Neural Networks

ACVAE: a Novel Self-adversarial Variational Auto-Encoder Combined with Contrast Learning for Time Series Anomaly Detection

--Manuscript Draft--

Manuscript Number:	NEUNET-D-22-01959R1
Article Type:	Article
Section/Category:	Learning Systems
Keywords:	Variational auto-encoder(VAE); Deep Generative Model; anomaly detection; Contrast learning; Multivariate Time Series; anomaly interpretation
Abstract:	Proposed a novel self-adversarial variational auto-encoder combined with contrast learning for anomaly detection. Explored the methods of anomaly interpretation and automatic threshold selection in our model. The generation model with contrast learning has higher detection accuracy and less time consumption. The adversarial mechanism and contrast learning are employed to make the model more suitable for anomaly detection and avoid the problem of of posterior collapse.

response

Dr. Wang Guoyin
Chongqing Key Laboratory of Computational Intelligence
Chongqing University of Posts and Telecommunications
Chongqing 400065
PR China
wanggy@cqupt.edu.cn

May. 1, 2023

Prof. Taro Toyoizumi
Editor-in-Chief
Neural Networks

Dear Prof. Taro Toyoizumi

Thank you very much for your reply and help. Thanks a lot for the reviewers' and your comments and the kind suggestions on our manuscript. We provide this cover letter to explain, point by point, the details of our revisions in the manuscript and our responses to the reviewers' and your comments as follows. Besides, we have carefully checked through the whole manuscript and corrected some grammar mistakes. We hope the revised paper would satisfy you and the reviewers. We are looking forward to hearing good news from you soon. Here we enclose the electronic file of the revised manuscript and its revision note. All changes are made to the revised manuscript and marked in red in the PDF file, then a summary of the revisions is shown as follows:

Editor's comments:

I am now at the stage in which I am able to make decision on this paper which is looking at an auto-encoder combined with contrast learning for time series anomaly detection. English proof reading is required for this paper to improve its readability. The reviewers have provided a list of requests for clarifications in which the authors need to provide in relation to the code used by the authors and the utilized dataset.

Revised: Thanks very much. We have revised the paper point-to-point according to the reviewers and your suggestions, the main revisions are listed as follows and presented in a pdf file named "Revision_ NEUNET-D-22-01959".

Reviewer #1:

- There are still some typos (e.g. "rather than customizing a model that suitable", "the ability to distinguish abnormal latent representations and constrains the decoder", "employed to analysis", "can achieve to", etc.) I suggest careful proofreading.

Revised: Thanks very much for your kind reminding. We fixed some obvious formatting errors and unified the writing format of technical terms. Meanwhile, we carefully examined the manuscript for grammatical problems and marked corrections in red. Here we list them as follows for reviewing conveniently:

Before (Page 1, Abstract, Paragraph 1, Line 3): ……rather than customizing a model that suitable for anomaly detection. ……
Revised (Page 1, Abstract, Paragraph 1, Line 3): ……rather than customizing a model suitable for anomaly detection. ……

Before (Page 3, Introduction, Paragraph 5, Line 9): ……the ability to distinguish abnormal latent representations and constrains the decoder ……
Revised (Page 3, Introduction, Paragraph 5, Line 9): ……the ability to distinguish between abnormal latent representations and constrains the decoder ……

Before (Page 9, Proposed Model, Contrastive part, Paragraph 1, Line 3): ……and then contrast learning is employed to analysis their relationships ……
Revised (Page 9, Proposed Model, Contrastive part, Paragraph 1, Line 3): ……and then introduce contrast learning to analyze their relationship.……

Before (Page 3, Introduction, Paragraph 6, Line 10): ……the interpretation accuracy of ACVAE can achieve to ……
Revised (Page 3, Introduction, Paragraph 6, Line 10): ……the interpretation accuracy of ACVAE can reach……

- The Related Work is too short. I suggest adding more details about the papers close to ACVAE and highlighting the differences between ACVAE and other works. If it is possible, citing other papers could help to create a deeper understanding of the scenario of this approach.

Revised: Thanks very much for your kind reminding. As you suggested, we add some new references to this manuscript, which are numbered as Ref. [26], [27], [29], [30], [31], the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 3, Related Work, Paragraph 2, Line 2: ……The Maximum Divergence Interval (MDI) [26] defines an unbiased Kullback-Leibler divergence that allows for ranking

regions of different size. It can unsupervisedly detect coherent spatial regions and time intervals, and performs well on synthetic and real data in different fields such as climate analysis, video surveillance, and text forensics. In addition, Trifunov et al. [27] proposed a novel attribution scheme for multivariate time series relying on counterfactual reasoning. Specifically, it detects anomalous intervals using the Maximally Divergent Interval (MDI) algorithm, replaces a subset of variables with their in-distribution values within the detected interval and observe if the interval has become less anomalous, by re-scoring it with MDI. ……

Page 4, Related Work, Paragraph 3, Line 1: Flach et al. [29] proposed a framework that combines feature extraction and anomaly detection algorithms, demonstrating that a carefully selected feature extraction step (such as subtracting seasonal cycles or dimensionality reduction) is more important than selecting a specific anomaly detection algorithm. The Graph-augmented normalizing flow approach (GANF) [30] mainly adds a Bayesian network to the constituent series. The graph is materialized as a Bayesian network, which models the conditional dependencies among constituent time series. A graph-based dependency decoder is designed to summarize the conditional information needed by the normalizing flow that calculates series density. Anomalies are detected through identifying instances with low density. The deep transformer network based anomaly detection and diagnosis model (TranAD) [31] uses attention- based sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data and uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability.

Ref: [26] B. Barz, E. Rodner, Y. G. Garcia, J. Denzler, Detecting regions of maximal divergence for spatio-temporal anomaly detection, IEEE transactions on pattern analysis and machine intelligence 41 (5) (2018) 1088–1101.
[27] V. T. Trifunov, M. Shadaydeh, B. Barz, J. Denzler, Anomaly attribution of multivariate time series using counterfactual reasoning, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2021, pp. 166–172.
[29] M. Flach, F. Gans, A. Brenning, J. Denzler, M. Reichstein, E. Rodner, S. Bathiany, P. Bodesheim, Y. Guanche, S. Sippel, et al., Multivariate anomaly detection for earth observations: a comparison of algorithms and feature extraction techniques, Earth System Dynamics 8 (3) (2017) 677–696.

[30] E. Dai, J. Chen, Graph-augmented normalizing flows for anomaly detection of multiple time series, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022.
[31] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, Proc. VLDB Endow. 15 (6) (2022) 1201–1214.

- In the sentence " It takes the normal Gaussian latent representation z", should that z be bold?

Revised: Thank you a lot. Based on your comments, we have double-checked the areas that need to be bolded in the text and revised them.

- It is not clear to me how the Transformation network works, what its components are, and how it is trained. I think that it is a key component in this proposal, so it should deserve more space and details.

Revised: Thanks a lot. Transformation network T is established based on Gaussian abnormal prior assumption, whose purpose is to generate an abnormal latent representation z_a similar to the normal potential representation z. It takes the normal Gaussian latent representation z encoded by normal data as input, and transforms z into a Gaussian abnormal latent representation z_a with different mean and variance. In practice, the network consists of fully connected layers, and here we choose three hidden layers to build the network. As you suggested, we have added some descriptions of the details of the transformation network, the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 16, Experiments, Implementation Details, Paragraph 3, Line 5: ……The transformation network consists of three fully connected layers, each with the same dimensionality, and the mean and variance will be input into the network as a whole after splicing……

- It seems that delta is not introduced in section 4.1.1 before using it in Equation 7.

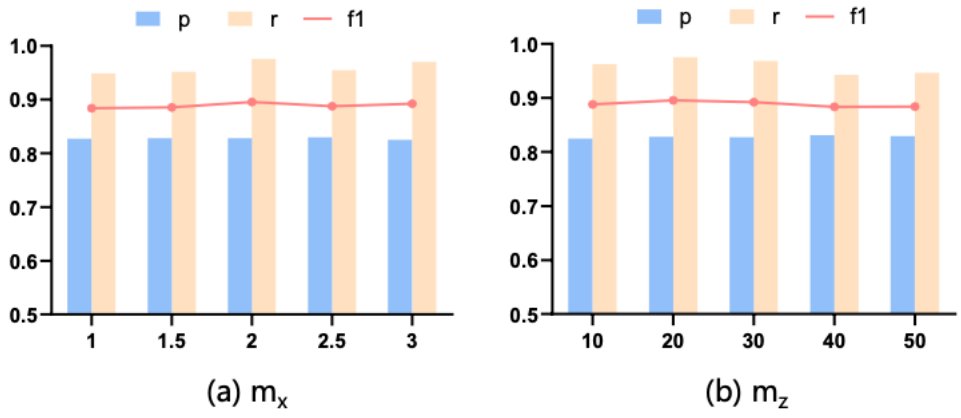
Revised: Thank you very much. Actually, including the symbols you mentioned, delta, phi and theta are the network parameters for T, E and G respectively. According to your suggestion, we have added some descriptions, the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 7, Proposed Model, Network Architecture, Paragraph 1, Line 4: ……δ, φ and θ are the network parameters of T, E and G, respectively. ……

- what is the reasoning behind the margins m_x and m_z? there is only one sentence to explain them but I think that a deeper explanation could help the reader.

Revised: Thank you very much. According to your suggestion, we have added some descriptions, the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

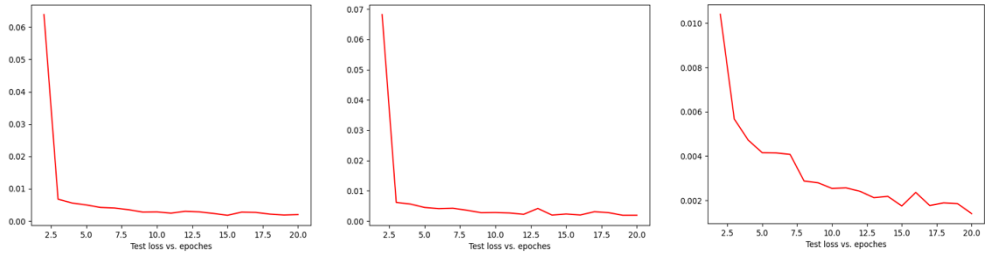
Page 19, Experiments, Sensitivity Analysis, Paragraph 2, Line 1: ……First, we tested the sensitivity of the model to the parameters mx and mz . Fig. 8 (a) summarizes the results about mx using five different ratios [1,1.5,2,2.5,3]. As for mz , we will choose between 10 and 50. Fig. 8 (b) shows the results. mx and mz have similar roles and are set to prevent part of the objective functions from becoming too small during the gradient descent. The model performance is not sensitive to mx and mz , and the choice of these parameters is not critical. Therefore, for convenience, we fix them to 2 and 20, respectively. ……



- since there is adversarial training between T and G, is it possible to have issues typical of GAN like mode collapse? Can one of them reach a point during training in which it fools the other easily?

Revised: Thank you very much, there is actually a difference between the adversarial mechanism in this model and the one in GAN. In GAN, there are only generators and discriminators, and the generators produce similar products, while the discriminators go to distinguish the products from the real inputs, which is a strong adversarial. In

contrast, in ACVAE, the adversarial is between T and G. And due to the design of dynamically changing weights in the loss function, it makes the model pay less attention to the adversarial loss after getting a better generative model. So the typical problems of GAN, such as pattern collapse, do not occur. A few convergence curves are listed to support the above statement. As for the effect of fooling, we can see it from the results of ablation experiment 3. As the results show, G succeeded in having some differentiation ability, which proves the success of the adversarial mechanism.



- In Equation 17, the loss L_{KLD} is not introduced before.

Revised: Thank you very much. According to your suggestion, we have added some descriptions, the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

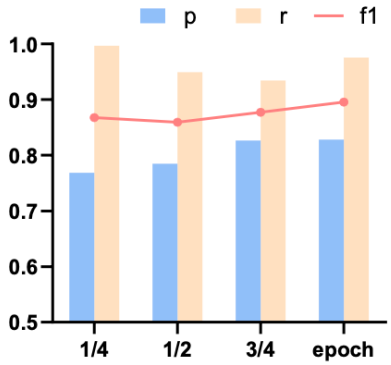
Page 6, Preliminary, Basics of Conv1d, VAE and Adversarial Mechanism, Paragraph 3, Line 6: ……The above can be abbreviated as $L_{KLD}(\mu, \sigma)$. ……

-In Equation 16, lambda is a hyperparameter that is set to the current training epoch. In this way, in the first epochs the loss L_E has no or low impact, while it gains strength in the next epochs. Similar reasoning but the other way around for L_D. How much impact had lambda for ACVAE? Can a fixed value of lambda work anyway or not?

Revised: Thank you very much. According to your suggestion, we have added some descriptions, the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 19, Experiments, Sensitivity Analysis, Paragraph 3, Line 1: ……Second, we explored the effect of λ on the model performance. As shown in Fig. 8 (c), the best results were achieved with dynamically varying lambda values. Such a setting allows

the model to ocus on different purposes at different stages, resulting in better training of the model. ……



(c) λ

- In Table 5, is ACVAE-D instead of ACVAE-G? ACVAE-G is not defined before.

Revised: Thank you very much. There is a writing error here, it should be only ACVAE-D. Based on your comments, we checked all the areas where ACVAE-D appears in the text and made changes.

- The plots are not uniform, it seems that they are generated with different tools. I suggest making them with the same tool/same style.

Revised: Thank you very much for your comment. We carefully examined the images in the text and converted some of them into tables. Here, we list them as follows for reviewing conveniently:

Table 7: The Accuracy HitRate@P% and NDCG@P% of Anomaly Interpretation

Methods	SMD					
	H@100%	H@150%	H@200%	N@100%	N@100%	N@200%
OmniAnomly	0.5054	0.6387	0.7028	0.4541	0.6185	0.6841
adVAE	0.4463	0.5252	0.6372	0.4854	0.5739	0.6928
TranAD	0.4782	**0.6937**	0.7249	**0.6022**	0.5916	0.7397
ACVAE	**0.5412**	0.6742	**0.7617**	0.5624	**0.6887**	**0.7864**

Table 8: Training Time (S/epoch) of SISVAE, ACVAE and adVAE on five datasets.

Method	SKAB	SMD	MSL	SMAP	AQD	UCR
SISVAE	87.7	307.48	504.25	1,129.63	146.22	36.87
ACVAE	3.39	10.42	21.97	52.11	4.21	1.27
adVAE	4.46	12.27	22.52	55.49	5.59	3.48

Table 9: Testing Time(S) of SISVAE, ACVAE and adVAE on five datasets.

Method	SKAB	SMD	MSL	SMAP	AQD	UCR
SISVAE	25.92	42.09	63.63	408.62	4.08	14.87
ACVAE	11.93	33.79	52.17	222.55	9.91	5.62
adVAE	35.14	71.93	174.38	575.09	33.81	9.74

- I did not get why P can assume values above 100 since it represents a percentage of G_{xt}.

Revised: Thank you very much, here p is not a parameter for G_{xt}, the product of p and the length of G_{xt} is used to select the range of A_{xt}. Let's take an example to illustrate this. Suppose there is a five-dimensional observation xt, its A_xt is {5, 2, 3, 4, 1}and G_xt is {5, 1}, if we take P = 150, then Top[⌊P % × |Gxt |⌋]= Top [150% ×2]=Top[3]. This means that we will pick the first three items in A_xt. The first three terms in A_xt are I_xt ={5, 2, 1}, and the intersection with Gxt is {5}. Thus, HitRate@100% = 1/2 = 0.5

- In the Conclusion, I suggest adding more details about the proposal of this paper and adding two examples of future works.

