Flight Data Anomaly Detection and Diagnosis with Variable Association Change

Sijie He*
hexxx893@umn.edu
University of Minnesota, Twin Cities

Hao Huang
haohuangcssbu@gmail.com
GE Global Research

Shinjae Yoo
shinjae.yoo@stonybrook.edu
Stony Brook University

Weizhong Yan, Feng Xue, Tianyi Wang
{yan,xue,wangt}@ge.com
GE Global Research

Chenxiao Xu
chenxiao.xu@stonybrook.edu
Stony Brook University

ABSTRACT

Aircraft sensors generate multivariate time series during flights, where each sensor corresponds to one variable. During normal operation mode, the associations (dependencies) among variables are mainly stationary. One type of flight anomaly that is of interest relates to variable association change. Detection and diagnosis of such type of anomaly need to pinpoint the time series, i.e., variables related to association change, which helps in understanding the underlying mechanisms of anomalies. However, it is hard to detect such change because the variable associations are usually unknown and complicated, and the anomalous samples are usually insufficient for learning the substandard association. In this work, we present a neural network that can 1) detect this type of anomalies given multivariate time series as input; 2) locate the association change by learning the nonlinear variable associations from both normal data and the detected anomalies. Specifically, we leverage the learned model from normal data to learn the faulty association of the anomalies. Experiments using simulated and real-world flight data show that our model outperforms existing methods in flight anomaly detection and diagnosis.

CCS CONCEPTS

• **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → **Anomaly detection**; **Temporal reasoning**;

KEYWORDS

time series anomaly detection; variable association

ACM Reference Format:
Sijie He, Hao Huang, Shinjae Yoo, Weizhong Yan, Feng Xue, Tianyi Wang, and Chenxiao Xu. 2021. Flight Data Anomaly Detection and Diagnosis, with Variable Association Change. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3412841.3441916

*Work done in part during internship at GE Global Research.

SAC '21, March 22–26, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8104-8/21/03…$15.00
https://doi.org/10.1145/3412841.3441916

1 INTRODUCTION

Multivariate time series are collected in aircraft, where each constituent univariate corresponds to one sensor or component in the aircraft systems. On each flight, samples are recorded with a $1Hz$ sampling rate, covering all flight phases including taxi, take-off, climb, cruise, descend, and landing. Effective anomaly detection and diagnosis on flight data are essential to ensure flight safety.

Under normal operation mode, variable associations (dependencies) are usually stationary [5, 13, 14]. Detecting anomaly with variable association change [14, 32, 39] is of great interest. It refers to pinpointing the time series and variable association related to the changes, so that proper actions can be taken in time to resolve the issues. Given normal multivariate time series as a training set and unlabeled time series as testing input, we aim to propose a model for flight data analysis that is capable of:

(1) learning the variable association in the normal time series;
(2) detecting any flight anomaly in the testing set;
(3) discovering how association change in the detected anomaly.

The major challenges we face are 1) the variable associations are usually unknown and complicated [6, 28], and 2) the detected anomaly samples are usually insufficient for learning the substandard associations. Traditional methods detect anomalies and their causes either through regression residuals with predefined kernels [23, 34], or distance-based metrics from inferred normal distributions [25, 32, 41], but without exploring variable association. Besides, selecting the appropriate kernel or distribution model requires a deep understanding of domain knowledge and, in many cases, is not even possible. Recent research tried to develop a more data-driven solution with autoencoder structure [39, 40]. But there is a lack of work in learning the variable association change, especially when the change is multivariate and nonlinear.

In this paper, we propose a solution by leveraging the information learned from normal data to anomalous variable association learning. Specifically, the variable association is acquired by learning a nonlinear Granger causal graph, where each node represents a variable in the time series, and each edge is directed and weighted, describing the Granger causality between the two connected nodes. As shown in Figure 1, by measuring the change between the two

Figure 1: We discover the variable association change by measuring the difference Δ between the variable association graph N learned from normal time series and the graph A learned from detected anomalous time series.

graphs learned from normal and anomalous time series, the model can discover the association change accordingly. We name our model ***D**etection and Diagnosis of **A**nomaly with **V**ariable **A**ssociation **C**hange*, or ***DAVAC***. Contributions of our work include:

(1) *DAVAC* can detect multivariate time series anomaly, and the underlying variable association change for flight data.
(2) *DAVAC* learns nonlinear variable association without any predefined kernel or data distribution assumption.
(3) *DAVAC* outperforms the popular baseline methods on flight anomaly detection and diagnosis by a significant margin.

2 RELATED WORK AND DISCUSSION

In the past decade, major advances have been made on anomaly detection [3, 8, 16–19, 26, 27, 31]. In this paper, we particularly focus on those for ***multivariate time series*** *and* ***providing diagnosis***. Research in [7] proposed a model to automatically learn log patterns from normal execution, and detect anomalies when log patterns deviate from the learned model. Some research target on finding variable association change in anomalies [5, 11, 32], by assuming that associations are all linear. In particular, work in [39] constructs multi-scale signature matrices by linear correlation and uses an autoencoder framework to measure the reconstruction error. But it does not apply to data with nonlinear correlation. Comparatively, our *DAVAC* aims at detecting nonlinear variable association without any linear assumption. Su et.al [35] proposed a stochastic recurrent neural network for multivariate time series anomaly detection, which as well provides interpretations based on the reconstruction probabilities of its constituent univariate time series. However, deeper insights about associations among variables are missing.

