Device-Specific Anomaly Detection Models for IoT Systems

Shahrzad Golestani
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
shg550@usask.ca

Dwight Makaroff
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada
makaroff@cs.usask.ca

***Abstract*—**

The Internet of Things (IoT) has transformed home automation, industry, and agriculture, yet security remains a major challenge. IoT systems comprise a wide range of devices generating vast and heterogeneous data. This paper investigates device-specific and device-type-specific anomaly detection models, highlighting the potential of leveraging unique traffic patterns from heterogeneous IoT devices. These models are compared to a single model trained on data from all devices, using eight different supervised and unsupervised One-Class Classifier (OCC) methods on two IoT-collected datasets.

Typically, real-world IoT devices generate normal traffic prior to any attack or intrusion. With an abundance of normal traffic and no labelled attack data, unsupervised learning becomes a suitable approach. The findings of this paper show that when using unsupervised methods, device-specific and device-type-specific models outperform single models, particularly when the data is dominated by one class. In this context, device/device-type models can be more effective for real-time anomaly detection by identifying attacks as deviations from the normal profiles established for each device or device type.

***Index Terms*—Internet of Things (IoT), Intrusion Detection System (IDS), Anomaly Detection, Device-based Model, One-Class Classifier (OCC), Unsupervised Learning.**

I. Introduction

The Internet of Things (IoT) aims to connect millions of smart devices worldwide. Number of IoT devices is rapidly increasing each year, but with this increase comes security and privacy challenges. Solutions like Intrusion Detection Systems (IDS) must adapt to IoT's unique characteristics, such as low-computation capabilities and diverse traffic types.

Traditional intrusion detection systems typically use a single model to analyze all IoT traffic for anomalies. In contrast, adopting a "Device-Specific" or "Device-Type-Specific" approach involves creating anomaly detection models for each individual device (e.g., camera A and camera B) or device-type (e.g., cameras and home assistants). The key research question is whether Device-Specific or Device-Type-Specific models improve an IDS's ability to detect anomalies compared to the conventional single model approach.

We build upon the premise that device-specific or device-type-specific models, encapsulating the distinctive traits and behaviours of individual IoT devices or those within specific groups, are likely to exhibit enhanced accuracy in identifying anomalous behaviour. The diversity in behavioural patterns across different devices or device-types is anticipated to contribute to a more effective intrusion detection system tailored to the intricacies of IoT systems.

Throughout the literature, most studies have used a single supervised or unsupervised model for anomaly detection; however, few have focused on device-specific or device-type-specific models in this area. A device-specific model is trained on data from a single IoT device, while a device-type-specific model is trained on collective data from several devices of the same type, such as cameras.

Supervised learning approaches require attack data and labelled datasets, which are costly and labour-intensive to generate. In large real-world IoT systems, devices primarily generate normal traffic. Unsupervised learning can leverage this abundance of normal traffic for anomaly detection using single or device-based models. This paper evaluates the accuracy of both supervised and unsupervised learning methods for anomaly detection using device-based models. We compare the performance of single, device-specific, and device-type-specific models using eight supervised Machine Learning (ML) and Deep Learning (DL) methods with unsupervised OCC methods. OCC methods, trained on normal traffic, detect anomalies by identifying deviations from the normal profile and output binary results, attack or normal. For comparison, supervised methods are implemented for binary classification. We address the following research questions:

1) What is the comparative accuracy difference between a single model for all IoT devices versus utilizing device-specific models tailored to individual devices?
2) Which modelling approach yields better accuracy outcomes: device-specific models tailored for each IoT device or device-type-specific models that apply a model to devices with similar type?

II. Related Work

Fingerprinting device behaviour can be used for both device identification and anomaly detection. Device identification has been extensively studied, but device behaviour fingerprinting for anomaly detection in IoT remains a relatively nascent field.

Only a few studies focused on profiling network traffic using device/device-type models trained on normal device behaviour, including one proposed by Sivanathan *et al.* [1], ComplexIoT [2], and DÏoT [3].

Sivanathan *et al.* [1] proposed a system that classifies IoT devices based on network activity, dynamically adapting to changes like firmware updates. Key traffic attributes are identified, traffic instances are grouped using K-means clustering and device behaviour is represented using unsupervised OCC models. ComplexIoT [2] classifies IoT traffic by assigning a trust score to each flow based on complexity and variance, leading to more precise anomaly detection boundaries for simpler devices, and generalized boundaries for more complex devices.