Revised: Thanks very much for your kind reminding. We revised the conclusion based on your comments. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 22, Conclusion, Paragraph 1, Line 4: ……With the constraint of two regularizers, the ACVAE is verified to be more suitable for anomaly detection tasks as its high accuracy and efficiency. Specifically, ACVAE introduces an adversarial mechanism that allows the decoder to be constrained and gain some discriminatory power. At the same time, ACVAE introduces contrast learning, which allows the encoder to obtain more training samples and be able to identify anomalies. In addition,

ACVAE provides a method for analyzing anomalies, i.e., identifying the main cause of anomalies based on the scores of each dimension……

Page 22, Conclusion, Paragraph 2, Line 1: For the future, we propose to explore the prior distribution of the method. Real-world data often present diverse distributions, and it is difficult to state that a normal distribution is the most applicable assumption for anomaly detection. We would also like to further extend the setting of anomaly scores. The traditional VAE-based anomaly score calculation treats each dimension equally, however, such a setting is clearly flawed when a variable is uncorrelated with other variables. In addition, it is a challenge to cope with non-smooth data.

Reviewer #2:

The paper tackles outlier detection in multivariate time-series data. There is an essential difference to anomaly detection in general since, in many applications, not an individual point in time is essential to identify, but an interval that shows the anomalous behaviour of a dynamic system. This summarises my first criticism. The authors have to discuss this point and indicate how their method can be extended to interval detection.

Revised: Thank you for your comments, for real applications it is not important to identify individual anomalous time points, we need to show intervals of anomalous behavior of dynamic systems. We describe this setup in detail in the Experiments section. Here, we list them as follows for reviewing conveniently:

Page 14, Experiments, Evaluation Metrics, Paragraph 1, Line 7: It is worth mentioning that in practical applications, anomaly observations often occur continuously and form anomaly segments. It is acceptable if an alarm can be triggered within the anomaly segment. Therefore [49] proposes a point adjustment method to calculate the performance. If the algorithm can detect any point in the anomaly segment, we consider that the segment is correctly detected and all observations in the segment are correctly detected as anomalies. The observations outside the anomaly segment are treated as usual. Many subsequent works [7],[9],[10], including this article, will also use point adjustment methods to calculate evaluation metrics.

Ref: [49] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al., Unsupervised anomaly detection via variational auto-encoder for seasonal

kpis in web applications, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 187–196.

[7] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.

[9] X. Chen, L. Deng, F. Huang, C. Zhang, Z. Zhang, Y. Zhao, K. Zheng, Daemon: Unsupervised anomaly detection and interpretation for multivariate time series, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 2225–2230.

[10] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.

The second criticism relates to the literature review. Most of the referenced works are old and cannot be seen as state-of-the-art or the basis for a proceeding in anomaly detection. Most importantly, a lot of important work is missing, like

- Barz et al.: Detecting Regions of Maximal Divergence for Spatio-Temporal Anomaly Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 41 (5) : 2019.
- Trifonov et al.: Anomaly Attribution of Multivariate Time Series using Counterfactual Reasoning. IEEE International Conference on Machine Learning and Applications (ICMLA). 2021
- Flach et al.: Multivariate anomaly detection for Earth observations: a comparison of algorithms and feature extraction techniques. Earth System Dynamics. 8 (3) : pp. 677-696. 2017
- Dai et al.: Graph-Augmented Normalizing Flows for Anomaly Detection of Multiple Time Series. International Conference on Learning Representations (ICLR) 2022.
- Tuli et al.: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. Proc. VLDB Endow. 2022

Revised: Thanks very much for your kind reminding. As you suggested, we add some new references to this manuscript, which are numbered as Ref. [26], [27], [29], [30], [31], the detailed description is marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 4, Related Work, Paragraph 2, Line 2: ……The Maximum Divergence Interval (MDI) [26] defines an unbiased Kullback-Leibler divergence that allows for ranking regions of different size. It can unsupervisedly detect coherent spatial regions and time intervals, and performs well on synthetic and real data in different fields such as climate analysis, video surveillance, and text forensics. In addition, Trifunov et al. [27] proposed a novel attribution scheme for multivariate time series relying on counterfactual reasoning. Specifically, it detects anomalous intervals using the Maximally Divergent Interval (MDI) algorithm, replaces a subset of variables with their in-distribution values within the detected interval and observe if the interval has become less anomalous, by re-scoring it with MDI. ……

Page 4, Related Work, Paragraph 3, Line 1: Flach et al. [29] proposed a framework that combines feature extraction and anomaly detection algorithms, demonstrating that a carefully selected feature extraction step (such as subtracting seasonal cycles or dimensionality reduction) is more important than selecting a specific anomaly detection algorithm. The Graph-augmented normalizing flow approach (GANF) [30] mainly adds a Bayesian network to the constituent series. The graph is materialized as a Bayesian network, which models the conditional dependencies among constituent time series. A graph-based dependency decoder is designed to summarize the conditional information needed by the normalizing flow that calculates series density. Anomalies are detected through identifying instances with low density. The deep transformer network based anomaly detection and diagnosis model (TranAD) [31] uses attention- based sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data and uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability.

Ref: [26] B. Barz, E. Rodner, Y. G. Garcia, J. Denzler, Detecting regions of maximal divergence for spatio-temporal anomaly detection, IEEE transactions on pattern analysis and machine intelligence 41 (5) (2018) 1088–1101.

[27] V. T. Trifunov, M. Shadaydeh, B. Barz, J. Denzler, Anomaly attribution of multivariate time series using counterfactual reasoning, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2021, pp. 166–172.

[29] M. Flach, F. Gans, A. Brenning, J. Denzler, M. Reichstein, E. Rodner, S. Bathiany, P. Bodesheim, Y. Guanche, S. Sippel, et al., Multivariate anomaly detection for earth observations: a comparison of algorithms and feature extraction techniques, Earth System Dynamics 8 (3) (2017) 677–696.

[30] E. Dai, J. Chen, Graph-augmented normalizing flows for anomaly detection of multiple time series, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022.
[31] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, Proc. VLDB Endow. 15 (6) (2022) 1201–1214.

Third, the selected benchmark datasets are not state-of-the-art. They are related to the referenced and compared with work but not to the current challenges in this area. I am recommending the UCR Anomaly Archive; if not suited, they have to explain in detail why this standard archive is not suited for their work. A recent paper compares several SotA methods (Rewicki et al.: Is It Worth It? Comparing Six Deep and Classical Methods for Unsupervised Anomaly Detection in Time Series. Applied Sciences. 13 (3) : 2023).

Revised: Thanks very much for your kind reminding. We add TranAD as a baseline and add a public dataset UCR, on the basis of which we have extended the discussion of the experimental results, and marked the relevant details in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 13, Experiments, Protocols and Settings, Public Datasets, Paragraph 1, Line 8: ……UCR is a dataset of multiple univariate time series that was used in KDD 2021 cup [54, 55]. We include only the datasets obtained from natural sources (the InternalBleeding datasets) and ignore the synthetic sequences. ……

Ref: [54] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, The ucr time series archive, IEEE/CAA Journal of Automatica Sinica 6 (6) (2019) 1293–1305.
[55] F. Rewicki, J. Denzler, J. Niebling, Is it worth it? comparing six deep and classical methods for unsupervised anomaly detection in time series (2023).

Page 14, Table 2:

Table 2: Statistics of the datasets

Dataset name	Subset number	Dimension number	Training set size	Testing set size	Anomaly ratio(%)
MSL	27	55	58317	73729	10.72
SMAP	55	25	135183	427617	13.13
SKAB	21	8	9401	22474	34.82
SMD	28	38	708405	708420	4.16
AQD	6	6	70200	70200	11.13
UCR	13	1	31700	65796	2.29

Page 14, Experiments, Protocols and Settings, Compared Methods, Paragraph 8, Line 1: …… 7)TranAD [31]. A reconstruction-based model that detects anomalies quickly by incorporating attention mechanisms and using adversarial training. ……

Ref: [31] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, Proc. VLDB Endow. 15 (6) (2022) 1201–1214.

Page 13, Table 5:

Methods	MSL			SMAP			SMD			SKAB			AQD			UCR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
IForest	0.5531	0.9654	0.7033	0.5337	0.5907	0.5608	0.1082	0.9992	0.1952	0.3537	1	0.5225	0.3255	0.9684	0.487	0.6482	0.7845	0.7098
OCSVM	0.5386	0.8999	0.6739	0.4656	0.7631	0.5783	0.2302	0.9570	0.3711	0.3482	1	0.5166	0.1063	1	0.192	0.5636	0.7213	0.6328
HBOS	0.6734	0.9687	0.7945	0.5963	0.5907	0.5935	0.2572	0.9718	0.4067	0.3861	1	0.5571	0.5328	0.8481	0.654	0.7363	0.8527	0.7902
VAE	0.3715	0.9997	0.5417	0.4181	1	0.5897	0.1844	0.9897	0.3108	0.7021	1	0.825	0.3681	0.7573	0.493	0.5473	0.9814	0.7027
DAGMM	0.5935	0.9673	0.7356	0.5666	0.9013	0.6958	0.5719	0.9849	0.7236	0.6214	0.9964	0.7654	0.4563	0.6082	0.521	0.5143	0.9624	0.6704
LSTM-NDT	0.5843	0.5642	0.5741	0.7246	0.9876	0.8359	0.5684	0.6438	0.6038	0.5862	1	0.7391	0.6285	0.5275	0.573	0.5492	0.8147	0.6561
OmniAnomly	0.9011	0.8647	0.8825	0.9246	0.5536	0.6925	0.8348	0.9107	0.8711	0.8404	0.9999	0.9133	0.7837	0.6712	**0.723**	0.7485	0.9425	0.8344
adVAE	0.9116	0.8669	0.8887	0.8018	0.9964	**0.8885**	0.7016	0.8576	0.7718	0.9121	1	0.9541	0.4430	0.6278	0.519	0.8738	0.9274	0.8998
SISVAE	0.9125	0.8875	0.8998	0.5302	0.7401	0.6178	0.5325	0.6549	0.5874	0.9874	0.9312	**0.9585**	0.1078	0.9464	0.193	0.7962	0.8512	0.8229
TranAD	0.8823	0.9222	**0.9018**	0.7856	0.9973	0.8789	0.8882	0.9026	**0.8953**	0.8531	1	0.9208	0.5011	0.8226	0.622	0.8907	1	**0.9422**
ACVAE-E	0.8435	0.3939	0.5370	0.7498	0.5907	0.6608	0.6833	0.8155	0.7436	0.9871	0.6140	0.7571	0.6690	0.7558	0.709	0.6894	0.9419	0.7961
ACVAE-D	0.8879	0.9163	0.8967	0.8265	0.9350	0.8774	0.8225	0.7171	0.7662	0.9852	0.6495	0.7529	0.3198	0.6750	0.434	0.8832	0.5029	0.6409
ACVAE	0.9157	0.9081	**0.9119**	0.8280	0.9756	**0.8958**	0.9374	0.9342	**0.9358**	0.9479	0.9813	**0.9643**	0.7341	0.7772	**0.755**	0.9254	0.9527	**0.9388**

Page 17, Experiments, Results and Discussions, Paragraph 5, Line 12: …… but its complex training method and failure to consider time dependence make it underperform on complex and diverse time-series datasets. TranAD performs slightly better than ACVAE on the UCR dataset, indicating the advantage of the attention mechanism for the univariate time series anomaly detection problem. To be able to extend the model to univariate time series anomaly detection, the attention mechanism may be a good direction. ……

Page 20, Table 6:

Table 6: Comparison of threshold selection

Evaluation metrics	MSL	SMAP	SMD	SKAB	AQD	UCR
F1 - KDE	0.9119	0.8958	0.9358	0.9643	0.7551	0.9388
F1 - Best	0.9228	0.9091	0.9461	0.9755	0.7692	0.9527

From a methodological point of view, the paper has few new ideas. It builds upon existing methods (VAE, GAN-like training, contrastive learning) and combines them innovatively. I like the idea of coming up with an automatic threshold selection scheme. However, I think this could be better elaborated. Also, I need help finding the explanation of the anomalies convincing. More evaluation is necessary to prove that this can be seen as a significant additional contribution to the suggested approach. One

of my main concerns is: (24) looks like a covariance computation and ignores the data points before x_t. This sounds not reasonable as an explanation module.

Revised: Thank you for your comments. Firstly, ACVAE introduces an adversarial mechanism, and we add a transformation network T so that it constitutes an adversarial relationship with the encoder and decoder, instead of constructing an encoder-discriminator adversary directly as in GAN. In this way, the decoder is able to receive two different inputs, thus gaining some discriminative power and being constrained to avoid a posteriori collapse. Second, we also introduce contrast learning for ACVAE, by feeding the reconstruction product to the encoder again and discriminating it, so that the encoder gets more training data than just normal data, and the encoder also gains some discriminative power. Finally, in order to train the model better, we set a dynamic weight in the objective function, which will change as the epoch increases. This setting makes the loss drop smoother and gives a better detection model. In summary, we believe that our work is novel.

Revised: Thank you for your comments. We revised the automatic-threshold-selection based on your comments. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 12, Proposed Model, Automatic Threshold Selection, Paragraph 5, Line 4: ……which means that the sample with anomaly score $S \geq Sa$ has at least $1 - \alpha$ probability of being abnormal. Because KDE decides the threshold by estimating the interval of normal data's anomaly scores, it is clear that using KDE to decide the threshold is more objective and reasonable than simply relying on human experience. A higher significance level α leads to a lower missing alarm rate, which means that models have fewer chances to mislabel outliers as normal data. On the contrary, a lower α means a lower false alarm rate. Therefore, the choice of the significance level α is a tradeoff. There is only one parameter α that needs to be tuned in the technique of KDE. ……

Revised: Thank you for your comments. Regarding the public notice (24), we just want to show that with the existing anomaly scores we can obtain the covariance of each dimension. We revised the anomaly-interpretation based on your comments. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 21, Experiments, Automatic Threshold Selection and Anomaly interpretation, Anomaly interpretation, Paragraph 1, Line 9: ……We also measure the Normalized Discounted Cumulative Gain (NDCG) [9]. NDCG@P% considers the same number of top predicted candidates as HitRate@P%……

Page 21, Experiments, Automatic Threshold Selection and Anomaly interpretation, Anomaly interpretation, Paragraph 3, Line 1: ……As shown in Table 7 where H and N correspond to HitRate and NDCG (with complete data used for model testing), …… In most P -selection schemes, the interpretation accuracy of ACVAE is higher than those of others, which reflects the superiority of our model.……

Table 7: The Accuracy HitRate@P% and NDCG@P% of Anomaly Interpretation

Methods	SMD					
	H@100%	H@150%	H@200%	N@100%	N@100%	N@200%
OmniAnomly	0.5054	0.6387	0.7028	0.4541	0.6185	0.6841
adVAE	0.4463	0.5252	0.6372	0.4854	0.5739	0.6928
TranAD	0.4782	**0.6937**	0.7249	**0.6022**	0.5916	0.7397
ACVAE	**0.5412**	0.6742	**0.7617**	0.5624	**0.6887**	**0.7864**

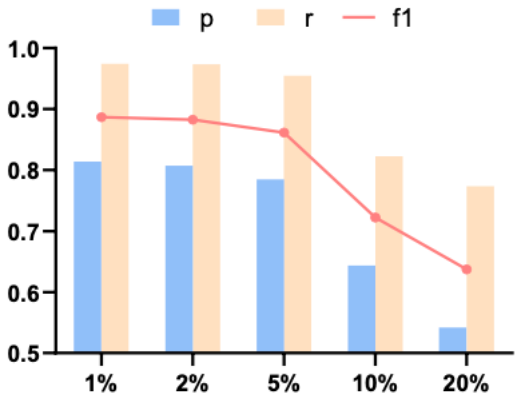
Ref: [9] X. Chen, L. Deng, F. Huang, C. Zhang, Z. Zhang, Y. Zhao, K. Zheng, Daemon: Unsupervised anomaly detection and interpretation for multivariate time series, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 2225–2230.

