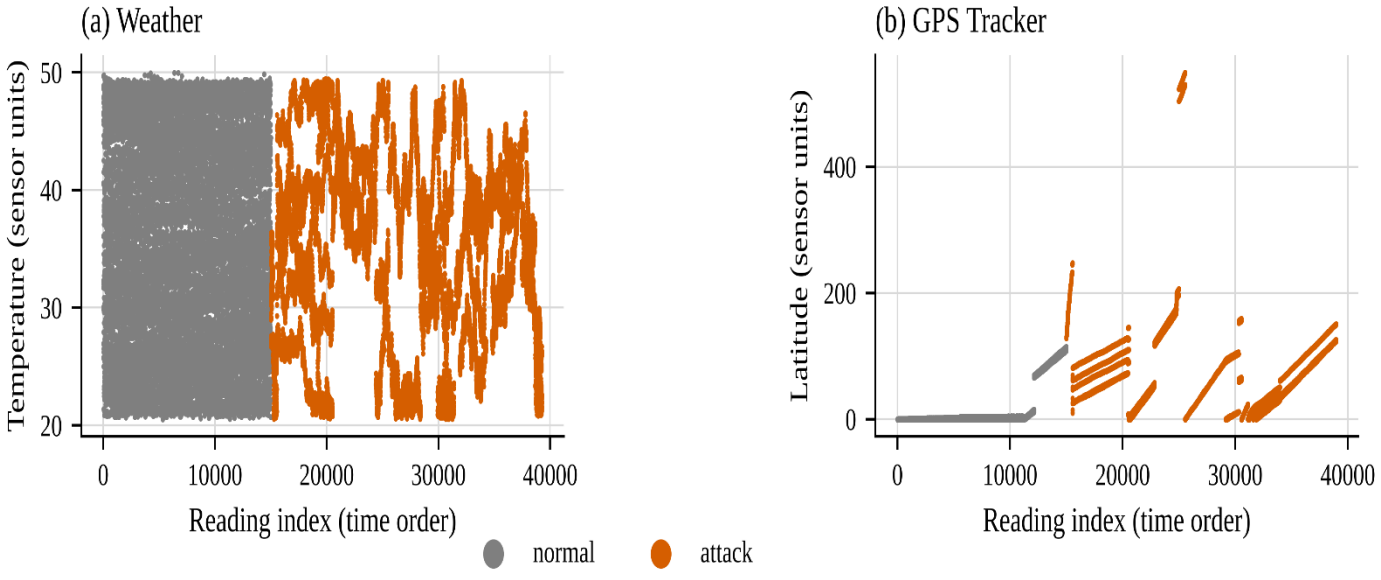

Fig. 5. Sensor values in time order for (a) the weather station and (b) the GPS tracker. Grey points are normal readings; orange points were recorded during attack windows.

3.4 Acquisition features: a shortcut, not a solution

Adding the inter-reading gap and burst count changed the picture sharply (Figs. 2b and 3b). AUROC rose to 0.97–1.00 for the Modbus controller and weather station, and to 1.00 for the Mahalanobis detector on the garage door. These gains come from a sampling artefact. In every TON_IoT telemetry file, normal readings arrive in bursts that share a timestamp, while attack-period readings arrive one to two seconds apart. A normal-only detector learns "several readings per second" as normal and flags anything slower.

The artefact also makes the detectors unstable. On the garage door, three of the five detectors scored AUROC values of 0.14–0.30, far *below* chance, so the scores of those detectors ran in the wrong direction. On the thermostat, four detectors were similarly inverted. False alarms on held-out normal data rose on most devices: on the Modbus controller to 0.16–0.55, on the weather station to 0.21–0.53, and on the GPS tracker to 0.85–0.88. Whether a sampling-rate feature helps or harms depends on accidents of how each capture was organised, which is exactly why such features should not be trusted on this data.

3.5 Which sensors are worth monitoring?

On the evidence of this benchmark, none of the seven devices supports normal-only attack detection from sensor values, and the reasons fall into three groups. Two devices, the garage door and motion light, report binary states that attacks do not change; no detector could use those readings. Three, the fridge, Modbus controller and thermostat, report richer values that attacks in this capture also left alone. Two, the GPS tracker and weather station, produce values that move over time, so a detector trained on an early stretch of normal data mostly detects the passage of time.