DÏoT [3] leverages a device-type-specific anomaly detection approach, comparing each device's behaviour to a specific device-type profile. The authors argue that a single model for all devices can lead to high false positives or reduced sensitivity due to IoT diversity. By employing dedicated models for each device type, DÏoT more accurately captures behaviour patterns, improving anomaly detection. This system uses Gated Recurrent Units (GRUs) within a federated learning framework, to aggregate profiles across devices. Comparing single vs. device type models, false positive rates of 0.67% and 0%, and true positive rates of 97.21% and 95.6% are achieved, respectively.

Our implementations assess whether the normal profile of a device or device-type is sufficiently precise for effective anomaly detection and whether unsupervised OCC methods outperform supervised learning.

III. Methodology

A. Learning Methods

This paper employs various state-of-the-art ML, DL, and OCC methods. The investigated ML and DL algorithms are Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Deep Neural Network (DNN), while the OCC methods include iForest (Isolation Forest), One-Class Support Vector Machine (OCSVM), Local Outlier Factor (LOF), and Deep Support Vector Data Description (DeepSVDD) [4].

iForest is an ensemble anomaly detection method that isolates anomalies by recursively partitioning the data to create a forest of trees [5]. **OCSVM** is an unsupervised ML algorithm designed for novelty detection that learns a decision boundary to encapsulate normal data points and identifies deviations as anomalies [6]. **LOF** is a density-based anomaly detection method that estimates data point density by measuring distances between points, identifying denser regions as normal and less dense regions as outliers [5]. **DeepSVDD** is an OCSVM-related technique that uses a hypersphere to separate data samples, leveraging neural networks to learn useful feature representations for anomaly detection [4].

B. Datasets

To assess IoT security solutions, datasets representing IoT behaviour (including normal and malicious behaviour) are needed. To implement the experiments, device IDs are required. Only some datasets provide device IDs in their feature vector. For others, the device IDs can be obtained from the provided PCAP (Packet Capture) files, if available.

1) N-BaIoT dataset: The N-BaIoT dataset [7] is collected from nine commercial IoT devices with scanning, junk spam, UDP and TCP flooding, ACK and SYN flooding attacks. Behavioural snapshots of network flows are captured for multiple time windows. There are 7,062,606 records, with 92% attack records. The record distribution among devices in this dataset is as follows: two devices hold approximately 5% of the data each, while the remaining seven devices hold between 10% and 16% of the data each.

2) CICIoT2023-Packet dataset: The CICIoT2023 dataset [8] was collected from 105 real IoT devices in a large IoT testbed, and includes 33 distinct attacks, categorized into seven groups: DDoS, DoS, reconnaissance, web-based, brute force, spoofing, and Mirai. To create a dataset with Device IDs, we generated CICIoT2023-Packet dataset from the PCAP files of this dataset, using Tcpdump, Scapy, Socket, Numpy, and Pandas Python packages with packet-level features. The generated dataset was 820GB, thus subsampling was performed, reducing the dataset to 9.8GB. All benign traffic was retained as the data predominantly consisted of attack records. For each attack subcategory, a maximum of 40,000 records were randomly selected. The dataset was then cleaned and preprocessed.

The generated dataset comprises data from 69 devices, with a highly non-uniform data distribution among different devices. Some devices are predominantly composed of normal records, while others are predominantly composed of attack records. Devices with minimal or no normal records were excluded because one-class classifiers need a sufficient amount of normal traffic to establish normal behaviour profiles, and ML/DL models require adequate samples from both normal and attack classes. The final dataset includes data from 62 devices belonging to seven groups. The data distribution is highly non-uniform, with most devices holding less than 2% of the data, while two devices hold 18% and 21%.

C. Performance Metrics and Experimental Design

We evaluate the performance of eight algorithms for binary classification using metrics, such as accuracy, precision, recall, and F1-score with a primary focus on accuracy and F1-score. Accuracy indicates the ratio of correct classifications on the entire test set, while F1-score is the harmonic mean of precision and recall. These metrics are extensively described in related works (e.g., [8]).

