Granger Causality for Time-Series Anomaly Detection

Huida Qiu, Yan Liu
Computer Science Department
University of Southern California
Los Angeles, CA 90089
Email: {huidaqiu, yanliu.cs}@usc.edu

Niranjan A Subrahmanya, Weichang Li
ExxonMobil Corporate Strategic Research
Annandale, NJ 08801
Email: {niranjan.a.subrahmanya, weichang.li}@exxonmobil.com

***Abstract*—Recent developments in industrial systems provide us with a large amount of time series data from sensors, logs, system settings and physical measurements, etc. These data are extremely valuable for providing insights about the complex systems and could be used to detect anomalies at early stages. However, the special characteristics of these time series data, such as high dimensions and complex dependencies between variables, as well as its massive volume, pose great challenges to existing anomaly detection algorithms. In this paper, we propose Granger graphical models as an effective and scalable approach for anomaly detection whose results can be readily interpreted. Specifically, Granger graphical models are a family of graphical models that exploit the temporal dependencies between variables by applying L_1-regularized learning to Granger causality. Our goal is to efficiently compute a robust "correlation anomaly" score for each variable via Granger graphical models that can provide insights on the possible reasons of anomalies. We evaluate the effectiveness of our proposed algorithms on both synthetic and application datasets. The results show the proposed algorithm achieves significantly better performance than other baseline algorithms and is scalable for large-scale applications.**

***Keywords*-Time Series Analysis, Anomaly Detection**

I. Introduction

Modern production systems in many industries are characterized by a high degree of complexities and uncertainties at multiple scales. As a result, it becomes challenging to detect the onset and analyze the cause of anomalies in these production processes, which, if undetected, could lead to severe economic, environmental and safety risks. Meanwhile, recent development in these industrial systems have provided us with a large amount of data from sensors, logs, system settings, physical measurements and so on. These data are extremely valuable for providing insights about the complex systems and could be useful to detect production anomalies at early stages.

The common practice for anomaly detection is to build a statistical model that captures the generation process of the normal data, then estimate the likelihood of a new observation based on this model and predict the data as an "anomaly" if the likelihood is below some threshold [5]. In addition, a number of models have also been developed to specifically handle time-series data. For instance, [14] studies change detection from a sequence of graphs based on a clustering technique for graph nodes, [15] addresses the task by change point detection in multivariate time series, and [11] utilizes discretization techniques to detect subsequences in long time series. These algorithms have been applied successfully in some applications of anomaly detection. However, several issues have not been fully addressed: (1) most models are very sensitive (to the observed data), which results in a high false alarm rate; (2) detection metrics such as a likelihood score is a single number that provides limited insights about the anomaly, while identifying the possible roots for anomalies is also important;(3) scalability is an important aspect, especially given high-dimensional data. Therefore, it is essential for us to pursue a robust, interpretable, yet efficient approach for time-series anomaly detection.

There usually exist two types of anomalies in multivariate time-series data: one type is the anomaly that occurs only within individual variables, referred to as "univariate anomaly"; the other type is the anomaly that occurs due to changes of temporal dependencies, referred to as "dependency anomaly". Univariate anomaly has been extensively studied [11], [15], while dependency anomaly is much more challenging to detect but common in the real applications. In this paper, we propose to investigate Granger graphical models, which uncover the temporal dependencies between variables in multivariate time-series data, as a novel and effective approach to detect dependency anomalies in time series data. In addition, we investigate several parallel implementations of our algorithms so that they are applicable to large scale applications. The experiments on one synthetic dataset and two application datasets demonstrate that our proposed model has the significant advantages on robustness, interpretability and scalability over existing approaches in time series anomaly detection.

II. Granger Graphical Models for Anomaly Detection

In this section, we first review Granger-lasso algorithm for uncovering temporal dependencies between time series, then describe in detail our proposed method via Granger graphical models.

A. Granger-Lasso Algorithm

A number of approaches have been investigated to define correlations or causalities for time series. Among them, Granger causality has gained tremendous success across many domains due to its simplicity, robustness, and extendibility [7], [4]. Most existing algorithms for detecting Granger causality are based on statistical significance tests, which is time-consuming and sensitive to the number of observations in the autoregression.

