Letter

Autonomous Recommendation of Fault Detection Algorithms for Spacecraft

Wenbo Li and Baoling Ning

Dear Editor,

This letter deals with the problem of algorithm recommendation for online fault detection of spacecraft. By transforming the time series data into distributions and introducing a distribution-aware measure, a principal method is designed for quantifying the detectabilities of fault detection algorithms over special datasets. Based on a sublinear time filtering method, an efficient algorithm for evaluating the detectabilities is designed. By combining the above techniques, RecAD is proposed for the recommendation of fault detection algorithms. Experimental results over typical datasets show that RecAD can select the detecting algorithm with better performance efficiently and the cost of the recommendation is rather small.

As a typical kind of autonomous intelligent system, spacecrafts are usually composed of many complex components, and each component is typically equipped with a certain number of sensors which will produce many kinds of telemetry data. Due to working in the extreme environment, spacecrafts tend to be failed or even damaged by the failure of a device or subsystem. To reduce the risk of those failures, a key task of spacecraft operations is anomaly detection that is to discover anomalies in the telemetry data.

There have been many research efforts focusing on anomaly detection over spacecraft telemetry data [1]–[3]. Out-of-limits (OOL) method is the most popular one due to its simplicity, low-cost and understandability [4], [5]. To overcome the limitations of the OOL methods, many data-driven anomaly detection methods have been introduced [6]–[9]. Recently, more and more deep anomaly detection methods are designed [2]. The most typical methods include reconstruction based approaches [10], generation based approaches [11], predication based approaches [4], etc. However, it has been found that no method can outperform others always [12], and a natural and feasible solution is to maintain several detection algorithms meanwhile and select the most proper one to detect anomalies according to the actual situations. Therefore, it is highly needed to study the problem of algorithm recommendation for detecting spacecraft anomalies.

Two principal challenges are identified.

1) The first one is the lack of labels and universal objective functions. Due to the limited computational resources of spacecrafts and the scarcity of anomalies, it is hardly possible to have access to any labels when online anomaly detection is processing, algorithm recommendation methods must work in an nearly total unsupervised way. Even worse, there does not exist a universal objective function that could guide algorithm recommendation.

2) The second one is the limited computation resources of online anomaly detection for spacecrafts. Because of the extreme working environments of spacecrafts, to enlarge the lifetime of spacecrafts as much as possible, only recommendation algorithms with enough high efficiency are allowed to be deployed.

In this letter, rising to the above challenges, using the ideas of measuring detectabilities by distributions and distinguishing distributions by sampling, an efficient automated algorithm recommendation method for detecting spacecraft anomalies is proposed. The main contributions include: a formal definition of the fault detection algorithm recommendation problem, a Kullback-Leibler (KL)-divergence based method for measuring the detectability of algorithms, a sublinear algorithm for efficiently estimating the measures and selecting the recommended detection algorithms, and a detailed experimental results to verify the effectiveness of the proposed method.

Notations and problem description: The telemetry data of a spacecraft is usually represented by a time series $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ where each $\mathbf{x}_t \in \mathbb{R}^m$ $(t \in [1,n])$ is an m-dimensional vector corresponding to the data on each dimension. In the following parts, for the sake of simplicity, the proposed method will be explained by assuming that the telemetry data has only one dimension, and we can denote X as $\{x_1, x_2, \dots, x_n\}$. It is not hard to verify that our method can be extended to the general cases trivially.

The goal of anomaly detection is to determine whether an observation x_t is an anomaly or not. Let $\mathcal{A} = \{A_1, A_2, \dots, A_h\}$ be the set of algorithms utilized for online anomaly detection. Each algorithm A_i is obtained by training over a special dataset D_i and previously selected among many potential algorithms as the best one. Then, given a new dataset D_{dec}, the problem of algorithm recommendation for anomaly detection (ARAD) is to select a detector/algorithm A_i with the best performance in $\mathcal{A}$.

