Detecting Causal Structure on Cloud Application Microservices Using Granger Causality Models

Qing Wang
Hybrid Cloud Service
IBM Research
Yorktown Heights, NY, US
qing.wang1@ibm.com

Larisa Shwartz
Hybrid Cloud Service
IBM Research
Yorktown Heights, NY, US
lshwart@us.ibm.com

Genady Ya. Grabarnik
Dept. Math & Computer Science
St. John's University
Queens, NY, USA
grabarng@stjohns.edu

Vijay Arya
Foundations of Trusted AI
IBM Research AI
Bangalore, KA, IN
vijay.arya@in.ibm.com

Karthikeyan Shanmugam
Foundations of Trusted AI
IBM Research AI
Yorktown Heights, NY, US
karthikeyan.shanmugam2@ibm.com

***Abstract*—The loosely-coupled microservices architecture has become increasingly popular due to the advantage of its modularity and elasticity in cloud applications. However, it also seriously complicates cloud management and degrades the performance of IT operations. Today, AI has been the locus of commerce and transactions, and transforming traditional IT operations for speed and growth. Inferring the dependencies among an application's microservices can greatly help SREs diagnose possible root causes of performance issues, which is a hard task due to the complex topology of microservices is often unknown in practice. Prior literature on detecting causal structure for cloud services requires significant application instrumentation, which rarely holds in reality. In this work, we leverage Granger causality models on just monitored log data of a microservice-based application to infer the impact of dependencies between microservices. We first describe the approach of modeling discrete log data as time series, and then formally define the Granger causality problem using both linear and nonlinear autoregressive models. Finally, we conduct an extensive comparative study to show the performance of the state-of-the-art linear and nonlinear (i.e., neural) Granger causality methods on both synthetic data and real-world log data from a publicly available benchmark microservice system. Our preliminary results indicate that neural Granger causality models outperform traditional Granger causality methods on both linear and nonlinear time series data, while for large linear time series, linear Granger causal models are more efficient with high accuracy. Using the real-world log data, we also demonstrate our interesting findings on inferred dependency graph of microservices by linear and neural Granger causality models.**

I. Introduction

Since digital transformation have accelerated across the globe in nearly every industry, AI for IT Operations (AIOps) has become a critical capability for any enterprise that aims to use rapidly growing IT data to assist its IT operations in providing reliability for its applications [10], [21], [28], [30], [32]. Nowadays, more and more companies have migrated their business systems from the monolithic service architecture to microservices architecture (Figure 1 shows a benchmark microservice application called TrainTicket [38], [39]).

These industrial microservice applications containing hundreds to thousands of microservices and unknown complex dependency relationships between them supported by extremely dynamic infrastructure, container deployments, for example, increasingly have lifespans of 10 seconds or less, making it challenging for site reliability engineers (SREs) to correctly diagnose incidents and timely fix them, or preemptively avert problems. By ingesting data from multiple sources such as applications, infrastructure, network, cloud and existing monitoring tools, AIOps solutions are inspired to provide a wide variety of functions to minimize service outages and assist SREs in their work. These functions include anomaly detection, event correlation, prediction and prevention of emerging incidents, reduction in false alarms or alert/ticket storms and root cause analysis. In interviews conducted with SREs, they identified diagnosis as the most difficult of the tasks. The majority of SREs pointed out that given right diagnosis, they would be able to quickly identify actions required to resolve the issue. Being able to troubleshoot a problem and to arrive to a diagnosis is often considered to be an innate skill [8]. A key prerequisite for a diagnosis, fault localization and root cause analysis, is the ability to determine causality of errors from log data, which contain details about an error: a resource, a microservice name and the timestamp it was registered.

The problem of inferring causal relationships from log data has been studied in the context of ISP networks [14], [15], data centers [17], and search engine query logs [24], but these work simply 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 is a challenge to replicate the results or to understand the advantages and disadvantages of different causal inference techniques which is necessary for providing a recommendation in the context of a new IT environment. Granger causality framework is known widely for its simplicity, robustness and extensibility, and commonly used in practice [6]. In this paper, we carefully conduct an extensive

