Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
©2023 ESREL2023 Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P475-cd

Interpretation of influential factors for AI-based anomaly detection

Sheng Ding

Institute for Automation Technology and Software System, University of Stuttgart, Germany.
E-mail: sheng.ding@ias.uni-stuttgart.de

Adrian Wolf

Institute for Automation Technology and Software System, University of Stuttgart, Germany.
E-mail: st161569@stud.uni-stuttgart.de

Andrey Morozov

Institute for Automation Technology and Software System, University of Stuttgart, Germany.
E-mail: andrey.morozov@ias.uni-stuttgart.de

Anomaly detection is a crucial task in a wide range of industries and domains. The ability to identify abnormal patterns and behaviors in time series data can help detect potential issues, prevent downtime, reduce maintenance costs, and improve the overall performance of systems and processes. This paper focuses on analyzing the significant factors that affect the accuracy and dependability of AI-based time series anomaly detection. The objective is to provide comprehensive insights into interpreting these factors and to explore their impact on the performance. Our study's outcomes shall assist researchers and practitioners in selecting the most appropriate approaches for anomaly detection tasks in diverse domains.

Keywords: Fault Detection, Anomaly Detection, Machine Learning, Deep Learning.

1. Introduction

With the increasing complexity of modern systems, traditional methods for anomaly detection are often insufficient or ineffective. As a result, recently there has been a growing interest in leveraging artificial intelligence (AI) techniques for time series anomaly detection. However, the interpretability of AI-based models is a critical issue that affects their trustworthiness. It is essential to ensure their effectiveness in real-world scenarios.

Contributions: In this paper, we answer the research question: "What influence do different components have on the performance of the whole anomaly detection pipeline?". Our toolbox, TSAD platform, was used to perform detailed evaluations of various elements of the anomaly detection pipeline (see Fig. 1) [a] Each time we observe and compare the results with one element (e.g. hyperparameters) changed and other elements fixed. The tests were performed on several datasets from different cyber-physical systems. We also present the results obtained by the dynamic switch mechanism.

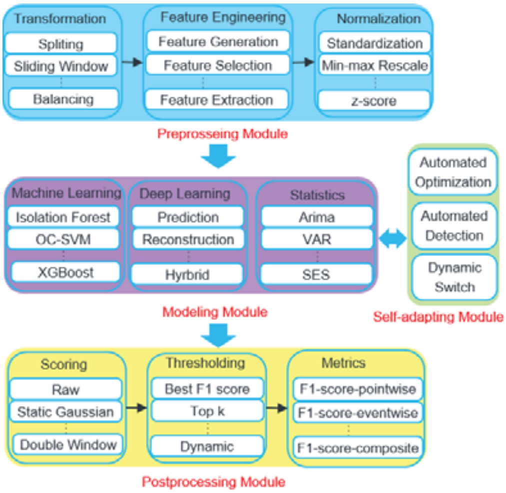


Fig. 1.: Our approach is to alter a single module at a time, determine the contribution of **influential factors** inside the anomaly detection system.

[a] www.github.com/mbsa-tud/tsad_platform

2. Influence of the input data

During experiments, we altered individual elements, while maintaining others constantly and compared the results. Our experiments involved 14 candidate TSAD algorithms (details of algorithms and datasets can be found in our prior work Ding et al. (2022)). The selection of model, training, and data-related hyperparameters, were used for the evaluations can be seen in Table 1. Some hyperparameters are inspired by a survey Braei and Wagner (2020) with minor modifications. Many parameters for Machine Learning (ML) algorithms were chosen by running several tests and optimization, like the values for k for the ML algorithms LOF and LDOF. We tested the algorithms on various datasets, in the following, we present the possible influence of different anomalies and attributes of the time series.

