Evaluation of Causal Inference Techniques for AIOps

Conference Paper · January 2021

DOI: 10.1145/3430984.3431027

Vijay Arya
IBM
101 PUBLICATIONS **2,146** CITATIONS

Karthikeyan Shanmugam
IBM
53 PUBLICATIONS **750** CITATIONS

Pooja Aggarwal
IBM
15 PUBLICATIONS **88** CITATIONS

Qing Wang
IBM Research
23 PUBLICATIONS **425** CITATIONS

Evaluation of Causal Inference Techniques for AIOps

V. Arya[1], K. Shanmugam[2], P. Aggarwal[1], Q. Wang[2], P. Mohapatra[1], S. Nagar[1]
IBM Research AI (India[1], USA[2])

ABSTRACT

Inferring causality of events from log data is critical to IT operations teams who continuously strive to identify probable root causes of events in order to quickly resolve incident tickets so that downtimes and service interruptions are kept to a minimum. Although prior work has applied some specific causal inference techniques on proprietary log data, they fail to benchmark the performance of different techniques on a common system or dataset. In this work, we evaluate the performance of multiple state-of-the-art causal inference techniques using log data obtained from a publicly available benchmark microservice system. We model log data both as a timeseries of error counts and as a temporal event sequence and evaluate 3 families of granger causal techniques: regression based, independence testing based, and event models. Our preliminary results indicate that event models yield causal graphs that have high precision and recall in comparison to regression and independence testing based granger methods.

KEYWORDS

Causal Inference, Granger causality, AIOps, IT Operations, log data

1 INTRODUCTION

AIOps (AI for IT Operations) solutions attempt to automate the monitoring of large complex enterprise application environments and their supporting infrastructure. By ingesting data from multiple sources such as applications, infrastructure, network, cloud and existing monitoring tools, these integrated solutions promise to provide a wide variety of functions to minimise service outages and assist site reliability engineering teams with rapid incident resolution [18]. These functions generally include anomaly detection, event correlation, prediction and prevention of emerging incidents, reduction in false alarms or alert/ticket storms and root cause analysis. A key component needed to implement most of these functions, especially fault localisation and root cause analysis, is the ability to determine causality of events from historical log data. In this work, we evaluate the performance of state-of-the-art granger causal inference techniques using temporal log data obtained from a benchmark microservice system [35].

While prior work has studied the problem of inferring causal relationships from log data in the context of ISP networks [13, 14], data centers [15], and search engine query logs [28], these works apply specific causal inference techniques on proprietary data and fail to compare the performance of different techniques on a common system or a publicly available log dataset. As a consequence it becomes a challenge to replicate results or understand the advantages and disadvantages of different causal inference techniques and recommend the right one in the context of a new IT environment. Moreover prior work models log data primarily as a time series and applies a variant of the PC algorithm [12] based on independence testing or a variant of regression based granger causality test [9] to infer the causal relationships among events. This modelling choice limits the range of causal inference techniques that can subsequently be applied on log data. In this work, we compare the performance of 3 families of granger causal inference techniques and their variants on a benchmark TrainTicket microservice system that is publicly available [33–35]: (a) PC algorithm based on independence tests and its variants (b) variants of regression based techniques, and (c) graphical event models [3, 10]. While in the first two approaches, the log data is modelled as a time series, in the third approach, the log data is modelled as a temporal event sequence and we leverage the recently proposed graphical event models to infer causal relationships. We present preliminary performance results of different granger causal inference techniques on a log dataset that spans half an hour and is obtained by injecting a fault in one of the microservices in the benchmark system which includes a total of 41 microservices. With the help of ground truth data, we compute the precision, recall, and F1 scores of the inferred causal graphs among a subset of impacted microservices. We view the main contributions of this work as follows:

- Using log data collected from a publicly available benchmark microservice system, we investigate the performance of different granger causal inference algorithms and present preliminary evaluation results for different variants of each algorithm. In addition to modelling the log data as a time series which is commonly used in prior work, we also model it as a temporal event sequence and present accuracy results for the recently proposed graphical event models.
- Our preliminary experimental results show that event models in general yield causal graphs that have high precision and recall in comparison to other approaches. However regression and independence testing based granger methods have parameters that may be fine tuned to improve performance.

