J-KOMA Journal of Computer Science and Applications

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Open Access

E-ISSN: 2620 - 4872
Vol.09, **No**.01
Doi: 10.21009/JKOMA.091.07

Recivied: 30 July 2026
Accepted: 30 Jult 2026
Published: 31 July 2026

Keywords:
IoT intrusion detection; data integrity; context shift; RT-IoT2022; Reproducibility.

***Correspondence Email:**
drzainababbas13@gmail.com

Integrity-Aware Evaluation of IoT Intrusion Detection Using Cleaned RT-IoT2022 and Context Holdout Testing

Zainab Abbas Fadhil
Network Department, Faculty of Engineering, Al-Iraqia University, Baghdad, Iraq.

Abstract
Machine learning has significantly improved intrusion detection performance for Internet of Things (IoT) networks; however, reported results are often influenced by hidden data quality issues and limited evaluation protocols. This study investigates the impact of dataset integrity and contextual generalization on IoT intrusion detection using the RT-IoT2022 benchmark. An integrity audit was first conducted to identify duplicate flow records and conflicting class labels before model development. The auditing process removed 5,208 inconsistent observations, producing a conflict-free dataset containing 117,909 network flows. Three baseline machine learning algorithms, namely Logistic Regression, Random Forest, and Extra Trees, were evaluated using stratified five-fold cross-validation under both complete and context-reduced feature profiles. Additional robustness assessments were performed through service-level holdout and attack-family holdout experiments to simulate previously unseen operational conditions. The cleaned dataset produced a Random Forest Macro-F1 score of 0.9968, while reducing context-specific features caused only a marginal performance decrease. Nevertheless, evaluations under withheld services and unseen attack families revealed substantial performance degradation for several traffic contexts, despite maintaining excellent results during conventional validation. These findings demonstrate that outstanding benchmark accuracy alone does not guarantee deployment robustness. The study highlights the importance of dataset integrity auditing, context-aware evaluation, uncertainty estimation, and prevalence-sensitive interpretation to obtain more reliable and reproducible assessments of IoT intrusion detection systems.

INTRODUCTION

IoT devices have moved into the ordinary network. Cameras, message brokers, smart lamps, building sensors, and gateways now operate beside conventional hosts, often with weaker update paths and less uniform administration. Traffic inspection is useful here, though it remains one control among authentication, device trust, segmentation, and patch management [1-4].

Flow logs are easy to use since a gateway can gather them without imposing a large agent on each machine. The evaluation is difficult. Identical logs may span a row partition; there can be different labels for the same vector; port/services can be used as shortcuts; and the next hold-out log may not have the same traffic pattern. These are separate faults and they should not be collapsed into one use of the word leakage [5-9]. Public IoT corpora cover different settings. N-BaIoT follows botnet behaviour, Bot-IoT was built for forensic traffic, TON_IoT joins telemetry and network sources, Edge-IIoTset includes industrial services,

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

CICIoT2023 contains large attack campaigns, and MQTT-IoT-IDS2020 focuses on message-oriented traffic [10-15]. Their percentages are not interchangeable because the capture layout, labels, and predictors differ. Work on RT-IoT2022 has examined compressed autoencoders, ordinary machine-learning comparisons, feature selection, statistical tests, and multiclass detection [15-20]. Table 1 places the present study beside that work without treating unlike protocols as numerical baselines. The objective is to determine how a strong binary result changes after an integrity audit, a reduction in port-and-service context, and complete withholding of selected services or attack families. The contribution is an evaluation protocol rather than a new classifier. It stays within RT-IoT2022 and does not stand in for enterprise-site, chronological, or zero-day validation.