The practical message for anyone designing device-level monitoring is that sensor values are only informative when the attack is aimed at the physical process, for example falsified readings, manipulated set-points or injected Modbus writes. Attacks on the surrounding network, the MQTT broker or the web front-end leave the readings untouched, and those were the attacks in this capture. A telemetry benchmark intended for value-based detection needs attacks of the first kind. Work on industrial control systems has made the same distinction between process-level and network-level attacks (Umer et al., 2022; Koay et al., 2023).

3.6 Drift is the main enemy of one-class telemetry detection

The GPS and weather results show that a one-class detector is only as good as its model of normal, and that model goes stale. Here normal data were recorded weeks before the attacks and, for the weather station, over only four minutes. In a deployment the same problem appears whenever seasons change, devices move or firmware is updated. The chronological protocol used here, which measures false alarms on *later* normal data, is a simple way to expose the problem. Random sampling of normal test data would have hidden it completely. Hewamalage et al. (2023) and Yang and Shami (2022) make the same argument for forecasting and IoT analytics in general.

3.7 The value of a negative result

Negative results are under-reported in security machine learning (Arp et al., 2024; Apruzzese et al., 2023a, 2023b). The finding that five of seven devices offer no detectable signal, and that the other two offer signals traceable to the capture process, is useful in two ways. It tells researchers that supervised accuracy on these files should not be read as evidence that the sensors respond to attacks. It also sets a clear target for the next generation of telemetry datasets, in line with wider calls for more careful collection and labelling of security data (Guerra et al., 2022): normal operation recorded throughout the attack period, identical sampling under all conditions, and attacks that touch the physical process.

3.8 Limitations

Only the curated train–test files were analysed. The full TON_IoT telemetry logs may include normal readings recorded during attack windows, which would allow a cleaner test of drift-free detection. Whether the

conclusions hold for telemetry from other testbeds is also open, given how poorly network detectors are known to transfer between datasets (Sarhan et al., 2022; Layeghy and Portmann, 2023). Each detector used one reasonable configuration rather than a tuned one. Tuning cannot rescue an AUROC of 0.50 on two-state devices, but it could change the weather and GPS numbers somewhat. Readings were scored one at a time; window-based or sequence-based detectors might pick up temporal patterns that point-wise detectors miss, although such detectors would face the same drift and sampling confounds. Finally, energy and latency on gateway hardware were not measured, since none of the detectors produced a signal worth deploying (Tekin et al., 2023).

CONCLUSION

A normal-only detector is the realistic choice for monitoring IoT devices in the field, and this study tested it on the seven TON_IoT telemetry streams. On five devices, five standard detectors could not tell attack-period readings from normal ones. On the GPS tracker, apparent detection came with four false alarms in every five normal readings. On the weather station, a moderate signal reflected the different character of a four-minute normal capture rather than the attacks. Adding reading-rate features produced near-perfect scores on some devices and inverted ones on others, all traceable to how the data were sampled. These are negative results, but informative ones: the results show what a device-telemetry benchmark must contain before value-based IoT attack detection can be properly evaluated.

Authorship Contribution

Reginald Ekene Ogu: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing.

FUNDING

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

The TON_IoT datasets analysed in this study are publicly available from IEEE DataPort (Moustafa, 2019; https://doi.org/10.21227/fesz-dm97) and from UNSW Canberra at https://research.unsw.edu.au/projects/toniot-datasets. The analysis scripts used in this study are available from the author on reasonable request.

Declarations

Declaration Of Competing Interest

The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration Of Generative Ai Use

During the preparation of this work, the author used Claude (Anthropic) to assist with data analysis code, figure preparation, drafting and language editing of the text, reference checking, and formatting in the journal template. The author reviewed, verified and edited all content and takes full responsibility for the accuracy, originality and integrity of the manuscript.

Ethics Approval

Not applicable.

