Evaluating Energy and Thermal Efficiency of Anomaly Detection Algorithms in Edge Devices

Felipe Pfeifer Rubin†, Paulo Silas Severo de Souza†, Wagner dos Santos Marques†,
Rômulo Reis de Oliveira†, Fábio Diniz Rossi*, and Tiago Ferreto†
†Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
*Federal Institute of Education, Science, and Technology Farroupilha (IFFAR), Alegrete, Brazil
Email: {felipe.rubin, paulo.silas, wagner.marques.001, romulo.reis}@acad.pucrs.br,
fabio.rossi@iffarroupilha.edu.br, tiago.ferreto@pucrs.br

***Abstract*—Although IoT delivers several benefits, it also raises concerns regarding privacy and security, from revenue disruption in industrial facilities to life-threatening situations caused by smart houses hacking. As a consequence, anomaly detection algorithms stand out to improve data reliability. However, little has been said about the implications of running these computationally expensive programs in hardware-constrained edge devices. Therefore, in this paper, we present an evaluation of six anomaly detection algorithms running in an edge device regarding performance, accuracy, temperature, and power consumption. The results showed that time complexity, resources demand, and detection approach directly impact on the feasibility of running anomaly detection algorithms in edge devices. Based on these results, we present a recommendation on which algorithms would best satisfy the requirements of several IoT environments.**

I. Introduction

The Internet of Things (IoT) encompasses a variety of applications. Smart classrooms equipped with embedded devices containing face recognition algorithms can give educators real-time feedback on students' motivation [1]. On agriculture, smart farms employing distributed sensors can obtain real-time information about soil moisture and temperature, helping farmers to improve crop production [2]. As a consequence, the requirements may vary according to several aspects, such as the number of users and specific characteristics of applications.

Despite differences in applications' requirements and demands, one of the most prominent aspects that IoT scenarios have in common is a large amount of data being generated rapidly [3] [4]. In order to provide efficient and reliable feedback, data processing becomes a priority task, while other tasks are often overlooked. As a consequence, sensors may be compromised by failures or even manipulated by attackers to induce erroneous decision-making [5] [6].

In this scenario, anomaly detection algorithms arise to ensure the integrity of the data. As most of these algorithms are computationally expensive, one could suggest executing them in the cloud to take advantage of high processing capabilities. However, the unstable connectivity of IoT scenarios makes this alternative unfeasible, as migrating data to the cloud could demand too much time [7]. Hence, anomaly detection algorithms are usually executed in embedded devices close to customers, at the edge of the network [8] [9] [10].

Although anomaly detection is a well-known research topic from a machine learning point of view [11] [12], little has been said about the implications of migrating these algorithms to the edge of the network, wherein devices are resource-constrained and battery-dependent. Accordingly, in this paper, we assess how anomaly detection algorithms affect the power consumption and temperature of embedded devices. We also highlight the following contributions:

- We present a discussion on the characteristics and requirements of different IoT scenarios and the role of anomaly detection algorithms on each of them;
- We analyze how characteristics such as time complexity and detection approach impact on the power consumption and temperature of embedded devices.

The remainder of this paper is organized as follows. In Section II we present the theoretical background that discusses the relevance of performing anomaly detection at the edge of the network. In Section III, we compare our study with other several investigations that focus on anomaly detection in IoT scenarios. In Section IV, we present details on the methodology we adopted. Section V is reserved for the results and its discussion, and in Section VI we present final remarks and directions for future investigations.

II. Background

The continuous development of areas such as Telecommunications brought about a new wave of electronic objects. From consumer electronics such as smart-phones to industrial appliances such as field sensors, networking capabilities are always present. This common networking capability became the foundation on which the Internet of Things stands [13].

The Internet of Things paradigm regards promoting the interaction between different devices in order to obtain, generate, or exchange data. An internet where every device is continuously interacting with another entails generating a tremendous amount of data every moment.

