Tandem Outlier Detectors for Decentralized Data

Marco Heyden
marco.heyden@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Jürgen Wilwer
juergen.wilwer@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Edouard Fouché
edouard.fouche@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Vadim Arzamasov
vadim.arzamasov@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Steffen Thoma
thoma@fzi.de
FZI Research Center for Information Technology
Karlsruhe, Germany

Sven Matthiesen
sven.matthiesen@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Thomas Gwosch
thomas.gwosch@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

ABSTRACT

Today, the collection of decentralized data is a common scenario: smartphones store users' messages locally, smart meters collect energy consumption data, and modern power tools monitor operator behavior. We identify different types of outliers in such data: *local*, *global*, and *partition outliers*. They contain valuable information, for example, about mistakes in operation. However, existing outlier detection approaches cannot distinguish between those types. Thus, we propose a "tandem" technique to join "local" and "federated" outlier detectors. Our core idea is to combine outlier detection on a single device with latent information about devices' data to discriminate between different outlier types. To the best of our knowledge, our method is the first to achieve this. We evaluate our approach on publicly available synthetic and real-world data that we collect in a study with 15 participants operating power tools.

CCS CONCEPTS

• **Computing methodologies → Anomaly detection**; *Distributed artificial intelligence.*

KEYWORDS

outlier detection, federated learning, edge computing

ACM Reference Format:
Marco Heyden, Jürgen Wilwer, Edouard Fouché, Vadim Arzamasov, Steffen Thoma, Sven Matthiesen, and Thomas Gwosch. 2022. Tandem Outlier Detectors for Decentralized Data. In *34th International Conference on Scientific and Statistical Database Management (SSDBM 2022), July 6–8, 2022, Copenhagen, Denmark.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3538712.3538748

1 INTRODUCTION

Edge computing decentralizes data analysis by moving computation tasks away from a central processing unit, closer to the edge of a network of edge devices. Each edge device, like a smart meter or power tool, collects data about its environment or usage. Thus, each device has a "local" data set which is a share of all the data generated in the network. We call it a *partition.*

Finding outliers in partitions is crucial for many applications including predictive maintainance [14, 23] and energy consumption analysis [11]. In such applications, outlier labels are often hard to obtain as it would require asking the owner of the edge device. Outlier detection for edge computing should thus be unsupervised.

Existing approaches for unsupervised outlier detection in decentralized data fall into three categories. First, *centralized* approaches transfer the partitions to a cloud instance for further processing [9] which is inefficient and raises privacy concerns. Second, *local* approaches only consider individual partitions without sharing information between peer devices. Third, *decentralized* approaches leverage Federated Learning (FL) [10] to collaboratively train a shared outlier detector [12, 17, 21]. FL does not require any exchange of raw data and is thus appealing if data is plenty or privacy-sensitive. We refer to outlier detection with FL as *federated* outlier detection.

We observe that local and federated outlier detection in isolation fail to identify the different types of outliers that exist in decentralized data, namely *local outliers*, *global outliers*, and *partition outliers*, as the 1-dimensional example in Figure 1 illustrates. Global outliers are observations that deviate from the data on all devices. Local outliers are observations that deviate from their partition, but do not deviate from data from other devices. Last, a partition can also be outlying (a partition outlier) if it is significantly different from other partitions. Consider the following real-world example:

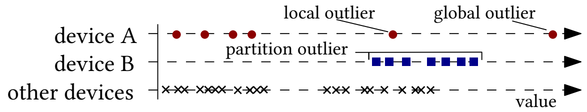


Figure 1: Outlier types in decentralized data

Example 1 (Connected power tools). *Workers on construction sites use multiple power tools of the same type for different tasks. For example, they keep a drill head for wood attached to device A and for metal to device B. As long as the workers use the drill heads for their intended materials, devices A and B record characteristic data patterns for the specific tasks. However, if a worker mistakenly uses device A to drill metal, the new data is different from the existing patterns in the partition. Such misuse leads to a local anomaly because*

SSDBM 2022, July 6–8, 2022, Copenhagen, Denmark

the recorded pattern is novel on device A, yet frequent on device B. In contrast, mistakes in power tool operation (e.g., drilling in an unintended material) may lead to a pattern that has not been observed on any device. Hence, these outliers are global.

Next, local observations may reflect a worker's skills, as unskilled workers tend to make systematic mistakes. Consequently, many observations on their device deviate from the data of other workers. We call this a partition outlier.