Nothing is said about non-stationary time series data. Since the model is trained beforehand on data, distribution shifts will generate anomalies. However, practical methods should be able to deal with such data, although non-stationary data is the most challenging situation for anomaly detection methods. My main criticism is twofold: First, training on some data (assumed to be "clean") limits the method. Second, normalising (preprocessing) the data (see 4.2.1) prevents the technique from detecting slight changes in the statistics over time.

Revised: Thank you for your comment. Since ACVAE requires training on normal datasets, it is important to investigate how ACVAE responds to contaminated training data. We conducted sensitivity experiments to investigate its impact on model performance. In practice, normal data is easy to collect, so the pollution rate is usually kept at a low level. Therefore, training on clean data is feasible. Our method has

achieved good results in this application field, but this does not mean that the exploration is over. The next direction of our work is to develop an anomaly detection method that can handle the presence of pollution and noise in training data. As for the preprocessing issue, we believe it is necessary to standardize data with different scales, which is beneficial for model training (this has been repeatedly demonstrated). But the overlooked subtle changes you mentioned do exist. This is a very interesting direction that deserves further consideration. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 20, Experiments, Sensitivity Analysis, Paragraph 4, Line 1: ……Since ACVAE requires training on normal datasets, it is necessary to study the response of the model to a contaminated training set. As shown in Fig. 8 (d), the model maintains good results at 5% noise, and the model performance decreases when the percentage of noise in the training set reaches 10%. Finally, a significant decrease in model performance can be seen for the high noise case (20%). This is because the VAE-based approach always tries to reduce the reconstruction loss of all training sets and the contaminated dataset also tends to affect the discriminative loss of ACVAE. However, in practice, normal data are easy to collect and it is unrealistic for the training set to show such high contamination. Therefore ACVAE is unlikely to face this situation during training. ……



(d) Contamination Ratio

The computation of the anomaly scores treats every dimension equally (see (20)). However, what will happen if one of the variables observed has nothing to do with the rest (or is white noise)? Could you handle this by the method? I would assume that the influence of this dimension should be zero for anomaly detection.

Revised: Thank you for your comment. The calculation of the anomaly score treats each dimension equally. Most VAE based models use this setup is considering the universal case. In real production, there are cases where a variable is uncorrelated (or white noise) with other variables. There are also many solutions for this particular type of problem. For example, consider using attention mechanisms or rescaling the anomaly score. Thus we need to synthesize the dataset manually or find a targeted real dataset and then consider handling the noise or try to find irrelevant variables to distinguish in the model design phase. This type of problem is worth exploring in depth as an important challenge in the field of anomaly detection. For this reason we include it in our conclusion as a future direction to explore. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 22, Conclusion, Paragraph 2, Line 3: ……We would also like to further extend the setting of anomaly scores. The traditional VAE-based anomaly score calculation treats each dimension equally, however, such a setting is clearly flawed when a variable is uncorrelated with other variables. In addition, it is a challenge to cope with non-smooth data. ……

I like to read a discussion about whether the authors see some limits of their approach in the used Gaussianity assumptions made across the paper.

Revised: Thank you for your comment. In fact, the advantage of the depth model is that no strict assumptions are made. However, to be able to generate the anomalous distribution we need for the intermediate process, we use anomalous prior assumptions. Although the Gaussian distribution is the most common distribution in nature, it is not fully representative of all real data sets. It is also not possible to simply generalize anomalies to a distribution. In the text we explain it this way - "Although negative samples are not necessarily true anomalies, as long as the encoder can identify them, the introduction of It means that this training method will give the model the ability to distinguish abnormal inputs. ". We are currently working on a work which is exactly about the distribution problem in the model. In practice, we analyze industrial datasets including environmental datasets, and we find that most of the industrial datasets show a log-normal distribution, which is a long-tailed distribution. We will refine the model based on this discovery and modify the measurement of distribution similarity. This type of problem is worth exploring in depth as an important challenge in the field of anomaly detection. For this reason we include it in our conclusion as a future direction

to explore. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 22, Conclusion, Paragraph 2, Line 1: ……For the future, we propose to explore the prior distribution of the method. Real-world data often present diverse distributions, and it is difficult to state that a normal distribution is the most applicable assumption for anomaly detection. ……

One comment on the overall structure of the paper. The presentation could be improved a lot. Since most parts come from existing work/literature, it is difficult to identify the individual factors generating novelty. Also, the figures (Fig 3, 4, and 5) could be improved. For me, they add nothing to the text.

Revised: Thank you for the above suggestions. Based on the reviewers' suggestions, we have revised the Introduction and Proposed Model sections to focus on the innovative nature of this work. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 3, Introduction, Paragraph 5, Line 13: ……We will choose five public datasets with different anomaly proportions, in addition to which we conduct a study with proprietary data from a location in central China to test whether the proposed method meets the training speed and high performance requirements proposed by the institution. The main contributions of this paper are summarized as follows: ……

Page 9, Proposed Model, Adversarial part, Paragraph 4, Line 9: ……Thus, we obtain a constrained decoder that achieves a certain degree of discriminative ability under the action of adversarial mechanisms and can alleviate posterior collapse. This decoder is more suitable for anomaly detection tasks. ……

Revised: Fig3 is the overall network structure of the model, which we refer to in the text to better introduce the algorithm flow. Fig4 is a pictorial presentation of two important mechanisms of the model, whose presence allows the reader to better understand the components of this work. Fig5 is a structural diagram of the two ablation models to illustrate the main structure of the ablation model under the removal of a component.

Finally, for the experiments, it remains unclear whether the authors used code provided

by the authors of the different methods or whether they re-implemented it. Also, I need a justification of the parameters in Table 4. How sensitive are the results concerning changes in exactly this topology/settings?

Revised: Thank you for your comments. All comparison models for this work use code provided by the authors or implemented on github at the following address:

IForest , OCSVM, HBOS and VAE - https://github.com/yzhao062/pyod

DAGMM - https://github.com/RomainSabathe/dagmm

LSTM-NDT - https://github.com/khundman/telemanom

OmniAnomly - https://github.com/NetManAIOps/OmniAnomaly

AdVAE - https://github.com/WangXuhongCN/adVAE

SISVAE - https://gitee.com/fake-cat/sisvae

TranAD - https://github.com/imperial-qore/TranAD

Revised: Thank you for your comments. We revised the conclusion based on your comments. The relevant details are marked in red in the manuscript. Here, we list them as follows for reviewing conveniently:

Page 20, Experiments, Sensitivity Analysis, Paragraph 5, Line 1: ……Finally, we investigate the effect of convolutional network structure. As shown in Fig. 8 (e), we considered three additional structures, V1(64(2,18,0), 128(8,1,0)), V2(64(4,4,0), 128(4,4,0), 256(4,4,0), 128(2,1,0)) and V4(8(4/2/1), 16(4/2/1), 32(4/2/1), 64(4/2/1), 128(4/2/1), 256(4/2/1), 256(2/1/1), 128(2/1/0)). From the results, it is clear that the model performance does not get better as the number of layers increases. Our currently selected structure V3 performs better, but it does not mean that it is the best choice. Therefore, the network structure tuning of neural networks is a complex process and it is worth exploring. ……

A summary of the revisions:
(1) The linguistic quality has been improved.
(2) All questions proposed by reviewers and editor are explained.
(3) Some experimental data have been added and the contents of the manuscript are more developed.

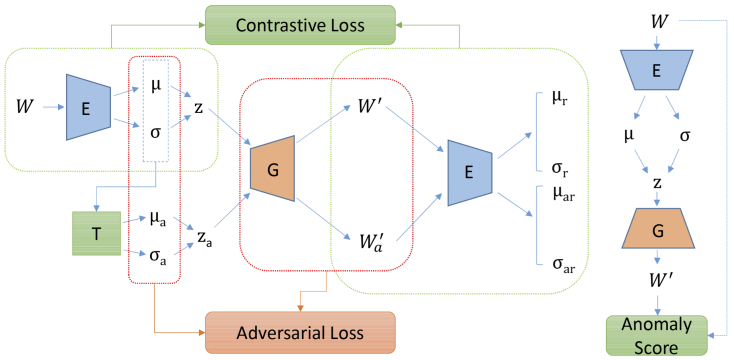
Thank you again for your help, we sincerely hope this paper can be accepted and published in Neural Networks and we are looking forward to hearing good news from you soon.

Yours Sincerely,
Xiaoxia Zhang, Shang Shi, HaiChao Sun, Degang Chen, Guoyin Wang, Kesheng Wu

Graphical Abstract

ACVAE: a Novel Self-adversarial Variational Auto-Encoder Combined with Contrast Learning for Time Series Anomaly Detection

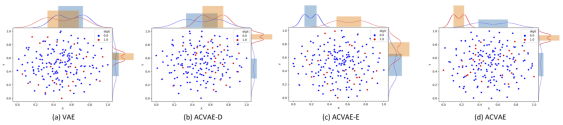
Xiaoxia Zhang, Shang Shi, HaiChao Sun, Degang Chen, Guoyin Wang, Kesheng Wu



(a) Training part

(b) Testing part

Methods	MSL			SMAP			SMD			SKAB			AQD		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
IForest	0.5531	0.9654	0.7033	0.5337	0.5907	0.5608	0.1082	0.9992	0.1952	0.3537	1	0.5225	0.3255	0.9684	0.4873
OCSVM	0.5386	0.8999	0.6739	0.4656	0.7631	0.5783	0.2302	0.9570	0.3711	0.3482	1	0.5166	0.1063	1	0.1922
HBOS	0.6734	0.9687	0.7945	0.5963	0.5907	0.5935	0.2572	0.9718	0.4067	0.3861	1	0.5571	0.5328	0.8481	0.6545
VAE	0.3715	0.9997	0.5417	0.4181	1	0.5897	0.1844	0.9897	0.3108	0.7021	1	0.825	0.3681	0.7573	0.4954
DAGMM	0.5935	0.9673	0.7356	0.5666	0.9013	0.6958	0.5719	0.9849	0.7236	0.6214	0.9964	0.7654	0.4563	0.6082	0.5214
LSTM-NDT	0.5843	0.5642	0.5741	0.7246	0.9876	0.8359	0.5684	0.6438	0.6038	0.5862	1	0.7391	0.6285	0.5275	0.5736
OmniAnomly	0.9011	0.8647	0.8825	0.9246	0.5536	0.6925	0.8348	0.9107	**0.8711**	0.8404	0.9999	0.9133	0.7837	0.6712	**0.7231**
adVAE	0.9116	0.8669	0.8887	0.8018	0.9964	**0.8885**	0.7016	0.8576	0.7718	0.9121	1	0.9541	0.4430	0.6278	0.5195
SISVAE	0.9125	0.8875	**0.8998**	0.5302	0.7401	0.6178	0.5325	0.6549	0.5874	0.9874	0.9312	**0.9585**	0.1078	0.9464	0.1936
ACVAE-E	0.8435	0.3939	0.5370	0.7498	0.5907	0.6608	0.6833	0.8155	0.7436	0.9871	0.6140	0.7571	0.6690	0.7558	0.7097
ACVAE-G	0.8879	0.9163	0.8967	0.8265	0.9350	0.8774	0.8225	0.7171	0.7662	0.9852	0.6495	0.7529	0.3198	0.6750	0.4340
ACVAE	0.9157	0.9081	**0.9119**	0.8280	0.9756	**0.8958**	0.9374	0.9342	**0.9358**	0.9479	0.9813	**0.9643**	0.7341	0.7772	**0.7551**



Highlights

ACVAE: a Novel Self-adversarial Variational Auto-Encoder Combined with Contrast Learning for Time Series Anomaly Detection

Xiaoxia Zhang, Shang Shi, HaiChao Sun, Degang Chen, Guoyin Wang, Kesheng Wu

- Proposed a novel self-adversarial variational auto-encoder combined with contrast learning for anomaly detection.
- Explored the methods of anomaly interpretation and automatic threshold selection in our model.
- The generation model with contrast learning has higher detection accuracy and less time consumption.
- The adversarial mechanism and contrast learning are employed to make the model more suitable for anomaly detection and avoid the problem of of posterior collapse.

ACVAE: a Novel Self-adversarial Variational Auto-Encoder Combined with Contrast Learning for Time Series Anomaly Detection

Xiaoxia Zhang[a,b,*], Shang Shi[a,b], HaiChao Sun[a,b], Degang Chen[c], Guoyin Wang[a,b,*], Kesheng Wu[d]

[a] *Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China*
[b] *Key Laboratory of Big Data Intelligent Computing, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China*
[c] *Department of Mathematics and Physics, North China Electric Power University, Beijing, 102206, China*
[d] *Computational Research Division,, Lawrance Berkeley National Laboratory, Berkeley, 94720, USA Í*

Abstract

Deep generative models have advantages in modeling complex time series and are widely used in anomaly detection. Nevertheless, the existing deep generative approaches mainly concentrate on the investigation of models' reconstruction capability rather than customizing a model suitable for anomaly detection. Meanwhile, VAE-based models suffer from posterior collapse, which can lead to a series of undesirable consequences, such as high false positive rate etc.. Based on these considerations, in this paper, we propose a novel self-adversarial variational auto-encoder combined with contrast learning, short for ACVAE, to address these challenges. ACVAE consist of three parts $\langle T, E, G\rangle$, wherein the transformation network T is employed to generate abnormal latent representations similar to those normal latent representations encoded by the encoder E, and the decoder G is used to distinguish the two representations. In the framework of this model, the normal reconstructions are considered as positive samples and abnormal reconstructions as negative samples, and the contrast learning is executed on the part E to measure the similarities between inputs and positive samples, dissimilarities between inputs and negative samples.Thus, an improved objective function is proposed by integrating two novel regularizers, one refers to adversarial mechanism and the other involves contrast learning, in which the encoder E and decoder G hold the capability to distinguish, and decoder G is constrained to mitigate the posterior collapse. We perform several experiments on five datasets, whose results show ACVAE outperforms state-of-the-art methods.

Keywords: Variational auto-encoder(VAE), deep generative model, anomaly detection, contrast learning, multivariate time series, anomaly interpretation

1. Introduction

The purpose of anomaly detection [1] is to identify data patterns that deviate significantly from expectations, and it helps to discover and analyze unexpected features that cannot be simply detected. Therefore, anomaly detection has attracted a lot of attention in many fields such as intrusion detection [2], healthcare [3], fault detection [4], environmental monitoring [5], and security inspection [6].

*Corresponding author: XiaoXia Zhang and Guoyin Wang

In recent years, with the development of the Internet, it is easy for various fields to accumulate a large amount of time series data in a short period of time, which brings challenges to the anomaly detection of time series data. Compared with general static data, for high-dimensional time-series big data, we need to consider not only the time-dependence of the data, but also the possible correlations among dimensions in a reasonable way. Therefore, this paper focuses more on anomaly detection for multivariate time series at the entity level rather than considering individual dimensional time series separately at the metric level [7, 8, 9, 10]. This is because it is labor-intensive to build and maintain a detection system for each dimension (e.g., the 55-dimensional MSL dataset in Table 2). And even if we get information about anomalies on individual dimensions, we have to use extensive domain knowledge to synthesize the anomalies on entities, which is a challenge.