In [1], the authors developed novel temporal dependency learning algorithms based on penalized regression to achieve neighborhood sparsity and demonstrated its effectiveness on both simulation data and application data. Given multivariate time series observations $\{X_i^t : t = 1, \ldots, T; i = 1, \ldots, p\}$, where T is the length of time series and p is the dimension, let $X_{t,L}^{Lagged}$ represent the concatenated vector of all the lagged variables (with a maximal lag of L) of up to time t, i.e. $\{x_j^{t-l} : j = 1, \ldots, p, l = 1, \ldots, L\}$. We can determine the temporal dependence between time series i and others by the following regularized regression:

$$\hat{\beta}_{\mathbf{i}}(\lambda) = \arg\min_{\beta_i}(\sum_{t=1}^{T} \|x_i^t - X_{t,L}^{Lagged}\beta_i\|^2 + \lambda\|\beta_{\mathbf{i}}\|_1), \quad (1)$$

where X_j Granger causes X_i if and only if at least one of the corresponding coefficients of X_j is non-zero. The major contribution of their work is to introduce L_1 regularized regression as an efficient method to recover the sparse Ganger causality relations from high-dimensional time series.

B. Granger Graphical Models for Anomaly Detection

Before delving into the details of our proposed algorithm, we first formally define the task of anomaly detection for multivariate time-series data: given p number of time series, X_1,...,X_p, we aim to find data points (indexed by time stamp) that significantly deviate from the normal pattern of the data sequence. Without loss of generality, we can transform the task into the following formulation: given two data sequences

$$\mathcal{D}^{(a)} = \{x_i^{(a),t} | i = 1, \ldots, p,\ t = 1, \ldots, T^{(a)}\},$$
$$\mathcal{D}^{(b)} = \{x_i^{(b),t} | i = 1, \ldots, p,\ t = 1, \ldots, T^{(b)}\}, \quad (2)$$

where $\mathcal{D}^{(a)}$ is the reference set, and $\mathcal{D}^{(b)}$ is the test set. We are interested in computing the anomaly score for $\mathcal{D}^{(b)}$ with respect to $\mathcal{D}^{(a)}$ for each variable, representing whether and how much each variable contributes to the difference between the two data sets. Notice that the length of the time-series $T^{(a)}$ and $T^{(b)}$ could be different, but the number of features in $\mathcal{D}^{(a)}$ and $\mathcal{D}^{(b)}$ must be the same. Theoretically, the longer the training time-series the better we can detect anomalies in the test data. However, as will be shown in our experiments, as long as $T^{(a)}$ and $T^{(b)}$ are comparable, our methods can yield reasonable results.

Univariate anomaly detection using Granger graphical models is a simple step: For observation of X_i in the test set $\mathcal{D}^{(b)}$, we can evaluate its likelihood based on the Granger graphical models built from the reference set $\mathcal{D}^{(a)}$ (since we have the assumption that the temporal dependencies remain the same). However, detecting temporal dependency anomaly is much more difficult. We propose an anomaly detection algorithm based on regularization, namely regularization-based Granger graphical models (GGM) algorithm, to solve the problem. The GGM algorithm consists of three steps: first, learning temporal causal graph of $\mathcal{D}^{(b)}$ by regularization; second, computing the anomaly scores of $\mathcal{D}^{(b)}$ using KL-divergence; third, determining anomaly by threshold cutoff and identifying potential causes. Next, we describe the algorithm in detail:

Step 1. Learning temporal causal graphs by regularization. Notice that the null hypothesis of dependency anomaly is that the temporal causal graphs of reference set $\mathcal{D}^{(a)}$ and test set $\mathcal{D}^{(b)}$ are the same. When learning the temporal graphs for $\mathcal{D}^{(b)}$, we can use the null hypothesis as an additional constraint in the optimization algorithm outlined in Granger-lasso algorithm. There are several possibilities to achieve this; we investigate two kinds of regularization terms and examine their effectiveness:

Neighborhood similarity: Given the reference set $\mathcal{D}^{(a)}$, we first learn the temporal dependency graph using eq(1). For test set $\mathcal{D}^{(b)}$, we can add additional regularization terms to impose the constraint that the values of $\beta^{(b)}$ should be zero (or nonzero) when the corresponding values of $\beta^{(a)}$ is zero (or nonzero). In other words, we can solve the following optimization problem to get the temporal graphs for $\mathcal{D}^{(b)}$:

$$\hat{\beta}_{\mathbf{i}}^{(b)}(\lambda) = \arg\min_{\beta_i^{(b)}}(\sum_{t=1}^{T} \|x_i^{(b),t} - X_{t,L}^{(b),Lagged}\beta_i^{(b)}\|^2), \quad (3)$$
$$\text{subject to } \sum_{j \in I_0} |\beta_{i,j}^{(b)}| \leq \epsilon_0,\ \sum_{j \in I_1} |\beta_{i,j}^{(b)}| \leq \epsilon_1,$$

where $I_0 = \{p : \beta_{i,p}^{(a)} = 0\}$ is the index set of zero entries in $\beta_i^{(a)}$ and $I_1 = \{p : \beta_{i,p}^{(a)} \neq 0\}$ is the index set of nonzero entries in $\beta_i^{(a)}$, ϵ_0 and ϵ_1 are constants. Intuitively, ϵ_0 should be significantly smaller than ϵ_1.

Coefficient similarity: Under the null hypothesis, another type of constraint we can put in the optimization function is that the values of the coefficients $\beta^{(b)}$ should be similar to those of $\beta^{(a)}$. Therefore we have the following optimization formulation to learn the temporal graphs for $\mathcal{D}^{(b)}$:

$$\hat{\beta}_{\mathbf{i}}^{(b)}(\lambda) = \arg\min_{\beta_i^{(b)}}(\sum_{t=1}^{T} \|x_i^{(b),t} - X_{t,L}^{(b),Lagged}\beta_i^{(b)}\|^2), \quad (4)$$
$$\text{subject to } \sum_{j} |\beta_{i,j}^{(b)} - \beta_{i,j}^{(a)}| \leq \epsilon,$$

where ϵ is a constant. We will discuss efficient solutions to eq(3, 4) in the next section.

Step 2. Computing anomaly scores. From information theoretic perspectives, the most natural difference measure between two distributions is the Kullback-Leibler (KL)

divergence. For a particular time-series (or feature) X_i, we can define its anomaly score as follows:

$$d_i^{\mathrm{ab}} \equiv \int \mathrm{d}x_i \, p_{(\mathrm{a})}(x_i|\boldsymbol{x}_L^{lagged}) \ln \frac{p_{(\mathrm{a})}(x_i|\boldsymbol{x}_L^{lagged})}{p_{(\mathrm{b})}(x_i|\boldsymbol{x}_L^{lagged})}, \quad (5)$$

where $p_{(\mathrm{a})}$ and $p_{(\mathrm{b})}$ are the underlying point process distributions with parameters estimated from $\mathcal{D}^{(a)}$ and $\mathcal{D}^{(b)}$ respectively (by maximum likelihood given the learned temporal graphs and underlying distribution over the graphs). If the underlying distribution is linear Gaussian models, i.e. $x_i^t|x_{t,L}^{Lagged} \sim N(x_{t,L}^{Lagged}\beta_i, \sigma^2)$, we can rewrite the anomaly score as follows:

$$\begin{aligned} d_i^{\mathrm{ab}} &= -(\ln|\sigma^{(a)}| - \ln|\sigma^{(b)}|) \\ &\quad - \frac{1}{2}(1 - \frac{1}{(\sigma^{(b)})^2}((\sigma^{(a)})^2 + ((\beta_i^{(a)} - \beta_i^{(b)})\Delta)^2)), \quad (6) \end{aligned}$$

where $(\beta_i^{(a)}, (\sigma^{(a)})^2)$ and $(\beta_i^{(b)}, (\sigma^{(b)})^2)$ are the coefficients and variance for $\mathcal{D}^{(a)}$ and $\mathcal{D}^{(b)}$ respectively, and Δ is the mean of the concatenate lagged variables.