The proposed RecAD: The intuitive idea of RecAD is to utilize the historical training information collected for the algorithms in $\mathcal{A}$. Given a set of training datasets $\mathcal{D} = \{D_1, D_2, \dots, D_h\}$, for each dataset D_i, it is assumed that all algorithms in $\mathcal{A}$ have been trained and tested over D_i, and the best one has been known. Besides D_{dec} and the algorithm set $\mathcal{A}$, the inputs of RecAD also include the matching pairs $\mathcal{M} = \{\langle A_i, D_j \rangle, \dots\}$, where a matching pair $\langle A_i, D_j \rangle$ means that the algorithm A_i performs better than others in $\mathcal{A}$ on D_j.

For the new dataset D_{dec}, the main idea of RecAD is to find out the most similar training dataset D_j and recommend the corresponding algorithm A_i to be the online detector. The whole procedure of the RecAD method is shown in Fig. 1. First, given the new data D_{dec}, after the regular preprocessing, two procedures, discretization and gram-extraction, are called to transform D_{dec} into the form proper for distribution-aware computation. Then, a procedure of computing the KL-divergence of D_{dec} with training datasets is invoked to select the recommended algorithm. Since computing KL-divergence is a rather expensive procedure, to achieve high-performance online fault detection, a filtering procedure with only sublinear time cost is utilized to prune impossible algorithms from $\mathcal{A}$ as many as possible. The details of the RecAD method is shown in Algorithm 1, where the main part (Lines 1−7) includes discretization and gram-extraction, and the function Match is invoked (Line 8) by RecAD finally.

Algorithm 1 RecAD (Recommendation for Anomaly Detection)

Input: A set of algorithms $\mathcal{A} = \{A_1, \dots, A_n\}$ and the corresponding training datasets $\mathcal{D} = \{D_1, \dots, D_n\}$ the new dataset D_{dec}.
Output: The recommended algorithm A.
1: Construct the discretized data $\hat{\mathcal{D}} = \{\hat{D}_i\}$ with $S \in \mathbb{N}^+$;
2: Construct the discretized data $\hat{D}_{\text{dec}}$ with $S \in \mathbb{N}^+$;
3: **for** each dataset $\hat{D} \in \hat{\mathcal{D}} \cup \hat{D}_{\text{dec}}$ **do**
4: Initialize $\hat{D}^{(l)}$ to be empty;
5: **for** each $\hat{x}_i \in \hat{D}$ such that $i \in [1, |\hat{D}| - l + 1]$ **do**
6: $\hat{x}_i^{(l)} \leftarrow \hat{x}_i \hat{x}_{i+1} \cdots \hat{x}_{i+l-1}$;
7: insert $\hat{x}_i^{(l)}$ to $\hat{D}^{(l)}$;
8: $A \leftarrow \arg\max_{A_i \in \mathcal{A}} \text{Match}(\hat{D}_i^{(l)}, \hat{D}_{\text{dec}}^{(l)})$;
9: **return** A;

Corresponding author: Baoling Ning.

Citation: W. Li and B. Ning, "Autonomous recommendation of fault detection algorithms for spacecraft," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 1, pp. 273–275, Jan. 2024.

W. Li is with the National Key Laboratory of Science and Technology on Space Intelligent Control, Beijing Institute of Control Engineering, Beijing 100094, China (e-mail: liwenbo_bice@163.com).

B. Ning is with the School of Data Science and Technology, Heilongjiang University, Harbin 150001, China (e-mail: ningbaoling2009@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123423

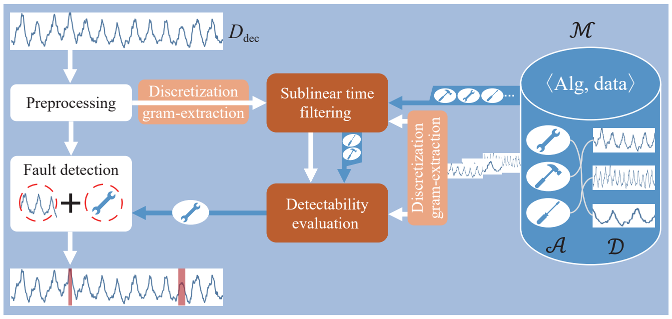

Fig. 1. Overview of the RecAD method.