In summary, outlier detection in decentralized data should (1) identify local, global and partition outliers, (2) work unsupervised, and (3) utilize device collaboration without raw data exchange.

Contributions

Our main contribution is an outlier detection framework that fulfills these requirements. **It combines local and federated anomaly detection in tandem** which allows to distinguish between the three outlier types without disclosing sensitive information.

We conduct a real-world study in the field of power tools. The open source[1]data set contains high-quality sensor readings from the usage patterns of power tools from 15 individuals.

Our experiments show the practical value of our tandem on the real world use case of smart power tools. The code for all experiments is available on GitHub[1] for reproducibility.

2 NOTATION

We consider a network $N = \{c_1, c_2, \dots, c_{|N|}\}$ with $|N|$ devices, or "clients". We associate a partition, i.e., a set of observations, $d_i = \{x_{i1}, x_{i2}, \dots, x_{i|d_i|}\}$ to each client $c_i \in N$ where $x_{ij} \in \mathbb{R}^d$. We call $D = d_1 \cup d_2 \cup \dots \cup d_{|N|}$ the global data set of size $|D|$. We assume that the observations $x_{ij} \in d_i$ come from an underlying distribution X_i. The joint distribution of all X_i is the global distribution X^G. Our goal is to distinguish three types of outliers:

Definition 1 (Partition Outlier). *A partition d_i is a partition outlier if the probability that d_i was sampled from $X^G \setminus X^i$ is less than α, i.e., $Pr(d_i \sim X^G \setminus X_i) < \alpha$.*

Definition 2 (Local outlier). *An outlier x_{ij} is local if it is outlying only with respect to d_i but not with respect to D.*

Definition 3 (Global outlier). *An outlier x_{ij} is global if it is outlying with respect to D.*

3 RELATED WORK

FL addresses decentralized training of neural networks, and Federated Averaging is arguably its most well known variant [10]: Each device trains a model on local data, which a server then averages into a global model. This process is repeated until convergence.

Existing work in FL for outlier detection tends to address outlier detection purely via FL [16, 17], or provides application specific solutions [12, 15]. However, no approach combines federated outlier detection with local outlier detection. Hence, they are unable to distinguish between the three outlier types that we listed earlier.

Previous work has used different notions of local and global outliers. In distributed settings, an anomaly is local if it occurs multiple times in a small geographical region [3, 20], e.g., if multiple devices that are close to each other observe the same, unusual phenomenon. Global outliers, in contrast, do not show such a spatial relationship. In our setting, devices can be mobile and their spatial relationship is irrelevant.

Last, [22] provide a definition similar to ours: local outliers are those in a single partition, and global outliers are outlying in the union of all partitions. Accordingly, an outlier could be global *and* local depending on the data available for outlier detection. Our definitions of local and global outliers in turn are mutually exclusive.

Next, our definition of "partition outlier" is related to the notions of "second-order outlier" [18], or "infected device" [12] in the literature. Existing approaches to detect such outliers share sensitive information such as individual observations [7], or mean and variance [6] of the data. In contrast, our approach only uses aggregated outlier scores which do not contain such sensitive information.

There exist several approaches for outlier detection in Wireless Sensor Networks, a related field. However, they are not suitable to our setting, as they either do not (1) handle multivariate data, (2) distinguish between different outlier types, (3) preserve privacy and security [8, 19] or (4) consider mobile devices [2, 5, 20, 22].

4 OUR APPROACH

Our approach consists of two outlier detectors per device referred to as F (federated) and L_i (local). They differ in the data used for outlier detection: L_i acts locally and only considers the partition d_i of the associated device. F is a federated outlier detector and identifies anomalies in the global data set D.

For each observation $x_{ij} \in d_i$, F and L_i return outlier scores, denoted as os^F_{ij} and os^L_{ij}, which are random samples from unknown outlier score distributions OS^F and OS^L_i.

Like a tandem, F and L_i work together to perform a common task: finding and distinguishing outliers in decentralized data. First, the clients fit L_i on the local data d_i and F through FL. Afterwards, they obtain sets of outlier scores $os^F_i = \{os^F_{i1}, os^F_{i2}, \dots\}$, $os^L_i = \{os^L_{i1}, os^L_{i2}, \dots\}$ for their data. Finally, each client identifies local and global outliers (see also Section 4.1) based on a client-specified detection thresholds ε^F_i and ε^L_i and evaluates if the local data is a partition outlier (see also Section 4.2).

Related work suggests that Autoencoders (AEs), a type of Neural Network that learns a compressed representation of the data, are particularly useful for outlier detection [13]. In addition, one can train them with FL. Hence we use them in our framework.