Table 1. Evaluation dimensions in recent RT-IoT2022 studies

Study	Main task	Dimension not central to the cited study	Dimension examined here
Sharmila & Nagapadma (2023a) [16]	Quantized autoencoder	Cross-context stress test	Integrity audit and disjoint holdouts
Airlangga (2024) [17]	Binary model comparison	Predictor-signature conflicts	Conflict-free evaluation set
Sharmila et al. (2024) [19]	Statistical model comparison	Unseen-family normal separation	Disjoint normal pools
Almohaimeed & Albalwy (2024) [18]	Feature selection	Whole-service withholding	Context-reduced profile
Dymora et al. (2026) [20]	Multiclass comparison	Duplicate-fold exposure	Explicit claim boundary

METHOD

The experiment used the RT-IoT2022 file released by the UCI Machine Learning Repository under CC BY 4.0 [21]. It holds 123,117 records, 83 predictors, and one traffic label. MQTT_Publish, Thing_Speak, and Wipro_bulb was mapped to normal traffic; the remaining labels were mapped to attack. Binary mapping matches a gateway escalation decision, but it does not test fine-grained attack diagnosis.

Predictor signature was created using all predictors, except the exported row index and class label. Duplicate refers to a predictor signature which is exactly the same, and no near-duplicate statement was made. Conflicting signature consists of more than one class labels. The 128 observations in the six conflicting groups were dropped, followed by dropping one copy of each repeated signature. The current curation makes use of the full labelled benchmark dataset and is not proposed as an online cleaning step. Rather, an additional sensitivity test included the retention of the majority label in five conflicting signatures, but dropped the one tie.

The analysis included the quantification of exposure under the standard five-fold row partitioning method. Given that D is the total number of copies beyond the first and k as the total number of folds independent of each other, another copy will be different from the original one with probability 1 − 1/k.

$$E_{\text{cross}} = 5202\left(1 - \frac{1}{5}\right) = 4161.6 \approx 4162 \quad (1)$$

The expected cross-fold copy count can be seen from equation (1). The raw split that was used was also tested directly using hashing to check if there were any rows that had the same predictor vector in the training fold. A diagnostic random-forest comparison of the raw and cleaned rows was done. This is where the effect of this split was measured; however, it does not serve as primary evidence since it uses raw folds with known copies.

Table 2. Integrity audit and evaluation design

Item	Value or rule
Raw file	123,117 records; 83 predictors
Duplicate audit	5,202 copies beyond the first (4.23%)

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Item	Value or rule
Label conflict	6 signatures; 128 records
Combined cleaning	5,208 rows removed; 117,909 retained
Clean class balance	105,898 attack; 12,011 normal
Feature profiles	Full: 83; context-reduced: 80
Internal validation	Five stratified folds; split seed 2026; fold-model seeds 2027–2031
Fixed-split copy exposure	4,202 extra copies crossed the anchor fold; 6.59–6.90% of test rows had an exact training copy
Service holdout	At least 500 records and both binary classes
Attack-family holdout	At least 500 attacks; disjoint 70/15/15 normal pools
Conflict sensitivity	Remove all six conflicting signatures versus majority-resolve five non-tied groups; one tied group removed
Uncertainty	1,000 class-stratified bootstrap resamples; Wilson FPR intervals

2.1 Fixed baseline models and internal validation

The full profile kept every predictor. The context-reduced profile removed id.orig_p, id.resp_p, and service while retaining protocol, timing, packet, payload, flag, subflow, active/idle, and window summaries. The name describes the deletion narrowly; it does not imply that every deployment-specific field vanished. Logistic regression supplied a linear baseline. Random forest and extremely randomised trees supplied non-linear ensembles. Each tree model used 150 trees, square-root feature sampling, a minimum leaf size of two, and balanced class weights. Logistic regression used the L-BFGS solver, C=1, balanced weights, and a 2,000-iteration limit. These were fixed baselines, not the outcome of an open-ended parameter search. Imputation, encoding, and scaling were fitted inside each training fold [22-25]. Five-fold stratified validation was run only on the conflict-free, de-duplicated file. Macro-F1 was primary. Balanced accuracy, MCC, false-positive rate (FPR), AUROC, and AUPRC were retained because attack prevalence was high and accuracy alone would hide the normal class [26-28]. Fold means and standard deviations are reported; paired fold differences are descriptive rather than inferential because five folds are not independent trials.