IoT devices can not process such a vast amount of data because contrary to general-purpose solutions, these devices are designed for specific purposes and have highly constrained resources; hence, other alternatives must be considered. Authors on [14] deliberate that Cloud Computing can provide the infrastructure to analyze, store, and monitor such amount of data.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on May 01,2023 at 19:58:20 UTC from IEEE Xplore. Restrictions apply.

Although Cloud Computing prevails as a solution for this significant amount of data, other problems arise with it. Of all possible problems, two might be considered the most challenging: latency and security. In the case of latency, applications that depend on obtaining up-to-date information are highly susceptible to become affected by delayed data arrival. In some cases, even with a dedicated link to transmit its data, the device's constraints might appear as a problem.

Security is also the subject of constant discussion in IoT. Due to the nature of the IoT, there exist many attack vectors. When transmitting data between devices, additional steps must be taken to ensure that sensitive information is not intercepted by attacks such as Man-in-the-middle [15]. Also, as many IoT devices are based far away from operators, ensuring that the device cannot be physically tampered with is a must. Besides malicious physical tampering, data might also be corrupted due to network issues or faulty components.

Considering the problems mentioned above, pre-processing appears as a solution. The concept of pre-processing is broad and may consist of several tasks. Simple filtering might reduce network latency by ensuring that only relevant packages will be transmitted to other devices. More sophisticated approaches that use anomaly detection algorithms could provide reassurance that the data being transmitted is secure and was not corrupted maliciously or not.

The concept of Big Data has bought many new insights regarding the importance of data analytics for the IoT [16]. The process of extracting useful information from large amounts of data provides more in-depth insights on applications and enables planing further developments [17]. In order to benefit from such remarkable insights, the data of which they are based on must be trusted.

Anomaly detection algorithms are used to detect if any data obtained from a given source could end up being an anomaly. Anomalies are patterns of data that do not conform with those around them [18]. They must be identified and purged before any data analysis takes place; otherwise, the results of such analysis would be meaningless.

Due to the granularity of data, a wide variety of anomaly detection algorithms exist. While many employ similar machine learning techniques, they have different requirements for the data input. The organization of the data is in regards to variables (multivariate or univariate), analysis model (online, offline, or both) and dimensional (spatial, temporal, or spatiotemporal). With such consideration, anomaly detection algorithms that employ machine learning may separated into three main groups:

- **Supervised Learning:** On supervised learning, prior knowledge, and labeled data about the patterns that will be explored already exists. These labeled patterns will be matched, and potential anomalies will be detected.
- **Reinforcement Learning:** Contrary to Supervised Learning has no available labeled data, yet knowledge of how the data should be labeled exists. As such, this technique requires the user to give feedback for every outcome, being positive if current or negative otherwise.
- **Unsupervised Learning:** In the case of unsupervised learning, there exist very few or no knowledge about the nature of the data. As such, other kinds of techniques must be used. One of which is clustering, where it is expected that patterns will appear closer to one another; therefore, nonconforming ones could be anomalies.

The use of anomaly detection algorithms is a common recurrence in cloud computing environments. An issue that rises with using these algorithms on edge devices is their resource requirements. Such kind of algorithm usually consists of complex mathematical computations and massive parallelism being supported by GPUs. In cloud computing environments, such resources are easily obtainable, but as stated before, IoT devices have very constrained resources. Even modern devices with embedded state-of-art GPUs might not be able to take advantage of some algorithms.

Therefore, in this paper, we are not concerned about the ends, as we understand that anomaly detection algorithms are mature enough to deliver accurate results. We are specifically interested in the means, i.e., what are the implications of executing these computationally expensive programs on edge devices' constrained hardware in terms of power consumption and heat.

III. Related Work

Several investigations have been undertaken to detect abnormal behaviors in different IoT scenarios.

Ukil et al. [8] present challenges and the relevance of anomaly detection techniques in several IoT health-care scenarios. The authors argue that anomaly detection plays an important role in IoT health-care scenarios. Abnormal data coming from compromised sensors could profoundly affect the effectiveness of health-care IoT systems or even put its users' lives in danger.

