Deep Anomaly Detection on Tennessee Eastman Process Data

Fabian Hartung, Billy Joe Franks, Tobias Michels, Dennis Wagner, Philipp Liznerski, Steffen Reithermann, Sophie Fellenz, Fabian Jirasek, Maja Rudolph, Daniel Neider, Heike Leitte, Chen Song, Benjamin Kloepper, Stephan Mandt, Michael Bortz, Jakob Burger, Hans Hasse, and Marius Kloft*

DOI: 10.1002/cite.202200238

Supporting Information available online

This paper provides the first comprehensive evaluation and analysis of modern (deep-learning-based) unsupervised anomaly detection methods for chemical process data. We focus on the Tennessee Eastman process dataset, a standard litmus test to benchmark anomaly detection methods for nearly three decades. Our extensive study will facilitate choosing appropriate anomaly detection methods in industrial applications. From the benchmark, we conclude that reconstruction-based methods are the methods of choice, followed by generative and forecasting-based methods.

Keywords: Anomaly detection, Benchmark, Chemical process data, Tennessee Eastman process, Time series

Received: December 16, 2022; *revised:* March 08, 2023; *accepted:* March 27, 2023

1 Introduction

Anomaly detection, i.e., detecting data that deviates from normality, is a fundamental method in machine learning and artificial intelligence. It is significant in many application domains, from detecting fake reviews in online shopping portals and bots in social networks to tumor detection and industrial fault detection. Anomaly detection is especially significant in safety-critical applications. While an undetected fake review in an online shopping portal may be harmless, failing to recognize anomalies in a chemical plant or a self-driving car may put lives at stake.

In chemical plants, most data is recorded during regular or problem-free operation – the normal data. Anomalies, in contrast, occur very rarely, and they can appear to the process or control engineers to be nominal behavior. Here, computing methodology naturally comes into play. Machine learning enables searching massive datasets and accurately detects anomalies, even when they are rare [1].

There is a large body of literature on detecting anomalies in chemical processes using machine learning [2–4]. Over the past three decades, the Tennessee Eastman process (TEP) has arisen as a litmus test for learning-based anomaly detection on chemical process data. Virtually any newly proposed method is benchmarked by default on the TEP dataset, originally recorded by Downs and Vogel in 1993 [5] using a model-based TEP simulator for data generation and modified by Rieth [6].

[1,2]Fabian Hartung, [1]Billy Joe Franks, [1]Tobias Michels, [1]Dennis Wagner, [1]Philipp Liznerski, [1]Steffen Reithermann, [1]Jun.-Prof. Dr. Sophie Fellenz, [1]Jun.-Prof. Dr.-Ing. Fabian Jirasek https://orcid.org/0000-0002-2502-5701, [3]PhD Maja Rudolph, [4]Prof. Dr. Daniel Neider, [1]Prof. Dr. Heike Leitte, [5]Dr. Chen Song, [5]Dr. Benjamin Kloepper, [6]Prof. Dr. Stephan Mandt, [7]Prof. Dr. Michael Bortz, [8]Prof. Dr.-Ing. Jakob Burger, [1]Prof. Dr.-Ing. Hans Hasse, [1]Prof. Dr. Marius Kloft (kloft@cs.uni-kl.de)

[1]Technische Universität Kaiserslautern, Gottlieb-Daimler-Straße 47, 67663 Kaiserslautern, Germany.

[2]BASF SE, Gas Treatment Technology, Carl-Bosch-Straße 38, 67063 Ludwigshafen am Rhein, Germany.

[3]Bosch AI, 2555 Smallman St, Pittsburgh, PA 15222, USA.

[4]Technische Universität Dortmund, August-Schmidt-Straße 1, 44227 Dortmund, Germany.

[5]ABB Corporate Research Center Ladenburg, Wallstadter Straße 59, 68526 Ladenburg, Germany.

[6]University of California, Irvine, 4228 Donald Bren Hall, Irvine 92697, USA.

[7]Fraunhofer ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany.

[8]Technische Universität München, Campus Straubing, Uferstraße 53, 94315 Straubing, Germany.

However, except for [7–12], all papers evaluate shallow unsupervised anomaly detection methods (not including neural networks) on the TEP dataset. But shallow machine learning is not adequate for complex, structured data, such as the time series occurring in chemical plants and the TEP. On such data, most of the many seminal advances in artificial intelligence during the last decade have been enabled by deep neural networks.