We can further define d_i^{ba} by reversing a and b in eq(5). Notice that d_i^{ab} and d_i^{ba} are quantities that measure the change in the neighborhood graph of the i-th node. The greater these quantities are, the greater change we have concerning X_i. Thus, given the assumption of neighborhood preservation, it is reasonable to define the anomaly score of the i-th variable as

$$a_i \equiv \max\{d_i^{\mathrm{ab}}, \, d_i^{\mathrm{ba}}\} \quad (7)$$

This definition naturally extends the one proposed in [10]. One of the drawbacks of that approach is that it simply uses the k-NN strategy for neighborhood selection. Also, due to a heuristic definition of the dissimilarity, it cannot detect anomalies caused by sign changes such as $x_i \to -x_i$. This information-theoretic definition of anomaly score can in principle detect any type of anomaly affecting the probability distribution.

In order to give insights to the time period during which an anomaly occurs, we can slide a window with a fixed length through the test series and compute anomaly scores for observations within this window. In this way we can not only determine if the whole test set is anomalous, but also provide details about anomalies within each time period. The length of the window depends on the balance between detection delay and detection precision. A short window length could lead to a shorter delay, while due to less information provided in smaller window, it could make our method less sensitive to anomalies.

Step 3. Determine anomaly by threshold cutoff and identifying potential causes. Once we have an anomaly score for a variable in test data, we need a threshold to decide whether the current observation should be considered as an anomaly. To obtain a threshold, we resort to the reference data and measure how much the normal time-series would score under our definition of anomaly score. Specifically, we slide the same window through the reference data and calculate the anomaly scores for each window. We use these scores to approximate the distribution of the anomaly scores which a normal time-series should have. Given a significance level α, we use the α-quantile of this distribution as the threshold cutoff to determine anomalousness of the variable in test data. If a time-series is found to be anomalous, we can discover the dependency changes by comparing the temporal dependency graphs of the reference data and test data or by directly examining the changes in the learned coefficients to identify potential causes for the anomaly.

Note that although our framework is introduced in a static setting, it can also be adapted to a streaming fashion. We can use two concatenated windows to slide through a stream of time-series, with the leading window as the test data and the tailing window as the reference data. When a new observation arrives, the two windows move forward for one time stamp. We use the previous coefficients to initialize the iterative optimization algorithms to learn a new set of coefficients. Based on the fact that anomalies are usually rare, most of the time the coefficients at two consecutive time stamps should't vary too much, which ensures that the optimization algorithms can usually converge very fast.

By now we have outlined the general framework of our Granger graphical models for time-series anomaly detection. Next we will investigate efficient solutions to the associated optimization problems in our method.

III. Stochastic & Parallel Anomaly Detection

In this section, we investigate stochastic optimization algorithms as accelerated solvers to penalized regression, which is the most computationally expensive part of our algorithm. We briefly introduce the candidate algorithms here, and report their efficiency results in Section IV-D.

1. Stochastic Subgradient Descent (*SGD*) randomly chooses a data point at each iteration and updates the coefficient β using the subgradient on the current point and a step size η_t satisfying the following constraint: $\eta_t \geq 0, \sum_{t=1}^{\infty} \eta_t = \infty, \sum_{t=1}^{\infty} \eta_t^2 = \delta < \infty$. We keep track of the lowest objective function in the updating process as well as the corresponding coefficients. It has been shown that SGD can almost surely converge to the optimal solution [2].

2. Stochastic Coordinate Descent (*SCD*) [13] is adapted from the shooting method. Instead of updating all coordinates, *SCD* uniformly picks only one coordinate in each iteration. The expectation of the objective function converge linearly to the optimal value in terms of iteration times.

3. Stochastic Subgradient Langevin Dynamics (*SGLD*) [16] combines mini-batch stochastic subgradient descent and Langevin Dynamics. At each iteration, besides the standard stochastic subgradient descent, a Gaussian noise $\epsilon_t = N(0, \eta_t)$ is added to the coefficients. The algorithm first goes through a stochastic optimization stage, then enters the Langevin Dynamics phase which generates samples approximating the posterior distribution of the coefficients.