Model	**Hyperparameter**	**Value/- s**
FC-AE	#Neurons in first fc layer	32
LSTM (r)	#Hidden units	64
CNN-LSTM (r)	#Hidden units #Filters in conv. layers	64 8
TCN-AE	#Filters in temporal conv. layers	32
GRU	#Hidden units	64
CNN-LSTM	#Hidden units #Filters in conv. layers	64 32
LSTM	#Hidden units	64
CNN (DeepAnT)	#Filters in first conv. layer	8
ResNet	#Filters in conv. layers	32
MLP	#Neurons in first fc layer	32
iForest	#Learners/iTrees	100
OC-SVM	-	-
LOF	k	100
LDOF	k	20

Table 1.: Selected values for model-related hyperparameters.

2.1. *Anomaly type*

To better understand the effects of different types of faults on the detection performance of different models, the F_1 and F_{c1} scores for three different fault types including noise, offset and stuck-at faults for the UAV and AVS datasets using the best F_1 score threshold are shown in Table 2 and Table 3. General observations regarding the different fault types can be summarized as such:

Noise faults: For the AVS datset, the FC-AE is by far the best model for noise faults and ML algorithms, while all achieving similar scores, generally performed worse compared to DL models. On the contrary, ML algorithms could keep up with DL models for the UAV dataset and the LOF even outperformed all other models.

Offset Faults: The Offset faults are dominated by the LSTM and Hybrid CNN-LSTM prediction models for the AVS dataset and by the TCN-AE for the UAV dataset. With a few exceptions, the scores have generally dropped compared to noise faults.

Stuck-at faults: Lastly, for stuck-at faults, LDOF and the FC-AE dominated for the AVS dataset while LOF completely outperformed all other models for the UAV dataset. When comparing these results to the scores for offset faults, another clear drop in performance can be noticed for most models.

Model	**Noise**		**Offset**		**Stuck-at**	
	F_1	F_{c1}	F_1	F_{c1}	F_1	F_{c1}
FC-AE	**0.9174**	**0.9801**	0.5565	0.6117	0.4448	**0.5989**
LSTM (r)	0.4557	0.8219	0.4018	0.4018	0.4062	0.4062
CNN-LSTM (r)	0.6821	0.9051	0.4014	0.4015	0.4051	0.4051
TCN-AE	0.893	0.9381	0.5147	0.6442	0.3998	0.4019
LSTM	0.814	0.9128	**0.6619**	0.712	0.4039	0.4113
CNN-LSTM	0.5577	0.8355	0.5878	**0.7138**	0.4127	0.4136
GRU	0.8766	0.9444	0.5419	0.6111	0.4666	0.474
CNN (DeepAnT)	0.8424	0.9142	0.5192	0.6419	0.3868	0.3869
ResNet	0.8091	0.8823	0.4952	0.5246	0.387	0.387
MLP	0.8646	0.9417	0.5007	0.568	0.4078	0.4231
iForest	0.5756	0.5796	0.5224	0.5506	0.4003	0.4008
OC-SVM	0.5525	0.5587	0.5285	0.5342	0.3937	0.4003
LOF	0.5519	0.5678	0.4511	0.4607	0.3984	0.3989
LDOF	0.5395	0.5514	0.4734	0.4761	**0.5541**	0.5726

Table 2.: F_1 and F_{c1} scores for noise, offset and stuck-at faults for different models. Results were obtained on the **AVS** dataset using the best F_1 score threshold for all models. Bold values indicate the best scores for each column.

The subtlety and deviation from normal data patterns determine the extent of performance degradation from noise faults to stuck-at faults in anomaly detection. Since the UAV and AVS datasets are predominantly noise-free under normal conditions, identifying noise faults is relatively straightforward. In contrast, offset faults involve a fixed deviation from normal data, resulting in a pattern that resembles normal data but with deviations in the middle parts that can be chal-