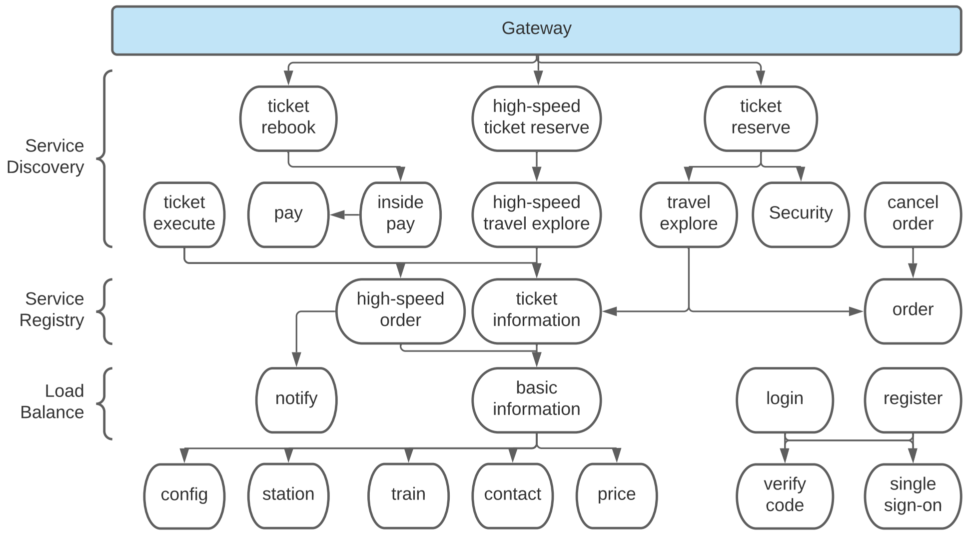

Fig. 1: The architecture of a benchmark microservice system called TrainTicket. It contains 41 microservices and 73 dependency relationships between microservices.

study using different linear and neural (i.e., nonlinear) Granger causal models to infer causal relationships on both synthetic datasets and real-world log data collected through TrainTicket system. Given the ground truth, we also demonstrate their performance using precision, recall, and F1 score as metrics. Our contributions include the following:

- We apply a binning technique on log data collected from a publicly available benchmark microservices system for further causality analysis.
- We provide the mathematical formulation of the problem of mining causal dependencies from time series data using linear and neural Granger causality models.
- We conduct extensive experiments to show the performance of linear and neural (i.e., nonlinear) Granger causality models on both synthetic time series data and real-world log data for detecting dependency structure between microservices.

The remainder of this paper is organized as follows. Section II briefly summarize relevant existing works. We describe the log data modeling and formulate the problem for identifying causal dependency using Granger causality in Section III. Extensive empirical evaluation results are reported in Section IV. Finally, we conclude our work and the future work in Section V.

II. Related Work

With the advent of AI techniques, AIOps is proposed to enhance IT operations and help IT operation teams to respond more quickly and even proactively to downtimes and outages [29], [33], [36], [37]. Mining temporal dependency structure among time series has been extensively studied in the modern IT system management [3], [34]. Jia et al. [12] use causal inference techniques to build a dependency graph among the application services for anomaly detection. [19] models causal dependencies among KPIs based on Granger causality to facilitate localizing faults in cloud system. It is still in its infancy using log data to uncover the hidden interactions of service architecture or system network in a distributed system due to lack of benchmark data.

Currently, two popular approaches (i.e., Bayesian network inference and Granger causality) prevail in the literature for causal relation inference. Since Granger causality is more straightforward and robust, we focus more on the approaches based on Granger causality. 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" relationship between time series A and B exists. Since the criterion is purely associational (against an otherwise interventional notion popular in other causal theories [11], [22]) but applied with the aid of arrow of time, this notion often is called Granger causality to distinguish it from interventional notions. Recent works employ sparse linear regression with a Lasso penalty [2] often called Lasso-Granger methods. Two linear Granger causality models [35] introduce BLR and Blasso, which model Grange causality from a Bayesian perspective can also capture the dynamic temporal dependency of time series data.

The above methods based on Granger causality assume linear time series and leverage the popular framework of VAR models [18], [25]. Neural networks in the architecture of MLP and LSTM units is used to model nonlinear dependency in the data [25]. In this paper, linear Granger causality models (i.e., BLR and Blasso) and neural Granger causality models (i.e., cMLP and cLSTM) are developed to explore their performance on detecting the dependency structure on both synthetic data and real-world log data.

III. Problem Modeling

In this section, we first describe the approach of modeling log data as multiple time series, and then mathematically define the Granger causality problem, one of the most widely used approaches for estimating causal relationships from time series.

A. Log Data Modeling

Benchmark Microservice Application. In this work, we use a publicly available benchmark TrainTicket microservice system[1] and deploy it on a Kubernetes cluster to collect log data. This system has about 41 microservices that 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 generate logs which may be normal or erroneous depending on how it is impacted by the fault in Figure 2(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.