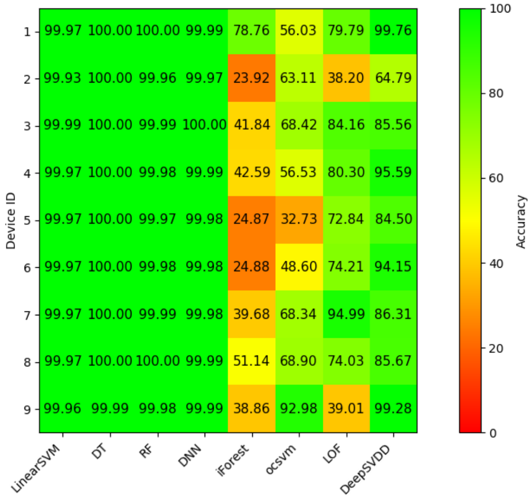
Records from each device (or device type) are randomly split into 70% train set and 30% test set. Various ML/DL/OCC algorithms are applied to the datasets. The binary classification results, averaged over five independent runs, include both the average and standard deviation. ML and DL methods are trained on the entire train set, while OCC methods are trained on the normal records from the train set. During evaluation, records are classified as benign (normal) or anomalous.

IV. Experiment Results

A. Device-Specific Models for Anomaly Detection

1) N-BaIoT Dataset: Nine device-specific models are trained on data from each of the nine devices of the N-BaIoT dataset using various supervised and unsupervised OCC methods. The accuracy and F1-score of binary classification for each of the nine models are averaged over five runs. The accuracy results are displayed in a heatmap in Figure 1. We observed the following results:

- ML/DL consistently outperformed OCC.
- Among OCC methods, DeepSVDD performed well across all devices, especially #1 and #9.
- LOF ranked after DeepSVDD, with moderate performance for most devices except #2 and #9.
- iForest showed the worst accuracy in almost all devices.



Device ID	LinearSVM	DT	RF	DNN	iForest	ocsvm	LOF	DeepSVDD
1	99.97	100.00	100.00	99.99	78.76	56.03	79.79	99.76
2	99.93	100.00	99.96	99.97	23.92	63.11	38.20	64.79
3	99.99	100.00	99.99	100.00	41.84	68.42	84.16	85.56
4	99.97	100.00	99.98	99.99	42.59	56.53	80.30	95.59
5	99.97	100.00	99.97	99.98	24.87	32.73	72.84	84.50
6	99.97	100.00	99.98	99.98	24.88	48.60	74.21	94.15
7	99.97	100.00	99.99	99.98	39.68	68.34	94.99	86.31
8	99.97	100.00	100.00	99.99	51.14	68.90	74.03	85.67
9	99.96	99.99	99.98	99.99	38.86	92.98	39.01	99.28

Fig. 1. Accuracy of Device-Specific Models (N-BaIoT dataset).

Table I presents the average scores of all device-specific models implemented using various algorithms, showcasing metrics along with their respective standard deviations (SD) across the nine models.

To evaluate the effectiveness of device-specific models versus a single model for all devices, the performance of the single model is compared with the average of all device-specific models, presented in Table II, highlighting the improved metrics. When using ML and DL algorithms, there were minor or no differences in accuracy and F1-score between the two types of models. However, significant differences emerged with OCC methods. Specifically, iForest and OCSVM showed notable improvements with device-specific models, with iForest's accuracy and F1-score nearly doubling, and OCSVM's metrics improving by 7.5% and 3%, respectively. In contrast, LOF and DeepSVDD performed worse with device-specific models, with LOF seeing a decrease in accuracy and F1-score by 29% and 21%, and DeepSVDD experiencing a decrease of 4% and 3%, respectively. Overall, ML and DL methods generally achieved higher accuracy. LinearSVM and DNN showed slight improvements with device-specific models, while other methods remained consistent. In terms of F1-scores, two algorithms remained unchanged, RF improved, and DNN declined when utilizing device-specific models.

TABLE I
Average scores of device-specific models (N-BaIoT dataset).

Method	**Accuracy**		**Precision**		**Recall**		**F1-Score**	
	%	SD	%	SD	%	SD	%	SD
LinearSVM	99.97	.0013	99.98	.0012	99.98	.0005	99.98	.0008
DT	100	.0002	100	.0001	100	.0001	100	.0001
RF	99.98	.0011	99.99	.0012	99.99	.0005	99.99	.0007
DNN	99.99	.0008	99.99	.0005	99.99	.0005	99.99	.0005
iForest	40.73	16.2	94.86	4.86	35.73	18.4	48.90	18.2
OCSVM	61.74	15.6	92.80	4.44	62.87	17.5	73.72	13.2
LOF	70.84	18.4	97.79	2.83	68.68	20.6	78.75	17.0
DeepSVDD	88.40	10.0	99.42	.0039	87.90	11.3	92.14	8.24

TABLE II
Single model vs. average of device-specific models: N-BaIoT.