Discretization and gram extraction: Given two time series data X and Y, since the elements of X and Y take real values, they can not be represented efficiently by discrete distributions. Therefore, they are first transformed into $\hat{X}$ and $\hat{Y}$ using discretization methods, and the corresponding distributions $p_{\hat{X}}$ and $p_{\hat{Y}}$ can be defined as follows.

Without loss of generality, suppose the domain of the values in X and Y is $[0,1]$ and a positive integer S is given in previous. First, the values in X and Y are discretized by replacing x_i and y_i with $\hat{x}_i = \lfloor x_i \times S \rfloor$ and $\hat{y}_i = \lfloor y_i \times S \rfloor$, and the obtained time series are denoted by $\hat{X}$ and $\hat{Y}$. Then, the corresponding distribution $p_{\hat{X}}$ can be defined by (1).

$$p_{\hat{X}}(a) = \frac{\sum_{1 \le i \le n_{\hat{X}}} \mathbf{1}(a = \hat{x}_i)}{n_{\hat{X}}}, \quad a \in [0, S] \tag{1}$$

where $n_{\hat{X}}$ is the length of the time series $\hat{X}$ and the function $\mathbf{1}(\cdot)$ returns 1 if the condition is satisfied and 0 otherwise.

Next, to capture the data dependencies over time, an extension for the time series data, named gram extraction, is introduced, whose idea is to use l-grams to replace the elements in a discretized distribution $\hat{X}$. Intuitively, the l-grams contains information of the l adjacent elements in the series, that is, they can describe the time dependencies of data within every intervals with length l. Given an element $\hat{x}_i$ in $\hat{X}$, the corresponding l-gram, denoted by $\hat{x}_i^{(l)}$, is the composited value $x_i x_{i+1} \cdots x_{i+l-1}$. Then, we can define the corresponding time series $\hat{X}^{(l)}$ as the list $\{\hat{x}_1^{(l)}, \hat{x}_2^{(l)}, \hat{x}_3^{(l)}, \ldots\}$, and the domain of elements in $\hat{X}^{(l)}$ is $\{0, 1, \ldots, S\}^l$.

Measuring the detectability: Here, the detectability of A_i on D can be represented by $\mathrm{Dec}(A_i, D)$. The method of measuring the detectability is motivated by the KL-divergence. Previous works (e.g., [13]) have shown that KL-divergence is a useful tool for the recommendation of supervised learning algorithms.

Generally speaking, given two probability distributions p and q, the KL-divergence is the measure of the relative difference between them, which can be calculated as $KL(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$.

Then, for a fixed gram size l, the detectability of A_i on D can be evaluated over the transformed data $\hat{D}^{(l)}$ and $\hat{D}_i^{(l)}$ as follows:

$$\mathrm{Dec}(A_i, D) = \mathrm{Match}(\hat{D}_i^{(l)}, \hat{D}^{(l)}) = 1/KL(p_{\hat{D}^{(l)}} \| p_{\hat{D}_i^{(l)}}). \tag{2}$$

Here, the larger the value of $KL(p_{\hat{D}^{(l)}} \| p_{\hat{D}_i^{(l)}})$ is, the smaller the *detectability* of A_i on D is. The task of calculating $\mathrm{Dec}(A_i, D)$ is implemented by the function Match whose details are shown in Algorithm 2 and will be explained in the following part.

Detectability evaluation: Obviously, the essential part of computing $\mathrm{Dec}(A_i, D)$ is calculating the corresponding KL-divergence. KL-divergence can be calculated directly according to the definition, and the computation time cost can be bounded by $O(n \log n)$. To satisfy the requirement of online algorithm recommendation and anomaly detection, the total cost of RecAD algorithm shown in Algorithm 1 should be reduced as much as possible. Therefore, propose a method to further filter unnecessary computation of $\mathrm{Dec}(A_i, D)$ and reduce the times of invoking KL-divergence computation.