Labeling 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., "HTTP 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 [7], [27], 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.

Binning logs as time series. We use different time bin sizes (10ms, 100ms, and 1sec) and count the number of error logs in each bin to obtain a time series of error counts corresponding to each impacted microservice (seen Figure 2(b)).

Our goal is to infer the causal graph among the microservices which have been impacted by the fault in order to help root cause analysis. Figure 2(c) shows a sample graph which explains how errors in one microservice are caused by errors in another microservice. In the following, we briefly provide the problem formulation of linear and neural Granger causality methods. Some important notations mentioned in this paper are summarized in Table I.

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

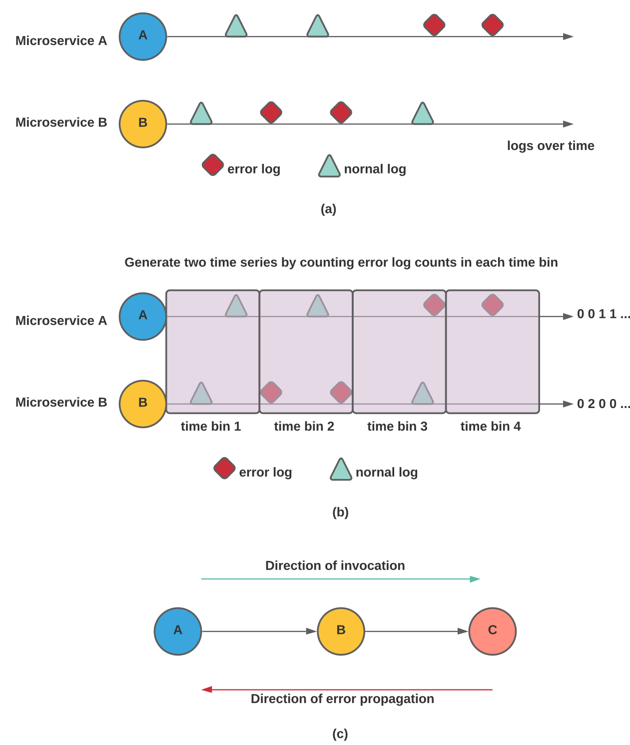

Fig. 2: (a): Log data after, (b) Modeling log data as multiple time series, and (c) A sample causal graph that we tend to infer which implies that errors in microservice A are caused by errors in microservice B are caused by errors in microservice C (i.e., root cause of errors is C).

B. Background and Temporal Causal Modeling with Granger Causality

Let $\mathbf{Y} = \{\mathbf{y}_i | 1 \leq i \leq K\}$ be a set of time series, where K is the number of time series in $\mathbf{Y}$ and $\mathbf{y}_i$ is the i^{th} time series. Assume $\mathbf{y}_{i,t} \in R$ to be the value of the i^{th} time series at time $t \in [0, T]$.

Definition 1 (Granger causality): Time series $\mathbf{y}_i$ "Granger-causes" time series $\mathbf{y}_j$, (or $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$) if and only if the regression for $\mathbf{y}_j$ using the past values of both $\mathbf{y}_j$ and $\mathbf{y}_i$ gains statistically significant improvement in terms of accuracy comparing with doing so with past values of $\mathbf{y}_j$ only.

1) Linear Granger Causality Modeling: The inference of Granger causality on time series data is typically studies using Vector Auto-Regression (VAR) model [18]. Let

$$\mathbf{y}_{\cdot,t} = (\mathbf{y}_{1,t}, ..., \mathbf{y}_{K,t})^\intercal \tag{1}$$

be a column vector containing the values of K time series at time t. Given the maximum time lag L, the VAR model is expressed as follows,

$$\mathbf{y}_{\cdot,t} = \sum_{l=1}^{L} (\mathbf{W}^l)^\intercal \mathbf{y}_{\cdot,t-l} + \epsilon, \tag{2}$$

TABLE I: Important Notations

Notation	Description
$\mathbf{Y}$	A set of time series.
K	The number of time series in $\mathbf{Y}$.
T	The length of time series.
L	The maximum time lag for VAR model.
s	The sparsity of the temporal dependency, denoted as the ratio of coefficients with zero value to K.
$\mathbf{y}_i$	The i^{th} time series.
$\mathbf{y}_{j,t}$	The value of j^{th} time series at time t.
$\mathbf{y}_{\bullet,t}$	A column vector containing the values of all time series at time t.
$\mathbf{x}_t$	A column vector built from all time series with time lag L at time t.
$\mathbf{W}^l$	The coefficient matrix for time lag l in VAR model.
$\mathbf{w}_j$	The coefficient vector used to predict j^{th} time series value in Bayesian Lasso model.
$\mathbf{w}_{j,t}$	The coefficient vector used to predict j^{th} time series value at time t in time-varying Bayesian Lasso model.
λ	The penalty parameters for $\mathbf{w}_j$.