	Method	**Accuracy**		**F1-score**	
		Single Model	Device model	Single Model	Device model
Supervised	LinearSVM	99.96	**99.97**	99.98	99.98
	DT	100	100	100	100
	RF	99.96	**99.98**	99.98	**99.99**
	DNN	99.99	99.99	**100**	99.99
Unsupervised	iForest	20.78	**40.73**	24.14	**48.90**
	OCSVM	57.42	**61.74**	71.53	**73.72**
	LOF	**99.21**	70.84	**99.57**	78.75
	DeepSVDD	**92.07**	88.40	**95.16**	92.14

Figure 2 shows the performance comparison between single and device-specific models by subtracting the accuracy of the single model from the average accuracy of device-specific models. Negative values (yellow to red) indicate the single model's superiority, while positive values (yellow to green) indicate the device-specific models' superiority. The results revealed the following insights:

- The difference in performance between single-model and device-specific approaches is minimal for ML/DL methods but varies significantly for OCC methods.
- Device-specific models using iForest consistently outperformed the single model across all devices.
- Device-specific models using LOF performed worse than the single model.
- When considering same-type devices, the algorithm type plays a more significant role in determining the best approach, rather than differences within the devices themselves.

2) CICIoT2023-Packet dataset: Models were trained on data from 62 devices using ML/DL/OCC. Table III displays metrics averaged over all device-specific models, along with standard

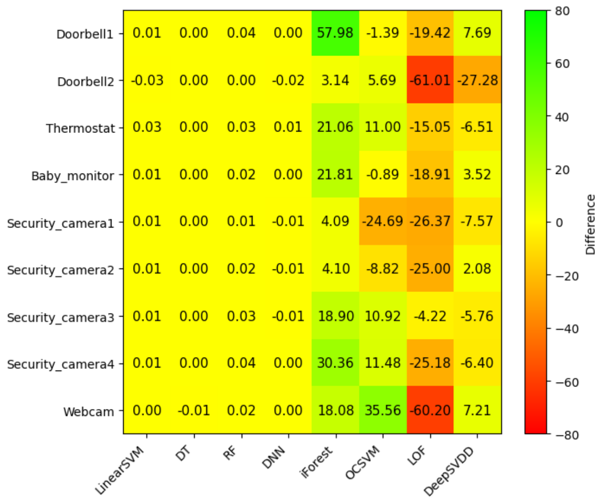


Fig. 2. Accuracy Difference: Single vs. Device-Specific Models (N-BaIoT).

deviations from the average across all models. Due to its resource-intensive nature, the OCSVM algorithm was executed on only 5% of the data for a single implementation; therefore, no standard deviation is provided. Similar to the N-BaIoT dataset results, ML and DL methods outperformed one-class classifiers.

TABLE III
AVERAGE SCORES OF DEVICE-SPECIFIC MODELS (CICIOT2023-PACKET).

Method	Accuracy		Precision		Recall		F1-Score	
	%	SD	%	SD	%	SD	%	SD
LinearSVM	81.56	10.3	84.53	15.3	55.18	31.2	61.79	28.5
DT	85.01	8.41	96.90	4.70	56.99	29.4	67.18	24.4
RF	84.01	8.78	94.04	6.61	56.47	30.0	65.59	25.4
DNN	81.88	10.1	55.35	30.9	61.30	28.5	77.65	23.9
iForest	62.90	14.7	58.95	23.5	39.01	17.7	43.78	17.9
OCSVM	58.98	-	55.02	-	77.46	-	60.06	-
LOF	77.38	14.5	95.51	11.0	44.12	26.8	56.29	24.8
DeepSVDD	74.64	14.8	83.35	23.2	39.47	28.2	49.18	28.0

Similar to the previous subsection, Table IV compares device-specific models with a single model for all devices. The results indicate that for ML and DL algorithms, a single model outperforms device-specific models in both accuracy and F1-score across all algorithms. However, for OCC, device-specific models using iForest, LOF, and DeepSVDD achieved higher accuracy than the single model, while only the device-specific models using OCSVM showed lower accuracy.