Narayanan, Mittal, and Joshi [9] developed OBD-SecureAlert, an anomaly detection mechanism that detects abnormal activities in new and 'on-road' vehicles. The data collected from different vehicle components is converted into a sequence of observation vectors. Every time a new observation is available, a sliding window containing previous observations moves, and the posterior probability of the new observation sequence is determined. If the estimated probability is below a threshold, the observation is perceived as an anomalous state.

On Trilles et al. [10], an environmental anomaly detection system is proposed. The system consists of three layers: content, services, and application. The content layer consists of multiple field sensors that are used to obtain air quality data. The sensors' data are sent to the services layer, where anomalies may be detected separately on each data stream using the CUSUM algorithm. The application layer provides a web interface were each sensor status is displayed along with any anomalous warning.

Due to the popularization of areas such as machine learning, several algorithms to detect anomalous sensor data have

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on May 01,2023 at 19:58:20 UTC from IEEE Xplore. Restrictions apply.

been proposed. However, choosing the wrong algorithm may delay or even compromise the detection process. Some studies also compare anomaly detection algorithms [19] [20] [6]. However, their analysis does not cover important aspects such as the impact of running anomaly detection algorithms in edge devices regarding temperature and power consumption. Both of these are relevant since edge devices usually operate in strict conditions.

IV. Methodology

A. Analyzed Algorithms

In this study, we did choose six anomaly detection algorithms with support to multivariate data sets from PyOD [21] (an overview of these algorithms is presented in Table I):

Isolation Forest (iForest) [22]. Uses random trees to detect anomalies based on the premise that after constructing isolation trees for a given data set, anomalies are isolated closer to the root of the tree while normal points are isolated in the further nodes. The anomaly detection is composed of two stages: training and evaluation. In the training stage, a given data set is separated into disjoint sub-sets from which isolation trees are constructed through continuous sub-sampling until either no more data is available or a cutoff value (tree height limit) is reached. The sub-sampling consists of designating points as left or right sub-trees using a randomly chosen attribute from the data set along with a randomly picked split-point between its minimum and maximum values. In the evaluation stage, test instances are passed through the isolation trees, and their anomaly scores are obtained.

K-Nearest Neighbors (kNN) [23]. Computes for each point of a data set the distance from its K^{th} nearest neighbor. Then, it selects the top N points with the maximum distances as outliers. In order to minimize the computational cost, a clustering algorithm is used to partition in disjoint subsets the input data. Partitions that cannot contain outliers are pruned. On the remaining partitions, outliers are computed.

Local Outlier Factor (LOF) [24]. Applies the concept of local density to determine if a point is an outlier. The local density of a point is calculated using the distance of its k-nearest neighbors. A point is considered an outlier if, by comparison, it has a lower density than its neighbors.

FindCBLOF [25]. Uses the Squeezer algorithm [26] on a given data set to obtain a set of clusters. Using the obtained clusters and two numerical parameters that are used to define a boundary, two new sets of clusters are derived: LC (Large Clusters) and SC (Small Clusters). For each record of the data set, if the record's cluster belongs to LC, the CBLOF value is calculated using the distance between the record and its cluster, otherwise (its cluster belong to SC) the distance is calculated using the minimal distance between the record and a cluster belonging to LC.

The Histogram-Based Outlier Score (HBOS) [27]. Firstly creates a univariate histogram for each feature on the data set. The histogram is created using different techniques considering the type of data that was provided (categorical or numerical). Then it normalizes the maximum height of each histogram to 1.0, which ensures an equal weight of each feature to the outlier score. Finally, the HBOS value (outlier score) of each instance of the data set is calculated using a formula that may be interpreted as the inverse of a discrete Naive Bayes probability model.

Angle-Based Outlier Detector (ABOD) [28]. Focuses on high-dimensional data, were common approaches tend to deteriorate due to the "curse of dimensionality." Contrary to popular algorithms, ABOD considers the variance of the angles between the different vectors of data objects as a property to measure if an object is an outlier. If the spectrum of observed angles for a point is broad, the point is surrounded by others and is probably located inside a cluster. Otherwise, it is believed to be outside of grouped sets of points; thus, it is considered an outlier. The measured distance between objects is used to normalize the results.