Table I
LIST OF ANOMALY DETECTION METHODS

Method	Description
PCA-T2	PCA projection with T^2 statistic
PCA-Q	PCA projection with Q statistic
DPCA-T2	Dynamic PCA projection with T^2 statistic
DPCA-Q	Dynamic PCA projection with Q statistic
RVL	Relevance vector learning method
DI	Modified distance measure
GGM-N	Our method using neighborhood similarity
GGM-C	Our method using coefficient similarity

4. Parallel Stochastic Coordinate Descent (*Shotgun*) [3] is a parallel implementation of *SCD*. Specifically, at each update, a random subset of P coordinates is selected, then P processors update each coordinate individually. Compared to *SCD*, at each iteration P coordinates are updated instead of one. *Shotgun* can also achieve linear convergence.

5. Parallel Stochastic Gradient Descent (*PSGD*) [12] randomly partitions the data, giving one partition to each processor, which sequentially uses each data point of its own partition to update β using a constant step size η. The coefficients from all processors are averaged as the final result.

Results on the efficiency of these algorithms are reported in Section IV-D.

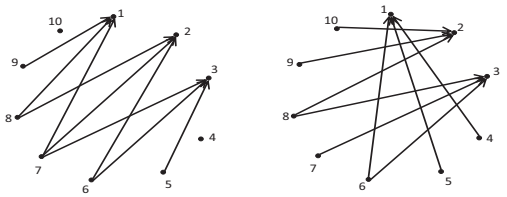
IV. EXPERIMENTS

In this section, we examine the effectiveness and efficiency of our proposed algorithms for time series anomaly detection on both synthetic and application datasets. For effectiveness, we compare our method with six state-of-art baseline methods, including PCA projection with T^2 statistic and Q statistic [8], dynamic PCA projection with T^2 statistic and Q statistic [8], relevance vector learning method [9], and modified distance measure [6]. For efficiency, we investigate five different lasso-type solvers as introduced in Section III for their suitability in our anomaly detection framework.

A. Dataset Description

The synthetic data is generated from vector auto-regression models with two coefficients C_1 and C_2 over 10 features and a lag of 3. It consists of 300 observations, in which the first 200 observations are generated from C_1 and the remaining 100 from C_2 (i.e., an anomaly occurs at time 201). The temporal dependence graphs of C_1 and C_2 are shown in Figure 1 (there are edges from node 1 to all other nodes, which are omitted in Figure 1). Notice that only the dependency structures for variables 1, 2 and 3 are changed. All coefficient value remains the same for C_1 and C_2 except those involved in the change of temporal dependence.

Figure 1. Temporal dependence graphs for synthetic data



(a) Dependence Graph for C_1 (b) Dependence Graph for C_2

We use the Tennessee Eastman process (*TEP*) data as our application dataset, which contains 52 time series representing measurements of temperature, pressures, flow rates, etc in a chemical process. The training data is free of anomalies with 500 observations. There are 21 preprogrammed faults in the *TE* process, giving us 21 test data sets. Each of them has 960 observations measured at an interval of 3 minutes, of which the first 160 are free of anomalies, and one of the 21 faults is introduced from time 161 to the end of observation. More details about the data can be found in [8]. In our experiment, we choose three most difficult data sets (as noted in [8]), with fault 4, fault 5 and fault 11 respectively, to demonstrate the effectiveness of our methods.

B. Results on Synthetic Data Set

In this experiment, the first 100 observations of the synthetic data set are used as reference data, and the remaining 200 as test data. The first half of the test data is normal while the remaining is anomalous. Table I lists all the anomaly detection methods used in the experiments. For our methods and *DI*, we use a window of size 10 to slide through the test data. For the PCA based methods, the number of principal components is determined using the smallest number needed to explain 95% of the total variance. A significance level of 95% is used for all significance test based methods.

Table II reports the recall (proportion of correctly detected alarms out of all true alarms), precision (proportion of correctly detected alarms out of all detected alarms) and F_1 value (defined as $F_1 = 2 \cdot \text{Recall} \cdot \text{Precision}/(\text{Recall} + \text{Precision})$) of all eight methods. Note that for *RVL*, *DI*, *GGM-N* and *GGM-C*, each variable has its own recall and precision rate, and we report their averaged rates. From the table we can see the significant advantages of our methods. Regarding recall rate, both *GGM-N* and *GGM-C* score very close to the highest one achieved by *DI*. Though having the highest recall rate, *DI* has a much lower precision than our methods. In terms of precision, *PCA-Q* and *DPCA-Q* achieve 100% while our performance of 99% is also competent, while the low recall rate of *PCA-Q* and *DPCA-Q* make them much less reliable. The low recall rates of *PCA*-based methods also indicate that anomalies cannot always be captured in the subspace where variance is maximized. The F_1 value indicates that our methods have the highest overall quality. Note that, with finely tuned parameters, *GGM-N*