where $\mathbf{W}^l$ is $K \times K$ coefficient matrix which specifies how lag l affects the future evolution of time series, and random noise ϵ is a $K \times 1$ vector. The nonzero value of $\mathbf{W}^l_{ij}$ indicates

$$\mathbf{y}_i \rightarrow_g \mathbf{y}_j.$$

A statistics test [2] is applied to determine the nonzero values in $\mathbf{W}^l$ over all lags based on the VAR model shown in Equation 2. However, the combinational explosion for the statistics test on time series pairs brings about its inefficiency for Granger causality inference, especially analyzing time series data with high dimension. This may be solved by adding a penalty item to a regression model for shrinking all $\mathbf{W}^l$ to zero. Specifically, the coefficient matrix $\mathbf{W}^l$ is obtained by minimizing the following objective function,

$$\min_{\{\mathbf{W}^l\}} \sum_{t=L+1}^{T} \parallel \mathbf{y}_{\bullet,t} - \sum_{l=1}^{L} (\mathbf{W}^l)^{\intercal} \mathbf{y}_{\bullet,t-l} \parallel_2^2 + \lambda \sum_{l=1}^{L} \parallel \mathbf{W}^l \parallel_F, \tag{3}$$

where $\| \cdot \|_F$ indicates L_F norm [2], and $\lambda > 0$ is the penalty parameter, which controls the sparsity level of the coefficient matrices. When $F = 2$, it represents L_2 norm, also known as ridge regression. While Lasso-Granger using L_1 norm (i.e., lasso regression) provides a more efficient and consistent way for addressing sparsity issue in high dimensional time series data.

To be simplified, we focus on the regression for one arbitrarily given variable $\mathbf{y}_j$, and the regression of other variables can be derived in a similar way. Let

$$\mathbf{x}_t = vec([\mathbf{y}_{\bullet,t-1}, \mathbf{y}_{\bullet,t-2}, ..., \mathbf{y}_{\bullet,t-L}]),$$

where $vec(\bullet)$ is an operator to convert a matrix into a vector by stacking column vectors. The L_F regression for the variable $\mathbf{y}_j$ is expressed as follows,

$$\min_{\mathbf{w}_j} \sum_{t=L+1}^{T} (\mathbf{y}_{j,t} - \mathbf{w}_j^{\intercal} \mathbf{x}_t)^2 + \lambda \parallel \mathbf{w}_j \parallel_F, \tag{4}$$

where $\mathbf{w}_j$ is the coefficient vector of the regression for the variable $\mathbf{y}_j$. Assuming $P = K * L$, both $\mathbf{x}_t$ and $\mathbf{w}_j$ are column vectors with the dimension $P \times 1$. Since Equation 4 provides only a linear autoregressive model with a penalty item for Granger causality problem, it is not suitable for nonlinear causal relations among time series in reality.

2) Nonlinear Granger Causality Modeling: In this section, we introduce a general formulation of the nonlinear autoregressive model [25]. Let us first define the nonlinear autoregressive function $f(\bullet)$ for Granger causality problem in time series analysis.

$$\mathbf{y}_{\bullet,t} = f(\mathbf{x}_t) + \epsilon, \tag{5}$$

where

$$\mathbf{x}_t = vec([\mathbf{y}_{\bullet,t-1}, \mathbf{y}_{\bullet,t-2}, ..., \mathbf{y}_{\bullet,t-L}])$$

and

$$\mathbf{y}_{\bullet,t} = (\mathbf{y}_{1,t}, ..., \mathbf{y}_{K,t})^{\intercal}.$$

For simplicity, Equation 6 can be written componentwise,

$$\mathbf{y}_{i,t} = f_i(\mathbf{x}_t) + \epsilon, \tag{6}$$

where the definition of function $f_i(\bullet)$ is how all time series with time lag L influence $\mathbf{y}_i$. According to Definition 1, if function $f_i(\bullet)$ depends on the past lags of $\mathbf{y}_j$, then $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$.

cMLP (Seen in Figure 3) and cLSTM are two nonlinear Granger causality models for time series data have been introduced in [25] using regularized neural networks (i.e., multilayer perceptrons (MLPs), LSTMs), which infer Granger causal relations using an optimization approach with sparse regularization similar to Equation 3.