In 2018, Ruff et al. [13] provided one of the earliest general-purpose deep learning-based approaches to anomaly detection. The paper triggered a wave of follow-up work, resulting in the new field of 'deep anomaly detection' [14]. Over the past four years, the detection error of unsupervised anomaly detection methods has been reduced drastically, from 35 % (best shallow method, 2017) to 1 % (best deep method, 2021) on CIFAR-10-AD, a standard anomaly detection benchmark dataset [13, 15]. Since then, deep anomaly detection approaches have been widely adopted in industrial practice.

Most recent breakthroughs in modern anomaly detection have been achieved on image data. However, the data in chemical plants – and particularly the TEP – are time series. Time series exhibit intriguing temporal interdependencies, well-suited for deep learning. Very recently, the first deep anomaly detection methods on time-series data were introduced, and their high potential tested on various benchmarks [16]. To date, there exist some 30 methods based on neural networks for anomaly detection on time series.

However, the research on the TEP has not caught up yet with these highly significant advances in unsupervised deep anomaly detection on time series. There exists no compelling up-to-date comparison of modern methods, most of which have been developed within the last two years. Thus, it is unclear which methods should ideally be used on such data to achieve maximal detection performance. Using inferior detection methods may lead to unnecessary errors or even put lives at risk when using them for real operation in plants.

With the present work, we intend to change this. This paper evaluates and compares all 27 unsupervised deep anomaly detection methods for time series existing to date, regarding their detection accuracy on the TEP data.[1)] The analysis represents the first – and by far the most comprehensive – comparison of modern unsupervised anomaly detection methods on chemical process data.

Our analysis also yields insights into which anomaly detection methods might be most suitable for application to real chemical process data. Establishing deep anomaly detection in real chemical processes would open the route for new, yet unexplored, ways to control these processes – with a perspective to advance autonomously running chemical processes.

1) Status mid/end 2022. We will maintain a website continuously updating the results for new methods proposed after the publication of this paper.

2 Related Work

Early papers on deep anomaly detection (AD) on times series were based on either reconstruction or forecasting objectives. Reconstruction-based approaches train an autoencoder (AE) on mostly normal training data so that the AE learns to compress and reconstruct normal data well. Samples not reconstructed well are considered anomalous. The deviation from the reconstruction to the input is the anomaly score [17–26]. In recent work, Rewicki et al. showed the advantage of AEs over other deep learning methods by comparing three classic machine learning methods and three deep learning-based methods for AD on time series. They concluded that, depending on the type of anomaly and computing time, classical approaches could outperform [54]. Forecasting-based models extrapolate a series' current and past data to predict future time steps. The anomaly score is the difference between the predicted and the actual future data [27–31]. Typically, both reconstruction and forecasting-based methods reconstruct each time step and aggregate their anomaly scores for an anomaly score of the entire time series.

Another branch of AD methods is based on generative models such as variational autoencoders (VAEs) [32–37] and generative adversarial neural networks (GANs) [38–43]. GANs jointly train two networks: a discriminator network to distinguish between accurate and generated data and a generator network to create samples that fool the discriminator. Anomaly scores are either based on the discriminator or are the deviation between the test sample and the best-fitting generated data sample. Some methods combine the above-mentioned methods to get the best parts from all worlds [44, 45].

Inspired by the success of supervised classifiers, there is also a paradigm called "one-class classification" [13]. This work trains a network to map normal samples to a hypersphere [13] or hyperplane [46] and anomalous data away from them. This paradigm has recently been used for AD on time series [44, 47]. A more direct application of classifiers for AD requires anomalous training samples. Since AD is typically unsupervised, these samples are not available. One approach to solve this issue is using random internet data as auxiliary anomalies during training. This approach is termed outlier exposure and is successful on images [15, 48]. However, pertinent data is unavailable for time series, so Goyal et al. proposed to train a network to distinguish between normal training data and synthetically generated anomalies [49]. The classifier's certainty for the anomalous class defines the anomaly score for test samples. The most recent approach to time series AD uses self-supervised learning [16]. This method designs an auxiliary training objective. Normal data samples are transformed, and the network has to predict which type of transformation has been applied. Since, for anomalous data, a correct prediction will be difficult, the value of the method's decision certainty is the anomaly score for test samples.