Table II
COMPARISON OF OVERALL ANOMALY DETECTION PERFORMANCE ON SYNTHETIC DATASET

Method	Recall	Precision	F-value
PCA-T2	0%	100%	0.00
PCA-Q	31%	**100%**	0.47
DPCA-T2	0%	100%	0.00
DPCA-Q	32%	**100%**	0.48
RVL	46%	65%	0.54
DI	**92%**	73%	0.81
GGM-N	89%	99%	**0.94**
GGM-C	89%	99%	**0.94**

and *GGM-C* perform almost the same, with only slightly different values in some anomaly scores.

Figure 3(a) through 3(c) give details of recall, precision and F_1 scores for each of the 10 features by *RVL*, *DI*, *GGM-N* and *GGM-C*. As we can see, *RVL* achieves very high recall and precision rates for features $1-3$, whose temporal dependencies are changed after the anomaly occurs. But for the remaining variables, since their coefficients and noise scale remain the same, simply using a regression model is not powerful enough to detect anomalies. *DI* generally performs best in terms of recall rates followed by our methods, while our methods demonstrates great advantage on precision. For *DI* method, changes in the mean, deviation and distribution all contribute similar amounts to the total score, while changes in the mean and deviation are also reflected in the distribution change, which makes *DI* very sensitive to fluctuation.

C. Results on TEP Simulation Data

In this experiment, we test all eight methods on the three *TEP* data sets with fault 4, fault 5 and fault 11 respectively. Note that when we calculate the recall and precision scores for *RVL*, *DI*, *GGM-N* and *GGM-C* we only look at those most affected variables in each fault, which are variable 51 for fault 4 and fault 11, and variable 22 for fault 5. A window size of 10 is used for the window-based methods.

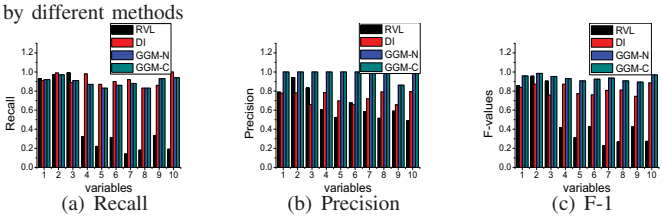
Table III reports the recall, precision and F_1 value. Similar to the previous experiment, the four *PCA*-based methods performs reasonably well in terms of precision, while quite poor on recall. These methods usually detect anomaly shortly after it occurs, but after some time the anomaly scores would drop below their thresholds, yielding low recall rates.

Fault 5 only causes a slight disturbance to variable 22, which goes back to almost the same behavior as before, making it difficult to persistently detecting fault after it takes place. Consequently, the recall rates for *RVL*, *DI*, *GGM-N* and *GGM-C* are lower than in the other two faults. The advantage of our methods lies in the fact that they can capture changes in dependencies between variables. Even though variable 22 appears to return to its normal behavior some time after the fault occurs, the dependence between this variable and the others are different from before, which enables our methods to significantly outperform the others

Table IV
PREDICTED RANK OF THE TRUE FEATURES RESPONSIBLE FOR FAULT 4, 5 AND 11 ON TEP DATASET

	Fault 4		Fault 5		Fault 11	
Method	**0-15h**	**15-40h**	**0-15h**	**15-40h**	**0-15h**	**15-40h**
PCA-CONT	11	9	26	33	9	9
PCA-RES	1	1	6	18	1	1
DPCA-CONT	10	9	32	34	12	13
DPCA-RES	1	1	6	17	1	1
RVL	1	1	12	37	1	1
DI	1	1	11	5	1	1
GGM-N	2	1	2	4	2	2
GGM-C	2	1	2	4	2	2

Figure 2. Detailed comparison of anomaly detection performance on synthetic dataset: recall, precision and F_1 scores on individual features by different methods



(a) Recall (b) Precision (c) F-1

in this case. Overall, our proposed methods achieve the best performance in detecting all three faults.