Model	**Noise**		**Offset**		**Stuck-at**	
	F_1	F_{c1}	F_1	F_{c1}	F_1	F_{c1}
FC-AE	0.7245	0.8092	0.3646	0.4236	0.31	0.3101
LSTM (r)	0.5194	0.5195	0.4668	0.5516	0.3167	0.3265
CNN-LSTM (r)	0.5513	0.5715	0.4827	0.5054	0.3179	0.3508
TCN-AE	0.5195	0.5195	**0.655**	**0.73**	0.3295	0.3317
LSTM	0.6634	0.7609	0.4917	0.5436	0.3157	0.3167
CNN-LSTM	0.6656	0.8609	0.4927	0.5576	0.3321	0.3491
GRU	0.6343	0.7861	0.4006	0.4713	0.3234	0.3317
CNN (DeepAnT)	0.6269	0.7989	0.3704	0.4394	0.3366	0.3603
ResNet	0.6983	0.8101	0.4364	0.4923	0.3312	0.3403
MLP	0.5306	0.5334	0.5288	0.4693	0.3106	0.3106
iForest	0.5294	0.5298	0.5109	0.4626	0.3046	0.3057
OC-SVM	0.6295	0.693	0.2132	0.2142	0.312	0.3124
LOF	**0.8525**	**0.8614**	0.1195	0.1683	**0.5618**	**0.6032**
LDOF	0.0209	0.5051	0.4936	0.5709	0	0

Table 3.: F_1 and F_{c1} scores for noise, offset and stuck-at faults for different models. Results were obtained on the **UAV** dataset using the best F_1 score threshold for all models. Bold values indicate the best scores for each column.

lenging to detect. This makes anomaly detection less effective than noise faults. Stuck-at faults are the most subtle type of anomaly, particularly in datasets with a smooth pattern like AVS and UAV datasets, which poses a higher detection challenge. On the other hand, for datasets with more sudden spikes or local fluctuations, noise faults may be more difficult, and stuck-at faults may be easier to identify.

In summary, no model can effectively detect all types of anomalies. Although only three types of anomalies were considered in this analysis, the results may vary with different types of anomalies, such as point, collective, or contextual anomalies.

2.2. *Anomaly length*

Another influential attribute of an anomaly, which can affect the accuracy of an anomaly detector, is its length. To illustrate the possible effect of the anomaly length, the smoothed F_1 scores for different anomaly lengths for the SMAP dataset can be observed in Fig. 2. In reality, the scores fluctuate way more as the depicted curves might suggest. This has to do with the fact that not only do the lengths of the anomalies vary, but also their overall pattern and severity. Outliers exist on both ends of the depicted range but the shown curve was modeled to show the rough average of all observed F_1 scores. Additionally it has to be noted that the x-axis values indicate the length of the longest anomaly found in each time series. Since most time series only contain one anomaly and if otherwise, only anomalies of roughly similar length, the results were still considered to be meaningful with regards to the influences of the anomaly length.

There are two general observations to be made regarding the models: *(i)* There is a decrease in performance for lengths from 1 to 500 observations, and *(ii)* scores increase beyond 500. There are several reasons for this trend. Firstly, short anomalies are often points where the value spikes out of the normal range, making them easy to detect. Secondly, anomalies in the middle range with lengths less than 500 are increasingly difficult to detect. Some of these time series contain sudden changes in data, which can lead to a lot of false positives, as shown in Fig. 3. As a result, the actual anomalies often receive lower anomaly scores, leading to poor detection performance. Lastly, anomalies longer than 500 timestamps tend to be more trivial for the SMAP dataset, as their values completely leave the range of what could be considered non-anomalous.

2.3. *Time series data pattern*

The detection performance can also be influenced by the overall pattern of the time series. To demonstrate potential impacts, we chose two highly distinct subsets of the MSL dataset and assessed the performance of all algorithms on them. Fig. 3 shows the ”p-11” subset, a smooth time series that rises constantly before dropping abruptly, the anomaly root cause is the rise and drop of the value occurs about 4 times faster. Fig. 4 shows the ”t-12” subset, a noisy time series, the anomaly root cause is (i) a higher noise level and (ii) a spike. In both figures, the topmost axes depict the anomalous time series, and the anomaly scores and detected anomalies for each model are shown below. All other axes show the model’s anomaly scores in blue, the best F_1 score threshold as a red horizontal line, and the detected anomalies as red areas. Green areas indicate the true anomaly.