3 Benchmarking Deep Time Series Anomaly Detection on the TEP

In this section, a more detailed explanation of the evaluation follows. First, we present the TEP data and explain the metrics used for the review. Finally, the implementation and evaluation protocol are presented.

3.1 TEP Dataset

TEP was based on an existing plant and the processes running in it. The data itself is synthetic, i.e., a simulation of the plant. It consists of five main modules, each a two-stage reactor, a condenser, a vapor-liquid separator, a stripper, and a reboiler, as well as 11 pneumatic valves, two pumps, and a compressor [50].

The version of the TEP data used here is available online [6] and is referenced in [51]. In addition to error-free data on which the algorithms are to be trained, it contains 20 different types of erroneous data sets and their complete simulation. Of these 21 data sets, there are 500 other runs, each of which is initialized with a different random value. The time points in each sample are generated every three minutes for 25 h for the training data and 48 h for the test data with 53 parameters.

3.2 Metrics

To compare and evaluate the examined algorithms with each other, a metric is necessary that measures the quality of the methods. Work on AD uses different evaluation metrics depending on the data. Some metrics, like the F1-score, require a binary decision, i.e., model outputs in {0, 1} where 0 denotes normal and 1 anomalous. Others, like the receiver operator characteristic or precision-recall curve, work with continuous anomaly scores. For AD on time series, the F1-score and area under the precision-recall curve are the most commonly used metrics, which is why we evaluate the methods in this paper using both.

An anomaly detector generates an anomaly score for each point in time of a time series. If this value exceeds a certain threshold, the respective method determines this point in time as an anomaly. The F1-score considers four options of evaluation for each time point: true positive (TP – a correctly detected anomaly), false negative (FN – an anomaly that was not detected), true negative (TN – a correctly identified normal point), and false positive (FP – a normal point mistakenly detected as an anomaly). With these four classes, two metrics can be calculated. One is precision, the proportion of TP among all detected anomalies (TP+FP), and the other is recall, the balance of TP anomalies among all true anomalies (TP+FN). Intuitively, precision describes the accuracy with which a detected anomaly is anomalous, and recall describes the accuracy with which the model detects true anomalies. The F1-score combines precision and recall in one metric, which can be calculated at every point of the time series. These F1-scores are averaged over the whole time series to receive the total F1-score.

The area under the precision-recall curve (AUPRC) can be used as a second metric for comparing methods. For every threshold, its respective recall and precision are calculated. As the threshold decreases, the recall increases to 1, which is plotted on the x-axis. The precision is plotted on the y-axis and can be arbitrary but generally decreases as the recall increases. The AUPRC measures the model's overall performance for any threshold. In essence, the higher the AUPRC, the higher the precision for any recall. In practice, there is a real-world cost associated with both FN and FP. Generally, the cost for undetected anomalies (FN) is higher than the cost of falsely detecting an anomaly (FP). However, the specific costs need to be defined case-by-case; therefore, the optimal threshold depends on the particular use case. The AUPRC is a good metric in case the specific costs are unknown since the higher the AUPRC, the lower these associated costs are expected to be.

3.3 Evaluation and Implementation

For an equal and fair evaluation of the considered methods, all methods were implemented in the same Python environment and were trained and evaluated using PyTorch [52]. Since some methods require an unlabeled validation set to adjust the parameters of the anomaly detector, a quarter of the training dataset was separated for this purpose. The test dataset was divided into five folds of equal size to adjust the hyperparameters of each method by optimizing them on each fold and evaluating the performance of the best model with the remaining folds. To avoid time dependencies, directly neighboring folds were excluded. Finally, all folds were averaged, the methods were compared using the best F1-score, and AUPRC received the best grid parameters. For better comparability, the size of the parameter grid of each method was chosen so that each one had a training and evaluation time of 24 h. In total, the evaluation contains 27 methods listed in Tab. 1. As proposed in [24], we added an Untrained-LSTM-AE as a baseline.