We also examine the performance of of our method in identifying the causes of anomalies. In the TEP dataset, we have the ground truth on which features are most closely related to each fault, i.e., feature 51 for fault 4 and 11, and feature 22 for fault 5. Table IV lists the rank of these features by different methods based on their predicted rank list for features responsible for the anomaly. For the *PCA*-based methods, variables are ranked-based on the *CONT* or *RES* measures [8]. For *RVL*, *DI*, *GGM-N* and *GGM-C* the ranking is based on how much by percentage the anomaly scores exceed their thresholds. We divide the time after the anomaly to two parts, 0 to 15 hours and 15 to 40 hours after the fault, which correspond to the time period shortly after the fault occurs and the period when some time has passed since the fault. It is obvious from Table IV that all the methods perform comparatively better for fault 4 and 11 than for fault 5. For fault 5, despite the variable's tendency of returning to normal after the introduction of the fault, the rankings based on our methods are persistently higher than the other methods in both time periods, which further proves the persistence of our fault detection performance.

D. Results on Scalability Analysis

In this experiment, we examine how our methods can be accelerated for large-scale applications. As discussed in Section III, we examine six algorithms to efficiently solve the associated optimization problems, including *SGD*, *SCD*, *SGLD*, *Shotgun* and *PSGD*. We test these algorithms on the *TEP* dataset and evaluate their performance on the anomaly detection accuracy given the same response time. Figure 3 plots the objective function of eq(3) in terms of the number of iterations and the runtime on the reference set of the *TEP* data. For *PSGD*, the number of data points distributed to each processor is counted as the number of iterations. All methods converge pretty fast to the minimum value. *Shotgun* shows some improvements over *SCD* by parallelization, and *SGLD* has almost the same performance as the basic *SGD*. Overall *PSGD* converges fastest both in terms of iteration and time. However, despite the fast convergence speed, we observe that its objective function value fluctuates a lot more than the other methods. Generally speaking, *PSGD* achieves faster convergence at the cost of incurring more noise.

Figure 4 reports the accuracy of anomaly detection using the five stochastic lasso solvers on *TEP* data with Fault

Table III
COMPARISON OF ANOMALY DETECTION PERFORMANCE ON TEP DATASET: RECALL, PRECISION AND F_1 SCORE FOR FAULT 4, 5 AND 11.

	Fault 4			Fault 5			Fault 11		
Method	**Recall**	**Precision**	F**-1**	**Recall**	**Precision**	F**-1**	**Recall**	**Precision**	F**-1**
PCA-T2	12.1%	88.2%	0.21	32.9%	95.3%	0.49	23.5%	91.7%	0.37
PCA-Q	8.9%	87.7%	0.16	29.1%	**95.9%**	0.45	8.6%	85.2%	0.16
DPCA-T2	14.0%	86.2%	0.24	34.5%	93.9%	0.5	22.9%	85.9%	0.36
DPCA-Q	8.6%	86.3%	0.16	28.4%	95.4%	0.44	8.1%	87.8%	0.15
RVL	**100%**	97%	0.98	22.6%	89.6%	0.36	83.9%	96.4%	0.90
DI	**100%**	95%	0.97	59.4%	89.8%	0.71	98.3%	95.4%	0.97
GGM-N	**100%**	98.8%	**0.99**	69.3%	90.1%	0.78	99.1%	**97.3%**	**0.98**
GGM-C	**100%**	**98.9%**	**0.99**	**70.5%**	89.5%	**0.79**	**99.3%**	**97.3%**	**0.98**

Figure 3. Scalability analysis on TEP dataset: the value of objective function over iterations (left) and over runtime (right).