2.2 Withheld contexts and uncertainty

The context-reduced random forest was stressed in two ways. For a service holdout, every record carrying that service was placed in the test set and the model was fitted on the remainder. The source value '-' was labelled Unassigned service, not treated as a protocol. Four services met the size and class requirements. Attack-family tests used a different construction. Normal records were divided once into 70% training, 15% validation, and 15% testing pools. The training set joined normal-training records with every attack family except the one withheld. Testing joined the fixed 1,802 normal-test records with the excluded family. Seven families met the 500-record threshold; the rare SSH brute-force and FIN-scan classes did not. Five random-forest seeds were fitted for each holdout and their probabilities were averaged as a soft-voting analysis ensemble. This was used to reduce dependence on one seed, not proposed as a lightweight deployment architecture. Macro-F1 and recall intervals came from 1,000 class-stratified resamples of the fixed test set. They are conditional, record-level intervals; training-set and traffic-session uncertainty are not folded into them. FPR intervals use the Wilson method on independent normal test records.

2.3 Calibration, prevalence, and reproducibility

A separate clean 70/15/15 stratified split checked probability calibration. Isotonic regression was fitted on validation probabilities only. Expected calibration error used ten equal-width bins; Brier score was retained

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

alongside it. A threshold was selected by validation macro-F1, then compared with the 0.5 rule on the untouched test portion.

To avoid reading benchmark precision as a fixed operational property, positive predictive value was projected over lower attack prevalence. With sensitivity Se, false-positive rate FPR, and assumed prevalence π, the calculation was:

$$PPV(\pi) = \frac{Se \times \pi}{Se \times \pi + FPR \times (1 - \pi)} \quad (2)$$

The projection assumes that sensitivity and FPR remain unchanged when prevalence moves; that assumption is useful for a sensitivity check, not a deployment forecast. The supplied package contains the verified study script, full rerun outputs, split indices, per-seed holdout metrics, confusion counts, pinned libraries, dataset hashes, and figure code. The exact UCI CSV used in the rerun matched the recorded SHA-256 value.

RESULTS AND DISCUSSION

3.1 Integrity exposure and internal behaviour

Raw split results proved that copy crossing was not just an abstract concern. In each of the five folds, 6.59–6.90% of test rows had an exact match to its predictor in the training set. Overall, 4,202 copies additionally crossed the fold of their initial appearance, close to the expected value of 4,161.6 from Equation (1). The diagnostic score was not in the predicted direction: random-forest macro-F1 reached 0.9965 on raw rows and 0.9968 on cleaned ones. The exposure was present, but no observable inflation occurred in the comparison. The former result is still diagnostic as the split is contaminated by definition.

Table 3 provides the clean internal results. While tree-based ensembles stuck near one, the linear benchmark stayed at 0.93 macro-F1. Excluding three context fields decreased random-forest macro-F1 on average by 0.00044 on paired folds and Extra Trees by 0.00046. In one fold, the difference slightly favored the reduced profile, hence it is small, not positive consistently. Conflict-policy sensitivity was even narrower: resolution by majority of five non-conflicted signatures gave the value of macro-F1 0.9965, against 0.9963 for removal of six conflicting signatures.