IV. Methodology and Empirical Study

With the purpose of demonstrating the performance of linear and neural Granger causality models, we conduct extensive experiments over both synthetic and real log data for further studying other AIOps projects (e.g., fault localization). The evaluation on each data is started with a brief description of the data and the evaluation methods, and followed by the presentation of the comparative experimental results between the linear and neural (i.e., nonlinear) Granger causality methods. Finally, we explore the capability of these models on real-world log data for inferring the causal structure for microservices systems in order to facilitate root causes analysis.

A. Linear and Neural Granger Causality Methods

In this empirical study, we demonstrate the performance of both linear and neural (i.e., nonlinear) Granger causality

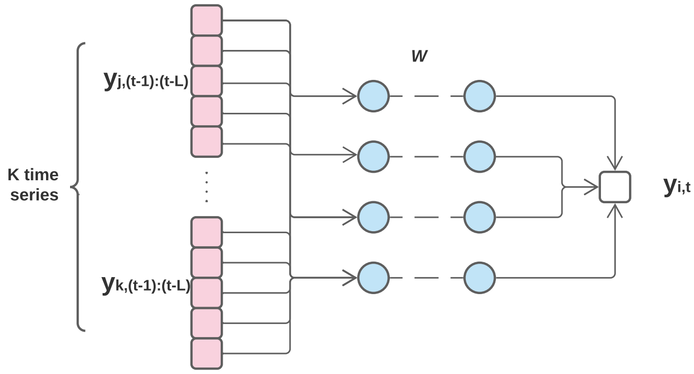

Fig. 3: Nonlinear Granger causality model with a multilayer perceptron (MLP). If the outgoing weights of time series j are penalized to zero, then time series j does not influence time series i.

algorithms including:

- `BLR`(q_0) [35]: It infers the temporal dependencies among time series using Bayesian Linear Regression with prior distribution $\mathcal{N}(\mathbf{0}, q_0^{-1}\mathbf{I}_d)$. It has been shown that the setting of the penalty parameter λ in ridge regression can be achieved by tuning q_0 accordingly [4].
- `BLasso`(λ) [35]: It applies Bayesian Lasso to learn the temporal dependencies, where λ is the L_1 penalty parameter [20]. It can effectively identify the sparse Granger casuality especially in high dimensions.
- `cMLP` [25]: It models the nonlinear Granger causality with a single multilayer perceptron layer. cMLPs is implemented using Adam optimizer and the activation function is ReLU (Rectified Linear Unit).
- `cLSTM` [25].: It uses LSTM architecture for Granger causality modeling which is well suited to modeling time series due to its capability of learning long-term dependencies. And it requires no maximum lag specification.

B. Evaluation Measures

`AUROC` Score: In order to verify the performance of the introduced methods for linear and nonlinear dependency detection, `AUROC`, the Area Under the ROC [5], is applied for performance evaluation due to its independence of priors, costs, and operating points [16]. The value of `AUROC` is the probability that the algorithm will assign a higher value to a randomly chosen existing edge than a randomly chosen non-existing edge in the temporal dependency structure. As we have mentioned in Section III, nonzero value of $\mathbf{W}_{ij}^l$ indicates $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$. It is reasonable to suppose that a higher absolute value of $\mathbf{W}_{ij}^l$ implies a larger likelihood of existing a causal dependency $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$. At each time t, an `AUROC` score of the algorithm is obtained by comparing its inferred temporal dependency structure with the ground truth at t.

C. Synthetic Data

The importance to conduct the empirical study over the synthetic data consists in the fact that the ground truth can be provided in advance so that the experimental conclusion can be shown by contrast.

Synthetic Data Generation: The time series data are generated with the linear VAR model and nonlinear Lorenz-96 model [13], where the coefficient value $\mathbf{W}_{ij}^l$ indicates the strength of dependency $\mathbf{y}_i \rightarrow_g \mathbf{y}_j$.

- **Linear VAR Data**: This synthetic data is generated with the VAR model. The coefficient holds a zero value, indicating no temporal dependency existing. The number of coefficient with zero value is determined by the sparsity s, which is the ratio of coefficients with zero value to the numbers of time series.
- **Lorenz-96 Data**: It is generated based on a nonlinear model of climate dynamics. The forcing constant F is used to determine the nonlinear level and chaos in the time series.