However, in actual applications, there are still some problems, the most prominent is the lack of labels. Labeling needs to be done manually using expert knowledge. Therefore, facing the number of data reaching millions, it is difficult for us to get enough labeled data for supervised learning. In this paper, we will focus on unsupervised methods to understand their practicality. This is also in line with the current trend of increasing research on unsupervised anomaly detection algorithms, such as AnRAD [11], AD-HKDE [12], GGM-VAE [13].

Recently, deep generative networks have received a lot of attention in the field of anomaly detection [14], which is a reconstruction-based approach whose core idea is to learn the probability distribution of normal data by training on anomaly-free data sets, and then when anomalous data are input, they will not reconstruct well, i.e., deviate from the normal data pattern,thus achieving anomaly detection.However, the anomaly detection methods based on deep generative models still have some problems:

1. A deep generative model is essentially a reconstruction model whose purpose is to reconstruct the data distribution rather than anomaly detection. Several prior works [7, 8, 9, 10] have focused on enhancing the reconstruction capabilities of the model rather than customizing a model suitable for anomaly detection. Thus, there is a serious problem here: these models learn only from the available normal samples, ignoring the importance of distinguishing anomalous data. When there are some anomalies that are close to the normal data distribution, these methods will have difficulty distinguishing between them [15]. From a macroscopic point of view, it is also difficult to learn latent representations that are valid for anomaly detection tasks by only enhancing the reconstruction capability.
2. There is a posterior collapse [16], which is a common problem in VAE-based models. VAE uses Kullback–Leibler divergence (KLD) to constrain the encoder, but not the decoder. Such a powerful decoder will ignore the information from z and reconstruct independently of z, resulting in a vanishing posterior [17], when the VAE simply maps $P(X)$ to the prior $p(z)(\mathcal{N}(0,1))$ instead of the true posterior distribution $p(z \mid x)$ [18].Once this occurs, the VAE will over-fit the existing distribution of normal data, resulting in undesirable consequences, such as normal data being misclassified as abnormal. Therefore, it is very important to constrain the decoder.

Based on theses considerations, we propose a novel self-adversarial variational auto-encoder combined with contrast learning (ACVAE), wherein two additional regularizers are proposed to construct the objective function by employing adversarial mechanism and contrast learning, which aims to solve the problems mentioned above. First, we adopt the anomalous prior mentioned in [19] (see Fig. 1), which is a weak assumption justified by the fact that the Gaussian distribution is one of the most commonly used distributions. As shown in Fig. 3, we set up an adversarial mechanism

to make the transformation network synthesize abnormal latent representation that are similar to the normal latent representation and make the decoder to distinguish them as much as possible, which gives the decoder the ability to distinguish between abnormal latent representations and constrains the decoder to avoid posterior collapse. Meanwhile, we introduce the contrast learning into encoder E to enhance its coding capabilities by treating normal reconstructions as positive samples and abnormal reconstructions as negative samples, which greatly increases the amount and variety of training data and gives the encoder the ability to distinguish abnormal inputs. We will choose five public datasets with different anomaly proportions, in addition to which we conduct a study with proprietary data from a location in central China to test whether the proposed method meets the training speed and high performance requirements proposed by the institution.

The main contributions of this paper are summarized as follows:

- We propose a novel self-adversarial variational auto-encoder combined with contrast learning (ACVAE) for anomaly detection. In the framework of this model, the adversarial mechanism and contrast learning are employed to make the model more suitable for anomaly detection and avoid the problem of posterior collapse. To the best of our knowledge, this is the first time that contrast learning is introduced into anomaly detection.

- We explore the methods of anomaly interpretation and automatic threshold selection in our model, wherein the results demonstrate the threshold extracted from KDE (a automatic threshold selection method) is almost same to the best threshold, and the interpretation accuracy of ACVAE can reach 0.76 which is higher than those of other methods.

- We conduct several experiments to testify the effectiveness of ACVAE, the results demonstrate our method outperforms the state-of-the-art methods on five public datasets and a real-world environmental dataset. Additionally, we exploit the time consumption of ACVAE and the results indicate the training and detection time of ACVAE is relatively small compared to other methods.

The rest of the paper is organized as follows: Section 2 briefly reviews the the work related to the anomaly detection. Section 3 introduces the preliminary knowledge used in this paper. Section 4 characterizes the concrete implementations of ACVAE. Extensive experimental results on detection and analysis are presented in Section 5. Section 6 concludes the paper.

2. Related Work

Anomaly detection in time series has been extensively studied and can be broadly classified into the following categories [20]: clustering-based methods [21], boundary-based methods [22], distance-based methods [23], and ensemble-learning-based methods [24].

In addition to traditional methods, deep learning-based anomaly detection methods are now the mainstream [25]. The Maximum Divergence Interval (MDI) [26] defines an unbiased Kullback-Leibler divergence that allows for ranking regions of different size. It can unsupervisedly detect coherent spatial regions and time intervals, and performs well on synthetic and real data in different fields such as climate analysis, video surveillance, and text forensics. In addition, Trifunov et al. [27] proposed a novel attribution scheme for multivariate time series relying on counterfactual reasoning. Specifically, it detects anomalous intervals using the MDI algorithm, replaces a subset of variables with their in-distribution values within the detected interval and observe if the interval has become less anomalous, by re-scoring it with MDI. The Deep Autoencoding Gaussian Mixture

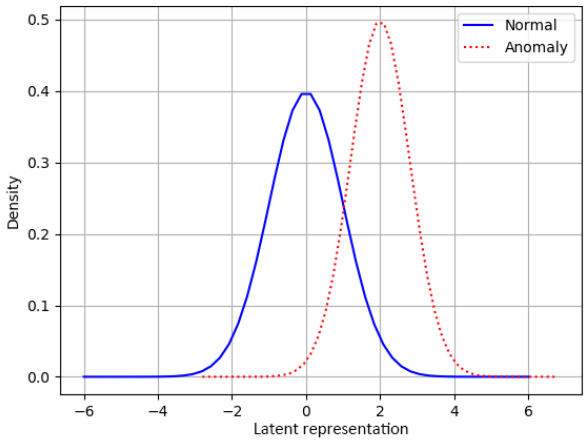


Figure 1: Gaussian anomaly prior assumption. (The prior distribution of normal data is close to the standard normal distribution, the abnormal prior distribution is a Gaussian distribution with unknown mean and variance, and there is overlap between the two.)

Model (DAGMM) [28] combines deep automatic encoding (AE) and Gaussian mixture model (GMM). It downscales the input to obtain the latent representation and then uses the GMM to estimate the density of the representation. However, this approach ignores the time dependence and thus is not suitable for time series anomaly detection.

Flach et al. [29] proposed a framework that combines feature extraction and anomaly detection algorithms, demonstrating that a carefully selected feature extraction step (such as subtracting seasonal cycles or dimensionality reduction) is more important than selecting a specific anomaly detection algorithm. The Graph-augmented normalizing flow approach (GANF) [30] mainly adds a Bayesian network to the constituent series. The graph is materialized as a Bayesian network, which models the conditional dependencies among constituent time series. A graph-based dependency decoder is designed to summarize the conditional information needed by the normalizing flow that calculates series density. Anomalies are detected through identifying instances with low density. The deep transformer network based anomaly detection and diagnosis model (TranAD) [31] uses attention- based sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data and uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability.

LSTM-VAE [32] uses LSTM to replace the network of encoder and decoder in VAE and obtains a model suitable for time series. One step further, SISVAE [33] replaces the LSTM with a more efficient GRU network and adds a smoothing regularization to make the model more robust to accept partial anomalies in the input. Most recently, [10] proposed a stochastic recurrent neural network for multivariate time series anomaly detection (OmniAnomaly), which models the time dependence and randomness of data through stochastic variable connection and Planar NF. However, these methods all focus on improving the ability to reconstruct the data distribution and do not attempt to design a model suitable for anomaly detection, and also suffer from the posterior collapse problem that VAE-based models all have. Although Ref. [19] starts to focus on this problem by using two adversarial components to optimize the model, its two-stage training process and complex loss function limit its wide applications, and it lacks consideration of time dependence.

Based on above descriptions, ACVAE is more suitable for time series anomaly detection, which

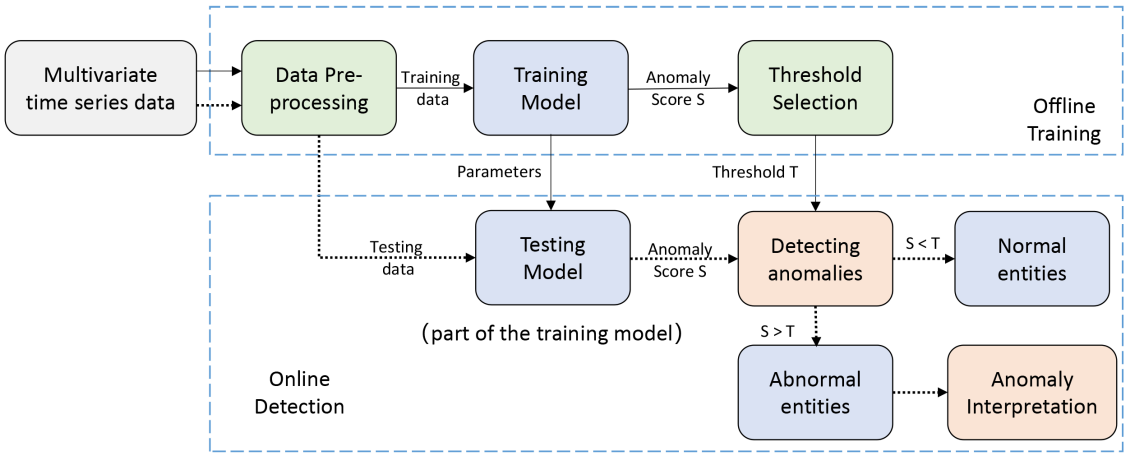


Figure 2: Framework of ACVAE.

not only focuses on time dependence but also tries to solve posterior collapse.

3. Preliminary

Before the discussions, we firstly introduce some notations that will be used in the following, $\boldsymbol{x}$ denotes a multivariate time variable in this paper, $\boldsymbol{x}_t$ represents the t-th time series of $\boldsymbol{x}$, and x_{it} is the entity of $\boldsymbol{x}$.

In this section, we will make a detailed statement on the task of multivariate time series anomaly detection and review some basic knowledge to facilitate the introduction of our method.

3.1. Problem Statement

A multivariate time series contains a series of time points:

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathrm{R}^{M \times N}, \tag{1}$$

where N is the length of $\boldsymbol{x}$, and the observation $\boldsymbol{x}_t = \left[x_t^1, x_t^2, \ldots, x_t^M\right] \in \mathrm{R}^M$ is an M-dimensional vector at a certain moment t $(t = 1, 2, \ldots, N)$ [8]. Obviously, univariate time series is a special case of multivariate one with $M = 1$. Therefore, we focus on anomaly detection and interpretation of multivariate time series, and that of univariate time series can be obtained in a similar way. Anomaly detection and interpretation are defined as follows:

- **Anomaly detection:** For the current entity $\boldsymbol{x}_t$, mark it as abnormal if its abnormal score is higher than the threshold.
- **Anomaly interpretation:** If an entity $\boldsymbol{x}_t$ is classified to be abnormal, it needs to further determine the top-k dimensions that cause the anomaly.

For time series modeling, historical data helps to better understand current data. Therefore, it is a better choice to additionally use historical observation data for anomaly detection [7, 8]. To model this kind of time dependency, we consider using sliding windows to process the data [34]. Given a current entity $\boldsymbol{x}_t$, the corresponding window is defined as $\boldsymbol{W}_{x_t}$:

$$\boldsymbol{W}_{x_t} = \{\boldsymbol{x}_{t-T+1}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t\}, \tag{2}$$

which represents the multivariate time series segment from $t-T+1$ to t, and T is the length of $\boldsymbol{W}_{x_t}$. As shown in Fig. 2, after the pre-processing module, $\boldsymbol{W}_{x_t}$ is used as input to calculate the anomaly score S_{x_t} of $\boldsymbol{x}_t$, and then S_{x_t} is employed to compare with a threshold to determine whether the entity $\boldsymbol{x}_t$ is abnormal. If $\boldsymbol{x}_t$ is classified to be abnormal, the main cause of the anomaly will be selected based on the ranking of the reconstruction error in each dimension.

3.2. Contrast learning

Contrast learning is a self-supervised learning method, which is characterized by not requiring manual labeling of category label information, and directly using the data itself as supervision information to learn the feature expression of sample data and use it for downstream tasks. The core idea is: It is not necessary to pay attention to every detail of the sample, as long as the learned characteristics can distinguish it from other samples [35],[36]. We can summarize its general paradigm:

$$\operatorname{score}\left(f(x), f\left(x^{+}\right)\right) \gg \operatorname{score}\left(f(x), f\left(x^{-}\right)\right), \tag{3}$$

where x^+ is a positive sample similar to x, x^- is a negative sample dissimilar to x, and score is a metric function to measure the similarity between samples. The meaning is that for any input x, the goal of contrast learning is to learn an encoder f to make x more similar to positive samples and different from negative samples.

3.3. Basics of Conv1d, VAE and Adversarial Mechanism

RNN [37] and its variants LSTM, GRU [38] are usually employed to process time series and capture the corresponding time dependence. However, it suffers from the problems of large training time consumption and complex training process. We therefore consider the use of one-dimensional convolution[9], which implies that the convolution kernels move in only one direction, and their weights are optimized for all data samples, which allows the network to understand the temporal correlation between all indicators [39].

Variational Auto-Encoder (VAE) is the product of combining the probabilistic graphical model of Bayesians and deep learning, which is also called the deep Bayesian model [40]. Suppose there is a dataset $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, VAE maps $\boldsymbol{x}$ to the latent representation $\boldsymbol{z}$ through dimension reduction, and then reconstructs $\boldsymbol{x}$ through $\boldsymbol{z}$. $p_\theta(\boldsymbol{z})$ is the prior distribution of the latent representation $\boldsymbol{z}$, and $\boldsymbol{x}$ will be sampled from the conditional distribution $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$, where θ is generative model parameters. In order to better calculate this conditional distribution, VAE introduces an inference network $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, where ϕ is inference model parameters.

Stochastic Gradient Variational Bayes (SGVB) [38] is a variational inference algorithm used to train the parameters θ and ϕ of VAE, which maximizes evidence of lower bound (ELBO) by representing the random variable $\boldsymbol{z}$ as a deterministic variable. ELBO is written below:

$$L(\theta, \phi; \boldsymbol{x}) = -D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right) + E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right], \tag{4}$$

where the D_{KL} in the first term represents the Kullback–Leibler divergence, which measures the distance between two distributions, and the prior distribution is generally chosen as the standard normal distribution $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; 0, 1)$.The above can be abbreviated as $\mathcal{L}_{KLD}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. The second term is the negative log-likelihood, or also known as the reconstruction loss, where the expectation can be calculated using Monte Carlo integration [41].