Table 3. Diagnostic raw-row and clean five-fold validation results

Model	Profile	Macro-F1	Bal. acc.	MCC	FPR
Random Forest	Raw full (diagnostic)	0.9965 ± 0.0006	0.9973 ± 0.0008	0.9930 ± 0.0011	0.0045 ± 0.0016
Logistic Regression	Full	0.9354 ± 0.0024	0.9820 ± 0.0012	0.8774 ± 0.0045	0.0077 ± 0.0014
Logistic Regression	Context-reduced	0.9329 ± 0.0031	0.9809 ± 0.0013	0.8729 ± 0.0055	0.0089 ± 0.0021
Random Forest	Full	0.9968 ± 0.0005	0.9968 ± 0.0009	0.9936 ± 0.0009	0.0057 ± 0.0020
Random Forest	Context-reduced	0.9963 ± 0.0007	0.9966 ± 0.0011	0.9927 ± 0.0014	0.0061 ± 0.0023
Extra Trees	Full	0.9951 ± 0.0008	0.9974 ± 0.0008	0.9902 ± 0.0016	0.0035 ± 0.0018
Extra Trees	Context-reduced	0.9946 ± 0.0008	0.9970 ± 0.0008	0.9893 ± 0.0017	0.0043 ± 0.0016

Clean comparison gives interpolation estimates within a single capture. They match the high percentages seen in previous comparisons done using the RT-IoT2022 approach, but such good agreement here does not determine behavior elsewhere or at other times [17, 19, 20]. The holdouts below take on a tougher question, still within the benchmark range.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

3.2 Complete services outside training

Context of Application was not faulty in one case. HTTP transmitted with minimal error; DNS forgot all attacks. SSL performed the reverse – all attacks were detected, but many legitimate SSL transmissions were flagged as attacks. Attacks accounted for the majority of the Unassigned category, and only 175 legitimate flows existed in this category, so the FPR is uncertain. Thus, Table 4 provides the number of flows in each class and the range of FPRs instead of accuracy.

Table 4. Context-reduced random forest with complete services withheld

Service	Train n	Test A/N	Macro-F1 [95% CI]	Attack recall [95% CI]	FPR [95% CI]
Unassigned service	19,712	98,022/175	0.660 [0.636, 0.682]	0.997 [0.997, 0.998]	0.503 [0.430, 0.576]
SSL	115,253	1,459/1,197	0.865 [0.852, 0.878]	1.000 [1.000, 1.000]	0.284 [0.259, 0.310]
DNS	108,472	5,570/3,867	0.869 [0.863, 0.875]	0.784 [0.774, 0.795]	0.007 [0.005, 0.011]
HTTP	114,620	794/2,495	0.994 [0.991, 0.997]	0.985 [0.976, 0.992]	0.001 [0.000, 0.003]

Both poor FPR examples are more significant than their high attack recall. An approach that identifies the attack category but let's through 25% or 50% of all normal instances to be examined has not transferred well. The statistics of the flow can continue to be useful even if the service is removed, but the composition of the traffic will affect the boundary the trees have learned [29-31].

3.3 Attack families absent from training

Two of the withheld families violated this trend. The UDP scan using Nmap had a recall of 0.354, and ARP poisoning had 0.468. The four scanning/flooding families found in the lower section of Table 5 were all predicted perfectly using the soft-voting prediction method, except for DDoS Slowloris which missed one flow. The test datasets used the same 1,802 normal observations, thus the FPR values are comparable.

Table 5. Disjoint attack-family holdouts with 1,802 normal test records

Withheld family	Train n	Attack n	Macro-F1 [95% CI]	Attack recall [95% CI]	FPR [95% CI]
Nmap UDP scan	111,723	2,582	0.600 [0.586, 0.613]	0.354 [0.335, 0.372]	0.008 [0.005, 0.014]
ARP poisoning	106,685	7,620	0.554 [0.546, 0.561]	0.468 [0.456, 0.479]	0.001 [0.000, 0.004]
DDoS Slowloris	113,773	532	0.990 [0.986, 0.995]	0.998 [0.994, 1.000]	0.008 [0.005, 0.014]
DoS SYN Hping	24,216	90,089	0.998 [0.996, 0.999]	1.000 [1.000, 1.000]	0.009 [0.006, 0.015]
Nmap Xmas tree scan	112,295	2,010	0.996 [0.994, 0.998]	1.000 [1.000, 1.000]	0.008 [0.005, 0.013]
Nmap OS detection	112,305	2,000	0.996 [0.994, 0.998]	1.000 [1.000, 1.000]	0.008 [0.005, 0.013]
Nmap TCP scan	113,303	1,002	0.994 [0.991, 0.997]	1.000 [1.000, 1.000]	0.008 [0.005, 0.014]