Performance Evaluation: We conduct the evaluation in terms of `AUROC` over four simulation VAR datasets, where $K \in \{5, 10\}$, $T \in \{1000, 10000\}$, $L = 1$, and $s = 0.2$, and four simulated nonlinear Lorenz datasets with $K = 5$, $F \in \{10, 40\}$ and $T \in \{500, 1000\}$. During our experiments, we employ grid search to find the optimal parameters for BLR and Blasso, and use the parameter settings (e.g., learning rate, regularization parameters, hierarchical group lasso penalty) from [25] for cMLP and cLSTM. In our experiments, 10 hidden units are configured for both single layer cMLP and cLSTM.

The performance evaluations on simulated linear VAR data are shown in Table II. Both linear and neural Granger causality methods perform well on this simulated linear datasets. As we expect, cMLP and cLSTM improve the their AUROC with larger T. And it is worth noting that cMLP performs better than cLSTM. Table III provides the comparison performance of nonlinear Lorenz-96 data using all mentioned models. As expected, neural Granger causality models outperform the linear Granger causality ones. The cMLP performs better than cLSTM when the simulated data is less chaos ($F = 10$). And with a large dataset ($T = 1000$), both the performance of cMLP and cLSTM have been improved.

TABLE II: `AUROC` comparisons of simulated VAR data with $s = 0.2$ on linear and neural Granger causality methods.

K T	5 1000	5 10000	10 1000	10 10000
BLR(1.0)	1.0	1.0	1.0	1.0
BLasso(1.0)	1.0	1.0	1.0	1.0
cMLP	0.92	**0.95**	0.62	0.64
cLSTM	0.52	0.78	0.63	0.73

TABLE III: AUROC comparisons of Lorenz-96 data with $K = 5$ on linear and neural Granger causality methods.

F	10	10	40	40
T	500	1000	500	1000
BLR(1.0)	0.75	0.75	0.48	0.48
BLasso(1.0)	0.78	0.75	$0.54_{(10.0)}$	$0.47_{(0.1)}$
cMLP	0.96	**0.98**	0.52	0.52
cLSTM	0.71	0.81	0.57	**0.57**

D. Dependencies Discovery in a Microservices System

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 microservices namely *ts-basic-service* is deleted from the system which results in four 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 time series, we experiment with 3 bin sizes: 10ms, 100ms, and 1000ms. In addition to models we mentioned above, another family of Granger causal discovery algorithms, namely Conditional independence (CI) testing, is introduced as well to infer the causal dependencies among the four impacted microservices. We compute precision, recall, and F_1 scores for all algorithms with respect to ground truth. Figure 4 shows the causal graphs inferred by one method with the best performance method in its algorithm family 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 relations that the algorithm fails to recover from log data. Table IV shows the results for all comparison approaches. We experiment with MMPC [26] in combination with two CI tests: partial correlation and RCoT [23]. We observe that MMPC yield better results in combination with **linear** CI tests (partial correlation) than with **nonlinear** CI tests such as RCoT. BLR(q_0) [4] and BLasso(λ) [20] are introduced as linear Granger causality models to infer the causal graph. In our experiments, the performance of the BLR (q_0) was not sensitive to q_0, so we set $q_0 = 1.0$. BLasso (λ) applies Bayesian Lasso to learn the causal graph, where λ weighs the L_1 penalty term. We also 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 practice, such a search procedure may be used to determine λ using training datasets that have associated ground truth information.

Methods	bin(ms)	Precision	Recall	F_1
MMPC (ParCorr)	10	0.45	1.0	0.62
MMPC (ParCorr)	10^2	0.8	0.8	**0.8**
MMPC (ParCorr)	10^3	1.0	0.4	0.57
MMPC (RCoT)	10	0.62	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
cMLP	10	1.0	0.60	**0.75**
cMLP	10^2	0.50	0.80	0.67
cMLP	10^3	0.56	1.0	0.71
cLSTM	10	0.45	1.0	0.62
cLSTM	10^2	0.41	1.0	0.59
cLSTM	10^3	0.38	1.0	0.56
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 IV: Performance results of different causal inference methods. PC models: MMPC with partial correlation and RCoT. Linear Granger causality models: BLR, BLasso. Neural Granger causality models: cMLP, cLSTM.

We also apply both neural Granger causality models (cMLP and cLSTM) with the optimal values of the regularization parameter λ on this dataset, which are tuned using ground truth information. We notice that cMLP obtains better results than cLSTM, which is consistent with the results on the simulated linear dataset, as well as MMPC models'. Furthermore, We observe that both bin size and model parameters play an important role. 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.