The most classic example of the adversarial mechanism is the adversarial generative network GAN [42], which uses a generator G to generate similar images, and then applys a discriminator D to distinguish them. In this way, a "game relationship" is formed between G and D, and we will

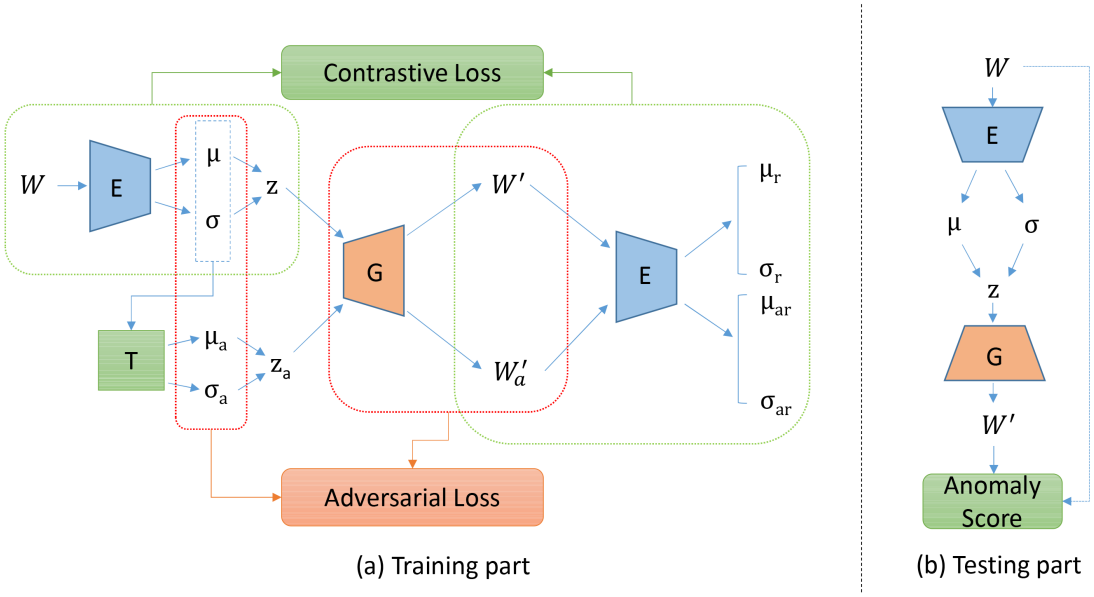


Figure 3: The network structure of ACVAE includes training part and testing part. (a) The training model consists of an encoder E, a decoder G, and a transformation network T. The red box indicates the adversarial part and the green one represents the contrastive part. (b) Encoder E and decoder G are used in the testing part to calculate the anomaly scores.

get an excellent generative model until the two of them reach a Nash equilibrium. The applications of this adversarial mechanism can also be seen on various VAE variant models [9, 43].

4. Proposed Model

In this section, we will introduce a novel time series anomaly detection method by integrating the thoughts of VAE and contrast learning, called Self-adversarial Variational Auto-Encoder Combined with Contrast Learning for Time Series Anomaly Detection (ACVAE). This section is starting with the network structure of the model, followed by the training part and the detection part, and finally we will introduce the automatic threshold selection and the anomaly inference.

4.1. Network Architecture

To design a VAE model that is suitable for anomaly detection, we adopt the Gaussian anomaly prior assumption [19] and use conv1d as the backbone to capture the complex relationships among multivariate time series. As shown in Fig. 4, the training part is composed of encoding network E, generating network G and transformation network T. δ, ϕ and θ are the network parameters of T, E and G, respectively. The model is more suitable for time series anomaly detection by integrating the contrastive loss and adversarial loss. The following we will introduce the model architecture in detailed based on these two parts.

4.1.1. Adversarial part

As discussed above, an overpowered decoder will ignore information from the latent representation $\boldsymbol{z}$ and thus reconstruct the input independently of $\boldsymbol{z}$., which is known as posterior collapse [16]. If the decoder is not restricted, the model will over-fit the distribution of normal data, resulting in some undesirable consequences [19]. To solve this problem, we obtain an efficient regularization

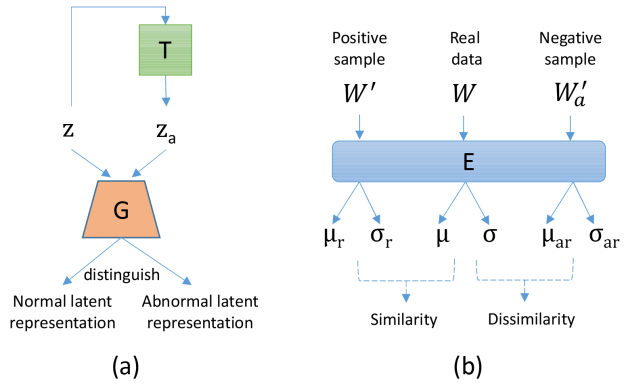


Figure 4: Adversarial mechanism and contrast learning of ACVAE. (a) $\boldsymbol{z}$ and $\boldsymbol{z}_a$ are normal and abnormal latent representations, respectively, and G is used to distinguish them. (b) Similarity: Positive samples and inputs, Dissimilarity: Negative samples and inputs. (Normal reconstruction is used as a positive sample and abnormal reconstruction is used as a negative sample.)

by introducing a confrontation between T and G. As shown in Fig. 3 (a), the input W is passed through an encoder E consisting of a multilayer convolutional network to compute the mean and variance of the latent representation $\boldsymbol{z}$, followed by a reparametrization tricks to obtain $\boldsymbol{z}$,

$$[\boldsymbol{\mu}, \boldsymbol{\sigma}] = E_{\phi}(\boldsymbol{W}), \tag{5}$$

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \text{ and } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6}$$

Transformation network T is established based on Gaussian abnormal prior assumption, whose purpose is to generate an abnormal latent representation $\boldsymbol{z}_a$ similar to the normal potential representation $\boldsymbol{z}$. It takes the normal Gaussian latent representation $\boldsymbol{z}$ encoded by normal data as input, and transforms $\boldsymbol{z}$ into a Gaussian abnormal latent representation $\boldsymbol{z}_a$ with different mean and variance,

$$\{\boldsymbol{z}_a; \boldsymbol{\mu}_a, \boldsymbol{\sigma}_a\} = T_{\delta}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}). \tag{7}$$

G tries to decode the two similar latent representations $\boldsymbol{z}$ and $\boldsymbol{z}_a$ into two different samples $\boldsymbol{W}^{'}$ and $\boldsymbol{W}^{'}_a$, respectively,

$$\boldsymbol{W}' = G_{\theta}(\boldsymbol{z}) \text{ and } \boldsymbol{W}'_a = G_{\theta}(\boldsymbol{z}_a). \tag{8}$$

As shown in Fig. 4 (a), T and G forms a kind of adversarial mechanism that can be used to generate and distinguish the latent representations $\boldsymbol{z}$ and $\boldsymbol{z}_a$.

Given a window $\boldsymbol{W}_{x_t} = \{\boldsymbol{x}_{t-T+1}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t\}$, the objective function of this adversarial process can be described as $\mathcal{L}_D$,

$$\mathcal{L}_D = \mathcal{L}_{D_T} + \mathcal{L}_{D_G}, \tag{9}$$

where

$$\begin{aligned} \mathcal{L}_{D_T} &= D_{KL}\left(\mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2\right)\right) \\ &= \int \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \log \frac{\mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right)}{\mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2\right)} \mathrm{d}z \\ &= \log \frac{\boldsymbol{\sigma}_a}{\boldsymbol{\sigma}} + \frac{\boldsymbol{\sigma}^2 + (\boldsymbol{\mu} - \boldsymbol{\mu}_a)^2}{2\boldsymbol{\sigma}_a^2} - \frac{1}{2}, \end{aligned} \tag{10}$$

and

$$\begin{aligned}\mathcal{L}_{D_G} &= \left[m_x - \mathcal{L}_{MSE}\left(G_\theta(\boldsymbol{z}), G_\theta\left(\boldsymbol{z}_a\right)\right)\right]^+ \\ &= \left[m_x - L_{MSE}\left(\boldsymbol{W}', \boldsymbol{W}'_a\right)\right]^+,\end{aligned} \tag{11}$$

where $[\cdot]^+ = \max(0, \cdot)$, m_x is a positive margin of the MSE target, its purpose is to ensure the value of the target is lower than m_x. $\mathcal{L}_{D_T}$ is the goal of the transformation network T and $\mathcal{L}_{D_G}$ is the target of generating network G, which denote two adversarial objectives, T tries to generate similar abnormal latent representation $\boldsymbol{z}_a$ to confuse G, while G is used to distinguish them as much as possible. After multiple iterations, T and G will eventually reach a balance. Thus, we obtain a constrained decoder that achieves a certain degree of discriminative ability under the action of adversarial mechanisms and can alleviate posterior collapse. This decoder is more suitable for anomaly detection tasks.

4.1.2. Contrastive part

The generative model VAE can model the distribution of the training data and generate the data similar to but different from themselves [44, 45]. Thus, we can get the positive and negative samples by combining the previous adversarial part, and then introduce contrast learning to analyze their relationship.

We add a re-encoder to the model structure, which is exactly the same as the encoder and shares parameters. Normal reconstruction $\boldsymbol{W}'$ and abnormal reconstruction $\boldsymbol{W}'_a$ pass this re-encoder to get their respective mean and variance,

$$\left[\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r\right] = E_\phi\left(\boldsymbol{W}'\right) \text{ and } \left[\boldsymbol{\mu}_{ar}, \boldsymbol{\sigma}_{ar}\right] = E_\phi\left(\boldsymbol{W}'_a\right). \tag{12}$$

As shown in Fig. 4 (b), we take the abnormal reconstruction $\boldsymbol{W}'_a$ as a negative sample and the normal reconstruction $\boldsymbol{W}'$ as a positive sample. Then we try to make the positive samples similar to the original inputs and the negative samples different from the original inputs. The objective function of the contrast process can be described as $\mathcal{L}_E$,

$$\mathcal{L}_E = D_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r^2\right)\right) + \left[m_z - D_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{\mu}_{ar}, \boldsymbol{\sigma}_{ar}^2\right)\right)\right]^+, \tag{13}$$

where D_{KL} denotes the Kullback-Leibler divergence, and $[\cdot]^+ = \max(0, \cdot)$, m_z is a positive margin, its purpose is same to that of m_x.

In this part, we try to add more data for training, not just use the available normal data to train the model. Although negative samples are not necessarily true anomalies, as long as the encoder can identify them, the introduction of contrast learning is useful to the model. It means that this training method will give the model the ability to distinguish abnormal inputs.

4.2. Offline Training

4.2.1. Data Preprocessing

Data pre-processing is an important step before training, which mainly includes data cleaning and data normalization.

Data cleaning: The Spectral Residual (SR) was first applied to significance testing tasks in computer vision [46], and it also performs well in univariate time series anomaly detection tasks [47]. To apply it to multivariate time series, we apply SR to each dimension of the series.

Data normalization: Let $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathrm{R}^{M \times N}$ be a multivariate time series with N time points and each of them with M dimensions. We use $\langle \boldsymbol{x}, \boldsymbol{c} \rangle$ to denote time series $\boldsymbol{x}$, where

Table 1: Anomaly detection framework based on reconstruction, $G(\cdot)$ represents the reconstruction model, and $R(G)$ is the regularizer

Method	$G(\cdot)$	$R\left(G\right)$
AE	$G_D(G_E(\boldsymbol{x}))$	/
VAE	$G_D(G_E(\boldsymbol{x}))$	$-D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right)$
SISVAE	$G_D(G_E(\boldsymbol{x}))$	$-D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right)+$ smoothness regularization
adVAE	$G_D(G_E(\boldsymbol{x}))$	$-D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right)+$ adversarial regularization
ACVAE	$G_D(G_E(\boldsymbol{x}))$	$-D_{KL}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right)+$ $\mathcal{L}_E + \mathcal{L}_D$

$\boldsymbol{c}$ denotes the corresponding features of $\boldsymbol{x}$. After data cleaning, we use min-max normalization to normalize $\langle \boldsymbol{x}, \boldsymbol{c} \rangle$,

$$\tilde{\boldsymbol{c}} = \frac{\boldsymbol{c} - \min(\boldsymbol{c})}{\max(\boldsymbol{c}) - \min(\boldsymbol{c})}, \tag{14}$$

where $\boldsymbol{c}$ is a dimension of a multivariate time series $\boldsymbol{x}$, and $\max\left(\boldsymbol{c}\right)$ and $\min\left(\boldsymbol{c}\right)$ are the maximum and minimum in that dimension, respectively.

4.2.2. Objective function of training

Anomaly detection framework based on reconstruction usually with a similar objective function,

$$L = \|\boldsymbol{x} - G(\boldsymbol{x})\|_2 + R(G), \tag{15}$$

where $G(\cdot)$ represents the reconstruction model, and $R(G)$ is the regularization term. As we can see from Table. 1, some of the latest reconstruction methods follow this general framework, and our method adds two additional regularizations on this basis: adversarial loss $\mathcal{L}_D$ and contrast loss $\mathcal{L}_E$. The objective function $\mathcal{L}_{ACVAE}$ is given as follows:

$$\mathcal{L}_{ACVAE} = \mathcal{L}_{VAE} + \frac{1}{\lambda}\mathcal{L}_D + \left(1 - \frac{1}{\lambda}\right)\mathcal{L}_E, \tag{16}$$

where

$$\mathcal{L}_{VAE} = \mathcal{L}_{MSE}\left(\boldsymbol{W}, \boldsymbol{W}'\right) + \mathcal{L}_{KLD}(\boldsymbol{\mu}, \boldsymbol{\sigma}), \tag{17}$$

$$\begin{aligned}\mathcal{L}_D =& D_{KL}\left(\mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2\right)\right) + \\ & \left[m_x - \mathcal{L}_{MSE}\left(\boldsymbol{W}', \boldsymbol{W}'_a\right)\right]^+,\end{aligned} \tag{18}$$

$$\begin{aligned}\mathcal{L}_E =& D_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r^2\right)\right) + \\ & \left[m_z - D_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{\mu}_{ar}, \boldsymbol{\sigma}_{ar}^2\right)\right)\right]^+.\end{aligned} \tag{19}$$

Based on above descriptions, $\mathcal{L}_{VAE}$ is the loss of VAE, and can be regarded as the base loss. $\mathcal{L}_D$ is the adversarial loss mentioned in the previous section, which will help decoder G gain the ability to distinguish the latent distribution of anomalies and to avoid the posterior collapse. $\mathcal{L}_E$ is the contrast loss, which will help ACVAE to become a more suitable model for anomaly detection. By adding training data, it enables encoder E to obtain the function of a discriminator to distinguish abnormal inputs. λ is a hyperparameter to measure the proportion of each regularizer in the loss,

it can be set as the value of the current training epoch in this paper to emphasize the model to concentrate more on generating abnormal latent representation and abnormal reconstruction in the early stages of training. When the model can generate reasonable positive and negative samples, contrast learning will be applied into the model to strengthen the encoding ability of encoder E. ACVAE with complex network structure can be well trained by designing such a loss function. The learning process is shown in Algorithm. 1.