4.1 Identification of local and global outliers

Finding outliers in a decentralized setting would require exchanging sensitive information between clients to estimate X^G based on the global data set D. It is thus not possible if privacy is a concern. Our method instead uses the empirical distribution of outlier scores, assuming that it is a good proxy for X_i and X^G.

F identifies outliers w.r.t. the global data set D. L_i, in contrast, finds outliers in an isolated partition d_i. As a result, F can only identify global outliers while L_i can find local and global outliers but is unable to distinguish them. Combining F and L_i is thus key to discriminate between local and global anomalies.

Our approach identifies and classifies an observation x_{ij} (1) as **global outlier** if both F *and* L_i identify it as an outlier, i.e., if $os^L_{ij} >$

[1] https://github.com/heymarco/TandemOutlierDetection

ε_i^L and $os_{ij}^F > \varepsilon_i^F$ and (2) as **local outlier** if the observation is only anomalous w.r.t. the local data set, i.e., if $os_{ij}^L > \varepsilon_i^L$ and $os_{ij}^F \leq \varepsilon_i^F$.

4.2 Identification of partition outliers

We observe that outlier scores in partition outliers are significantly higher than those of normal devices if they use identical outlier detectors. Hence, we can identify partition outliers by evaluating for which clients the outlier scores obtained with F – which is identical for all c_i – deviate significantly from the rest. For that we use a decentralized variant of the one-sided Mann-Whitney-U test:

1. ScoreAggregation: Each client c_i divides the sorted federated outlier scores into bins of size b and computes the mean for each bin. It transfers the set of means, os_i^*, to the server. This step reduces data transmission b-fold and masks information about point outliers.
2. ServerEvaluation: Next, the server computes the p-value for each client c_i using the Mann-Whitney-U statistic between os_i^* and $(os_1^* \cup os_2^* \cup \ldots \cup os_{|N|}^*) \setminus os_i^*$.
3. ClientEvaluation: At last, a client evaluates if $p_i < \alpha_i$, a client-dependent significance level. For brevity, we will refer to α_i as α from now on.

Note that step (1) introduces a trade-off between statistical power and data transmission – large choices of b provide effective data compression and small choices improve the test's statistical power. We will give a recommendation for b in Section 5.1.

5 EXPERIMENTS

Next, we evaluate our approach. As an ablation, we compare to *local* and *federated* outlier detection on the synthetic data. Our results are the average of 10 repetitions with the random seed set to the repetition index.

We use Autoencoders (AE) as F and L_i with one hidden layer of size $\eta \cdot d$ [13], with $\eta = 0.7$ on synthetic data and $\eta = 0.4$ on power tool data. We apply ReLu activation in the hidden and Sigmoid in the output layer on the synthetic data. The output layer uses a linear activation function on the power tool data set due to standard normalized input. In FL, the clients train for one local epoch $E = 1$ and 20 communication rounds with a batch size of $B = 32$.

5.1 Evaluation on Synthetic Data

5.1.1 Local and global outliers. We generate synthetic data from a 10-component Gaussian mixture (GMM) with 10 dimensions representing X^G. We use 30 clients with $|d_i| = 1000$ observations sampled from five randomly chosen GMM-components representing X^i. Hence, the global data set $D = d_1 \cup d_2 \cup \ldots \cup d_{30}$ has ten components, while each d_i only contains five components.

We sample from those patterns not present in the local data to introduce local outliers and random observations for global outliers. Each partition receives 4 % outliers. We vary the ratio between local and global outliers over the course of 6 experiments.

For our approach, we set ε_i^F and ε_i^L to the 96th percentiles of os_i^F and os_i^L. For the baselines, i.e., local and federated outlier detection, we also use the 96th percentile. Further, we assume that local outlier detection always predicts *local* outliers and federated outlier detection always *global* outliers, as suggested in [22].

Figure 2 reports precision, recall, and F1 score. We consider a classification correct iff the detector classified the exact label, i.e., *inlier*, *local outlier* or *global outlier*.

One can see that our approach outperforms federated and local outlier detection, achieving a higher precision and recall, except if the data contains solely local outliers. To our surprise, the tandem even outperforms federated outlier detection if the data contains only global outliers – a task at which we expected the federated approach to excel.