Sublinear algorithms for filtering: In this part, a sublinear algorithm for filtering the candidate algorithms is introduced. The filtering algorithm can reduce the times of divergence computation and improve the performance of RecAD significantly with only small extra costs. Intuitively, if the new data D_{dec} is quite different from a special training dataset D_i, that is the corresponding KL-divergence is quite large, the matched algorithm of D_i can be filtered.

Algorithm 2 Match (Detectability Evaluation)

Input: The new dataset D_{dec}, a threshold $\gamma \in (0,1)$, a training data D^*, and an input $\epsilon \in (0,1)$.
Output: The detectability of A^* on D_{dec}.
1: $Z \leftarrow \{a | p_{D_{\text{dec}}}(a) < \gamma\}$;
2: Construct D by removing data points in Z from D_{dec};
3: Let Z' be the domain of D;
4: Construct D' by removing data points not in Z' from D^*;
5: Let $k = \frac{2\log n + \log\frac{2}{\epsilon}}{1+\epsilon}$, where n is the domain size;
6: $B_0 \leftarrow \{a : D'(a) < \frac{\epsilon}{2n}\}$;
7: **for** j from 1 to $k-1$ **do**
8: $\quad B_j \leftarrow \{a : \frac{\epsilon(1+\epsilon)^{j-1}}{2n} \le D'(a) < \frac{\epsilon(1+\epsilon)^j}{2n}\}$;
9: $count = 0$;
10: **for** each random sample d from D **do**
11: $\quad$ Let B_i be the one satisfying $d \in B_i$;
12: $\quad$ Insert d to the multiset S_i;
13: $\quad$ **if** $|S_i| == C \times \frac{\sqrt{n}}{\epsilon^2}$ **then**
14: $\quad\quad$ **if** $(\|p_{S_i}\|_2^2 > (1+\epsilon^2)/|B_i|)$ and $(D'(B_i) \ge \epsilon/k)$ **then**
15: $\quad\quad\quad$ **return** 0;
16: $\quad\quad$ **else**
17: $\quad\quad\quad$ $count = count + 1$;
18: $\quad$ **if** $count == k$ **then**
19: $\quad\quad$ **break**;
20: **return** $1/KL(p_{D^*} \| p_{D_{\text{dec}}})$;

The algorithm Match provided in this part is highly motivated by [14]. The intuitive idea of Match is to transform an arbitrary distribution D^* to a set of uniform distributions and check whether the differences between the given distribution and the uniform distributions are too large. The details are shown in the main part in Algorithm 2. The first step is to remove the data points with low frequencies from D_{dec} and remove the unrelated data points from D^* (Lines 1−4). The datasets obtained are represented by D and D', respectively. The second step is called bucketing (Lines 5−8). After the bucketing, intuitively, the constraints $D'(B_0) \le \epsilon/2$ and $\|D'_{B_j} - U_{B_j}\|_2 \le \epsilon/(2\sqrt{|B_j|})$ are satisfied. After bucketing, the algorithm Match works as follows. Random samples from D are collected into different buckets. When the size of some bucket B_i is large enough (Line 13), it will be checked whether the constrained distribution of D on B_i is far from an uniform distribution and the suppose size of B_i on D' is large enough (Line 14). If both of the two conditions are satisfied, the input dataset will be filtered by returning 0 to represent the corresponding detectability. After all buckets have been checked (Line 18), the value of $1/KL(p_{D^*} \| p_{D_{\text{dec}}})$ will be returned (Line 20) by invoking a trivial method for calculating KL-divergence exactly. Obviously, the cost of the filtering procedure is determined by the sample size, according to [14], it can be bounded by $O(\sqrt{n}\log n/\epsilon^6)$ in a high probability. Therefore, the time cost will be bounded by $\tilde{O}(\sqrt{n}/\epsilon^6)$ obviously, and the time cost of the filtering will be sublinear.