Fig. 3 shows the result with a window size of 100. When analyzing the ”t-12” subset, all anoma-

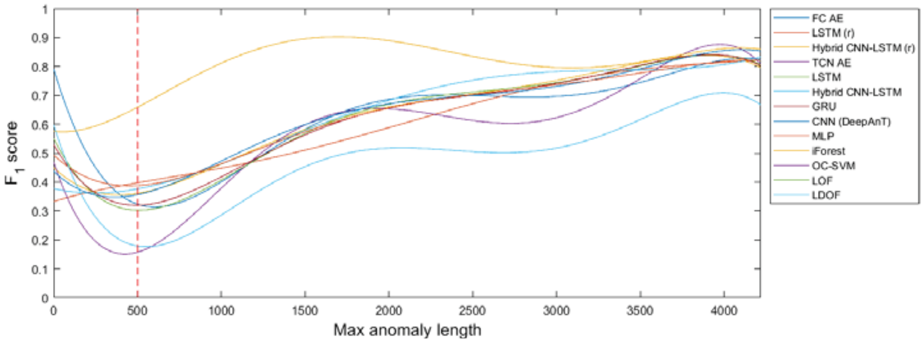


Fig. 2.: Comparison of F_1 scores for different anomaly lengths for the SMAP dataset using the best F_1 score threshold. The value on the x-axis indicates the length of the longest anomaly in the tested time series.

lies are detected by DL models event wisely. But ML models (the last three axes) perform better detection, especially LOF, and LDOF. The normal sudden drops result in inevitable spikes in the anomaly scores, resulting in false positives. At every time window, DL assigns each point a unique anomaly score, while ML assigns a single score vector to the entire subsequence. Therefore only DL models can do detection in a streaming way. If the window size decrease, ML models will raise worse detection.

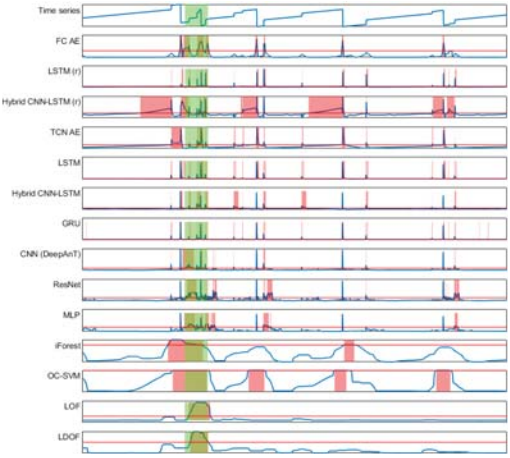


Fig. 3.: Anomaly detection for the "t-12" dataset

Compared to "t-12" this time DL model performs better when analyzing the "p-11" subset. Fig. 3 shows the result using the point-wise best F_1 score threshold, the threshold got set very low to improve the true positives. It is beneficial for the ML models, as if the threshold is a bit higher, the iForest and OC-SVM will fail to detect the second anomaly and LDOF will fail to detect the first anomaly. By setting a higher threshold, the DL models could receive almost perfect event-wise, point-adjusted and composite scores.

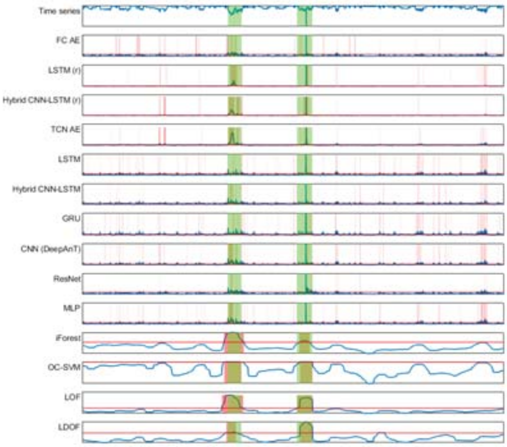


Fig. 4.: Anomaly detection for the "p-11" dataset

3. Influence of algorithms

3.1. *Univarite datasets*

Table 4 shows the F_1 scores for all tested models/algorithms for the univariate datasets (MSL, SMAP, AVS and UAV) using the best F_1 score threshold.