Figure 1 combines attack recall with FPR without needing to redraw either table. The low-recall families are separate from the others despite having low error for the normal class. High AUROC does not fix the issue since ranking and a set decision threshold address different questions, and the unseen family could have little in common with the geometry from the other attacks [30, 32].

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

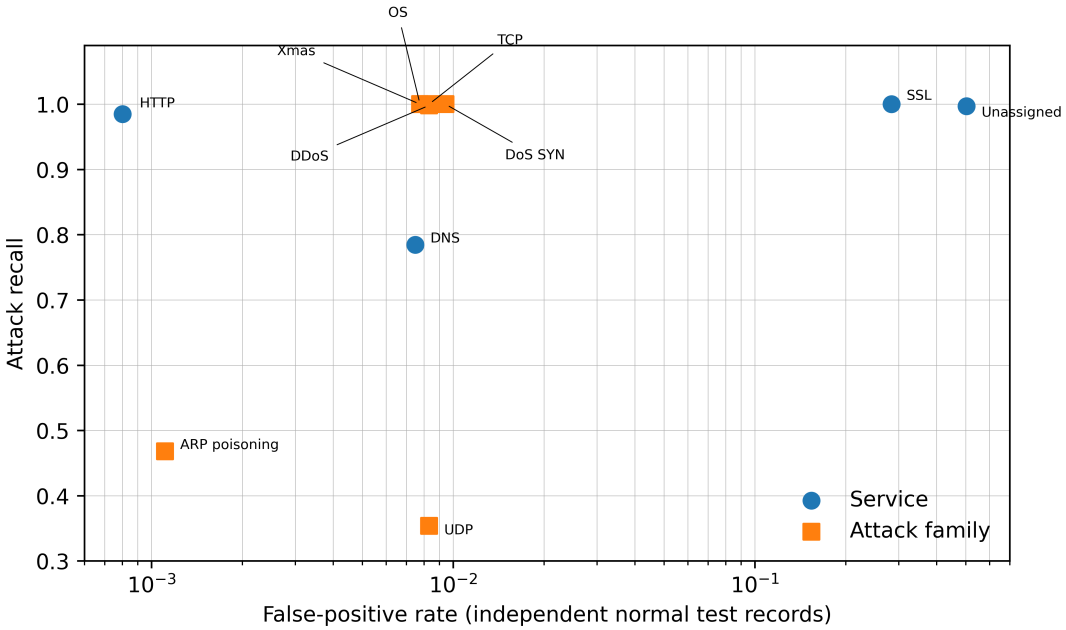
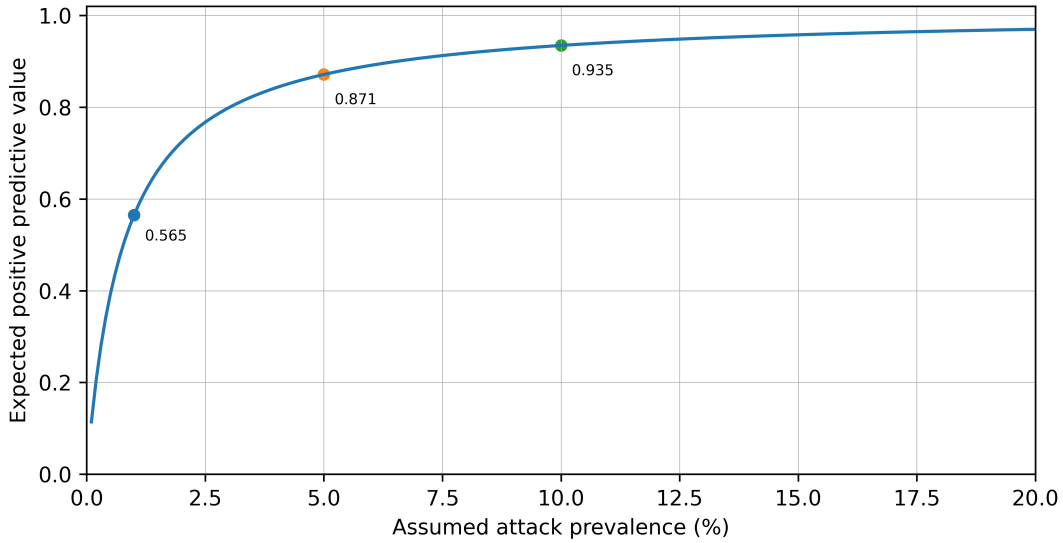