Algorithm 1 Training ACVAE Model

Input: Normal sequence of windows $\boldsymbol{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T\}$.
Parameter: $\boldsymbol{\phi}_E$, $\boldsymbol{\theta}_G$, $\boldsymbol{\delta}_T \leftarrow$ Initialize network parameters.
Output: An encoder E and decoder G.

while not converged **do**
  $\boldsymbol{w} \leftarrow$ Random mini-batch from $\boldsymbol{W}$.
  $[\boldsymbol{\mu}\ ,\ \boldsymbol{\sigma}] = E_\phi(\boldsymbol{w})$ //encoding
  $[\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a] = T_\delta(\boldsymbol{\mu}, \boldsymbol{\sigma})$ //transforming
  $\boldsymbol{z} \leftarrow$ Samples from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$
  $\boldsymbol{z}_a \leftarrow$ Samples from $\mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a^2)$
  $\boldsymbol{w}' = G_\theta(\boldsymbol{z})$ and $\boldsymbol{w}'_a = G_\theta(\boldsymbol{z}_a)$ //decoding
  $[\boldsymbol{\mu}_r\ ,\ \boldsymbol{\sigma}_r] = E_\phi(\boldsymbol{w}')$ //re-encoding
  $[\boldsymbol{\mu}_{ar}\ , \boldsymbol{\sigma}_{ar}] = E_\phi(\boldsymbol{w}'_a)$ //re-encoding
  Calculating $\mathcal{L}_{ACVAE}$
  Update parameters $\boldsymbol{\phi}$, $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$
end while

4.3. Online Detection

As shown in Fig. 3 (b), the test model only uses the trained encoder E and decoder G. Its overall structure is the same as that of a conventional VAE. Since historical observations help to understand the current entity, we will use window $\boldsymbol{W}$ as the input to the model. This means that we will use $\boldsymbol{x}_t$ and its previous $T-1$ observations to compute the anomaly score of $\boldsymbol{x}_t$. Given a sequence $\boldsymbol{W}_{x_t} = \{\boldsymbol{x}_{t-T+1}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t\}$ as input, the encoder calculates the mean and variance of the latent representation $\boldsymbol{z}$ as output. After sampling to get $\boldsymbol{z}$, it is sent to the decoder to get the reconstruction of the input sequence. Then we will calculate the anomaly score of $\boldsymbol{x}_t$.

The error between the input and reconstruction reflects the difference between the test data and the normal data patterns learned during the model training period. The anomaly score is defined as follows [19]:

$$S_{x_t} = \boldsymbol{e}_{\mathrm{t}} \boldsymbol{e}_{\mathrm{t}}{}^T = (\boldsymbol{x}_t - \boldsymbol{x}'_t)(\boldsymbol{x}_t - \boldsymbol{x}'_t)^T, \tag{20}$$

where $\boldsymbol{e}_t \in \mathrm{R}^M$ is the error vector and $\boldsymbol{x}'_t$ is the reconstruction of $\boldsymbol{x}_t$.

A higher anomaly score means that $\boldsymbol{x}_t$ is harder to reconstruct, which demonstrate the more it deviates from the normal data pattern, the more likely we considerate it is to be abnormal. A threshold is usually used to measure the level of abnormal scores, and in the next section we describe how to automatically select the appropriate threshold.

The process of calculating anomaly scores and detecting anomalies is shown in Algorithm. 2.

4.4. Automatic Threshold Selection

There are various ways to select the thresholds. In some early works [48, 49], the importance of thresholds selection was ignored, because reconstruction error is an abstract concept, and it

Algorithm 2 Calculating Anomaly Scores and Detecting Anomalies

Input: sequence of windows $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T\}$ trained model $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, threshold S_a.
Output: Anomaly score $\boldsymbol{s} = \{S_1, \ldots, S_T\}$ and Labels $\boldsymbol{y} = \{y_1, \ldots, y_T\}$.

for $t = 1$ to T **do**
  $[\boldsymbol{\mu}_t,\ \boldsymbol{\sigma}_t] = E_\phi(\boldsymbol{w}_t)$ //encoding
  $\boldsymbol{z}_t \leftarrow$ Samples from $\mathcal{N}(\boldsymbol{\mu}_t, {\boldsymbol{\sigma}_t}^2)$
  $\boldsymbol{w}_t{}' = G_\theta(\boldsymbol{z}_t)$ //decoding
  $S_t = (\boldsymbol{x}_t - \boldsymbol{x}_t')(\boldsymbol{x}_t - \boldsymbol{x}_t')^T$ // calculating scores
  if $S_t \geq S_a$ **then**
    $y_t = 1$
  else
    $y_t = 0$
  end if
end for

is difficult to determine its thresholds. Some studies used cross-validation methods to select the thresholds [32, 50], but in fact it is very difficult to construct a large enough validation set. There are other methods to select the optimal threshold based on the evaluation metrics [51], but this is impractical and difficult to apply in real-life situations. Therefore, it is particularly important to design a module to select the thresholds automatically.

We employ kernel density estimation (KDE) technique [52] to select decision threshold, which is usually used to estimate the unknown density function with the advantage of not requiring assumptions on the data distribution.

As shown in Fig. 2, during offline training, we use the training data to calculate a series of anomaly scores $\boldsymbol{s} = \{S_1, \ldots, S_T\}$. KDE estimates its probability density function (PDF) $p(s)$ as

$$p(s) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{s - s_i}{h}\right), \tag{21}$$

where N is the size of the training anomaly score set, h is the bandwidth, s_i is the anomaly score of the training data, $K(\cdot)$ is the kernel function, and the Gaussian kernel function is selected in the KDE model in this paper.

After obtaining the PDF of the training anomaly score, its cumulative distribution function (CDF) $F(s)$ can be calculated by the following equation:

$$F(s) = \int_{-\infty}^{s} p(s)\mathrm{d}s. \tag{22}$$

Given a significance level α and CDF, we can find a suitable threshold S_a by the following method,

$$F(S_a) = 1 - \alpha, \quad \alpha \in [0, 1], \tag{23}$$

which means that the sample with anomaly score $S \geq S_a$ has at least $1 - \alpha$ probability of being abnormal. Because KDE decides the threshold by estimating the interval of normal data's anomaly scores, it is clear that using KDE to decide the threshold is more objective and reasonable than simply relying on human experience. A higher significance level α leads to a lower missing alarm rate, which means that models have fewer chances to mislabel outliers as normal data. On the contrary, a lower α means a lower false alarm rate. Therefore, the choice of the significance level

α is a tradeoff. There is only one parameter α that needs to be tuned in the technique of KDE. The specific settings will be described in the experimental section.

4.5. Anomaly Interpretation

Once an entity $\boldsymbol{x}_t$ is identified as to be abnormal, we will analyze the main cause of the anomaly based on its anomaly score on each dimension. Therefore, we need to get the anomaly score for each dimension, and the anomaly score of an entity can be expressed as

$$\begin{aligned} S_{x_t} &= \left(\boldsymbol{x}_t - \boldsymbol{x}'_t\right)\left(\boldsymbol{x}_t - \boldsymbol{x}'_t\right)^T \\ &= \sum_{j=1}^{M} \left(\boldsymbol{x}_t^j - \boldsymbol{x}_t'^j\right)\left(\boldsymbol{x}_t^j - \boldsymbol{x}_t'^j\right)^T \\ &= \sum_{j=1}^{M} S_{x_t}^j, \end{aligned} \tag{24}$$

where M is the number of dimensions, and $S_{x_t}^j$ is the anomaly score of j-th dimension of the entity $\boldsymbol{x}_t$.

The anomaly scores of each dimension are sorted in descending order, and we consider the top-k dimensions with larger scores as the main anomaly causes.

5. Experiments

In this section, we first introduce the dataset and evaluation metrics, and then we conduct several experiments to demonstrate the effectiveness of our model.

5.1. Protocols and Settings

5.1.1. Public Datasets

Our experiments use six real-world datasets: SMD (Server Machine Dataset) [10] is a five-week-long dataset collected and published by a large Internet company. SMAP (Soil Moisture Active Passive satellite) and MSL (Mars Science Laboratory rover) [8] is a public data from NASA and its label is marked by experts. SKAB (Skoltech Anomaly Benchmark) [53] is a public dataset collected by IIot testbed system which is located in Skoltech and published on GitHub. AQD (Air Quality Dataset) is a 36-month-long air quality dataset for a city in China. The data sources come from monitoring stations in six regions including six major pollutants. We divide AQD into two sets of equal size: the first half is used as the test set and the second half as the training set. UCR is a dataset of multiple univariate time series that was used in KDD 2021 cup [54, 55]. We include only the datasets obtained from natural sources (the InternalBleeding datasets) and ignore the synthetic sequences. Table . 2 shows the details of these datasets.

5.1.2. Evaluation Metrics

Anomaly detection is similar to the two-class classification problem, thus we chose Precision (P), Recall (R) and F1 score($F1$) to evaluate the model's performance:

$$\begin{aligned} \text{Precision } &= \frac{TP}{TP+FP}, \text{ Recall } = \frac{TP}{TP+FN}, \\ F1 \text{ score} &= 2 \cdot \frac{\text{Precision } \cdot \text{ Recall}}{\text{Precision } + \text{ Recall}}, \end{aligned} \tag{25}$$

Table 2: Statistics of the datasets

Dataset name	Subset number	Dimension number	Training set size	Testing set size	Anomaly ratio(%)
MSL	27	55	58317	73729	10.72
SMAP	55	25	135183	427617	13.13
SKAB	21	8	9401	22474	34.82
SMD	28	38	708405	708420	4.16
AQD	6	6	70200	70200	11.13
UCR	13	1	31700	65796	2.29

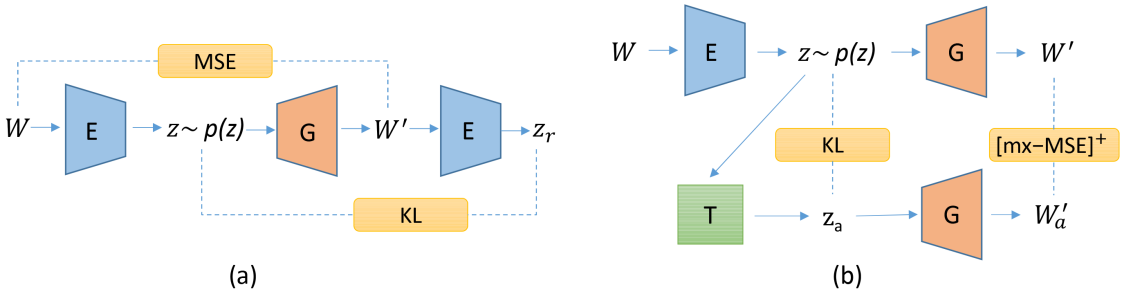


Figure 5: The architecture of the ablation model. (a) ACVAE-E, the model only adds regularization items to Encoder. (b) ACVAE-D, the model only adds regularization items to Decoder.

where TP is true positive and FP is false positive, and FN is false negative. To evaluate the "global" performance of the model, we use the results of multiple tests to calculate the evaluation metrics:

$$P = \frac{1}{N}\sum_{i=1}^{N} P_i, R = \frac{1}{N}\sum_{i=1}^{N} R_i, F1 = 2 \cdot \frac{P \cdot R}{P + R}, \tag{26}$$

where N is the number of tests, we set N to 10 in this paper, P_i and R_i denotes the Precision and Recall of i-th ($i = 1, 2, \ldots, N$) test, respectively. It is worth mentioning that in practical applications, anomaly observations often occur continuously and form anomaly segments. It is acceptable if an alarm can be triggered within the anomaly segment. Therefore [49] proposes a point adjustment method to calculate the performance. If the algorithm can detect any point in the anomaly segment, we consider that the segment is correctly detected and all observations in the segment are correctly detected as anomalies. The observations outside the anomaly segment are treated as usual. Many subsequent works [7],[9],[10], including this article, will also use point adjustment methods to calculate evaluation metrics.

5.1.3. Compared Methods

To verify the effectiveness of the model, we compare ACVAE with five recent deep learning methods:

1)**VAE** [56]. A classic unsupervised generative model that marks the observations which cannot be well reconstructed as anomalies by learning the "normal mode" of the data.

Table 3: parameter settings

Parameter	Default value
batch size	50
sequence length	128
epoch	20
optimizer	Adam
momentums of Adam	0.9 (β1), 0.999 (β2)
learning rate	0.0003

2)**DAGMM** [28]. A prediction-based model that combines deep autoencoding (AE) and Gaussian mixture model (GMM), consisting of a compression network and an estimation network.

3)**LSTM-NDT** [57]. A prediction-based model that uses Long Short-Term Memory (LSTMs) as the backbone network for prediction and treats the deviation from the predicted value as an abnormal value.

4)**OmniAnomly** [10]. A reconstruction-based model that models the time dependence and randomness of data through stochastic variable connection and Planar NF.

5)**AdVAE** [19]. A reconstruction-based model that reconstructs the input data by proposing two adversarial processes.

6)**SISVAE** [33]. A model based on reconstruction, by proposing smoothness-inducing prior to better reconstruct the input data.

7)**TranAD** [31]. A reconstruction-based model that detects anomalies quickly by incorporating attention mechanisms and using adversarial training.

There are also three traditional methods:

1)**IForest** [24]. An ensemble-learning-based approach for finding points that are sparsely distributed and far away from high-density groups.

2)**OCSVM** [22]. A boundary-based approach that attempts to define a boundary around normal class data and determines data that is far away from the boundary as abnormal.

3)**HBOS** [58]. A distance-based approach divides the sample into intervals according to characteristics, and the interval with a small number of samples is the anomaly.

We further compare two variants of our model:

1)**ACVAE-E**. The model ACVAE-E without considering T but with adding a re-encoder is tested to show the effectiveness of the adversarial mechanism in ACVAE.

2)**ACVAE-D**. The model ACVAE-D without considering re-encoder is performed to validate the effectiveness of the contrast learning in ACVAE.

Fig. 5 demonstrates the architectures of two ablation models, they have the same parameters and network structure as ACVAE. The objective functions of ACVAE-E and ACVAE-D are as follows:

$$\mathcal{L}_{ACVAE\text{-}E} = \mathcal{L}_{VAE} + D_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) \| \mathcal{N}\left(\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r^2\right)\right), \tag{27}$$

$$\mathcal{L}_{ACVAE\text{-}D} = \mathcal{L}_{VAE} + \mathcal{L}_D. \tag{28}$$

5.1.4. Implementation Details

For OCSVM, we use the RBF core. Because it has an important parameter *nu* that needs to be tuned, we will select it in $\{0.01, 0.05, 0.1, 0.15, 0.2\}$. The parameter that IForest needs to

Table 4: The size and number of kernel in each layer

Encoder	Decoder
32(4/2/1)	512(4/1/0)
64(4/2/1)	256(4/2/1)
128(4/2/1)	128(4/2/1)
256(4/2/1)	64(4/2/1)
512(4/2/1)	32(4/2/1)
128(4/1/0) (μ) and 128(4/1/0) (σ)	M (4/2/1)

tune is the number of decision trees N selected from $\{50, 100, 200, 300\}$. For HBOS, the number of bins will be selected from $\{5, 10, 20, 30, 50\}$. All three traditional methods mentioned above are implemented on a common detection framework, PyOD [59].

For deep learning methods, the configurations are referred to the official open-source code. It is worth mentioning that the data preprocessing, batch size and sliding window settings (if used) are consistent with this paper. And except for SISVAE, all other methods have modules for automatic threshold selection, and we follow this principle to select the threshold. For SISVAE, it chooses the optimal threshold value as much as possible.

Table 3 shows the parameters used in the experiment. If there is no special description, the default values will be used. Kaiming uniform weight initialization was used for all networks. And ACVAE's network is composed of conv1d, the details of which are shown in Table 4. Give an example to illustrate its meaning: $M(4/2/1)$ means the number of filters is M (Number of dimensions of input data), the size of filter is 4, the stride is 2 and the padding is 1. The transformation network consists of three fully connected layers, each with the same dimensionality, and the mean and variance will be input into the network as a whole after splicing. During the entire model training and testing process, we have three hyperparameters that need to be tuned: m_x, m_z and α. The functions of m_x and m_z are the same and they are only used to prevent the negative term from being too small. It has been verified in [19] that the model is not sensitive to m_x and m_z, thus we follow the previous work and set m_x and m_z to 2 and 20, respectively. The significance level α is set to 0.01 of SMD and MSL, 0.04 of SMAP and SKAB, and 0.1 of AQD.