The balance of this paper is organised as follows. Section 2 briefly describes the benchmark microservice system and modelling of log data. Section 3 provides an overview of 3 families of granger causal inference techniques. Section 4 presents experimental results and we conclude in section 5 with directions for future work.

2 LOG DATA AND MODELLING

Benchmark Microservice System. In this work, we use a publicly available benchmark TrainTicket microservice system[1] and

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

[1] https://github.com/FudanSELab/train-ticket

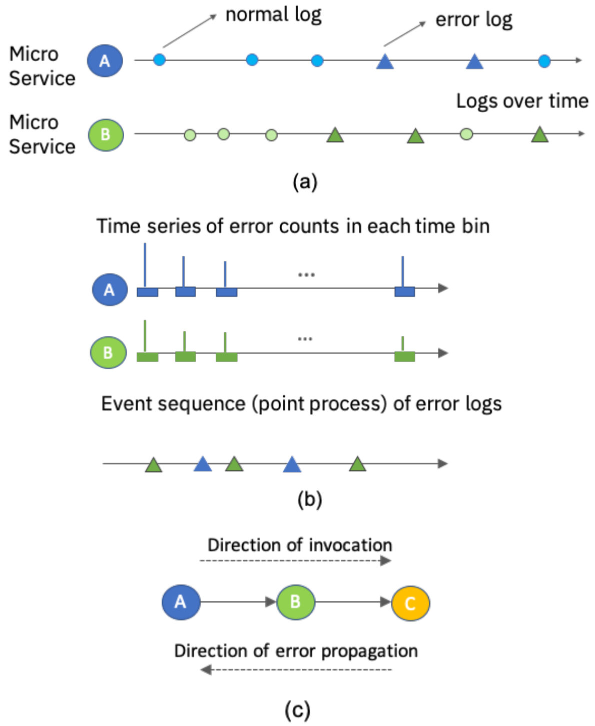
normal log
error log
Micro Service
A
Micro Service
B
Logs over time
(a)
Time series of error counts in each time bin
A
B
...
Event sequence (point process) of error logs
(b)
Direction of invocation
A
B
C
Direction of error propagation
(c)

Figure 1: (a): Log data, (b) Modeling log data as a time series or event sequence, and (c) A sample causal graph that we wish to infer which implies that errors in microservice A are caused by errors in microservice B are caused by errors in microservice C (i.e. root cause of error is C).

deploy it on a Kubernetes cluster in order to collect log data. This system has about 41 microservices that work together and allow users to reserve train tickets, make payments, enter station, etc. Several microservices in the TrainTicket system interact with one another resembling microservice systems commonly seen in large enterprise applications. This system is run for about half an hour during which bookings made by multiple users is simulated. A fault is injected in one of the microservices which impacts a subset of other microservices in the system. Each microservice continues to emit logs which may be normal or erroneous depending on how it is impacted by the fault (e.g. see Fig. 1(a)). We consider logs from those microservices which emit at least one erroneous log and attempt to construct a causal graph among these impacted microservices.

Labelling of log messages. The logs collected from the system are available in JSON format with each log having about 71 fields including attributes such as the name of the microservice, container id, log line message, and so on. We use a dictionary based classifier that looks for error patterns in various fields such as log level and log line messages (e.g. '500 Internal Server Error') to accurately distinguish whether a log is normal or erroneous. We manually validate the accuracy of our labels by comparing the logs emitted before and after the fault and by considering the underlying microservice architecture (this is generally unavailable in real enterprise environments). As opposed to prior work that uses machine learning techniques to distinguish anomalous logs [6, 31], we use the above domain specific approach in order to avoid introducing errors in labelling of logs so that we can first evaluate the performance of the different causal inference methods in the absence of any label noise.

Modeling logs as timeseries. We consider different time bin sizes (10ms, 100ms, 1sec) and count the number of error logs in each bin to obtain a timeseries of error counts corresponding to each impacted microservice (e.g. see figure 1(b)).