Figure 3 denotes a comparison between the average of device-specific models and the single model. The following insights are derived:

- The single model based on ML/DL algorithms exhibited higher accuracy across all device types except for NextGeneration devices.
- For NextGeneration devices, holding 43% of all data and also dominated by one class (i.e., attack class), device-specific models consistently outperformed the single model across all algorithms.

TABLE IV
SINGLE MODEL VS. AVERAGE OF DEVICE-SPECIFIC MODELS (CICIOT2023-PACKET).

	Method	Accuracy		F1-score	
		Single Model	Device model	Single Model	Device model
Supervised	LinearSVM	**83.92**	81.56	**86.85**	61.79
	DT	**90.28**	85.01	**91.80**	67.18
	RF	84.55	84.01	**87.80**	65.59
	DNN	**87.89**	81.88	**89.64**	77.65
Unsupervised	iForest	48.12	**62.90**	32.28	**43.78**
	OCSVM	**77.63**	58.98	**83.76**	60.06
	LOF	69.01	**77.38**	**65.48**	56.29
	DeepSVDD	53.63	**74.64**	37.88	**49.18**

- Generally, device-specific models using OCC methods outperformed the single model across all algorithms, except for OCSVM.

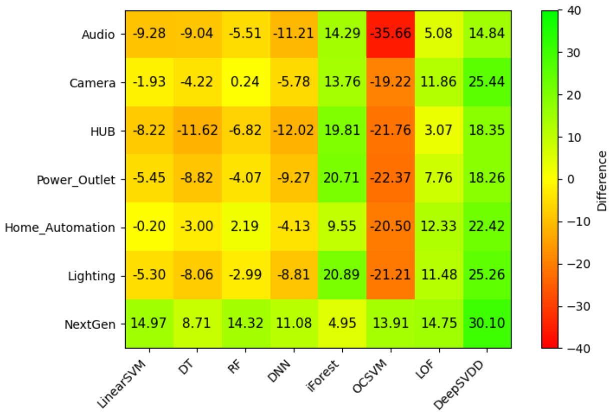


Fig. 3. Accuracy Difference: Single vs. Average of Device-Specific Models (CICIoT2023-Packet).

B. Device-Type-Specific Models

1) N-BaIoT Dataset: The N-BaIoT dataset includes nine devices across five categories: two doorbells, one thermostat, one baby monitor, four security cameras, and one webcam. Device-type-specific anomaly detection models are trained on data from each category. Below are the key observations from these results on accuracy of device-type-specific models, aligning with findings on device-specific models for the N-BaIoT dataset.

- ML/DL methods consistently outperformed OCC methods, demonstrating superior performance.
- Among OCC methods, OCSVM and DeepSVDD performed well across most device types, with OCSVM achieving 92-94% accuracy and DeepSVDD exceeding 99% accuracy.
- iForest and LOF displayed notably low accuracy across all types of devices.

Comparing device-type-specific models to a single model in Figure 4, the difference is negligible for supervised ML/DL methods. However, one-class classifiers showed significant variation. Particularly, iForest and OCSVM models generally performed better with device-type-specific models, sometimes by a considerable margin. In contrast, LOF and DeepSVDD showed higher accuracy with a single model.

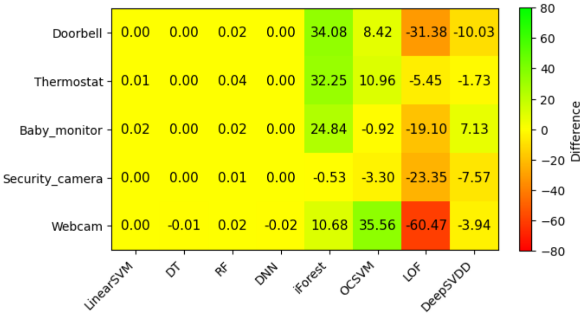


Fig. 4. Accuracy Difference: Single vs. Device-Type-Specific Models (N-BaIoT).

Figure 5 compares device-type-specific models to device-specific models for categories with multiple devices, which are doorbells (device #1 and #2) and security cameras (devices #5 to #8). The main insights are summarized as follows:

- Minimal difference is observed between these model types with supervised ML and DL methods.
- For OCC methods, devices #1 and #2 had better accuracy with device-type-specific models, while devices #5 to #8 performed better with device-specific models. Thus, the optimal model choice depends on the device type.