Model	MSL	SMAP	AVS	UAV
FC-AE	0.5102	0.6005	**0.5513**	0.4042
LSTM (r)	0.5083	0.4644	0.2999	0.3158
Hybrid CNN-LSTM (r)	0.5107	0.4998	0.4636	0.3665
TCN-AE	0.4872	0.4144	0.4442	0.3219
LSTM	0.4698	0.5141	0.4516	0.3517
Hybrid CNN-LSTM	0.4445	0.4854	0.3884	0.3872
GRU	0.462	0.5114	0.4542	0.317
CNN (DeepAnT)	0.4285	0.4929	0.4262	0.3326
MLP	0.4162	0.5219	0.4496	0.4139
iForest	**0.6964**	**0.6787**	0.3951	0.3126
OC-SVM	0.5183	0.6596	0.4286	0.3145
LOF	0.5874	0.5175	0.4179	0.3574
LDOF	0.424	0.4323	0.4976	**0.4935**

Table 4.: Average F_1 scores for all **univariate** datasets using the best F_1 score threshold. Bold values indicate the best scores for each dataset.

For both the MSL and SMAP dataset, the iForest is the clear winner. While the FC-AE clearly dominates the AVS dataset, the LDOF algorithm performs best for the UAV dataset. While this table might suggest that classic ML methods outperform DL models, Fig. 5 paints a different picture. It shows boxplots for the F_1 scores for all models/algorithms using the best F_1 score threshold. The results are combined for all tested univariate datasets. The red lines indicate the median scores and 50% of all scores fall within the blue boxes. Classic ML methods no longer seem to outperform, but the iForest is still the clear winner for the category of ML. Predictive models show similar performance overall with the exception of the Hybrid CNN-LSTM predictor as it generally performed worse. The performance of reconstructive models shows a lot more fluctuation across models. The Hybrid CNN-LSTM reconstructor for example has the highest median F_1 score, while the TCN-AE shows a very big variance. In general, no model can be considered as the best model overall.

3.2. *Multivariate datasets*

Table 5 compares the results for the SMD dataset using the best F_1 score threshold for the two methods of anomaly detection for multivariate time series: **Aggregated scores** and **aggregated detection**. The method of aggregating the channel-wise scores generally improves the performance of the DL models compared to aggregating the channel-wise detections. For all DL models, higher F_1 scores were obtained by aggregating the scores before applying the threshold. The same observation can however not be made for the F_{c1} scores. Since the results were obtained by using the best F_1 score threshold and point-wise and composite scores don't necessarily correlate, this is considered irrelevant. The classic ML algorithms (iForest, OC-SVM, LOF and LDOF) were not trained/tested separately for each channel and for this dataset, the time series was only split into subsequences for DL models. The iForest and OC-SVM could however still keep up or even outperform DL models. LOF and LDOF performed worse compared to all other models.

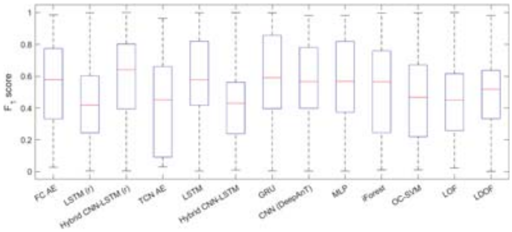


Fig. 5.: Boxplots for F_1 scores for all **univariate** datasets.

Model	Aggregated scores		Aggregated detection	
	F_1	F_{c1}	F_1	F_{c1}
FC-AE	0.4574	0.549	0.4022	0.4711
LSTM (r)	0.4548	0.5781	0.4278	0.5018
Hybrid CNN-LSTM (r)	0.4639	0.4812	0.391	0.6197
TCN-AE	**0.5017**	**0.6765**	0.4855	0.6238
LSTM	0.4917	0.5773	0.4691	**0.6335**
Hybrid CNN-LSTM	0.4752	0.6001	0.4242	0.455
GRU	0.4617	0.5813	0.4062	0.6131
CNN (DeepAnT)	0.483	0.566	0.436	0.6049
MLP	0.4791	0.5677	0.4445	0.5894
iForest	0.4937	0.5723	0.4937	0.5723
OC-SVM	0.4985	0.568	**0.4985**	0.568
LOF	0.2923	0.4054	0.2923	0.4054
LDOF	0.1884	0.2148	0.1884	0.2148

Table 5.: F_1 and F_{c1} scores for all models for the multivariate **SMD** dataset. For DL models the two scoring methods, aggregated detection and aggregated scores, are compared. ML models show the same values for both since only one method of detection was used. Bold values indicate the best scores for each category.