Theorem 1: Given $\epsilon \in (0,1)$, Match will take $O(\sqrt{n}\log n/\epsilon^6)$ samples from D_{dec}, and with a high probability, the candidate data D^* filtered by Match has a divergence larger than $1/(2\ln 2)\epsilon^2$.

Experimental results: This part introduces the experimental results

Datasets: Seven real life datasets, SWaT, WADI, DMDS, SKAB, MSL, SMAP, and SMD, are used. They are also often used by previous works[4], [10], and the details can be found in [12].

Algorithms: The fault detection algorithms considered to be the candidates of the recommendation methods in this letter are PCA,

UAE [15], LSTM-AE [16], TCN-AE [17], LSTM-VAE [18], MSCRED [19], BeatGAN [20], and NASALSTM [4].

Results: The comparison of algorithms utilized in the experiments and the recommendation results are shown in Table 1, where the comparison is based on the F1-scores [12]. It is found that the UAE method outperforms other methods in five of the datasets used, TCN-AE is the best detection algorithm on SMD, and LSTM-VAE performs best on WADI. The recommendation methods are shown in the last row of the table, and the algorithm recommended is denoted by RecAD. It can be found that the RecAD algorithm can not always find out the best algorithm, but it can select the best one on six datasets. Also, RecAD can indeed avoid the worst algorithm efficiently, for example, on SKAB, the algorithm recommended by RecAD is not the best but it has comparative performance.

Table 1. The Result of Recommendation

	DMDS	MSL	SKAB	SMAP	SMD	SWaT	WADI
PCA	0.5339	0.4067	0.5524	0.3793	0.5344	0.5314	0.3747
UAE	**0.6378**	**0.5111**	**0.5550**	**0.4793**	0.5501	**0.5713**	0.5105
LSTM-AE	0.5999	0.4481	0.5418	0.4536	0.5271	0.5163	0.4265
TCN-AE	0.5989	0.4354	0.5488	0.3873	**0.5800**	0.4732	0.5126
LSTM-VAE	0.5939	0.3910	0.5439	0.2988	0.5427	0.4456	**0.5758**
BeatGAN	0.5391	0.4531	0.5437	0.3732	0.5479	0.4777	0.4908
MSCRED	0.2906	0.3944	0.5526	0.3724	0.4145	0.4315	0.3253
NASALSTM	0.1284	0.4715	0.5339	0.4280	0.3879	0.1398	0.1058
RecAD	UAE	UAE	LSTM-VAE	UAE	TCN-AE	UAE	LSTM-VAE

To verify the efficiency of RecAD, we selected 10 data slices generated from the given datasets, and ran both the RecAD algorithms with and without filtering procedures. The time costs of them are compared and shown in Fig. 2, where the labels of x-axis represent different data slices, the time cost taken by the no-filter method is standardized to be 1, and the values of y-axis represent the ratio of time costs between RecAD with and without filtering procedures. It can be found that the filtering procedure proposed by this paper can improve the performance of RecAD significantly in most cases. The only exceptional instance is the third data slice, where the cost using filtering is 6% more than the one not using filtering, because the data slice is too common to rule out any algorithms by the filtering procedure and the extra cost is caused by the filtering procedure.

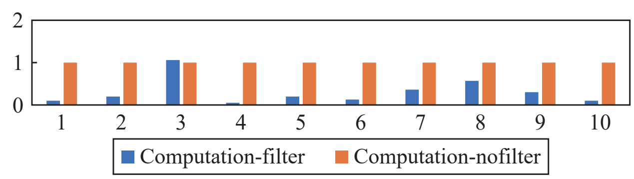

Fig. 2. Efficiency of the filtering procedure.

To verify the end-to-end performance of RecAD, three kinds of time costs are compared. The first one is All-Check representing the cost of using all algorithms to check anomalies, the second one is Rec-Check representing the cost of using only the recommended algorithm, and the third one is RecAd representing the cost of selecting the recommended algorithm. The detailed results are shown in Table 2. It can be found that the procedure of algorithm recommendation only takes few costs and the time costs can be hugely reduced by the strategy of only running the recommended algorithm.