Table I. Specifications about the evaluated anomaly detection algorithms.

Algorithm	Type	Time Complexity
Angle-Based Outlier Detector	Probabilistic	$O(n^3)$
Local Outlier Factor	Proximity-Based	$O(n^2 \times k)$
K-Nearest Neighbors	Proximity-Based	$O(n^2)$
Isolation Forest	Outlier Ensembles	$O(n)$
Cluster Based Local Outlier Factor	Proximity-Based	$O(n)$
Histogram-Based Outlier Score	Proximity-Based	$O(n)$

B. Experimental Setup

During the tests, we used the ForestCover [29], which is a well-known multivariate data set. This data set contains the forest cover type from cartographic variables from four areas with minimal human-caused disturbances of the Roosevelt National Forest (United States). Its data is useful for researchers interested in aspects such as biodiversity.

Our experimental setup comprises an NVIDIA Jetson TX2 Developer Kit[1], which is a power-efficient embedded AI computing device suitable for several IoT applications such as robots, drones, smart cameras, and portable medical devices [30]. Besides, Jetson TX2 has thermal and power consumption sensors that make it suitable for our experimentation. Table II presents more details about this embedded platform.

Table II. Technical specifications of the NVIDIA Jetson TX2 Developer Kit.

Components	Specifications
Processors	HMP Dual Denver 2/2MB L2 + Quad ARM A57/2MB L2
GPU	NVIDIA PASCAL (256 CUDA cores)
Memory	8GB 128bit LPDDR4 59.7GB/s
Networking	1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth
Operating System	Linux Ubuntu 16.04.5 LTS (kernel 4.4.38-tegra)

V. Results and Discussion

The results presented next are the average of 5 executions of each algorithm with a standard deviation lower than 3%. The assets of our experimentation are available in our GitHub repository[2]. We also monitored CPU and memory usage during the execution of the algorithms. Besides, we used the Pearson

[1]Jetson TX2. <developer.nvidia.com/embedded/buy/jetson-tx2>.

[2]Experiments assets. <github.com/paulosevero/outlier_detec_comp>.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on May 01,2023 at 19:58:20 UTC from IEEE Xplore. Restrictions apply.

Correlation Coefficient to verify the impact of resources usage on the results of performance, power consumption, temperature, and EDP. Table III summarizes the correlations among each of these metrics.

Table III. CORRELATION AMONG DIFFERENT METRICS.

Metrics	Pearson Correlation Coefficient
Temperature and Execution Time	0.99
Temperature and EDP	0.92
RAM Usage and Power Consumption	0.87
CPU Usage and Temperature	0.82
CPU Usage and Execution Time	0.77
CPU Usage and Power Consumption	0.75
Temperature and Power Consumption	0.33
RAM Usage and Temperature	-0.03
RAM Usage and Execution Time	-0.07

A. Accuracy

We adopted Precision at n ($P@n$) as the accuracy measure, which is given by Equation 1 and shows the number of relevant instances among the retrieved results. The algorithm that presented the best accuracy results was Isolation Forest, with $P@n = 0.11$. This result shows the effectiveness of Isolation Forest in handling problems widely discussed in anomaly detection landscape such as Swamping and Masking effects [22]. Swamping consists of identifying normal instances as anomalies. It usually occurs when normal points are too close to the anomalous ones. Masking refers to considering anomalous instances as normal data and often occurs when there are too many anomalies.

$$Precision = \frac{|\text{Relevant Data} \cap \text{Retrieved Data}|}{|\text{Retrieved Data}|} \quad (1)$$

Different from the other algorithms like Local Outlier Factor, which presented the worst accuracy result ($P@n = 0.56$), Isolation Forest focuses on identifying anomalies rather than profiling normal points. Since anomalies are less frequent and different from normal observations, Isolation Forest randomly partitions data in decision trees (called isolation trees), so anomalies can be identified as the instances closer to the root of the tree since fewer splits are necessary to isolate them. Figure 1 [A] presents the accuracy results of the evaluated algorithms.