3.4 Results

Tab. 1 shows the experiments' results, implemented methods, and a reference to their original publications. The methods are ranked according to performance, and the results are rounded to four decimal places. The rankings are computed with the exact results. With few exceptions, both metrics and their associated rankings show similar results. It can only be observed for GMM-GRU-VAE, LSTM-AE-OC-SVM, and TCN-S2S-P differences of more than ten places in their order. The BeatGAN, TCN-S2S-AE, and

Table 1. All results of the experiments run for this evaluation. In the first column all methods are listed with their respective source (papers they were initially published). The second column groups them into reconstruction-based, forecasting-based, generative-VAE/GAN-based, and hybrid methods. The following columns list the F1-score, the AUPRC, rankings for both, and a mean ranking of these two. That ranking gives the order of the table to find the best results at the top.

Method	Method type	F1-score	F1-score ranking	AUPRC	AUPRC ranking	Total ranking
BeatGAN [38]	Generative-GAN-based	0.9699	1	0.9896	2	1
TCN-S2S-AE [23]	Reconstruction-based	0.9632	3	0.9914	1	2
Dense-AE [22]	Reconstruction-based	0.9631	4	0.9880	3	3
LSTM-AE [25]	Reconstruction-based	0.9506	5	0.9861	4	4
LSTM-P [27]	Forecasting-based	0.9693	2	0.9824	8	5
MSCRED [21]	Reconstruction-based	0.9353	7	0.9842	5	6
Donut [33]	Generative-VAE-based	0.9450	6	0.9829	7	7
LSTM-VAE [32]	Generative-VAE-based	0.9334	11	0.9831	6	8
OmniAnomaly [36]	Generative-VAE-based	0.9336	9	0.9808	12	9
SIS-VAE [37]	Generative-VAE-based	0.9335	10	0.9790	14	10
Untrained-LSTM-AE [24]	Reconstruction-based	0.9333	13	0.9792	13	11
LSTM-DVAE [34]	Generative-VAE-based	0.9333	16	0.9811	11	12
USAD [22]	Reconstruction-based	0.9333	12	0.9779	16	13
GMM-GRU-VAE [35]	Generative-VAE-based	0.9291	21	0.9815	10	14
TCN-S2S-P [30]	Forecasting-based	0.9172	23	0.9821	9	15
LSTM-MAX-AE [20]	Reconstruction-based	0.9333	18	0.9786	15	16
LSTM-AE-OC-SVM [44]	Hybrid	0.9337	8	0.9511	26	17
LSTM-VAE-GAN [40]	Generative-GAN-based	0.9333	14	0.9735	20	17
GenAD [25]	Reconstruction-based	0.9333	19	0.9755	19	19
TadGAN [41]	Generative-GAN-based	0.9333	15	0.9690	23	19
STGAT-MAD [26]	Reconstruction-based	0.9267	22	0.9767	17	21
Mad-GAN [39]	Generative-GAN-based	0.9333	17	0.9621	24	22
MTAD-GAT [45]	Hybrid	0.9097	25	0.9758	18	23
DeepANT/TCN-P [29]	Forecasting-based	0.9114	24	0.9712	22	24
GDN [31]	Forecasting-based	0.9078	26	0.9722	21	25
LSTM-2S2-P [28]	Forecasting-based	0.9327	20	0.9171	27	25
THOC [47]	Hybrid	0.9074	27	0.9618	25	27

Dense-AE methods score best. The weakest performers are GDN, LSTM-2S2-P, and THOC. It should be noted that Untrained-LSTM-AE, proposed above as a baseline, ends up in the upper midfield.

4 Discussion and Conclusion

Even though a generative model was ranked first in these experiments, it can be concluded that the reconstruction-based methods performed best on average, followed by the generative and, finally, the forecasting-based models. Even the proposed baseline, which belongs to the reconstruction-based methods, achieved an above-average ranking.

For future work, a few more things need to be investigated. On the one hand, it has to be considered that the TEP data are synthetic. Despite the simulation's quality, chemical processes are multifaceted, and, especially with real data, other parameters may play a role that cannot be simulated this way. All methods have yielded high scores. That could be related to the studied synthetic data with defined synthetic faults introduced in a fault-free run. The task will be considerably more challenging for actual chemical process data, but the present study is a starting point to

tackle this problem. The challenge here will be in uncovering the data and correctly labeling the anomalies in that data. On the other hand, additional metrics should be taken into account. The F1-score and AUPRC are a reasonable basis for comparison but cannot assess longer periods and interdependent points, as with time series [24, 53].

The benchmarking in this paper can guide further research and practitioners in selecting a suitable method for anomaly detection on chemical time series.