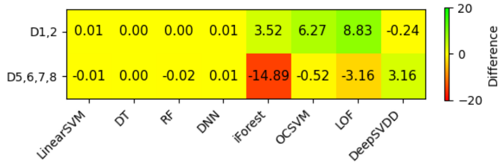


Fig. 5. Accuracy Difference: Device-Type-Specific vs. Avg. of Device-Specific models (N-BaIoT).

2) *CICIoT2023-Packet Dataset:*

This dataset is comprised of 62 devices in seven categories. There are seven audio devices, 14 cameras, six Hub devices, 14 power outlets, eight home automation devices, six lights, and seven next generation devices.

A model per device type is trained and evaluated for each of the seven device groups. Figure 6 shows the accuracy of these device-type models using various ML/DL/OCC algorithms. Due to difficulties with large training sets, OCSVM results were not obtained for this dataset. Camera and NextGen device groups performed exceptionally well across almost all algorithms, likely due to their large data share (43% for NextGen and 34% for cameras) and the dominance of one class, 99% attack records for NextGen and 84% normal records for cameras. This data volume and class dominance enhanced OCC method performance.

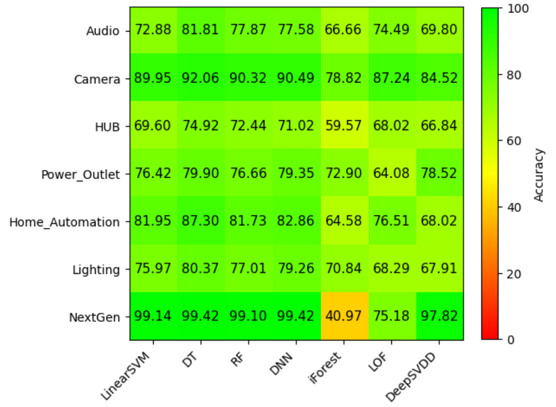


Fig. 6. Device-Type-Specific Model Accuracy (CICIoT2023-Packet).

When comparing single and device-type-specific models (Figure 7), the NextGen and camera device groups performed better with device-type-specific models. In other groups, the single model excelled with supervised ML/DL methods, whereas device-type-specific models outperformed with iForest and DeepSVDD. Consistent with the N-BaIoT dataset results, LOF resulted in outperforming the single model.

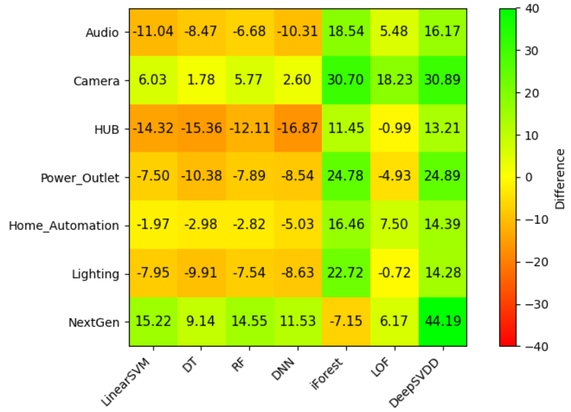


Fig. 7. Accuracy Difference: Single vs. Device-Type-Specific Models (CICIoT2023-Packet).

Comparing device-type and device-specific models shows varied performance by device group. In the camera group, device-type models outperformed device-specific models across all algorithms due to the high volume of data from devices 11 and 13. In contrast, device-specific models consistently outperformed device-type models in the Hub device group across all ML/DL/OCC methods.

V. Discussion

The experiments presented in this paper provide insights into the performance of various modelling approaches, including single, device-specific, and device-type-specific models. Supervised ML and DL methods generally showed high accuracy in anomaly detection for device-specific models. However, OCC

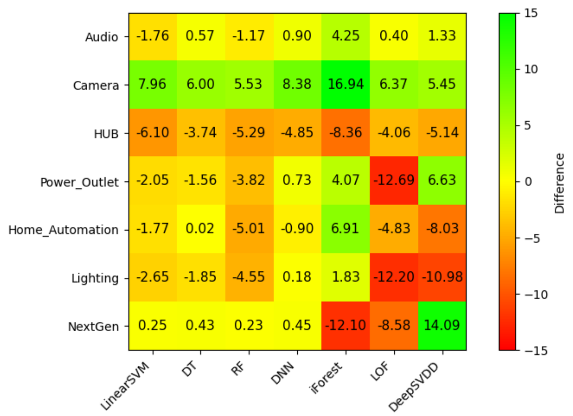


Fig. 8. Accuracy Difference: Device-Type-Specific vs. Device-Specific Models (CICIoT2023-Packet).

methods yielded diverse results, with DeepSVDD and then LOF performing acceptably across all datasets. DeepSVDD, which is an integration of OCC and DL, and LOF excel in establishing more accurate boundaries for normal profiles.