From all these experimental observations from both simulated time series data and real datasets, we could conclude that when K is small, cMLP is the best candidate since it performs well on both linear and nonlinear time series data even with a small T. cLSTM model works better to capture more complicated nonlinear dependencies. And BLR and BLasso methods have very good performance (e.g., accuracy, scalability and efficiency) on linear time series data when T is large enough.

V. Conclusion and Future Work

Industrial microservice systems always have hundreds to thousands of microservices and complex dependency rela-

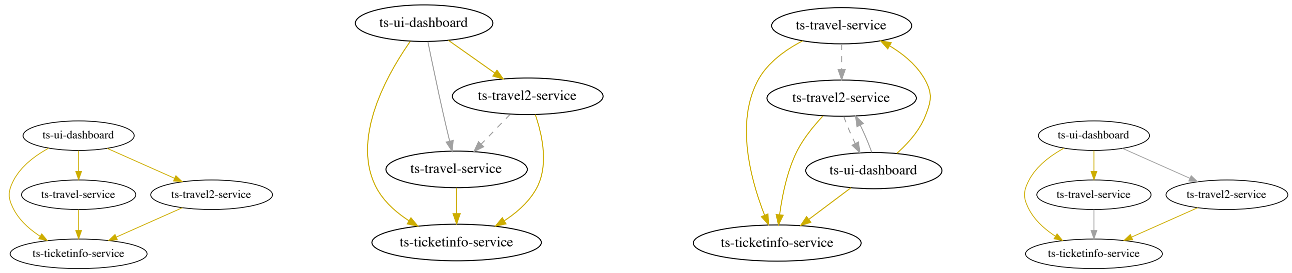

Fig. 4: Causal graphs inferred using different linear and nonlinear Granger methods. (left to right): Ground truth, MMPC(ParCorr, bin=100ms), BLR(1.0, bin=100ms), and cMLP(bin=10ms) model. False +ves (superfluous causal relationships) are marked by dashed gray edges and false -ves (missing causal relationships) with gray edges.

tionships among them, which poses an unique challenge for IT operations team timely determining the root causes and troubleshooting [1], [31]. Practically, many cloud applications suffer from limited observability and unknown topology making it very difficult to localize a fault. Thus, there is a need for particular machine learning models to identify the causal structure between microservices from observational data. In this work, we carefully study the performance of both linear and neural Granger causal techniques using just log data collected from a benchmark microservice system. Our experimental results clearly show that neural Granger causality models can accurately detect Granger causal relations in both linear and nonlinear settings. Linear Granger causal models are more efficient on large linear time series with high accuracy.

In the future work, we will extend our analysis to multiple datasets that involves different types of faults and a larger number of microservices for the fault localization problem. The sensitivity of algorithms to both length and granularity of time series and effects of log label noise in timing information of logs are needed to be studied as well.

References

[1] P. Aggarwal, A. Gupta, P. Mohapatra, S. Nagar, A. Mandal, Q. Wang, and A. Paradkar. Localization of operational faults in cloud applications by mining causal dependencies in logs using golden signals. In *International Conference on Service-Oriented Computing*, pages 137–149. Springer, 2020.

[2] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 66–75. ACM, 2007.

[3] V. Arya, K. Shanmugam, P. Aggarwal, Q. Wang, P. Mohapatra, and S. Nagar. Evaluation of causal inference techniques for aiops. In *8th ACM IKDD CODS and 26th COMAD*, pages 188–192. 2021.

[4] C. M. Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[5] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[6] D. Cheng, M. T. Bahadori, and Y. Liu. Fblg: A simple and effective approach for temporal dependence discovery from time series data. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 382–391. ACM, 2014.

[7] M. Du, F. Li, G. Zheng, and V. Srikumar. 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, page 1285–1298, New York, NY, USA, 2017. Association for Computing Machinery.

[8] Google. Sre book. https://landing.google.com/sre/sre-book/chapters/effective-troubleshooting/, 2020. [ONLINE].

[9] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[10] J. Hwang, L. Shwartz, Q. Wang, R. Batta, H. Kumar, and M. Nidd. Fixme: Enhance software reliability with hybrid approaches in cloud. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 228–237. IEEE, 2021.

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

[12] T. Jia, P. Chen, L. Yang, Y. Li, F. Meng, and J. Xu. An approach for anomaly diagnosis based on hybrid graph model with logs for distributed services. In *2017 IEEE International Conference on Web Services (ICWS)*, pages 25–32. IEEE, 2017.