3.3. *Influence of hyperparameters*

To illustrate the possible effect of choosing the correct hyperparameters, the FC-AE was optimized using the platform's built-in bayesian optimization function to achieve the best possible F_1 score. Through our experiments, most hyperparameters do not have a great impact on the detection performance, but changing the number of neurons and the size of the sliding windows/subsequences w will raise different results. Therefore they are considered the most relevant. Table 6 compares the hyperparameters and obtained F_1 scores before and after running the optimization for 30 iterations.

Hyperparameter	**Initial**	**Optimized**
#Neurons	32	10
Window size	100	396
Score	**Initial**	**Optimized**
F_1	0.2873	**0.4015**
F_{e1}	**0.5**	0.0126
F_{p1}	0.2875	**0.6286**
F_{c1}	0.2873	**0.4812**

Table 6.: Hyperparameters and achieved F_1 scores before and after optimizing the window size and number of neurons for the best F_1 score for the FC-AE. The optimization was run for 30 iterations. Bold values indicate the column-wise best scores.

All F_1 scores with the exception of the event-wise F_{e1} score dramatically improved after the optimization process. This inconsistency exists because even though the total number of false positive points declined, the optimized model detected a lot more false positive events, thus lowering the event-wise scores dramatically. Since the optimization algorithm tried to find hyperparameters resulting in the best possible F_1 score, the improvement of the F_1 score was expected. Better F_{p1} and F_{c1} scores are a welcomed side effect that wasn't necessarily expected, as the different types of evaluation metrics don't always correlate with each other. When comparing the hyperparameters as they were before and after optimization, it becomes apparent that the window size has changed more dramatically compared to the number of neurons.

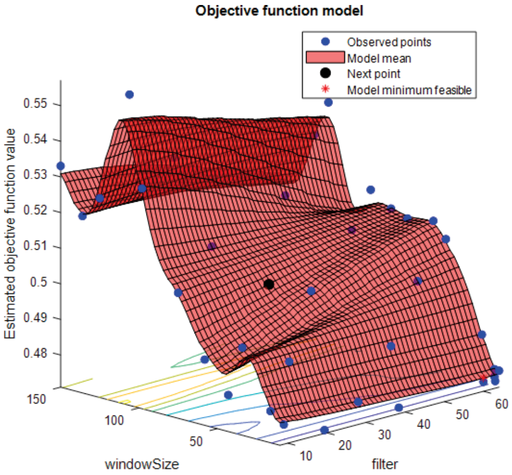


Fig. 6.: Example of optimizing the window size w of the subsequences and the number of filters for the CNN-LSTM.

Fig. 6 shows an example of the optimization process for the CNN-LSTM. The red area indicates the mean of all observed scores for the different combinations of window size w and number of filters. It is rather flat when varying the filters and a much steeper slope can be observed for different values for w. This again indicates a greater impact on the window size as for the FC-AE. In general, it can be concluded, that the setting of the correct **window size** is amongst the most crucial aspects of configuring a good anomaly detection pipeline.

4. Influence of postprocessing methods

4.1. *Influence of thresholding methods*

One of the most significant factors affecting detection performance is the thresholding method. Varying the threshold can dramatically change the resulting scores. To illustrate this, Fig. 7 and Fig. 8 show the F_1 and F_{c1} scores for all tested models/algorithms for six different thresholds: The top-k, dynamic, best F_1 score, best F_{e1} score, best F_{p1} score and best F_{c1} score threshold.