B. Performance

According to the results presented in Figure 1 [B], the algorithm that presented the best performance results was HBOS, being able to detect the outliers in $3.83s$. As presented in Table I, this algorithm has a linear time complexity $O(n)$, which we believe was the main reason for such a result.

It is worth noting that CBLOF and iForest also have linear time complexity. However, as stated by Goldstein and Dengel [27], HBOS can detect outliers in large multivariate data sets faster than other algorithms by computing a histogram to each feature of the data set, scoring them individually, and combining them in the end. On the other hand, ABOD presented the worst performance due to its cubic time complexity $O(n^3)$, which is caused since several iterations are required for computing the anomaly score of each instance in the data set.

C. Power Consumption

Regarding power consumption (Figure 1 [C]), Histogram-based Outlier Score was the algorithm with better results, consuming $170.91mW$. Considering the high correlation between resource usage and power consumption presented in Table III, we can see that the energy-saving achieved by HBOS was influenced by its low CPU (71.04%) and memory ($1144.08MB$) demand. On the other hand, Isolation Forest, that presented the worst power consumption result, was the algorithm that most impacted in the device's memory usage ($1386.98MB$).

D. Temperature

As resource usage affects the device's temperature, we realized that CPU usage impacted the device's heat during the algorithm's execution ($Correlation = 0.82$, as shown in Table III). Besides, we found a high correlation between execution time and temperature, i.e., the longer the algorithms took to execute, the higher was their impact on the device's heat since the evaluated algorithms involve computing-intensive tasks.

As a consequence of such correlations, the algorithm that presented less impact on the device's temperature was CBLOF, which uses the Squeezer algorithm to define the number of clusters without imposing a bottleneck to IO operations and requires only one scan over the data set for computing the anomaly score of instances. On the other hand, ABOD was the algorithm that most generated heat to the embedded device due to its higher execution time. All the results regarding temperature are shown in Figure 1 [D].

E. Energy-Delay Product

One of the well-known strategies to prolong the battery life of embedded systems is through power-saving modes that focus on minimizing the device's power consumption at the cost of reducing the application's performance. In this context, there is a concern regarding the balance between performance and power saving [31]. Therefore, in our analysis, we also considered the Energy-Delay Product (EDP), which is given by $Performance \times Power\ Consumption$, to evaluate the impact of the chosen algorithms to this trade-off.

The results showed that performance was the metric that most impacted in the EDP results, since HBOS, which was the algorithm that demanded less time to process data, achieved the best results, and ABOD, that presented the poorer performance was the algorithm with worst EDP index. Besides, as shown in Figure 1 [E], the EDP results presented a high correlation with the heat imposed on the device, which showed that as better the algorithms manage to balance the trade-off between performance and power saving, the better they can achieve thermal-efficiency.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on May 01,2023 at 19:58:20 UTC from IEEE Xplore. Restrictions apply.