All the above deep learning methods are implemented on PyTorch [60] 1.6, except for Omni-Anomly which is implemented on TensorFlow 1.12 used by the original authors, and these methods are trained on a single NVIDIA TITAN RTX GPU.

5.2. Results and Discussions

Table 5 shows the results of the model performance comparison. According to the experimental results, it can be seen ACVAE outperforms all baselines on the four public datasets. Meanwhile, ACVAE is better than other baselines in terms of robustness because its precision and recall are higher than 0.73 on all datasets, which cannot be done by other methods.

Overall, the three traditional methods perform poorly with ordinary results on some datasets and bad results on others, which is because these methods are proposed based on the strict assumptions on the data. Once the data is inconsistent with the assumptions of the traditional methods or is high-dimensional, the performance of the methods is impaired. This is corroborated by the performance on the SMD dataset, which is a large dataset consisting of 28 subsets. The diverse data of SMD struggles to meet strict assumptions and lead to the poor results.

However, neural networks generally do not make strict assumptions, but rather an approximation to a function. Therefore, neural network-based methods perform better and more stable in high dimensional big data.

The results of VAE are predictable, as we discussed earlier if no restrictions are placed on the decoder, the model will overfit the distribution of normal data, which precisely means that the FP will become high, thus the precision of VAE is very low. DAGMM doesn't consider the time dependence of the sequence, and its input is only a point instead of a window, which is obviously its weakness. And the latent representation dimension of the model must be set to 1, which limits the model's ability to learn the latent space. The results also demonstrate this, with the best performance on SKAB, which has the smallest amount of data and dimensionality, and poor performance on SMAP. LSTM-NDT is a prediction based model. However, due to external factors, some time series are inherently unpredictable[61], and the prediction of time series may not be accurate, so the prediction based model may not be appropriate. For MSL, various behaviors have different laws, so it is difficult to predict. Some time series in SMD (for example, TCP retransmission) are also unpredictable due to uncontrollable factors (for example, complex and dynamic network environment). Therefore, LSTM-NDT performs poorly on these two datasets.

SISVAE, adVAE, and OmniAnomaly are three methods that have achieved suboptimal performance on different datasets, but they still have some problems. SISVAE is a method that considers using smoothness regularization to constrain the encoder, but it also does not constrain the decoder, and this constraint is likely to have negative effects when a dataset has significant trends and seasonality. For example, the pollutant pm2.5 in AQD has a decreasing trend due to human control and shows a higher value at night and in winter, therefore it is not surprising if you use such a simple smoothing on the AQD dataset and get such results. Although OmniAnomly considers temporal correlation and stochasticity, it mainly focused on reconstruction rather than anomaly detection, such models also include SISVAE and those designed based on VAE. adVAE adds an adversarial mechanism to make the model more suitable for anomaly detection, but its complex training method and failure to consider time dependence make it underperform on complex and diverse time-series datasets. TranAD performs slightly better than ACVAE on the UCR dataset, indicating the advantage of the attention mechanism for the univariate time series anomaly detection problem. To be able to extend the model to univariate time series anomaly detection, the attention mechanism may be a good direction.

In summary, our model ACVAE not only attempts to solve the posterior collapse problem, but also equips both the encoder and decoder with the ability to recognize anomalies by adding two regularizers and introducing contrast learning. We believe that combining contrast learning and generative models is a worthwhile direction to explore and has potential applications in anomaly detection.

5.3. Ablation Study

We designed ablation experiments to facilitate a more detailed analysis of the proposed method. Since the structure of the variant ACVAE is improved based on VAE, we choose VAE as the baseline. And if there are no special instructions, this experiment will be carried out on the SMD dataset.

1) Since two additional regularization terms are added to the objective function of ACVAE, we would like to demonstrate that these two components can enhance the model. Therefore, we propose two variants of ACVAE: ACVAE-E and ACVAE-D (Fig. 5). ACVAE-E only added re-encoder, trying to illustrate the situation without a adversarial mechanism. ACVAE-D only retains the transformation network T to show that there is no contrast learning. According to the results

Table 5: Performance Comparison of All algorithms on five datasets

Methods	MSL			SMAP			SMD			SKAB			AQD			UCR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
IForest	0.5531	0.9654	0.7033	0.5337	0.5907	0.5608	0.1082	0.9992	0.1952	0.3537	1	0.5225	0.3255	0.9684	0.4873	0.6482	0.7845	0.7098
OCSVM	0.5386	0.8999	0.6739	0.4656	0.7631	0.5783	0.2302	0.9570	0.3711	0.3482	1	0.5166	0.1063	1	0.1922	0.5636	0.7213	0.6328
HBOS	0.6734	0.9687	0.7945	0.5963	0.5907	0.5935	0.2572	0.9718	0.4067	0.3861	1	0.5571	0.5328	0.8481	0.6545	0.7363	0.8527	0.7902
VAE	0.3715	0.9997	0.5417	0.4181	1	0.5897	0.1844	0.9897	0.3108	0.7021	1	0.825	0.3681	0.7573	0.4954	0.5473	0.9814	0.7027
DAGMM	0.5935	0.9673	0.7356	0.5666	0.9013	0.6958	0.5719	0.9849	0.7236	0.6214	0.9964	0.7654	0.4563	0.6082	0.5214	0.5143	0.9624	0.6704
LSTM-NDT	0.5843	0.5642	0.5741	0.7246	0.9876	0.8359	0.5684	0.6438	0.6038	0.5862	1	0.7391	0.6285	0.5275	0.5736	0.5492	0.8147	0.6561
OmniAnomly	0.9011	0.8647	0.8825	0.9246	0.5536	0.6925	0.8348	0.9107	0.8711	0.8404	0.9999	0.9133	0.7837	0.6712	**0.7231**	0.7485	0.9425	0.8344
adVAE	0.9116	0.8669	0.8887	0.8018	0.9964	**0.8885**	0.7016	0.8576	0.7718	0.9121	1	0.9541	0.4430	0.6278	0.5195	0.8738	0.9274	0.8998
SISVAE	0.9125	0.8875	0.8998	0.5302	0.7401	0.6178	0.5325	0.6549	0.5874	0.9874	0.9312	**0.9585**	0.1078	0.9464	0.1936	0.7962	0.8512	0.8229
TranAD	0.8823	0.9222	**0.9018**	0.7856	0.9973	0.8789	0.8882	0.9026	**0.8953**	0.8531	1	0.9208	0.5011	0.8226	0.6228	0.8907	1	**0.9422**
ACVAE-E	0.8435	0.3939	0.5370	0.7498	0.5907	0.6608	0.6833	0.8155	0.7436	0.9871	0.6140	0.7571	0.6690	0.7558	0.7097	0.6894	0.9419	0.7961
ACVAE-D	0.8879	0.9163	0.8967	0.8265	0.9350	0.8774	0.8225	0.7171	0.7662	0.9852	0.6495	0.7529	0.3198	0.6750	0.4340	0.8832	0.5029	0.6409
ACVAE	0.9157	0.9081	**0.9119**	0.8280	0.9756	**0.8958**	0.9374	0.9342	**0.9358**	0.9479	0.9813	**0.9643**	0.7341	0.7772	**0.7551**	0.9254	0.9527	**0.9388**

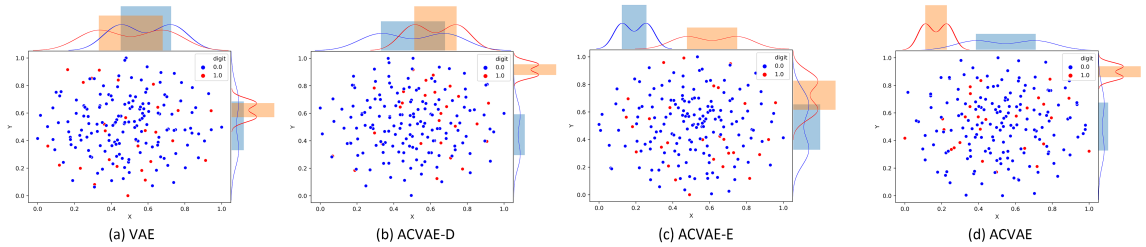


Figure 6: T-SNE visualization of the latent space of the SMD dataset. (Blue dots indicate latent representations of normal data and red dots indicate latent representations of abnormal data. To show the results more visually, the KDE curves are plotted at the top and right of the figure.)

in Table 5, both variants show performance improvements relative to VAE, which illustrates the effectiveness of the two regularizers.

Table 5 demonstrates ACVAE-D performs better than ACVAE-E on most datasets, but it does not indicate that the adversarial mechanism is far better than contrast learning. This is because without the transformation network T, it is difficult for us to find a negative sample, thus ACVAE-E is an incomplete model. ACVAE-E performs better on the AQD dataset because AQD is a complex multi-factor influenced environment dataset with more diverse data patterns, while ACVAE-D with adversarial mechanism is difficult to achieve better results on some subsets, i.e., difficulty in reaching equilibrium. Thus, the above results indicate contrast learning can help the model converge better to some extent.

2) ACVAE is a model more suitable for anomaly detection, and we want encoder to have the ability to identify anomalies as well, thus we designed a potential spatial visualization experiment to demonstrate this improvement, with a data source from a subset of SMD. As shown in Fig. 6, ACVAE has better recognition ability, because it can be clearly seen from the KDE curve that the latent representation $\boldsymbol{z}$ is separated in space, which is more conducive to recognition by the subsequent decoder. Theoretically, we expect ACVAE-E perform better, as mentioned before that ACVAE-E is difficult to find negative samples, it only reflects the effect of positive samples to ACVAE.

3) We design an experiment to verify whether G can independently distinguish the latent

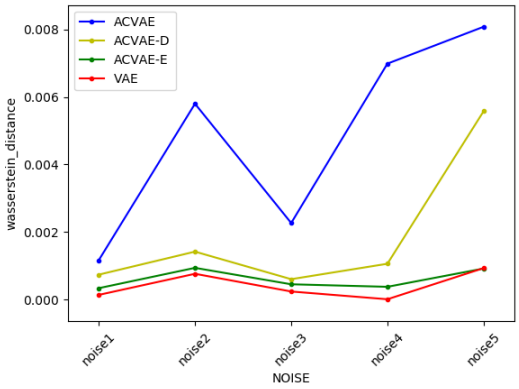


Figure 7: Wasserstein distance of anomaly scores between S and S_a. (We add five types of noises: (1) uniform distribution $\mathcal{U}(0,\ 1)$; (2) Gaussian distribution $\mathcal{N}(0,\ 1)$; (3) Constant value 0.5; (4) Multiply the value of the first half of the dimension by 0.5; (5) The value of the second half of the dimension is set to 0.)

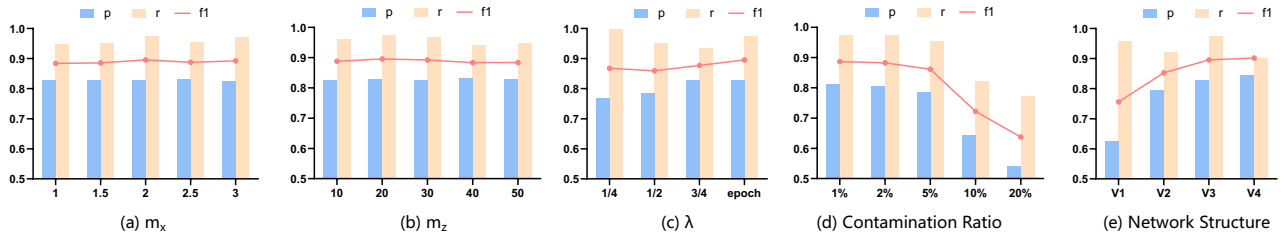


Figure 8: Sensitivity Analysis. (a) Objective function parameter m_x. (b) Objective function parameter m_z. (c) Objective function parameter λ. (d) Training dataset contamination ratio. (e) Convolutional network structure.

representation. First, normal data $\boldsymbol{x}$ from a subset of SMD is used as input to obtain the normal latent representation $\boldsymbol{z}$. Then we synthesize the abnormal latent representation $\boldsymbol{z}_a$ by adding five types of noise, so that we have six latent representations including $\boldsymbol{z}$. Then they are fed into G to obtain the reconstruction and calculate the corresponding abnormal scores S and S_a. S represents the anomaly score of $\boldsymbol{z}$, which is calculated from $G(\boldsymbol{z})$ and $\boldsymbol{x}$. S_a represents the anomaly score of $\boldsymbol{z}_a$, and the difference between it and S will be used to measure the discrimination ability of G. The larger the difference, the better the detection ability. As shown in Fig. 7, the recognition performance of ACVAE is the best, followed by ACVAE-D, and there is basically no difference between ACVAE-E and VAE.

In conclusion, ACVAE is indeed a suitable model for anomaly detection. The ablation study proves that two regularizations we proposed are effective and their combination will get better results.

5.4. Sensitivity Analysis

In this section, we investigated the effects of different parameters and factors on the performance of ACVAE. All experiments were completed using the SMAP dataset.

First, we tested the sensitivity of the model to the parameters m_x and m_z. Fig. 8 (a) summarizes the results about m_x using five different ratios [1,1.5,2,2.5,3]. As for m_z, we will choose between 10 and 50. Fig. 8 (b) shows the results. m_x and m_z have similar roles and are set to prevent

Table 6: Comparison of threshold selection

Evaluation metrics	MSL	SMAP	SMD	SKAB	AQD	UCR
F1 - KDE	0.9119	0.8958	0.9358	0.9643	0.7551	0.9388
F1 - Best	0.9228	0.9091	0.9461	0.9755	0.7692	0.9527

Table 7: The Accuracy HitRate@P% and NDCG@P% of Anomaly Interpretation

Methods	SMD					
	H@100%	H@150%	H@200%	N@100%	N@100%	N@200%
OmniAnomly	0.5054	0.6387	0.7028	0.4541	0.6185	0.6841
adVAE	0.4463	0.5252	0.6372	0.4854	0.5739	0.6928
TranAD	0.4782	**0.6937**	0.7249	**0.6022**	0.5916	0.7397
ACVAE	**0.5412**	0.6742	**0.7617**	0.5624	**0.6887**	**0.7864**

part of the objective functions from becoming too small during the gradient descent. The model performance is not sensitive to m_x and m_z, and the choice of these parameters is not critical. Therefore, for convenience, we fix them to 2 and 20, respectively.

Second, we explored the effect of λ on the model performance. As shown in Fig. 8 (c), the best results were achieved with dynamically varying lambda values. Such a setting allows the model to focus on different purposes at different stages, resulting in better training of the model.

Since ACVAE requires training on normal datasets, it is necessary to study the response of the model to a contaminated training set. As shown in Fig. 8 (d), the model maintains good results at 5% noise, and the model performance decreases when the percentage of noise in the training set reaches 10%. Finally, a significant decrease in model performance can be seen for the high noise case (20%). This is because the VAE-based approach always tries to reduce the reconstruction loss of all training sets and the contaminated dataset also tends to affect the discriminative loss of ACVAE. However, in practice, normal data are easy to collect and it is unrealistic for the training set to show such high contamination. Therefore ACVAE is unlikely to face this situation during training.