Supporting Information

Supporting Information for this article can be found under https://doi.org/10.1002/cite.202200238. There is an overview and explanation of all methods evaluated in this paper.

Acknowledgment

Part of this work was conducted within the DFG research unit FOR 5359 on Deep Learning on Sparse Chemical Process Data and the BMWK project KEEN (01MK20014U, 01MK20014L). Open access funding enabled and organized by Projekt DEAL.

Abbreviations

AD	anomaly detection
AE	autoencoder
AUPRC	area under the precision-recall curve
ELBO	evidence lower bound
FN	false negative
FP	false positive
GAN	generative adversarial neural networks
GRU	gated recurrent units
LSTM	long short-term memory
MAE	mean absolute error
MLP	multilayer perceptron
MSE	mean squared error
NN	neural network
TCN	temporal convolutional network
TEP	Tennessee Eastman process
TN	true negative
TP	true positive
VAE	variational autoencoder

References

[1] S. Garg, A. Singh, S. Batra, N. Kumar, M. S. Obaidat, in *Proc. of the 36th IEEE Global Communications Conf.*, IEEE, Piscataway, NJ **2017**, 1–7. DOI: https://doi.org/10.1109/GLOCOM.2017.8255025

[2] G. S. Chadha, A. Rabbani, A. Schwung, in *Proc. of the 17th International Conf. on Industrial Informatics*, IEEE, Piscataway, NJ **2019**, 214–219. DOI: https://doi.org/10.1109/INDIN41052.2019.8972172

[3] I. Monroy, G. Escudero, M. Graells, *19th European Symposium on Computer Aided Process Engineering* **2009**, *26*, 255–260. DOI: https://doi.org/10.1016/S1570-7946(09)70043-4

[4] B. Song, Y. Suh, *J. Loss Prev. Process Ind.* **2019**, *57*, 47–54. DOI: https://doi.org/10.1016/j.jlp.2018.08.010

[5] J. J. Downs, E. F. Vogel, *Comput. Chem. Eng.* **1993**, *17 (3)*, 245–255. DOI: https://doi.org/10.1016/0098-1354(93)80018-I

[6] A. C. Rieth, B. D. Amsel, R. Tran, M. B. Cook, *Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation*, Harvard Dataverse, **2017**. DOI: https://doi.org/10.7910/DVN/6C3JR1

[7] G. S. Chadha, I. Islam, A. Schwung, S. X. Ding, *Sensors* **2021**, *21 (16)*, 5488. DOI: https://doi.org/10.3390/s21165488

[8] P. Spyridon, Y. S. Boutalis, in *Proc. of the 16th European Control Conference*, IEEE, Piscataway, NJ **2018**, 691–696. DOI: https://doi.org/10.23919/ECC.2018.8550560

[9] S. Plakias, Y. S. Boutalis, *Comput. Ind.* **2022**, *142*, 103743. DOI: https://doi.org/10.1016/j.compind.2022.103743

[10] X. Yang, D. Feng, in *Proc. of the 5th Int. Conf. on Control, Automation and Robotics*, IEEE, Piscataway, NJ **2019**, 644–648. DOI: https://doi.org/10.1109/ICCAR.2019.8813415

[11] F. Neubürger, Y. Saeid, T. Kopinski, *Variational-Autoencoder Architectures for Anomaly Detection in Industrial Processes*, July **2021**.

[12] G. S. Chadha, A. Rabbani, A. Schwung, in *Proc. of the 17th Int. Conf. on Industrial Informatics*, IEEE, Piscataway, NJ **2019**, 214–219. DOI: https://doi.org/10.1109/INDIN41052.2019.8972172

[13] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep One-Class Classification, *Proc. Mach. Learn. Res.* **2018**, *80*, 4393–4402.

[14] L. Ruff, J. Kauffmann, R. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. Dietterich, K.-R. Müller, *Proc. IEEE* **2021**, *109 (5)*, 1–40. DOI: https://doi.org/10.1109/JPROC.2021.3052449

[15] P. Liznerski, L. Ruff, R. Vandermeulen, B. Franks, K.-R. Müller, M. Kloft, Exposing Outlier Exposure: What Can Be Learned From Few, One, and Zero Outlier Images, *Trans. Mach. Learn. Res.* **2022**, *11*, 1–47.