REFERENCES

1. Al-Hawawreh, M., Sitnikova, E., Aboutorab, N., 2022. X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things. IEEE Internet of Things Journal, 9(5), 3962–3977. https://doi.org/10.1109/JIOT.2021.3102056
2. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A., 2020. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access, 8, 165130–165150. https://doi.org/10.1109/ACCESS.2020.3022862
3. Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V., Di Franco, F., 2023a. The role of machine learning in cybersecurity. Digital Threats: Research and Practice, 4(1), 1–38. https://doi.org/10.1145/3545574
4. Apruzzese, G., Laskov, P., Schneider, J., 2023b. SoK: Pragmatic assessment of machine learning for network intrusion detection. In: Proceedings of 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P), pp. 592–614. https://doi.org/10.1109/EuroSP57164.2023.00042
5. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K., 2024. Pitfalls in machine learning for computer security. Communications of the ACM, 67(11), 104–112. https://doi.org/10.1145/3643456
6. Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., den Hartog, F.T.H., 2022. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet of Things Journal, 9(1), 485–496. https://doi.org/10.1109/JIOT.2021.3085194
7. Bovenzi, G., Aceto, G., Ciuonzo, D., Montieri, A., Persico, V., Pescapé, A., 2023. Network anomaly detection methods in IoT environments via deep learning: A fair comparison of performance and robustness. Computers & Security, 128, 103167. https://doi.org/10.1016/j.cose.2023.103167
8. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J., 2000. LOF: Identifying density-based local outliers. In: Proceedings of 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104. https://doi.org/10.1145/342009.335388
9. Chatterjee, A., Ahmed, B.S., 2022. IoT anomaly detection methods and applications: A survey. Internet of Things, 19, 100568. https://doi.org/10.1016/j.iot.2022.100568
10. DeMedeiros, K., Hendawi, A., Alvarez, M., 2023. A survey of AI-based anomaly detection in IoT and sensor networks. Sensors, 23(3), 1352. https://doi.org/10.3390/s23031352
11. D'hooge, L., Verkerken, M., Volckaert, B., Wauters, T., De Turck, F., 2022. Establishing the contaminating effect of metadata feature inclusion in machine-learned network intrusion detection models. In: Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2022). Lecture Notes in Computer Science, vol. 13358. Springer, Cham, pp. 23–41. https://doi.org/10.1007/978-3-031-09484-2_2
12. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H., 2022. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281–40306. https://doi.org/10.1109/ACCESS.2022.3165809
13. Guerra, J.L., Catania, C., Veas, E., 2022. Datasets are not enough: Challenges in labeling network traffic. Computers & Security, 120, 102810. https://doi.org/10.1016/j.cose.2022.102810
14. Gyamfi, E., Jurcut, A., 2022. Intrusion detection in Internet of Things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets. Sensors, 22(10), 3744. https://doi.org/10.3390/s22103744
15. Haque, S., El-Moussa, F., Komninos, N., Rajarajan, M., 2023. A systematic review of data-driven attack detection trends in IoT. Sensors, 23(16), 7191. https://doi.org/10.3390/s23167191
16. Heidari, A., Jabraeil Jamali, M.A., 2023. Internet of Things intrusion detection systems: A comprehensive review and future directions. Cluster Computing, 26(6), 3753–3780. https://doi.org/10.1007/s10586-022-03776-z
17. Hewamalage, H., Ackermann, K., Bergmeir, C., 2023. Forecast evaluation for data scientists: Common pitfalls and best practices. Data Mining and Knowledge Discovery, 37(2), 788–832. https://doi.org/10.1007/s10618-022-00894-5
18. Kapoor, S., Narayanan, A., 2023. Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