Finally, we investigate the effect of convolutional network structure. As shown in Fig. 8 (e), we considered three additional structures, V1(64(2,18,0), 128(8,1,0)), V2(64(4,4,0), 128(4,4,0), 256(4,4,0), 128(2,1,0)) and V4(8(4/2/1), 16(4/2/1), 32(4/2/1), 64(4/2/1), 128(4/2/1), 256(4/2/1), 256(2/1/1), 128(2/1/0)). From the results, it is clear that the model performance does not get better as the number of layers increases. Our currently selected structure V3 performs better, but it does not mean that it is the best choice. Therefore, the network structure tuning of neural networks is a complex process and it is worth exploring.

5.5. Automatic Threshold Selection and Anomaly interpretation

5.5.1. Effect of KDE

Manual adjustment of thresholds in applications is impractical, therefore an effective method of automatic threshold selection is necessary. As seen in Table 6, the $F1$ obtained by KDE is only slightly lower than the optimal $F1$, whose difference is between 0.0103 and 0.0141. Considering the less time consumption, a slight reduction in $F1$ is acceptable.

Table 8: Training Time (S/epoch) of SISVAE, ACVAE and adVAE on five datasets.

Method	SKAB	SMD	MSL	SMAP	AQD	UCR
SISVAE	87.7	307.48	504.25	1,129.63	146.22	36.87
ACVAE	3.39	10.42	21.97	52.11	4.21	1.27
adVAE	4.46	12.27	22.52	55.49	5.59	3.48

5.5.2. Anomaly interpretation

To detect the causes of the anomaly, for the detected anomalies, we rank the anomaly scores of their dimensions in descending order A_{x_t}, and the dimension with a higher score will be considered as the main cause of the anomaly. We will follow [10] and use HitRate@P% to measure the interpretation performance of the anomaly detection algorithm. The metric is defined as follows:

$$\text{HitRate@P\%} = \frac{\text{Hit@ } [P\% \times |G_{x_t}|]}{|G_{x_t}|}, \tag{29}$$

where G_{x_t} is the ground truth array containing labels about the main cause, $|G_{x_t}|$ denotes its length, and P can be set to $[100, 150, 200]$. The following we give an example to illustrate how to calculate this metric. Suppose there is a five-dimensional observation $\boldsymbol{x}_t$, its A_{x_t} is $\{5, 2, 1, 4, 3\}$ and G_{x_t} is $\{5, 1\}$, if we take $P = 100$, then Top$[\lfloor P\% \times |G_{x_t}| \rfloor]$=Top[2]. The first two terms in A_{x_t} are I_{x_t}=$\{5, 2\}$, and the intersection with G_{x_t} is $\{5\}$. Thus, HitRate@100% = 1/2 = 0.5. We also measure the Normalized Discounted Cumulative Gain (NDCG) [9]. NDCG@P% considers the same number of top predicted candidates as HitRate@P%.

Since only SMD provides the ground truth array for anomaly interpretation, this section will only be discussed on the SMD dataset. We cannot choose SISVAE, VAE and DAGMM for comparison. This is because of the poor performance of SISVAE and VAE on SMD and the unavailability of reconstruction errors for each dimension of DAGMM.

As shown in Table 7 where H and N correspond to HitRate and NDCG (with complete data used for model testing), the average interpretation accuracy of all detected anomalies increases as P increases, because more G_{x_t} can be included in I_{x_t}. In most P-selection schemes, the interpretation accuracy of ACVAE is higher than those of others, which reflects the superiority of our model.

5.6. Further Comparison with SISVAE and adVAE

5.6.1. Efficiency

Since we want to obtain an online anomaly detection model, the detection time is also an important metric in addition to accuracy. And the model needs to be trained frequently, then the training time needs to be considered as well. We evaluated the training time for each epoch of the training dataset and the detection time for the test dataset. It should be noted that the SMD and AQD training sets in this section are a subset.

The training time of ACVAE shown in Table 8 is much less than that of SISVAE, which is because the training process of CNN is faster than that of GRU. However, it is also because we chose CNN while adVAE uses a simple fully connected layer, which makes the training time of ACVAE and adVAE unable to pull apart. As shown in Fig. 9, the convergence of ACVAE is better than the other two methods, which shows that ACVAE can be trained more efficiently in a shorter time. In addition, the detection time of ACVAE is much smaller than the other methods (see Fig. 9).

Table 9: Testing Time(S) of SISVAE, ACVAE and adVAE on five datasets.

Method	SKAB	SMD	MSL	SMAP	AQD	UCR
SISVAE	25.92	42.09	63.63	408.62	4.08	14.87
ACVAE	11.93	33.79	52.17	222.55	9.91	5.62
adVAE	35.14	71.93	174.38	575.09	33.81	9.74

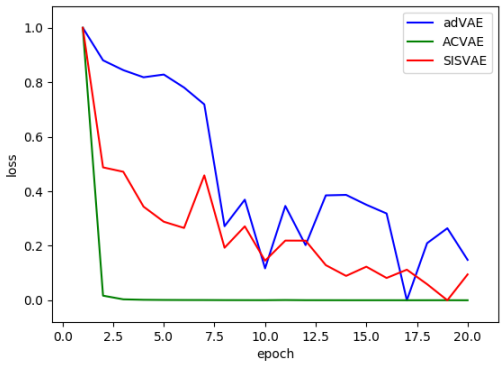


Figure 9: Convergence curve of adVAE, ACVAE and SIVAE. (We normalize the loss curve of each method and draw them together for easy observation.)

Table 10: Mean and standard deviation of experimental results

Dataset	SISVAE	ACVAE	adVAE
SKAB	0.9697±0.0199	0.9642±0.0112	0.9539±0.0153
SMD	0.5425±0.2027	0.9324±0.0404	0.7578±0.1792
MSL	0.8917±0.0200	0.9117±0.0168	0.8883±0.0240
SMAP	0.6177±0.0262	0.8958±0.0016	0.8958±0.0016
ADQ	0.1936±0.0086	0.7551±0.0289	0.5195±0.0216

5.6.2. Stability

The experimental results have been compared well in Table 5, but we found that the F1 differences of the models are not very large on some datasets. To present the results more clearly, we show the mean and variance of the results in Table 10. We can see that ACVAE achieves a better mean result with a small variance. This indicates that our proposed ACVAE has better stability, which makes it better to be applied in a realistic environment.

6. Conclusion

In this paper, we propose a novel self-adversarial variational auto-encoder combined with contrast learning (ACVAE) for time series anomaly detection. With the constraint of two regularizers, the ACVAE is verified to be more suitable for anomaly detection tasks as its high accuracy and efficiency. Specifically, ACVAE introduces an adversarial mechanism that allows the decoder to

be constrained and gain some discriminatory power. At the same time, ACVAE introduces contrast learning, which allows the encoder to obtain more training samples and be able to identify anomalies. In addition, ACVAE provides a method for analyzing anomalies, i.e., identifying the main cause of anomalies based on the scores of each dimension. Extensive experimental results show that ACVAE outperforms state-of-the-art methods on five datasets and can be applied to various application environments, such as environmental monitoring and server detection.

For the future, we propose to explore the prior distribution of the method. Real-world data often present diverse distributions, and it is difficult to state that a normal distribution is the most applicable assumption for anomaly detection. We would also like to further extend the setting of anomaly scores. The traditional VAE-based anomaly score calculation treats each dimension equally, however, such a setting is clearly flawed when a variable is uncorrelated with other variables. In addition, it is a challenge to cope with non-smooth data.

Funding

This work is supported by the National Key R&D Program of China (2019YFB2103000), the State Key Program of National Nature Science Foundation of China (61936001), the Natural Science Foundation of Chongqing (cstc2019jcyj-cxttX0002, cstc2020jcyj-msxmX0737, cstc2021ycjh-bgzxm0013), the Key Cooperation Project of Chongqing Municipal Education Commission (HZ2021008), the Science and Technology Research Program of Chongqing Education Commission of China (KJQN201900638).

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (3) (2009) 1–58.

[2] G. Osada, K. Omote, T. Nishide, Network intrusion detection based on semi-supervised variational auto-encoder, in: European Symposium on Research in Computer Security, Springer, 2017, pp. 344–361.

[3] M. Hauskrecht, I. Batal, M. Valko, S. Visweswaran, G. F. Cooper, G. Clermont, Outlier detection for patient monitoring and alerting, Journal of biomedical informatics 46 (1) (2013) 47–55.

[4] P. Cui, C. Zhan, Y. Yang, Improved nonlinear process monitoring based on ensemble kpca with local structure analysis, Chemical Engineering Research and Design 142 (2019) 355–368.

[5] D. J. Hill, B. S. Minsker, Anomaly detection in streaming environmental sensor data: A data-driven modeling approach, Environmental Modelling & Software 25 (9) (2010) 1014–1022.

[6] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Asian conference on computer vision, Springer, 2018, pp. 622–637.

[7] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.

[8] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 387–395.

[9] X. Chen, L. Deng, F. Huang, C. Zhang, Z. Zhang, Y. Zhao, K. Zheng, Daemon: Unsupervised anomaly detection and interpretation for multivariate time series, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 2225–2230.

[10] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.

[11] Q. Chen, R. Luley, Q. Wu, M. Bishop, R. W. Linderman, Q. Qiu, Anrad: A neuromorphic anomaly detection framework for massive concurrent data streams, IEEE transactions on neural networks and learning systems 29 (5) (2017) 1622–1636.

[12] M. Kerpicci, H. Ozkan, S. S. Kozat, Online anomaly detection with bandwidth optimized hierarchical kernel density estimators, IEEE Transactions on Neural Networks and Learning Systems (2020).

[13] Y. Guo, T. Ji, Q. Wang, L. Yu, G. Min, P. Li, Unsupervised anomaly detection in iot systems for smart cities, IEEE Transactions on Network Science and Engineering 7 (4) (2020) 2231–2242.

[14] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, arXiv preprint arXiv:1901.03407 (2019).

[15] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. v. d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1705–1714.

[16] A. Goyal, A. Sordoni, M.-A. Côté, N. R. Ke, Y. Bengio, Z-forcing: Training stochastic recurrent networks, arXiv preprint arXiv:1711.05411 (2017).

[17] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, P. Abbeel, Variational lossy autoencoder, arXiv preprint arXiv:1611.02731 (2016).

[18] I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, Y. Bengio, A hierarchical latent variable encoder-decoder model for generating dialogues, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[19] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, Y. Yang, advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection, Knowledge-Based Systems 190 (2020) 105187.

[20] M. Gupta, J. Gao, C. C. Aggarwal, J. Han, Outlier detection for temporal data: A survey, IEEE Transactions on Knowledge and data Engineering 26 (9) (2013) 2250–2267.

[21] I. Kiss, B. Genge, P. Haller, G. Sebestyén, Data clustering-based anomaly detection in industrial control systems, in: 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, 2014, pp. 275–281.

[22] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural computation 13 (7) (2001) 1443–1471.

[23] W. A. Chaovalitwongse, Y.-J. Fan, R. C. Sachdeo, On the time series k-nearest neighbor classification of abnormal brain activity, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 37 (6) (2007) 1005–1016.

[24] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (1) (2012) 1–39.

[25] A. A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for iot time-series data: A survey, IEEE Internet of Things Journal 7 (7) (2019) 6481–6494.

[26] B. Barz, E. Rodner, Y. G. Garcia, J. Denzler, Detecting regions of maximal divergence for spatio-temporal anomaly detection, IEEE transactions on pattern analysis and machine intelligence 41 (5) (2018) 1088–1101.

[27] V. T. Trifunov, M. Shadaydeh, B. Barz, J. Denzler, Anomaly attribution of multivariate time series using counterfactual reasoning, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2021, pp. 166–172.

[28] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International conference on learning representations, 2018.

[29] M. Flach, F. Gans, A. Brenning, J. Denzler, M. Reichstein, E. Rodner, S. Bathiany, P. Bodesheim, Y. Guanche, S. Sippel, et al., Multivariate anomaly detection for earth observations: a comparison of algorithms and feature extraction techniques, Earth System Dynamics 8 (3) (2017) 677–696.

[30] E. Dai, J. Chen, Graph-augmented normalizing flows for anomaly detection of multiple time series, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022.

[31] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, Proc. VLDB Endow. 15 (6) (2022) 1201–1214.

[32] D. Park, Y. Hoshi, C. C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robotics and Automation Letters 3 (3) (2018) 1544–1551.

[33] L. Li, J. Yan, H. Wang, Y. Jin, Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder, IEEE transactions on neural networks and learning systems 32 (3) (2020) 1177–1191.

[34] T. J. Sejnowski, C. R. Rosenberg, Parallel networks that learn to pronounce english text, Complex systems 1 (1) (1987) 145–168.

[35] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, N. Saunshi, A theoretical analysis of contrastive unsupervised representation learning, arXiv preprint arXiv:1902.09229 (2019).

[36] J. Jeong, J. Shin, Training gans with stronger augmentations via contrastive discriminator, arXiv preprint arXiv:2103.09742 (2021).

[37] M. Fraccaro, S. K. Sønderby, U. Paquet, O. Winther, Sequential neural models with stochastic layers, arXiv preprint arXiv:1605.07571 (2016).

[38] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[39] R. Assaf, I. Giurgiu, J. Pfefferle, S. Monney, H. Pozidis, A. Schumann, A. Explainable, A. Dhurandhar, K. Shanmugam, et al., An anomaly detection and explainability framework using convolutional autoencoders for data storage systems., in: IJCAI, 2020, pp. 5228–5230.

[40] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[41] P. R. Christian, G. Casella, Monte carlo statistical methods, NY: Springer-Verlag (1999).

[42] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: An overview, IEEE Signal Processing Magazine 35 (1) (2018) 53–65.

[43] W. Chen, H. Xu, Z. Li, D. Pei, J. Chen, H. Qiao, Y. Feng, Z. Wang, Unsupervised anomaly detection for intricate kpis via adversarial training of vae, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 1891–1899.

[44] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).

[45] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[46] X. Hou, L. Zhang, Saliency detection: A spectral residual approach, in: 2007 IEEE Conference on computer vision and pattern recognition, Ieee, 2007, pp. 1–8.

[47] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, Q. Zhang, Time-series anomaly detection service at microsoft, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 3009–3017.

[48] M. Sölch, J. Bayer, M. Ludersdorfer, P. van der Smagt, Variational inference for on-line anomaly detection in high-dimensional time series, arXiv preprint arXiv:1602.07109 (2016).

[49] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al., Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 187–196.

[50] S. Suh, D. H. Chae, H.-G. Kang, S. Choi, Echo-state conditional variational autoencoder for anomaly detection, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 1015–1022.

[51] R. Assaf, I. Giurgiu, J. Pfefferle, S. Monney, H. Pozidis, A. Schumann, A. Explainable, A. Dhurandhar, K. Shanmugam, et al., An anomaly detection and explainability framework using convolutional autoencoders for data storage systems., in: IJCAI, 2020, pp. 5228–5230.

[52] A. Gramacki, J. Gramacki, Fft-based fast bandwidth selector for multivariate kernel density estimation, Computational Statistics & Data Analysis 106 (2017) 27–45.

[53] I. D. Katser, V. O. Kozitsin, Skoltech anomaly benchmark (skab), `https://www.kaggle.com/dsv/1693952` (2020). doi:10.34740/KAGGLE/DSV/1693952.

[54] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, The ucr time series archive, IEEE/CAA Journal of Automatica Sinica 6 (6) (2019) 1293–1305.

[55] F. Rewicki, J. Denzler, J. Niebling, Is it worth it? comparing six deep and classical methods for unsupervised anomaly detection in time series (2023).

[56] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lecture on IE 2 (1) (2015) 1–18.

[57] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding (2018).

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

[59] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, arXiv preprint arXiv:1901.01588 (2019).

[60] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019) 8026–8037.

[61] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection (2016).