For *variable association learning*, major related work are:

• ***Granger Causality.*** The method in [2] builds a regression model on two variables with F-tests on the residuals. Later works [12, 38] involve all the variables by Vector Auto-regression (VAR). However, the VAR method is inherently linear. In contrast, our *DAVAC* does not rely on any distribution or kernel assumption.

• ***Transfer Entropy.*** It was firstly proposed in [33] for learning pairwise association without distribution assumption and linear setting. Such method is extended in the work [22] to be more robust and computationally efficient. But this type of methods only detect pairwise association without considering multivariate interaction, and thus share similar limitations of Granger causality.

• ***Graph Learning.*** Research such as [1, 4, 32] learn variable association by adding lasso or ridge regularization to VAR. The variable association depends on whether or not the weighted sum of the two variables across all timestamps is close to zero. However, it can only learn linear temporal relationships. Comparatively, our *DAVAC* can capture complicated nonlinear relationships.

3 MOTIVATION AND BACKGROUND

3.1 Motivation

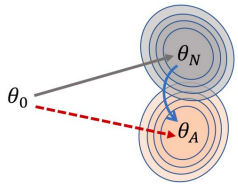


Figure 2: Given one learning model, θ_0 denotes the initial parameter configuration, and θ_N (θ_A) denotes the configuration that performs well in learning normal (anomalous) pattern. Due to the robust design of aircrafts, anomalies usually only affect a small portion of the system, so θ_A is not far from θ_N. Hence learning θ_A from θ_N is more efficient than from scratch (θ_0) given a limited amount of anomalies.

One major challenge in anomaly detection is that the anomaly samples are usually insufficient for learning the substandard variable associations. Given a limited amount of detected anomalies, instead of learning the underlying association from scratch, we propose to leverage the information learned from normal data. Our motivation is illustrated in Figure 2. Learning a model is about adjusting all the parameters (collectively referred to θ) to optimize the objective function. Let θ_0 denote the initial configuration, and θ_N (θ_A) denote the best configuration for learning normal (anomalous) variable association. Due to the robust design of modern aircraft, anomalies usually only affect a small portion of the system. The variable associations of normal and anomaly are not dramatically different. Therefore, θ_A is not far away from θ_N. Accordingly, reaching θ_A from θ_N is more efficient than from scratch (θ_0) with a limited amount of anomalies. Hence in our method, 1) *normal and anomalous models share the same structure*, and 2) *anomalous model is initialized by the parameters learned from normal data*. Such initialization provides a "warm start" for learning anomalous association.

3.2 Notation

From each flight, we sample time series using sliding window with fixed length. We denote *normal* time series data for training as $X = \{X(1,*), \cdots, X(m,*)\} \in \mathbb{R}^{m \times n}$ with m variables and n timestamps, where $X(i,*) \in \mathbb{R}^{1 \times n}$ is the time series associated to the i-th variable. $X(*,t_j) = \{X(1,t_j), \cdots, X(m,t_j)\} \in \mathbb{R}^{m \times 1}$ denotes the values of all m variables, where $X(i,t_j)$ is the value of the i-th variable at time t_j. The variable association graph for normal data is denoted as $N \in \mathbb{R}^{m \times m}$ with non-negative elements, where $N(i,j)$ denotes the strength of association on j-th variable from i-th variable. Similarly, the *anomalous* data and the corresponding association graph is denoted by X' and A.

3.3 Multivariate Granger Causality

In this paper we use nonlinear Granger causality to represent variable association. Granger causality was originally performed on pair-wise variables [2, 15]. Later research [1, 20, 32, 37] approach it in multivariate way using VAR. Particularly, it is a regression model to predict $X(*, t_n)$ with $X(*, t_1 : t_{n-1})$:

$$X(*, t_{j+1}) = \sum_{r=1}^{\ell} A_r X(*, t_{j+1-r}) + \epsilon(*, t_{j+1}), \tag{1}$$

where $\epsilon(*, t_{j+1}) \in \mathbb{R}^{m\times 1}$ is a white Gaussian noise at time t_{j+1}, ℓ is the time order, and A_r is the Granger causal graph for each time lag r. Time series $X(i, *)$ is called a *Granger cause* of time series $X(j, *)$ if there is a significant residual reduction with the random variable i. This does not guarantee but implies that at least one of the elements $A_r(i, j)$ is significantly larger than zero (in absolute value) and this is widely used in various temporal causal inference algorithms [1, 9, 20, 21, 32, 37].

However, Equation (1) is purely linear and does not capture any nonlinear association between variables. Comparatively, our *DAVAC* model targets to discover more general and nonlinear Granger causality in a data-driven way.