Comparing single vs. device/device-type models on the N-BaIoT dataset revealed similar performance when using ML/DL methods. On the CICIoT2023-Packet dataset, the single model generally outperformed device/device-type models with ML/DL methods, except for NextGen and camera devices. For these two groups, which make up 77% of the dataset (NextGen at 43%, camera at 34%), device/device-type models consistently outperformed the single model. Both groups are dominated by one class, 99% of NextGen data is attack records and 84% of camera data is normal. This suggests that device/device-type models are particularly effective for anomaly detection when devices have ample data and are dominated by a single type of traffic pattern. The effectiveness of device-specific vs. device-type-specific models depends more on device types and their data rather than on the algorithm used.

The results differ between the two datasets due to their distinct data distributions. The N-BaIoT dataset has a more balanced data distribution among devices, while the CICIoT2023-Packet dataset has significant data imbalances.

To overcome the cold start issue in distributed IDS for IoT devices with limited initial data, a strategy involves preloading devices with pre-trained models based on device types. These models, initially trained on extensive data, undergo fine-tuning using local data through transfer learning. Michau and Fink [9] proposed an unsupervised transfer learning framework using OCC for industrial applications.

VI. Conclusion and Future Work

In pursuit of a practical Intrusion Detection System (IDS) tailored to the needs of Internet of Things (IoT), device-specific and device-type-specific models were evaluated and compared with a single model for all devices. While a single model requires less storage space, device/device-type models trained on data from individual IoT devices or groups of similar devices prove to be beneficial in IoT systems where heterogeneous devices generate unique traffic patterns.

Device/device-type models are superior to single models when either the data is dominated by one class or when using one-class classifiers. IoT systems predominantly produce benign traffic before any attacks occur. By training device/device-type models on an IoT device's stream of traffic, an IDS agent can build a normal profile and then detect any deviations from the normal profile as intrusions. In this context, use of One-Class Classifier (OCC) methods emerges as a viable solution instead of supervised Machine Learning and Deep Learning methods, which rely on pre-generated and labelled datasets. Attack traffic is usually scarce, and collecting attack data is difficult and costly, making supervised methods impractical in large IoT systems.

Future work will focus on the application of device/device-type models in a Federated Learning (FL) setting to enable knowledge sharing through model parameter weights while preserving data privacy. Feature analysis will also be considered. Additionally, we also plan to evaluate OCC models on a dataset dominated by normal traffic, contrasting with the attack-dominant N-BaIoT and the imbalanced CICIoT2023-Packet datasets, to assess their accuracy in detecting anomalies in IoT systems with predominantly normal traffic.

References

[1] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, "Detecting Behavioral Change of IoT Devices Using Clustering-Based Network Traffic Modeling," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7295–7309, Mar. 2020.

[2] K. Haefner and I. Ray, "ComplexIoT: Behavior-Based Trust For IoT Networks," in *Trust, Privacy and Security in Intelligent Systems and Applications*, Los Angeles, CA, Dec. 2019, pp. 56–65.

[3] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," in *Distributed Computing Systems*, Dallas, TX, Jul. 2019, pp. 756–767.

[4] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep One-Class Classification," in *ICML*, vol. 80, Stockholm, Sweden, Jul. 2018, pp. 4393–4402.

[5] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: an Intelligent Anomaly-based Intrusion Detection System for IoT Edge Devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, Jan. 2020.

[6] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised Intelligent System based on One Class Support Vector Machine and Grey Wolf Optimization for IoT Botnet Detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 2809–2825, 2020.

[7] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, July-September 2018.

[8] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, pp. 1–26, Jun. 2023.

[9] G. Michau and O. Fink, "Unsupervised Transfer Learning for Anomaly Detection: Application to Complementary Operating Condition Transfer," *Knowledge-Based Systems*, vol. 216, pp. 1–9, Jan. 2021.