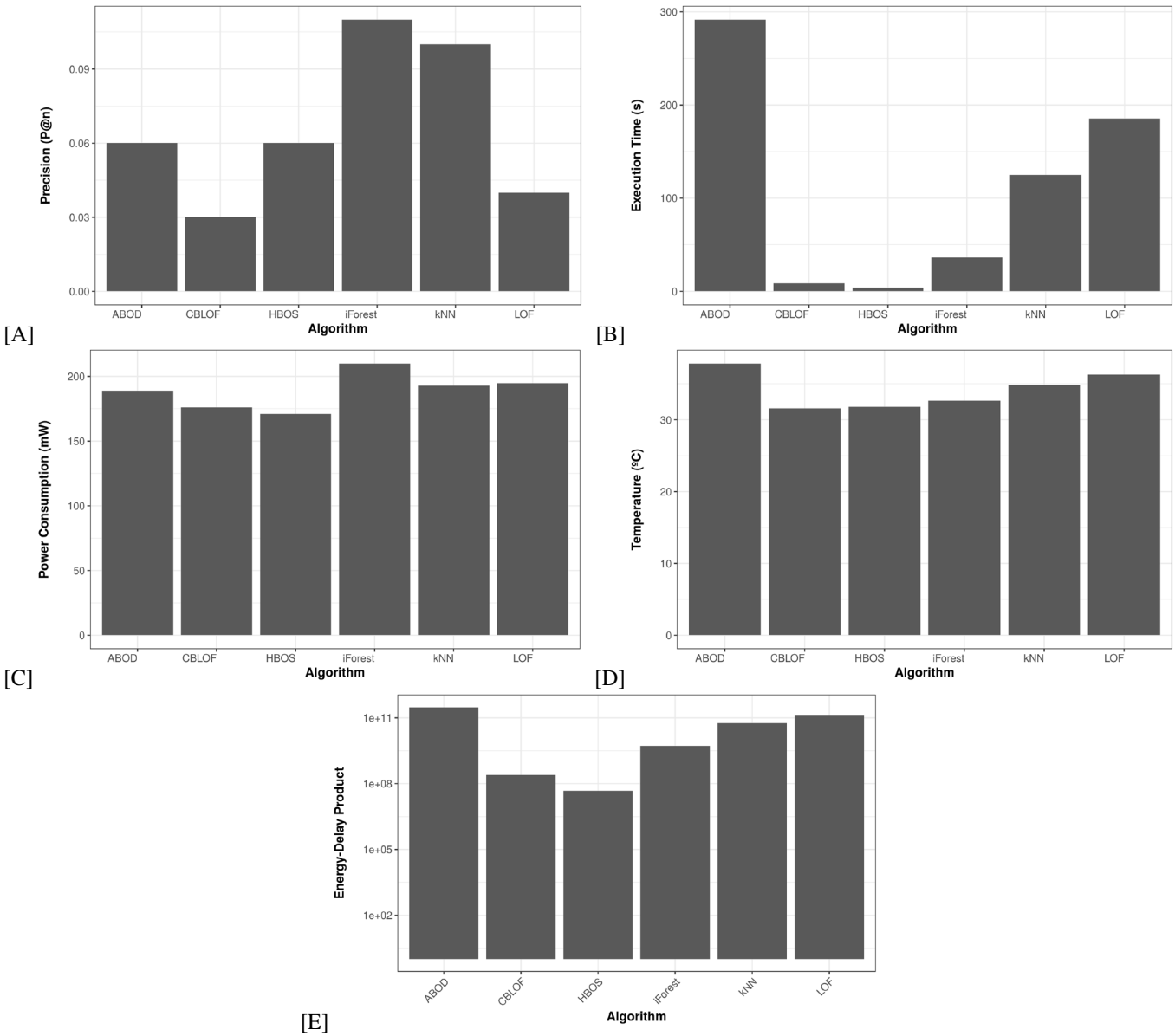

Figure 1. Comparison of six anomaly detection algorithms running on an embedded device regarding accuracy, performance, power consumption, and temperature.

VI. Conclusions and Future Work

The advances of areas like computing and engineering drove into the advent of the Internet of Things, in which electronic devices called embedded systems such as sensors and actuators are distributed across the environment in order to collect data and help in decision-making based on events. In this context, there is a concern about ensuring the security of IoT networks since embedded systems used to collect and pre-process data usually are not physically protected against attackers. Thus, anomaly detection techniques have been used to improve the security of embedded systems and the reliability of the data being processed.

Embedded systems usually have constrained resources that may hamper the use of too expensive anomaly detection algorithms. Besides, these devices are typically battery-dependent, and high operating temperatures may impact their efficiency or even reduce their lifetime. Some previous investigations compared anomaly detection techniques; however, to the best of our knowledge, none of these studies cover the analysis of power consumption and temperature. In this sense, we present an analysis of six anomaly detection algorithms running on an embedded device, and we discuss the impact of each of these techniques regarding power consumption, temperature, accuracy, and performance.