19. Koay, A.M.Y., Ko, R.K.L., Hettema, H., Radke, K., 2023. Machine learning in industrial control system (ICS) security: Current landscape, opportunities and challenges. Journal of Intelligent Information Systems, 60(2), 377–405. https://doi.org/10.1007/s10844-022-00753-1
20. Layeghy, S., Portmann, M., 2023. Explainable cross-domain evaluation of ML-based network intrusion detection systems. Computers and Electrical Engineering, 108, 108692. https://doi.org/10.1016/j.compeleceng.2023.108692
21. Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation forest. In: Proceedings of 2008 Eighth IEEE International Conference on Data Mining (ICDM), pp. 413–422. https://doi.org/10.1109/ICDM.2008.17
22. Moustafa, N., 2019. ToN_IoT datasets. [Data set]. IEEE Dataport. https://doi.org/10.21227/fesz-dm97
23. Moustafa, N., 2021. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustainable Cities and Society, 72, 102994. https://doi.org/10.1016/j.scs.2021.102994
24. Naz, N., Khan, M.A., Alsuhibany, S.A., Diyan, M., Tan, Z., Khan, M.A., Ahmad, J., 2022. Ensemble learning-based IDS for sensors telemetry data in IoT networks. Mathematical Biosciences and Engineering, 19(10), 10550–10580. https://doi.org/10.3934/mbe.2022493
25. Naz, N., Khan, M.A., Khan, M.A., Khan, M.A., Jan, S.U., Shah, S.A., Arshad, Abbasi, Q.H., Ahmad, J., 2023. A comparison of ensemble learning for intrusion detection in telemetry data. In: Advances on Intelligent Computing and Data Science. Lecture Notes on Data Engineering and Communications Technologies, vol. 179. Springer, Cham, pp. 451–462. https://doi.org/10.1007/978-3-031-36258-3_40
26. Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A., 2023. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors, 23(13), 5941. https://doi.org/10.3390/s23135941
27. Pang, G., Shen, C., Cao, L., van den Hengel, A., 2021. Deep learning for anomaly detection: A review. ACM Computing Surveys, 54(2), 1–38. https://doi.org/10.1145/3439950
28. Poonkuzhal, S., Shobana, M., Jeyalakshmi, J., 2023. A deep transfer learning approach for IoT/IIoT cyber attack detection using telemetry data. Neural Network World, 33(4), 225–244. https://doi.org/10.14311/NNW.2023.33.014
29. Rosenblatt, M., Tejavibulya, L., Jiang, R., Noble, S., Scheinost, D., 2024. Data leakage inflates prediction performance in connectome-based machine learning models. Nature Communications, 15(1), 1829. https://doi.org/10.1038/s41467-024-46150-w
30. Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Müller, K.-R., 2021. A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE, 109(5), 756–795. https://doi.org/10.1109/JPROC.2021.3052449
31. Russell, P., Elsayed, M.A., Nandy, B., Seddigh, N., Zincir-Heywood, N., 2023. On the fence: Anomaly detection in IoT networks. In: Proceedings of NOMS 2023 – IEEE/IFIP Network Operations and Management Symposium, pp. 1–4. https://doi.org/10.1109/NOMS56928.2023.10154271
32. Saba, T., Rehman, A., Sadad, T., Kolivand, H., Bahaj, S.A., 2022. Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering, 99, 107810. https://doi.org/10.1016/j.compeleceng.2022.107810
33. Sarhan, M., Layeghy, S., Portmann, M., 2022. Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection. Big Data Research, 30, 100359. https://doi.org/10.1016/j.bdr.2022.100359
34. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471. https://doi.org/10.1162/089976601750264965
35. Shi, T., McCann, R.A., Huang, Y., Wang, W., Kong, J., 2024. Malware detection for Internet of Things using one-class classification. Sensors, 24(13), 4122. https://doi.org/10.3390/s24134122
36. Tekin, N., Acar, A., Aris, A., Uluagac, A.S., Gungor, V.C., 2023. Energy consumption of on-device machine learning models for IoT intrusion detection. Internet of Things, 21, 100670. https://doi.org/10.1016/j.iot.2022.100670
37. Umer, M.A., Junejo, K.N., Jilani, M.T., Mathur, A.P., 2022. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. International Journal of Critical Infrastructure Protection, 38, 100516. https://doi.org/10.1016/j.ijcip.2022.100516

38. Verkerken, M., D'hooge, L., Wauters, T., Volckaert, B., De Turck, F., 2022. Towards model generalization for intrusion detection: Unsupervised machine learning techniques. Journal of Network and Systems Management, 30(1), 12. https://doi.org/10.1007/s10922-021-09615-7
39. Whalen, S., Schreiber, J., Noble, W.S., Pollard, K.S., 2022. Navigating the pitfalls of applying machine learning in genomics. Nature Reviews Genetics, 23(3), 169–181. https://doi.org/10.1038/s41576-021-00434-9
40. Yang, L., Shami, A., 2022. IoT data analytics in dynamic environments: From an automated machine learning perspective. Engineering Applications of Artificial Intelligence, 116, 105366. https://doi.org/10.1016/j.engappai.2022.105366