Figure 1. Recall–FPR operating map for withheld services and attack families

3.4 Calibration and low-prevalence reading

Isotonic calibration reduced ten-bin expected calibration error from 0.00126 to 0.00058, whereas Brier score was reduced from 0.000963 to 0.000980. Threshold selected through validation is 0.440. For the non-calibrated test part, both calibrated and uncalibrated thresholds achieved the same F1 score of 0.9963 when compared with uncalibrated threshold of 0.5 which had a F1 score of 0.9960. Prevalence altered interpretation more sharply. Using sensitivity 0.9992 and FPR 0.00777 in Equation (2), expected positive predictive value was 0.565 at 1% attacks, 0.871 at 5%, and 0.935 at 10%. Figure 2 shows the continuous projection. The curve does not predict a particular enterprise network, but it shows why a high benchmark score can still generate a meaningful normal-alert load.



Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Figure 2. Expected positive predictive value under assumed attack prevalence

3.5 Boundaries of the evidence

RT-IoT2022 has no defensible enterprise-site identifier and no chronological field suitable for a site or temporal split. The present holdouts are therefore stress tests inside one testbed. Binary mapping also hides attack-to-attack confusion, and rare SSH brute-force and FIN-scan classes were too small for a stable family holdout. No live throughput, memory demand, analyst workload, or containment action was measured. Cross-dataset and edge-testbed work is still needed before a broader transfer claim can be considered [33-36].

CONCLUSION

This study presented an integrity-aware evaluation framework for machine learning-based IoT intrusion detection using the RT-IoT2022 dataset. An integrity audit identified duplicate predictor signatures and conflicting class labels, resulting in a cleaned dataset that enabled a more reliable assessment of model performance. Experimental results showed that tree-based ensemble models, particularly Random Forest, achieved excellent performance during conventional cross-validation, while removing context-specific features such as source port, destination port, and service produced only a marginal reduction in Macro-F1 score. However, more challenging evaluations revealed that high benchmark accuracy did not necessarily translate into robust real-world performance. Complete service holdout and unseen attack-family experiments exposed substantial degradation for several traffic contexts, especially for SSL, Unassigned traffic, Nmap UDP Scan, and ARP Poisoning, indicating limited generalization under distribution shifts. These findings demonstrate that benchmark evaluation should extend beyond conventional random data partitioning by incorporating integrity auditing, context-aware validation, uncertainty estimation, and prevalence-sensitive performance interpretation. Although the proposed evaluation remains limited to a single benchmark dataset and binary classification, it provides a more transparent and reproducible protocol for assessing IoT intrusion detection systems and establishes a stronger foundation for future cross-dataset and real-world deployment studies.