[16] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, M. Rudolph, Neural Transformation Learning for Deep Anomaly Detection Beyond Images, *Proc. Mach. Learn. Res.* **2021**, *139*, 8703–8714.

[17] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, L. S. Davis, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Piscataway, NJ **2016**, 733–742. DOI: https://doi.org/10.1109/CVPR.2016.86

[18] W. Luo, W. Liu, S. Gao, in *Proc. of the 10th IEEE Int. Conf. on Multimedia and Expo*, IEEE, Piscataway, NJ **2017**, 439–444. DOI: https://doi.org/10.1109/ICME.2017.8019325

[19] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, *Lstm-based encoder-decoder for multi-sensor anomaly detection*, arXiv:1607.00148, **2016**. DOI: https://doi.org/10.48550/arXiv.1607.00148

[20] A. H. Mirza, S. Cosan, in *Proc. of the 26th Signal Processing and Communications Applications Conf.*, IEEE, Piscataway, NJ **2018**, 1–4. DOI: https://doi.org/10.1109/SIU.2018.8404689

[21] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N. V. Chawla, in *Proc. of the 33th AAAI Conf. on Artificial Intelligence*, AAAI Press, Washington, DC **2019**, 1409–1416. DOI: https://doi.org/10.1609/aaai.v33i01.33011409

[22] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, in *Proc. of the 26th Int. Conf. on Knowledge Discovery & Data*