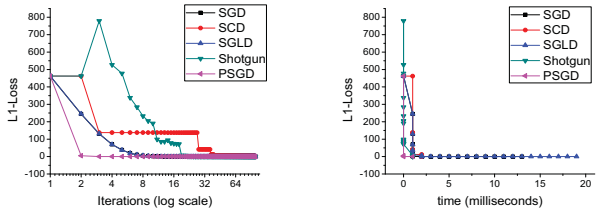
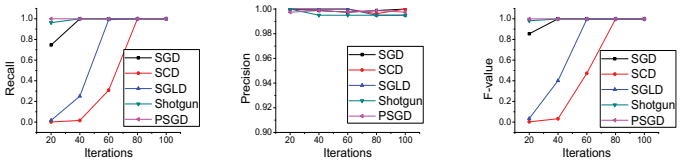


Figure 4. Scalability Analysis: anomaly detection performance of each lasso solvers over the number of search iterations.



(a) Recall (b) Precision (c) F-value

4. When the number of iterations is small, *SGD*, *SGLD* and *SCD* are still a long way from convergence. Their anomaly thresholds stay above their anomaly scores at most points, allowing only a few anomalous points to exceed their thresholds, which results in very low recall rates and high precision. *Shotgun* and *PSGD* are much more tolerant for low iterations, with *PSGD* slightly better. In general, the F_1 score in Figure 5(c) is dominated by the recall rate, and *PSGD* achieves the best overall performance. Considering the fact that *PSGD* enjoys fast convergence while suffers more from fluctuation, we can see our anomaly detection framework does not require very high accuracy in the optimization. As long as a lasso solver can quickly capture the general dependence between variables, it would be a good fit for our framework.

V. CONCLUSIONS

In this paper, we proposed a novel method based on Granger causality to detect anomalies regarding dependency changes in multi-variate time series. The key idea is to build a Granger graphical model on a reference data set, pose a constraint on the model of the test data set assuming it has the same temporal dependence as the reference data, and measure the difference between the two distributions inferred from reference and test model using KL-divergence. We also investigated several stochastic and parallel optimization algorithms to speed up our method. The empirical results verified the effectiveness of our method in accuracy and persistence. For future work, we are interested in relaxing the linear assumption in Granger causality, as well as anomaly detection with missing observations or even hidden variables.

ACKNOWLEDGEMENT

This research was supported by the NSF research grants IIS-1134990 and the U.S. Defense Advanced Research Projects Agency (DARPA) under the Anomaly Detection at Multiple Scales (ADAMS) program, Agreement Number W911NF-11-C-0200. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency.

REFERENCES

[1] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical Granger methods. In *Proc. of SIGKDD*, 2007.
[2] S. Boyd. Stochastic subgradient methods. Lecture slides and notes for EE364b. Stanford University, Spring quarter 2007-2008.
[3] J. Bradley, A. Kyrola, D. Bickson and C. Guestrin. Parallel Coordinate Descent for L1-Regularized Loss Minimization. In *ICML*, 2011.
[4] A. Brovelli, M. Ding, A. Ledberg, Y. Chen, R. Nakamura, and S. L. Bressler. Beta oscillations in a large-scale sensorimotor cortical network: Directional influences revealed by granger causality. In *Proc. of PNAS*, 101(26):9849–9854, 2004.
[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, 2009.
[6] L. Chiang and R. Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(2):159 – 178, 2003.
[7] C. Diks and V. Panchenko. Modified hiemstra-jones test for Granger non-causality. Technical Report 192, Society for Computational Economics, 2004.
[8] E. L. Russel, L.H. Chiang and R. D. Braatz. *Data-driven techniques for fault detection and diagnosis in chemical process*. Springer-Verlag, 1st edition, 2000.
[9] R. Fujimaki, T. Yairi, and K. Machida. An anomaly detection method for spacecraft using relevance vector learning. In *PAKDD*, 2005.
[10] T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. In *ICDM*, 2007.
[11] E. Keogh and J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Proc. of ICDM*, 2005.
[12] A. J. S. Martin Zinkevich, Markus Weimer and L. Li. Parallelized stochastic gradient descent. In *Proc. of NIPS*, 2010.
[13] S. Shalev-Shwartz and A. Tewari. Stochastic Methods for l1-regularized Loss Minimization. In *Proc. of ICML*, 2009.
[14] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proc. of SIGKDD*, 2007.
[15] J.-i. Takeuchi and K. Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE Trans. on Knowl. and Data Eng.*, 18:482–492, 2006.
[16] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proc. of ICML*, 2011.