Modeling logs as an event sequence. We construct a tuple sequence $\{(t_i, l_i)\}$ where t_i refers to the time at which the error log of microservice l_i occurs. We consider unique tuples only, i.e., if more than one error log of the same microservice is recorded at the same time, only one tuple is retained. However if error logs of different microservices occur at the same time, all of these are retained.

Our goal is to infer the causal graph among the microservices which have been impacted by the fault in order to aid root cause analysis. Figure 1(c) shows a sample graph which explains how errors in one microservice are caused by errors in an other microservice. The next section briefly reviews the three families of causal inference techniques that we evaluate in our experiments.

3 OVERVIEW OF GRANGER CAUSALITY FOR TIME SERIES AND EVENT SEQUENCES

The intuitive idea of Granger causality [9] is that if the time series A *Granger causes* time series B, the past of A has additional information about the future of B over and above the information contained in the past of time series B. This criterion could be used to verify if a "causal" relationships between time series A and B exists. Since the criterion is purely associational (against an otherwise interventional notion popular in other causal theories [11, 19]) but applied with the aid of arrow of time, this notion often is called Granger causality to distinguish it from interventional notions.

3.1 Regression Based Approaches

One of the classic approaches for a Granger causal test is to linearly regress B_t on $A_{t-1:t-p}, B_{t-1:t-p}$ for some lag p and compare the residue with that of regression of B_t on $B_{t-1:t-p}$ alone. Distributional test involving residues finally acts a Granger causal test. [8] was one of the earliest works that studied the above idea and proposed distributional tests for residues in linear regression. The above linear regression based test can also be done in the frequency domain using the theory of Auto-regressive processes. The time domain tests and frequency domain tests culminated in the MVGC toolbox quite popular in neuroscience [2]. The above test can be extended to the case when A is a multivariate time series. Moving away from testing methods, recent works employ sparse linear regression with a LASSO penalty [1, 32] often called Lasso-Granger methods. [16] observed that lagged variables belonging to the same time series must be grouped together and advocate a group lasso penalty with linear regression. Recently, neural networks in the form of LSTM units is used to model non linearity in the data and a hierarchical group lasso penalty is used along with an LSTM model [29].

Although the literature on regression based techniques in vast, for our experimental comparison we use methods that conduct causal inference from a Bayesian perspective: Bayesian Linear [4] and Bayesain lasso regression [17].

3.2 Independence Testing Based Approaches

From information theory perspective, one of the important signatures of Granger causality for time series is the following test: Time series A *does not* Granger cause B if for some set of time series $\mathbf{Z}$, the following quantity is zero for some lag p:

$$D(A \rightarrow B|\mathbf{Z}) = \sum_{t=p}^{T} I(B_t; A_{t-1:t-p}|B_{t-1:t-p}, \mathbf{Z}_{t-1:t-p}) \qquad (1)$$

Here, $I(\cdot;\cdot|\cdot)$ is the conditional mutual information and D is sometimes called directed mutual information. If D from A to B given $\mathbf{Z}$ is zero, one can delete the directed edge between A and B. [7, 22] formalized this and showed that one can recover a Bayesian Network on the time series if the data is generated according to a *strictly causal* manner (instantaneous influence is forbidden) and if there are no unobserved latent variables (causal sufficiency) under some mildly technical regularity conditions. In practice, with a single realization of the time series and when the time series has not attained stationarity, it is not directly possibly to test directed mutual information. In practice, the above condition is tested [23] approximately using off the shelf conditional independence (CI) testers that check if $I(X;Y|W)$ is 0 or not on a set of i.i.d samples X_i, Y_i, W_i with some joint distribution. To use it, we supply a data table with rows indexed by t and the columns substituted as follows: $X_t = B_t$, $Y_t = A_{t-1:t-p}$ and $W_t = [\mathbf{Z}_{t-1:t-p}, B_{t-1:t-p}]$.