Mining, Association for Computing Machinery, New York **2020**, 3395–3404. DOI: https://doi.org/10.1145/3394486.3403392
[23] M. Thill, W. Konen, T. Bäck, in *Proc. of the 9th Int. Conf. on Bioinspired Methods and Their Applications* (Eds: B. Filipic, E. A. Minisci, M. Vasile), Springer, Cham, Switzerland **2020**, 161–173. DOI: https://doi.org/10.1007/978-3-030-63710-1_13
[24] S. Kim, K. Choi, H.-S. Choi, B. Lee, S. Yoon, in *Proc. of the 36th AAAI Conf. on Artificial Intelligence*, AAAI, Washington, DC **2022**, 7194–7201. DOI: https://doi.org/10.1609/aaai.v36i7.20680
[25] X. Hua, L. Zhu, S. Zhang, Z. Li, S. Wang, D. Zhou, S. Wang, C. Deng, *GenAD: General representations of multivariate time series for anomaly detection*, arXiv:2202.04250, **2022**. DOI: https://doi.org/10.48550/arXiv.2202.04250
[26] J. Zhan, S. Wang, X. Ma, C. Wu, C. Yang, D. Zeng, S. Wang, in *Proc. of the 47th Int. Conf. on Acoustics, Speech, and Signal Processing*, IEEE, Piscataway, NJ **2022**, 3568–3572. DOI: https://doi.org/10.1109/ICASSP43922.2022.9747274
[27] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, in *Proc. of the 23rd European Symposium on Artificial Neural Networks*, Ciaco, Louvain-la-Neuve **2015**, ES2015-56.
[28] P. Filonov, A. Lavrentyev, A. Vorontsov, *Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model*, arXiv:1612.06676, **2016**. DOI: https://doi.org/10.48550/arXiv.1612.06676
[29] M. Munir, S. A. Siddiqui, A. Dengel, S. Ahmed, *IEEE Access* **2018**, *7*, 1991–2005. DOI: https://doi.org/10.1109/ACCESS.2018.2886457
[30] Y. He, J. Zhao, *J. Phys.: Conf. Ser.* **2019**, *1213 (4)*, 042050. DOI: https://doi.org/10.1088/1742-6596/1213/4/042050
[31] A. Deng, B. Hooi, in *Proc. of the 35th AAAI Conf. on Artificial Intelligence*, AAAI, Washington, DC **2021**, 4027–4035. DOI: https://doi.org/10.1609/aaai.v35i5.16523
[32] M. Sölch, J. Bayer, M. Ludersdorfer, P. van der Smagt, *Variational inference for on-line anomaly detection in high-dimensional time series*, arXiv:1602.07109, **2016**. DO: https://doi.org/10.48550/arXiv.1602.07109
[33] H. Xu, Y. Feng, J. Chen, Z. Wang, H. Qiao, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, in *Proc. of the 27th Int. World Wide Web Conf.* (Eds: P.-A. Champin, F. Gandon, L. Médini), Association for Computing Machinery, New York **2018**. DOI: https://doi.org/10.1145/3178876.3185996
[34] D. Park, Y. Hoshi, C. C. Kemp, *IEEE Rob. Autom. Lett.* **2018**, *3 (3)*, 1544–1551. DOI: https://doi.org/10.1109/LRA.2018.2801475
[35] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, P. Li, Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach, *Proc. Mach. Learn. Res.* **2018**, *95*, 97–112.
[36] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, in *Proc. of the 25th Int. Conf. on Knowledge Discovery & Data Mining* (Eds: A. Teredesai, V. Kumar), Association for Computing Machinery, New York **2019**, 2828–2837. DOI: https://doi.org/10.1145/3292500.3330672
[37] L. Li, J. Yan, H. Wang, Y. Jin, *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *32 (3)*, 1177–1191. DOI: https://doi.org/10.1109/TNNLS.2020.2980749
[38] B. Zhou, S. Liu, B. Hooi, X. Cheng, J. Ye, in *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence*, Springer, Singapore **2019**, 4433–4439. DOI: https://doi.org/10.24963/ijcai.2019/616
[39] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, in *Proc. of the 28th Int. Conf. on Artificial Neural Networks*, Springer, Cham, Switzerland **2019**, 703–716. DOI: https://doi.org/10.1007/978-3-030-30490-4_56
[40] Z. Niu, K. Yu, X. Wu, *Sensors* **2020**, *20 (13)*, 3738. DOI: https://doi.org/10.3390/s20133738
[41] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, K. Veeramachaneni, in *Proc. of the 8th Int. Conf. on Big Data*, IEEE, Piscataway, NJ **2020**, 33–43. DOI: https://doi.org/10.1109/BigData50022.2020.9378139
[42] M. Sabokrou, M. Khalooei, M. Fathy, E. Adeli, in *Proc. of the 31th Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2018**, 3379–3388. DOI: https://doi.org/10.1109/CVPR.2018.00356
[43] W. Liu, W. Luo, D. Lian, S. Gao, in *Proc. of the 31th Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2018**, 6536–6545. DOI: https://doi.org/10.1109/CVPR.2018.00684
[44] M. S. Elsayed, N.-A. Le-Khac, S. Dev, A. D. Jurcut, in *Proc. of the 16th Symposium on QoS and Security for Wireless and Mobile Networks* (Eds: C. Li, A. Mostefaoui), The Association for Computing Machinery, New York **2020**, 37–45. DOI: https://doi.org/10.1145/3416013.3426457
[45] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, in *Proc. of the 20th Int. Conf. on Data Mining*, IEEE, Piscataway, NJ **2020**, 841–850. DOI: https://doi.org/10.1109/ICDM50108.2020.00093
[46] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, *Neural Comput.* **2001**, *13 (7)*, 1443–1471. DOI: https://doi.org/10.1162/089976601750264965
[47] L. Shen, Z. Li, J. T. Kwok, in *Proc. of the 34th Int. Conf. on Neural Information Processing Systems*, Curran Associates, Red Hook, NY **2020**, 13016–13026, Article 1092.
[48] D. Hendrycks, M. Mazeika, T. Dietterich, *Deep Anomaly Detection with Outlier Exposure*, arXiv:1812.04606, **2018**. DOI: https://doi.org/10.48550/arXiv.1812.04606
[49] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, P. Jain, DROCC: Deep Robust One-Class Classification, *Proc. Mach. Learn. Res.* **2020**, *119*, 3711–3721.
[50] G. Manca, *IEEE Access* **2021**, *9*, 154965–154974. DOI: https://doi.org/10.1109/ACCESS.2021.3128695
[51] C. Rieth, B. Amsel, R. Tran, M. Cook, in *Advances in Human Factors in Robots and Unmanned Systems* (Ed: J. Chen), **2018**, 52–63. DOI: https://doi.org/10.1007/978-3-319-60384-1_6
[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, **2019**, *32*.
[53] K. Doshi, S. Abudalou, Y. Yilmaz, *Tisat: Time series anomaly transformer*, arXiv:2203.05167, **2022**. DOI: https://doi.org/10.48550/arXiv.2203.05167
[54] F. Rewicki, J. Denzler, J. Niebling, *Appl. Sci.* **2023**, *13 (3)*, 1778. DOI: https://doi.org/10.3390/app13031778