The Histogram-Based Outlier Score algorithm presented the best results regarding performance and power consumption due to its lower resources requirements and its linear time complexity. On the other hand, Isolation Forest presented the highest accuracy due to its approach that focuses on identifying anomalies instead of profiling normal points.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on May 01,2023 at 19:58:20 UTC from IEEE Xplore. Restrictions apply.

As future work, we intend to develop a resources management tool that will employ anomaly detection techniques to analyze the resource usage of multiple inter-connected edge devices. Based on the feedback of anomaly detection algorithms, our tool will use consolidation and load balancing strategies to improve devices' resource utilization.

VII. Acknowledgements

This work was supported by the PDTI Program, funded by Dell Computadores do Brasil Ltda (Law 8.248/91).

References

[1] A. Majeed and M. Ali, "How internet-of-things (iot) making the university campuses smart? qa higher education (qahe) perspective," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 646–648.

[2] J. chun Zhao, J. feng Zhang, Y. Feng, and J. xin Guo, "The study and application of the iot technology in agriculture," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 2, July 2010, pp. 462–465.

[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[4] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[5] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25 – 37, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517300802

[6] P. S. S. de Souza, W. dos Santos Marques, F. D. Rossi, G. da Cunha Rodrigues, and R. N. Calheiros, "Performance and accuracy trade-off analysis of techniques for anomaly detection in iot sensors," in *Information Networking (ICOIN), 2017 International Conference on*. IEEE, 2017, pp. 486–491.

[7] B. Zhang, N. Mor, J. Kolb, D. S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E. Lee, and J. Kubiatowicz, "The cloud is not enough: Saving iot from the cloud," in *7th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 15)*, 2015.

[8] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "Iot healthcare analytics: The importance of anomaly detection," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, March 2016, pp. 994–997.

[9] S. N. Narayanan, S. Mittal, and A. Joshi, "Obd_securealert: An anomaly detection system for vehicles," in *Smart Computing (SMARTCOMP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[10] S. Trilles, Ò. Belmonte, S. Schade, and J. Huerta, "A domain-independent methodology to analyze iot data streams in real-time. a proof of concept implementation for anomaly detection from environmental data," *International Journal of Digital Earth*, vol. 10, no. 1, pp. 103–120, 2017.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[12] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[13] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[15] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "Sok: Security evaluation of home-based iot deployments," in *IEEE S&P*, 2019, pp. 208–226.

[16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[17] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the internet of things using big data analytics," *Comput. Netw.*, vol. 101, no. C, pp. 63–80, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2015.12.023

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, 07 2009.

[19] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous, "A comparative study of anomaly detection techniques for smart city wireless sensor networks," *Sensors*, vol. 16, no. 6, p. 868, 2016.

[20] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[21] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *arXiv preprint arXiv:1901.01588*, 2019. [Online]. Available: https://arxiv.org/abs/1901.01588

[22] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ser. ICDM '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 413–422. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2008.17

[23] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 427–438. [Online]. Available: http://doi.acm.org/10.1145/342009.335437

[24] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/342009.335388

[25] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[26] ——, "Squeezer: an efficient algorithm for clustering categorical data," *Journal of Computer Science and Technology*, vol. 17, no. 5, pp. 611–624, 2002.

[27] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

[28] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 444–452. [Online]. Available: http://doi.acm.org/10.1145/1401890.1401946

[29] S. Rayana, "ODDS library," pp. 1–1, 2016. [Online]. Available: http://odds.cs.stonybrook.edu

[30] S. Mittal, "A survey on optimized implementation of deep learning models on the nvidia jetson platform," *Journal of Systems Architecture*, 2019.

[31] T. S. Chis and P. G. Harrison, "Performance-energy trade-offs in smartphones," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2016, pp. 127–135.

Authorized licensed use limited to: Pontificia Universidade Catolica do Rio Grande do Sul (PUC/RS). Downloaded on May 01,2023 at 19:58:20 UTC from IEEE Xplore. Restrictions apply.