The manner in which the tests of the form $D(A \rightarrow B|\mathbf{Z})$ will be used to construct the graph is highly analogous to the PC [12, 24, 25] and MMPC [30] algorithms used for non time series based causal structure discovery. PC has two popular variants applied to the time series case: a) PCMCI algorithm in [23] and b) Modified PC algorithm in [5]. MMPC algorithm from the i.i.d setting has been adapted to the time series case in [20].

We use 3 i.i.d CI testers namely G^2, ParCorr (Partial Correlation Test) [23] and RCoT (Randomized Conditional Correlation Test) [27]. G^2 CI test is based on G^2 statistic, ParCorr is a fast test based in linear modeling assumptions, and RCoT is a tester that works even for non linear relations based on Kernel techniques. We select a subset of the CI testing algorithms: PC [12], Modified PC (denoted PCMod) [5], and MMPC [30].

3.3 Graphical Event Models (GEMS)

An event A is a random sequence of time points with label A on a time line. Granger causality for multiple event sequences (multiple labeled points on a common timeline) is modelled by a Graphical Event Model (GEM). In this, a set of events is modelled by a multivariate inhomogenous Poisson process, where instantaneous intensity (Poisson rate) of event A at time t is a function of the history of the *parental events* $\text{Pa}(A)$, i.e., intensity function at time t is a function of the time sequence of occurrences of event types in $\text{Pa}(A)$ before t. The tuples $(A, \text{Pa}(A))$ form a directed graph that constitutes the structure of the GEM. Any GEM model with a smooth conditional intensity function has been shown to be approximated by a class of GEMs where the conditional intensity function is a function of indicators each of which indicates if some parental event occurred in a specific time bin in the past or not [10]. They also propose score based method to search over all such GEMs, whose intensity functions depend on indicators of event occurrences in various time bins in the past, that converges to the true GEM from this family in asymptotical limit. A score based method assigns a scalar value to a graph structure (or generally a GEM structure) which is a combination of a likelihood term and a complexity term (related to the sparsity of the graph). The score maximizing graph is chosen at the end of a forward backward greedy search procedure. However, implementing this score method in practice for the general class of GEMs in [10] is infeasible due to search over time binning parameters in addition to graph search. A practical method was proposed in [3] where the authors make a simplifying proximal assumption (resulting in a Proximal GEM denoted PGEM). They assume that the conditional intensity function depends only on indicators of event occurrences of each of the parental events in a single window (per parent event) of a certain length in the immediate past. A scored based method over only graph structures was proposed where the window search (for each of the candidate parents given a graph) was incorporated in the likelihood maximization involving the intensity functions. This decoupling and searching over only the graphs makes this practical and scalable. In our experiments, since the raw error logs are just event sequences, we use PGEM search procedure to learn the graph structure of influences.

4 EXPERIMENTAL RESULTS

We now present experimental evaluation results based on log data collected from the TrainTicket microservice system. In order to introduce a fault, one of the microservies namely ts-basic is deleted from the system which results in 4 microservices emitting error logs namely: ts-ui-dashboard, ts-travel2-service, ts-travel-service, and ts-ticketinfo-service. We filter out error logs along with their timing information to construct (a) time series of error counts corresponding to each impacted microservice and (b) a temporal event sequence $\{(t_i, l_i)\}$ which records the time t_i at which microservice l_i emits an error log. A total of 266 error logs were emitted. For timeseries, we experiment with 3 bin sizes: 10, 100, and 1000ms. We use the 3 families of granger causal discovery algorithms described in Section 3 namely: Conditional independence (CI) testing based, regression based, and GEMs to infer the causal dependencies among the 4 impacted microservices. We compute precision, recall, and F_1 scores for all algorithms with respect to ground truth.

Figure 2 shows the causal graphs inferred by one variant of each algorithm along with ground truth information. The gold coloured edges represent a match with ground truth. The false positives or type I errors are marked by a grey edge and indicate superfluous causal relationships inferred. The false negatives or type II errors are marked by a dashed grey edge and indicate causal relationships that the algorithm fails to recover from log data.