The most important observations can be summarized as such:

Best F score thresholds: The best F_{e1} score and best F_{p1} score threshold impact the F_1 and

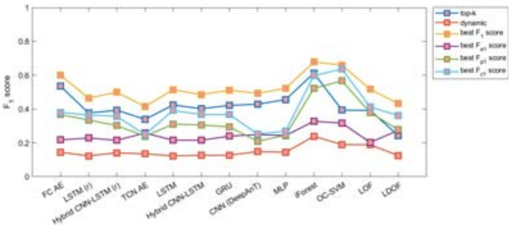


Fig. 7.: Comparison of F_1 scores for different thresholding methods for the SMAP dataset.

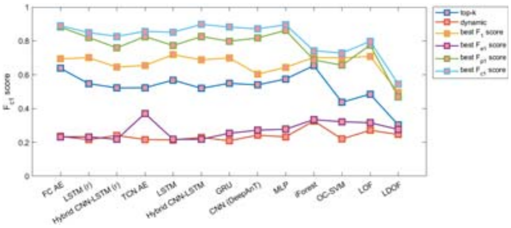


Fig. 8.: Comparison of F_{c1} scores for different thresholding methods for the SMAP dataset.

F_{c1} scores in different ways. These thresholds are all supervised, meaning they require a labeled anomalous validation set V_A to be computed.

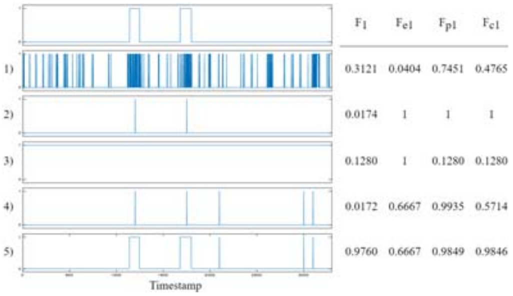
Top-k threshold: The top-k threshold seems to have a much higher correlation to the F_1 score than the F_{c1} score. Since it is a supervised threshold that gets set by using a labeled anomalous validation set V_A, it shares the same flaws as the best F score thresholds.

Non-parametric, dynamic threshold: The dynamic threshold requires no prior knowledge of the distribution of anomaly scores, but will always detect at least one anomaly per window, making false positives inevitable for these tests. Furthermore, the dynamic threshold's necessary parameters are highly dependent on the context, making it unsuitable for an automated anomaly detection system.

4.2. *Influence of evaluation metrics*

It can be observed from Fig. 7 and Fig. 8, that achieving a high composite F_{c1} score is generally easier than achieving a high point-wise F_1 score. While the maximum observed F_1 score has a value of about 0.7, F_{c1} scores often reached a value of 0.9. This difference exists because the point-wise metrics are stricter when compared to event-wise, point-adjusted, or composite metrics.

Fig. 9 shows a synthetic test case to compare different evaluation metrics. The topmost row displays the ground truth labels, followed by several synthetic detected labels. The table on the right shows the F_1, F_{e1}, F_{p1} and F_{c1} scores achieved for each scenario. When comparing the different metrics, it becomes apparent that they can vary dramatically even for the same scenario. Most noticeable are the following differences:



	F_1	F_{e1}	F_{p1}	F_{c1}
1)	0.3121	0.0404	0.7451	0.4765
2)	0.0174	1	1	1
3)	0.1280	1	0.1280	0.1280
4)	0.0172	0.6667	0.9935	0.5714
5)	0.9760	0.6667	0.9849	0.9846

Fig. 9.: Comparison of evaluation metrics for different synthetic test scenarios. The value of each point is 1 (high) for anomalies and 0 (low) for a fault-free detection.

(1) The first scenario depicts a random detection, but most detected anomalies fall within true anomalous sections. The point-adjusted F_{p1} score rewards this scenario the most.
(2) Just one observation within each true anomalous segment is labeled as anomalous, anomaly detected, still resulting in a very poor point-wise F_1 score.
(3) The entire time series is detected as an anomaly. While this results in a perfect F_{e1} score of 1, all other scores are rather bad.
(4) This scenario is similar to scenario 2) with the addition of three false positive points. The scores range from 0.0172 for the F_1 score to an almost perfect F_{p1} score.
(5) If all anomalies are detected entirely with only a small number of false positive points, all scores are close to perfect with the exception of the event-wise F_{e1} score.