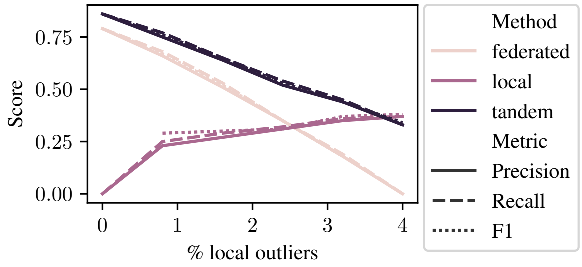


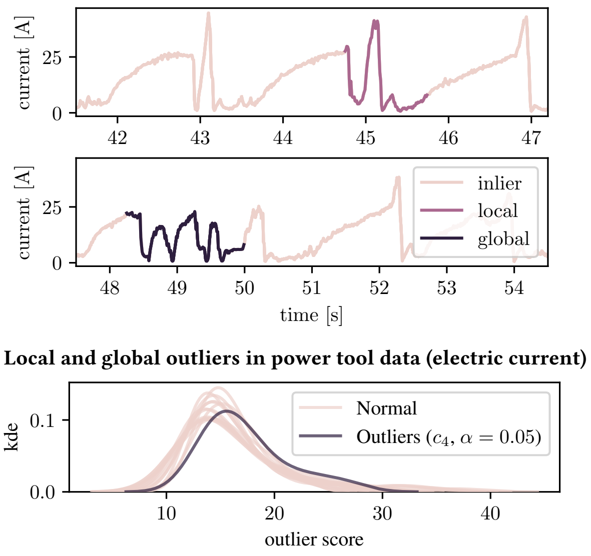
Figure 2: Results for local and global outliers in synthetic data. The amount of global outliers is 4% minus % local outliers.

$\lvert d_i\rvert$	b	shift [std]				
		0.0	0.25	0.5	0.75	1.0
100	1	0.39	0.36	0.14	0.16	0.09
	10	0.47	0.41	0.28	0.27	0.14
	100	0.45	0.39	0.26	0.25	0.13
1000	1	0.68	0.34	0.18	**0.02**	**0.00**
	10	0.66	0.34	0.22	**0.04**	**0.02**
	100	0.60	0.41	0.31	0.12	0.08
	1000	0.66	0.38	0.30	0.09	0.07
10000	10	0.56	0.19	**0.01**	**0.00**	**0.00**
	100	0.55	0.30	0.10	**0.03**	**0.00**
	1000	0.53	0.38	0.26	0.10	**0.03**
	10000	0.55	0.28	0.10	0.06	**0.03**

Table 1: Detection of partition outliers: p-values for different partition sizes, b, and shift distances. $p < \alpha = 0.05$ are bold.

5.1.2 Partition outliers. We use the same setting as before and shift the data of c_0 by *shift* standard deviations. Table 1 reports the p-values for c_0 for different choices of b, $|d_i|$, and *shift*.

One observes that our approach returns high p-values if *shift* = 0.0, i.e., if the partition is an inlier. Also, one notices that, for a fixed b, p decreases the larger *shift*. This confirms that p is indeed a good indicator of the similarity between $X^G \setminus X_0$ and X_0. Last, one can see that our test becomes more sensitive the larger $|d_i|$ and the smaller b. Hence, we suggest to choose b smaller, say $b = 10$, for smaller partitions and larger, e.g., $b > 100$, for larger partitions.



(a) Local and global outliers in power tool data (electric current)

(b) Outlier score distributions in power tool data (electric current)

Figure 3: Evaluation on power tool data

5.2 Power Tool Case Study

We conducted a user study with 15 participants operating a cordless screwdriver PDC 18/4 Quaddrive by Festool. An attached data logger [4] recorded battery current and voltage, and inertial measurement units (IMUs) measured accelerations and angular velocities. We extracted $d = 1,665$ features using TSFEL [1] after removing inactive phases of the power tool.

First, we examine local and global anomalies for one of the power tool operators, c_{15}. Figure 3a shows the battery current for two exemplary anomalies. A normal screwing process, e.g., in the interval $[50.5s, 52.5s]$, shows a characteristic pattern of increasing current which spikes when the screw head enters the wood. This pattern is absent in the identified global anomaly. In the video, we could see that the operator slipped off the screw head.

In contrast, the local outlier in the graph has a high similarity to regular executions. Further investigation did not reveal any significant mistakes except minor deviations from the usual usage patterns, which were confirmed by video analysis. Local anomaly detection alone would have raised a false alarm in such a case.