Table 1 documents the results for all algorithms. Firstly note that using correlation metric does not help recover the true causal dependencies (row 1). Next we experiment with 3 CI testing algorithms: PC [12], PCMod [5], and MMPC [30] in combination with 3 CI tests: G^2, partial correlation, and RCoT [26]. We observe that advanced PC variants PCMod and MMPC yield better results in

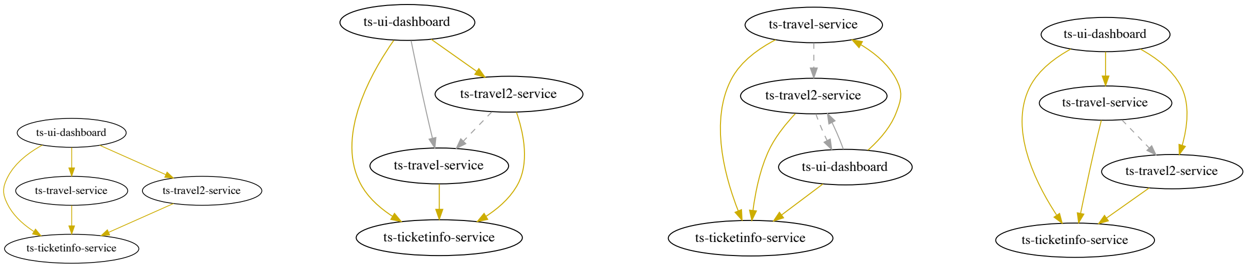

Figure 2: Causal graphs inferred using different granger methods. (left to right): Ground truth, CI testing based MMPC(ParCorr, bin=100ms), regression-based BLR(1) with bin=100ms, and event model based PGEM. False +ves (superfluous causal relationships) are marked by dashed gray edges and false -ves (missing causal relationships) with gray edges.

combination with linear CI tests (G^2 and partial correlation) than with non linear CI tests such as RCoT.

We use 2 regression based algorithms to infer causal dependencies between microservices: BLR(q_0) [4] and BLasso(λ) [17]. BLR(q_0) uses Bayesian linear regression with prior distribution $\mathcal{N}(\mathbf{0}, {q_0}^{-1}\mathbf{I}_d)$, where d is the number of time series and q_0 is used to tune the penalty term in ridge regression. In our experiments, the performance of the algorithm was not sensitive to q_0, so we set $q_0 = 1.0$. BLasso(λ) applies Bayesain Lasso to learn the causal graph, where λ weighs the L_1 penalty term. We vary λ from 0.01 to 1000 and present the average case results corresponding to $\lambda = 1.0$. Additionally, using ground truth information, we determine the λ value that yields the highest F_1 score for our dataset. We observe that BLasso($\lambda = 10$) with bin size 10ms yields the highest accuracy. In practise, such a search procedure may be used to determine λ using training datasets that have associated ground truth information.

We use the proximal graphical event model [3] to infer causal relationships from event sequence data. PGEM is largely parameter free and yields high precision and recall with only one false positive edge (Fig. 2).

We observe that both bin size and model parameters play an important role for regression and CI testing based granger methods. All methods uniformly perform worse for larger bin size of 1000ms (inter-arrival times between error logs in this dataset vary significantly with a mean of 2224ms and std of 4959ms). Additionally, both methods have parameters that may be fine tuned to improve accuracy with the help of training datasets that have ground truth information.