Understanding which aspects are emphasized by each metric is crucial. The F_1 score rewards a model that detects most anomalies with fewer false positives, but it can still perform well even with numerous false positives. The F_{e1} score favors detectors that overlap with at least one true anomalous segment and have few false positives, but large detected segments can still achieve a perfect score. The F_{p1} score is beneficial to single true positive points within an anomalous segment, potentially detecting random points as anomalous. The F_{c1} score rewards detectors with high point-wise precision and event-wise recall. Choosing an appropriate metric depends on the ideal anomaly detector's expectations and the specific context.

5. Influence of a dynamic switch mechanism

No model is best for all contexts. The dynamic switch mechanism represents an attempt to solve this problem. Time series of the anomalous testing set were randomly assigned into the train and test set for the dynamic switch, equally distributing them across both. After the models were trained, for each time series of the training set the best model was computed by comparing the F_1 scores. The subsequent training of the DNN classifier and the evaluations on the test set lead to the results shown in Table 7 for the AVS dataset.

Model	F_1	F_{e1}	F_{p1}	F_{c1}
Dynamic Switch	**0.6917**	0.9637	**0.7504**	**0.7363**
FC-AE	0.5209	0.8337	0.6763	0.6408
LSTM (r)	0.3889	0.9139	0.4097	0.3943
CNN-LSTM (r)	0.4575	0.6646	0.5953	0.5631
TCN-AE	0.4213	0.6226	0.5662	0.5164
LSTM	0.4581	0.6419	0.531	0.504
CNN-LSTM	0.4428	0.7011	0.5587	0.5166
GRU	0.4535	0.6076	0.5354	0.5061
CNN (DeepAnT)	0.4443	0.5646	0.4949	0.4816
ResNet	0.4695	0.5818	0.5492	0.5222
MLP	0.4607	0.5668	0.5436	0.5155
iForest	0.4018	0.6591	0.4188	0.4154
OC-SVM	0.4018	0.7045	0.4316	0.4194
LOF	0.41	0.9432	0.4458	0.4245
LDOF	0.4932	**0.9752**	0.5078	0.501

Table 7.: Results for the dynamic switch and the individual models for the **AVS** dataset. Bold values indicate the best scores.

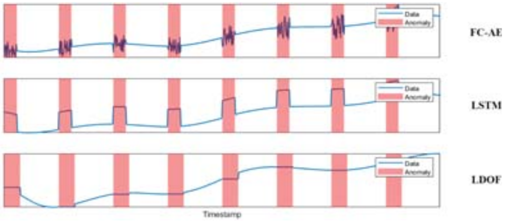


Fig. 10.: Examples of models recommended/selected by the dynamic switch for the corresponding time series of the AVS dataset.

Fig. 10 shows three examples of the models selected by the dynamic switch for the AVS dataset. It recommends the FC-AE for noise faults, the LSTM for offset faults and the LDOF dominated for stuck-at faults. The recommendations by the dynamic switch are consistent with the former analysis (Section. 2.1). The recommendation of an anomaly detection method shows very promising results and may help in configuring a better anomaly detection pipeline.

6. Conclusion

The interpretation of influential factors plays a vital role in ensuring the reliability of anomaly detection results. Therefore, this paper aims to present a comprehensive analysis of the influential factors for AI-based time series anomaly detection and provide insights into their interpretation. The findings of this study shall guide researchers and practitioners in selecting the most suitable approaches for anomaly detection tasks in different domains.

References

Braei, M. and S. Wagner (2020). Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*.

Ding, S., S. Ayoub, and A. Morozov (2022). Tool paper: Time series anomaly detection platform for matlab simulink. In C. Seguin, M. Zeller, and T. Prosvirnova (Eds.), *Model-Based Safety and Assessment*, Cham, pp. 204–218. Springer International Publishing.