Table 2. The Result of Time Costs (in seconds)

	DMDS	MSL	SKAB	SMAP	SMD	SWaT	WADI
All-Check	175.8	30.3	25.1	109.3	344.8	553.1	737.7
Rec-Check	36.7	5.1	5.3	17.9	65.5	97.3	170.5
RecAd	4.1	2.3	3.6	3.0	3.1	2.9	3.2

Conclusion: This letter has investigated the problem of algorithm recommendation for online anomaly detection of spacecrafts. Using the idea of measuring detectabilities by distributions, RedAD is proposed to support efficient automated algorithm recommendation for detecting spacecraft anomalies. Experimental results show that the proposed method is effective and efficient.

Acknowledgments: This work was supported by the National Key R&D Program of China (2021YFB1715000) and the National Natural Science Foundation of China (U1811461, 62022013, 12150007, 62103450, 61832003, 62272137).

References

[1] L. Basora, X. Olive, and T. Dubot, "Recent advances in anomaly detection methods applied to aviation," *Aerospace*, vol. 6, no. 11, pp. 1–27, 2019.

[2] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2022.

[3] J. Chen, J. Sun, and G. Wang, "From unmanned systems to autonomous intelligent systems," *Engineering*, vol. 12, no. 5, pp. 16–19, 2022.

[4] K. Hundman, V. Constantinou, C. Laporte, *et al*., "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. KDD*, 2018, pp. 387–395.

[5] X. Wang, J. Sun, G. Wang, *et al*., "Data-driven control of distributed event-triggered network systems," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 2, pp. 351–364, 2023.

[6] Y. Gao, T. Yang, M. Xu, and N. Xing, "An unsupervised anomaly detection approach for spacecraft based on normal behavior clustering," in *Proc. ICICTA*, 2012, pp. 478–481.

[7] K. Li, Y. Wu, S. Song, *et al*., "A novel method for spacecraft electrical fault detection based on FCM clustering and wpsvm classification with PCA feature extraction," in *Proc. Institution of Mechanical Engineers*, 2017, pp. 98–108.

[8] X. Liu, J. Li, H. Gao, *et al.*, "On the hardness of labeled correlation clustering problem: A parameterized complexity view," *Theor. Comput. Sci.*, vol. 609, no. 3, pp. 583–593, 2016.

[9] B. Pilastre, L. Boussouf, S. D'Escrivan, *et al.*, "Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning," *Signal Process*., vol. 168, pp. 1–10, 2020.

[10] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. KDD*, 2019, pp. 2828–2837.

[11] L. Xu, L. Zheng, W. Li, *et al*., "NVAE-GAN based approach for unsupervised time series anomaly detection," CoRR, vol. abs/2101.02908, pp. 1–8, 2021.

[12] A. Garg, W. Zhang, J. Samaran, *et al*., "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Trans. Neural Networks Learning Systems*, vol. 33, no. 6, pp. 2508–2517, 2022.

[13] C. M. Hurvich and C.-L. Tsai, "Regression and time series model selection in small samples," *Biometrika*, vol. 76, pp. 297–307, 1989.

[14] T. Batu, L. Fortnow, E. Fischer, *et al*., "Testing random variables for independence and identity," in *Proc. FOCS*, 2001, pp. 442–451.

[15] D. E. Rumelhart, G. Hinton, R. Williams, *et al*., "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, USA: MIT Press, 1986, vol. 1, pp. 318–362.

[16] P. Malhotra, A. Ramakrishnan, G. Anand, *et al.*, "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv preprint arXiv: 1607.00148, 2016.

[17] S. Bai, J. Kolter, V. Koltun, *et al*., "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv: 1803.01271, 2018.

[18] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[19] C. Zhang, D. Song, Y. Chen, *et. al*., "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proc. AAAI*, 2019, pp. 1409–1416.

[20] B. Zhou, S. Liu, B. Hooi, *et al*., "BeatGAN: Anomalous rhythm detection using adversarially generated time series," in *Proc. IJCAI*, 2019, pp. 4433–4439.