Method	bin(ms)	Precision	Recall	F_1
correlation	10-10^3	0	0	**0**
PC (G^2, RCOT)	10	0.43	0.6	0.5
PC (G^2, RCOT)	10^2	0.33	0.4	0.36
PC (G^2, RCOT)	10^3	0.5	0.4	0.44
PCMod (G^2)	10	0.71	1.0	**0.83**
PCMod (G^2)	10^2	1.0	0.6	0.75
PCMod (G^2)	10^3	0	0	0
PCMod (RCOT)	10	0	0	0
PCMod (RCOT)	10^2	0.5	0.2	0.28
PCMod (RCOT)	10^3	0.4	0.4	0.4
MMPC (ParCorr)	10	0.45	1.0	0.625
MMPC (ParCorr)	10^2	0.8	0.8	**0.8**
MMPC (ParCorr)	10^3	1.0	0.4	0.57
MMPC (RCOT)	10	0.625	1.0	0.76
MMPC (RCOT)	10^2	0.33	0.2	0.25
MMPC (RCOT)	10^3	0.66	0.4	0.5
BLR(1.0)	10-10^2	1.0	0.6	**0.75**
BLR(1.0)	10^3	0.75	0.6	0.66
BLasso(1.0)	10	0.66	0.8	0.72
BLasso(1.0)	10^2-10^3	0.75	0.6	0.66
PGEM	NIL	0.83	1.0	**0.88**
After tuning parameters using ground truth information				
BLasso(10.0)	10	1.0	1.0	**1.0**
BLasso(10.0)	10^2	1.0	0.6	0.75
BLasso(10.0)	10^3	1.0	0.4	0.57

Table 1: Performance results of different causal inference methods. Correlation (no causation). PC variants: PC [12], PCMod [5], MMPC [21]. Granger causality variants: BLR [4], BLasso [17]. Graphical Event models: PGEM [3]

5 DISCUSSIONS AND FUTURE WORK

AIOps solutions are crucial to IT Operations teams today who manage infrastructure and applications running across complex on-prem, container, and multi-cloud environments. In order to minimise downtimes, these teams need to continually identify probable root causes of events by inferring causal relationships from log data. In this work, we benchmarked the performance of multiple causal inference techniques using log data available from the TrainTicket microservice system and by modelling log data both as a timeseries of error counts and as a temporal event sequence. Our preliminary results show that graphical event model based granger techniques yield causal graphs that have high F_1 score. The accuracy of regression and CI testing based granger methods is dependent on parameters which need to be fine tuned to improve performance.

In future work, we will extend our analysis to multiple datasets that involve different types of faults and larger number of microservices. We plan to study the sensitivity of algorithms to both length and granularity of time series, log label noise, and effects of jitter in timing information of logs. We also plan to compare the computational scalability of methods and evaluate the performance of neural network based causal inference algorithms.

REFERENCES

[1] Andrew Arnold, Yan Liu, and Naoki Abe. 2007. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 66–75.

[2] Lionel Barnett and Anil K Seth. 2014. The MVGC multivariate Granger causality toolbox: a new approach to Granger-causal inference. *Journal of neuroscience methods* 223 (2014), 50–68.

[3] Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. 2018. Proximal Graphical Event Models. In *Advances in Neural Information Processing Systems 31*. 8136–8145.

[4] Christopher M Bishop. 2006. Pattern recognition. *Machine Learning* 128 (2006).

[5] John W Cook, David Danks, and Sergey M Plis. [n.d.]. Learning dynamic structure from undersampled data.

[6] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1285–1298. https://doi.org/10.1145/3133956.3134015

[7] Michael Eichler. 2006. Graphical modelling of multivariate time series with latent variables. *Preprint, Universiteit Maastricht* (2006).

[8] John F Geweke. 1984. Measures of conditional linear dependence and feedback between time series. *J. Amer. Statist. Assoc.* 79, 388 (1984), 907–915.

[9] C. W. J. Granger. 1969. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica* 37, 3 (1969), 424–438.

[10] Asela Gunawardana and Chris Meek. 2016. Universal Models of Multivariate Temporal Point Processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*, Vol. 51. PMLR, Cadiz, Spain, 556–563.

[11] Guido W Imbens and Donald B Rubin. 2010. Rubin causal model. In *Microeconometrics*. Springer, 229–241.

[12] Markus Kalisch and Peter Bühlmann. 2007. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm. *J. Mach. Learn. Res.* 8 (May 2007), 613–636.

[13] S. Kobayashi, K. Otomo, and K. Fukuda. 2019. Causal analysis of network logs with layered protocols and topology knowledge. In *2019 15th International Conference on Network and Service Management (CNSM)*. 1–9.