Next, we evaluate partition outliers. Figure 3b shows the distribution of os_i^F for each participant. We use our recommended $b = 10$ as $|d_i| \in [240, 718]$. Our approach identifies c_4 as partition outlier. The graph shows that the outlier scores of c_4 are indeed significantly higher than those of the other devices. However, the anomaly of c_4 would have been undetectable without incorporating knowledge about other clients. In our study, we used the information to gain knowledge about the power tool usage of non-professionals.

6 CONCLUSION

This paper presents a tandem approach for unsupervised outlier detection in decentralized data. We show that existing approaches, i.e., local and federated anomaly detection, fail to classify local and global outliers reliably, while our approach does. Also, we provide a technique to identify partition outliers with a client-specific level of confidence. We conduct a user study with 15 participants in the power tools domain, demonstrating the applicability of our approach in practice.

In the future, we plan to extend our approach the data stream setting and apply it to applications in healthcare and energy.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) Research Training Group GRK 2153: *Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation.*, and by the German Federal Ministry of Education and Research (BMBF), project number 16SV8280.

REFERENCES

[1] M. Barandas, D. Folgado, et al. 2020. TSFEL: Time Series Feature Extraction Library. *SoftwareX* (2020).
[2] Sourabh Bharti, K. K. Pattanaik, and Anshul Pandey. 2020. Contextual outlier detection for wireless sensor networks. *JAIHC* (2020).
[3] H. H. W. J. Bosman, G. Iacca, et al. 2017. Spatial anomaly detection in sensor networks using neighborhood information. *Inf. Fusion* (2017).
[4] M. Dörr, J. Peters, and S. Matthiesen. 2021. Data-Driven Analysis of Human-Machine Systems – A Data Logger and Possible Use Cases for Field Studies with Cordless Power Tools. In *IHIET*.
[5] L. Fang and S. Dobson. 2014. Data Collection with In-network Fault Detection Based on Spatial Correlation. In *ICCAC*.
[6] M. Gabel, A. Schuster, and D. Keren. 2014. Communication-Efficient Distributed Variance Monitoring and Outlier Detection for Multivariate Time Series. In *IPDPS*.
[7] H. Kumarage, I. Khalil, et al. 2013. Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. *JPDC* (2013).
[8] H. Kumarage, I. Khalil, and Z. Tari. 2015. Granular Evaluation of Anomalies in Wireless Sensor Networks Using Dynamic Data Partitioning with an Entropy Criteria. *IEEE Trans. Computers* (2015).
[9] T. Luo and S. G. Nagarajan. 2018. Distributed Anomaly Detection Using Autoencoder Neural Networks in WSN for IoT. In *ICC*.
[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.
[11] R. Moghaddass and J. Wang. 2018. A Hierarchical Framework for Smart Grid Anomaly Detection Using Large-Scale Smart Meter Data. *IEEE Trans. Smart Grid* (2018).
[12] T. D. Nguyen, S. Marchal, et al. 2019. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In *ICDCS*.
[13] D. Popovic, E. Fouché, and K. Böhm. 2019. Unsupervised Artificial Neural Networks for Outlier Detection in High-Dimensional Data. In *ADBIS*.
[14] J. Rabatel, S. Bringay, and P. Poncelet. 2011. Anomaly detection in monitoring sensor data for preventive maintenance. *Expert Syst. Appl.* (2011).
[15] R. A. Sater and A. B. Hamza. 2020. A Federated Learning Approach to Anomaly Detection in Smart Buildings. *CoRR* (2020).
[16] J. Schneible and A. Lu. 2017. Anomaly detection on the edge. In *MILCOM*.
[17] S. Singh, S. Bhardwaj, et al. 2021. Anomaly Detection Using Federated Learning. In *ICAIA*.
[18] S. Suthaharan, M. Alzahrani, et al. 2010. Labelled data collection for anomaly detection in wireless sensor networks. In *ISSNIP*.
[19] X.-Y. Xiao, W.-C. Peng, et al. 2007. Using sensorranks for in-network detection of faulty readings in wireless sensor networks. In *MobiDE*.
[20] X. Yu, H. Lu, et al. 2020. An adaptive method based on contextual anomaly detection in Internet of Things through wireless sensor networks. *IJDSN* (2020).
[21] K. Zhang, Y. Jiang, et al. 2021. Federated Variational Learning for Anomaly Detection in Multivariate Time Series. *CoRR* abs/2108.08404 (2021).
[22] Y. Zhang, N. Meratnia, and P. J. M. Havinga. 2010. Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Commun. Surv. Tutorials* (2010).
[23] P. Zhao, M. Kurihara, et al. 2017. Advanced correlation-based anomaly detection method for predictive maintenance. In *ICPHM*.