[13] A. Karimi and M. R. Paul. Extensive chaos in the lorenz-96 model. *Chaos: An interdisciplinary journal of nonlinear science*, 20(4):043105, 2010.

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

[15] S. Kobayashi, K. Otomo, K. Fukuda, and H. Esaki. Mining causality of network events in log data. *IEEE Transactions on Network and Service Management*, 15(1):53–67, 2018.

[16] T. Landgrebe and R. Duin. A simplified extension of the area under the roc to the multiclass domain. In *Seventeenth annual symposium of the pattern recognition association of South Africa*, pages 241–245, 2006.

[17] C. Liang, T. Benson, P. Kanuparthy, and Y. He. Finding needles in the haystack: Harnessing syslogs for data center management, 2016.

[18] H. Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[19] L. Mariani, C. Monni, M. Pezzé, O. Riganelli, and R. Xin. Localizing faults in cloud systems. In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*, pages 262–273, 2018.

[20] T. Park and G. Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[21] S. Paskin. https://www.bmc.com/blogs/what-is-aiops/, 2020.

[22] J. Pearl. *Causality*. Cambridge university press, 2009.

[23] E. V. Strobl, K. Zhang, and S. Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery, 2017.

[24] Y. Sun, K. Xie, N. Liu, S. Yan, B. Zhang, and Z. Chen. Causal relation of queries from temporal logs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 1141–1142. ACM, 2007.

[25] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox. Neural granger causality for nonlinear time series. *arXiv preprint arXiv:1802.05842*, 2018.

[26] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

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

[28] Q. Wang. Intelligent data mining techniques for automatic service management. In *FIU Electronic Theses and Dissertations, https://digitalcommons.fiu.edu/etd/3883*. FIU, 2018.

[29] Q. Wang, T. Li, S. Iyengar, L. Shwartz, and G. Y. Grabarnik. Online it ticket automation recommendation using hierarchical multi-armed bandit algorithms. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 657–665. SIAM, 2018.

[30] Q. Wang, L. Shwartz, G. Y. Grabarnik, M. Nidd, and J. Hwang. Leveraging ai in service automation modeling: From classical ai through deep learning to combination models. In *International Conference on Service-Oriented Computing*, pages 186–201. Springer, 2019.

[31] Q. Wang, C. Zeng, S. Iyengar, T. Li, L. Shwartz, and G. Y. Grabarnik. Aistar: an intelligent system for online it ticket automation recommendation. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1875–1884. IEEE, 2018.

[32] Q. Wang, W. Zhou, C. Zeng, T. Li, L. Shwartz, and G. Y. Grabarnik. Constructing the knowledge base for cognitive it service management. In *2017 IEEE International Conference on Services Computing (SCC)*, pages 410–417. IEEE, 2017.

[33] C. Zeng, T. Li, L. Shwartz, and G. Y. Grabarnik. Hierarchical multi-label classification over ticket data using contextual loss. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–8. IEEE, 2014.

[34] C. Zeng, L. Tang, T. Li, L. Shwartz, and G. Y. Grabarnik. Mining temporal lag from fluctuating events for correlation and root cause analysis. In *Network and Service Management (CNSM), 2014 10th International Conference on*, pages 19–27. IEEE, 2014.

[35] C. Zeng, Q. Wang, W. Wang, T. Li, and L. Shwartz. Online inference for time-varying temporal dependency discovery from time series. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1281–1290. IEEE, 2016.

[36] C. Zeng, W. Zhou, T. Li, L. Shwartz, and G. Y. Grabarnik. Knowledge guided hierarchical multi-label classification over ticket data. *IEEE Transactions on Network and Service Management*, 14(2):246–260, 2017.

[37] W. Zhou, W. Xue, R. Baral, Q. Wang, C. Zeng, T. Li, J. Xu, Z. Liu, L. Shwartz, and G. Ya. Grabarnik. Star: A system for ticket analysis and resolution. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2181–2190, 2017.

[38] X. Zhou, X. Peng, T. Xie, J. Sun, C. Ji, W. Li, and D. Ding. Fault analysis and debugging of microservice systems: Industrial survey, benchmark system, and empirical study. *IEEE Transactions on Software Engineering*, 2018.

[39] X. Zhou, X. Peng, T. Xie, J. Sun, C. Xu, C. Ji, and W. Zhao. 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*, pages 323–324. ACM, 2018.