[14] S. Kobayashi, K. Otomo, K. Fukuda, and H. Esaki. 2018. Mining Causality of Network Events in Log Data. *IEEE Transactions on Network and Service Management* 15, 1 (2018), 53–67.

[15] Chen Liang, Theophilus Benson, Partha Kanuparthy, and Yihua He. 2016. Finding Needles in the Haystack: Harnessing Syslogs for Data Center Management. arXiv:cs.NI/1605.06150

[16] Aurelie C Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. 2009. Grouped graphical Granger modeling methods for temporal causal modeling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 577–586.

[17] Trevor Park and George Casella. 2008. The bayesian lasso. *J. Amer. Statist. Assoc.* 103, 482 (2008), 681–686.

[18] Seth Paskin. 2020. https://www.bmc.com/blogs/what-is-aiops/.

[19] Judea Pearl. 2009. *Causality*. Cambridge university press.

[20] Bernat Guillen Pegueroles, Bhanukiran Vinzamuri, Karthikeyan Shanmugam, Steve Hedden, Jonathan D Moyer, and Kush R Varshney. 2018. Structure learning from time series with false discovery control. *arXiv preprint arXiv:1805.09909* (2018).

[21] Bernat Guillen Pegueroles, Bhanukiran Vinzamuri, Karthikeyan Shanmugam, Steve Hedden, Jonathan D Moyer, and Kush R Varshney. 2018. Structure learning from time series with false discovery control. *arXiv preprint arXiv:1805.09909* (2018).

[22] Christopher J Quinn, Negar Kiyavash, and Todd P Coleman. 2015. Directed information graphs. *IEEE Transactions on information theory* 61, 12 (2015), 6887–6909.

[23] Jakob Runge. 2018. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28, 7 (2018), 075310.

[24] Peter Spirtes and Clark Glymour. 1991. An Algorithm for Fast Recovery of Sparse Causal Graphs. *Social Science Computer Review* 9, 1 (1991), 62–72.

[25] Peter Spirtes, Clark Glymour, and Richard Scheines. 2001. *Causation, Prediction, and Search, 2nd Edition*. MIT Press Books, Vol. 1. The MIT Press.

[26] Eric V. Strobl, Kun Zhang, and Shyam Visweswaran. 2017. Approximate Kernel-based Conditional Independence Tests for Fast Non-Parametric Causal Discovery. arXiv:stat.ME/1702.03877

[27] Eric V Strobl, Kun Zhang, and Shyam Visweswaran. 2019. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *Journal of Causal Inference* 7, 1 (2019).

[28] Yizhou Sun, Kunqing Xie, Ning Liu, Shuicheng Yan, Benyu Zhang, and Zheng Chen. 2007. Causal Relation of Queries from Temporal Logs. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, 1141–1142.

[29] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily Fox. 2018. Neural granger causality for nonlinear time series. *arXiv preprint arXiv:1802.05842* (2018).

[30] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning* 65, 1 (2006), 31–78.

[31] R. Vaarandi, B. Blumbergs, and M. Kont. 2018. An unsupervised framework for detecting anomalous messages from syslog log files. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. 1–6.

[32] Chunqiu Zeng, Qing Wang, Wentao Wang, Tao Li, and Larisa Shwartz. 2016. Online inference for time-varying temporal dependency discovery from time series. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 1281–1290.

[33] X. Zhou, X. Peng, T. Xie, J. Sun, C. Ji, W. Li, and D. Ding. 2018. Fault Analysis and Debugging of Microservice Systems: Industrial Survey, Benchmark System, and Empirical Study. *IEEE Transactions on Software Engineering* (2018), 1–1.

[34] Xiang Zhou, Xin Peng, Tao Xie, Jun Sun, Chao Ji, Dewei Liu, Qilin Xiang, and Chuan He. 2019. Latent error prediction and fault localization for microservice applications by learning from system trace logs. In *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019*. ACM, 683–694.

[35] Xiang Zhou, Xin Peng, Tao Xie, Jun Sun, Chenjie Xu, Chao Ji, and Wenyun Zhao. 2018. Benchmarking microservice systems for software engineering research. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018*. ACM, 323–324.