REFERENCE

[1] Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008

[2] Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. Journal of Network and Computer Applications, 88, 10–28. https://doi.org/10.1016/j.jnca.2017.04.002

[3] Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys & Tutorials, 21(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

[4] Karanja, E. M., Masupe, S., & Jeffrey, M. G. (2021). A survey on Internet of Things security: Requirements, challenges, and solutions. Internet of Things, 14, 100129. https://doi.org/10.1016/j.iot.2019.100129

[5] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. 2010 IEEE Symposium on Security and Privacy. 305–316. https://doi.org/10.1109/SP.2010.25

[6] Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. ACM Transactions on Knowledge Discovery from Data, 6(4), 15. https://doi.org/10.1145/2382577.2382579

[7] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

[8] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, 20. https://doi.org/10.1186/s42400-019-0038-7

[9] Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150. https://doi.org/10.1002/ett.4150

[10] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2018). N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

[11] Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of a realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Future Generation Computer Systems, 100, 779–796. https://doi.org/10.1016/j.future.2019.05.041

[12] 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

[13] Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., & Bellekens, X. (2021). Machine learning based IoT intrusion detection system: An MQTT case study. International Networking Conference, 73–84. https://doi.org/10.1007/978-3-030-64758-2_6

[14] 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

[15] Pinto Neto, E. C., 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

[16] Sharmila, B. S., & Nagapadma, R. (2023a). Quantized autoencoder intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022. Cybersecurity, 6, 41. https://doi.org/10.1186/s42400-023-00178-5

[17] Airlangga, G. (2024). Comparative analysis of machine learning models for intrusion detection in Internet of Things networks using the RT-IoT2022 dataset. MALCOM: Indonesian Journal of Machine Learning and Computer Science, 4(2), 656–662. https://doi.org/10.57152/malcom.v4i2.1304

[18] Almohaimeed, M., & Albalwy, F. (2024). Enhancing IoT network security using feature selection for intrusion detection systems. Applied Sciences, 14(24), 11966. https://doi.org/10.3390/app142411966

[19] Sharmila, B. S., Nandini, B. M., Kavitha, S. S., & Srivatsa, A. (2024). Performance evaluation of parametric and non-parametric machine learning models using statistical analysis for RT-IoT2022 dataset. Journal of Scientific & Industrial Research, 83(8), 864–872. https://doi.org/10.56042/jsir.v83i8.7437

[20] Dymora, P., Mazurek, M., Łukiewicz, K., & Bokota, K. (2026). Comparative analysis of machine learning models for multiclass anomaly detection in IoT network traffic using the RT-IoT2022 dataset. Advances in Science and Technology Research Journal, 20(4), 45–59. https://doi.org/10.12913/22998624/213807

[21] Sharmila, B. S., & Nagapadma, R. (2023b). RT-IoT2022 [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5P338

[22] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20), 4396. https://doi.org/10.3390/app9204396

[23] Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data, 7, 105. https://doi.org/10.1186/s40537-020-00379-6

[24] Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. Journal of Big Data, 7, 41. https://doi.org/10.1186/s40537-020-00318-5

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

[25] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. IEEE Access, 8, 222310–222354. https://doi.org/10.1109/ACCESS.2020.3041951
[26] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239
[27] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432
[28] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21, 6. https://doi.org/10.1186/s12864-019-6413-7
[29] Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. 2018 IEEE Security and Privacy Workshops, 29–35. https://doi.org/10.1109/SPW.2018.00013
[30] Moustafa, N., Turnbull, B., & Choo, K.-K. R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things. IEEE Internet of Things Journal, 6(3), 4815–4830. https://doi.org/10.1109/JIOT.2018.2871719
[31] Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2020). CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet of Things Journal, 8(5), 3242–3254. https://doi.org/10.1109/JIOT.2020.3002255
[32] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2018.23204
[33] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419. https://doi.org/10.1016/j.jisa.2019.102419
[34] Hajiheidari, S., Wakil, K., Badri, M., & Jafari Navimipour, N. (2019). Intrusion detection systems in the Internet of Things: A comprehensive investigation. Computer Networks, 160, 165–191. https://doi.org/10.1016/j.comnet.2019.05.014
[35] Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, 18. https://doi.org/10.1186/s42400-021-00077-7
[36] 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