4 THE PROPOSED DAVAC MODEL

The proposed *DAVAC* model approaches anomaly detection and the normal Granger graph through regression setting, which uses $X(*, t_{j-\ell+1} : t_j)$ to predict $X(*, t_{j+1} : t_{j+c})$, with learning normal Granger graph as a downstream module. The regression model is then used to detect anomalies that have high prediction residuals. A separate model is initialized with the parameters from the normal regression model, trained using the detected anomaly, and learns the corresponding anomalous Granger graph in the same way.

We describe normal regression modeling first. Figure 3 illustrates the model that consists of three modules. Section 4.1 focuses on Module 1 that explores temporal nonlinearity, section 4.2 introduces Module 2 that aims to discover the Granger causal graph, and section 4.3 explains Module 3 that explores intervariable nonlinearity and regresses prediction targets. In the end, we explain the learning of anomalous data in Section 4.4.

4.1 Module 1: Learning Temporal Features

In Module 1, we apply a residual neural network (ResNet) [10] to learn the univariate nonlinearity. By setting $G_0 = X(*, t_{j-\ell+1} : t_j)$, a typical ResNet proposed in [10] is defined as:

$$G_q = ReLU\big(G_{q-1}B_q + I(G_{q-1})\big), \quad q = 1, 2, ..., Q, \tag{2}$$

where $G_q \in \mathbb{R}^{m\times p}$ denotes the output of the q-th residual layer, $B_q \in \mathbb{R}^{p\times p}$ notes the weight matrix in the q-th layer (the first weight $B_1 \in \mathbb{R}^{\ell\times p}$), I is an identity mapping, and Q is the total number of residual layers. The core idea of ResNet is to introduce an identity shortcut connection that skips one or more layers if they have no contribution to the final target. Therefore, the *DAVAC* model goes deeper as it will not produce a training error greater than its shallower counterparts.

As the ResNet goes deeper, we can discover more complicated temporal nonlinearity. But in practice, it is difficult to know which layer (level of nonlinearity) is more relevant to the underlying Granger causality. Usually, the relevance to the Granger causality varies in different datasets. Therefore we concatenate the output of each ResNet layer and pass it through a fully connected layer for feature extraction. Specifically, the output of all the Q ResNet layers $(G_1, \cdots, G_Q)$ are concatenated as

$$G_{\text{all}} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_Q \end{bmatrix} \in \mathbb{R}^{pQ\times m}, \tag{3}$$

which is passed to a fully connected layer for aggregating nonlinear features learned by the ResNet. The output of Module 1 is denoted as $\widetilde{G} \in \mathbb{R}^{s\times m}$, where s is the number of extracted features that represent all the useful temporal univariate nonlinearity that contribute to the final regression objective. Each column $\widetilde{G}(*, i)$ contains the univariate nonlinear features learned only from $X(i, t_{j+\ell-1} : t_j)$, where $i = 1, ..., m$.

It is worth emphasizing that Module 1 only learns the nonlinearity on the univariate level without rolling information among variables. Such rolling should only involve the Granger causes of each variable, which is done in the causality learning in Module 2.

4.2 Module 2: Learning Granger Causal Graph

Module 2 learns the Granger causal graph N from normal time series. The output $\widetilde{G} \in \mathbb{R}^{s\times m}$ of Module 1 represents the temporal nonlinearity on univariate level. Intuitively, Module 2 finds all the contributing temporal features from $\widetilde{G}(*, i)$ for predicting $X(i, t_{j+1} : t_{j+c})$ later. That can be done by learning Granger causal graph N:

$$\Gamma(*, i) = \widetilde{G}N(*, i), \quad i = 1, 2, ..., m. \tag{4}$$

Each column $N(*, i)$ in N tells the Granger causes of variable $X(i, *)$, where rows with larger values indicate the variables with higher association on $X(i, *)$ in terms of regression. Therefore, the output Γ of Module 2 is the Granger cause variables learned on the intervariate level for each variable.

Note that, although Equation 4 is linear, the learning process before and after are all nonlinear, therefore the N depicts nonlinear associations among variables.

4.3 Module 3: Time Series Regression

The input of Module 3 is $\Gamma \in \mathbb{R}^{s\times m}$, of which $\Gamma(*, i)$ contains a combination of temporal nonlinearity from the Granger causes of i-th variable. The goal of Module 3 is to roll these input in a nonlinear way, and use them to estimate the regression objectives $X(i, t_{j+1} : t_{j+c})$, where c is the length of the prediction target.

The first part of Module 3 consists of a series of fully connected layers, and the j-th layer output is defined as:

$$H_j = tanh(H_{j-1}W_j + b_j), \quad j = 1, 2, ..., D \tag{5}$$

where $H_{j-1} \in \mathbb{R}^{m\times d_{j-1}}$ is the output of layer $j-1$, $W_j \in \mathbb{R}^{d_{j-1}\times d_j}$ is a weight matrix, and b_j is a bias vector. We set $H_0 = \Gamma^T$ (transpose of Γ), and the final output is $H_D \in \mathbb{R}^{m\times d_D}$. We choose $tanh$ as activation function in Equation (5) since it yields the best performance.

The second part is designed for time series regression. The i-th row of H_D contains all of the contributing nonlinear forms from the

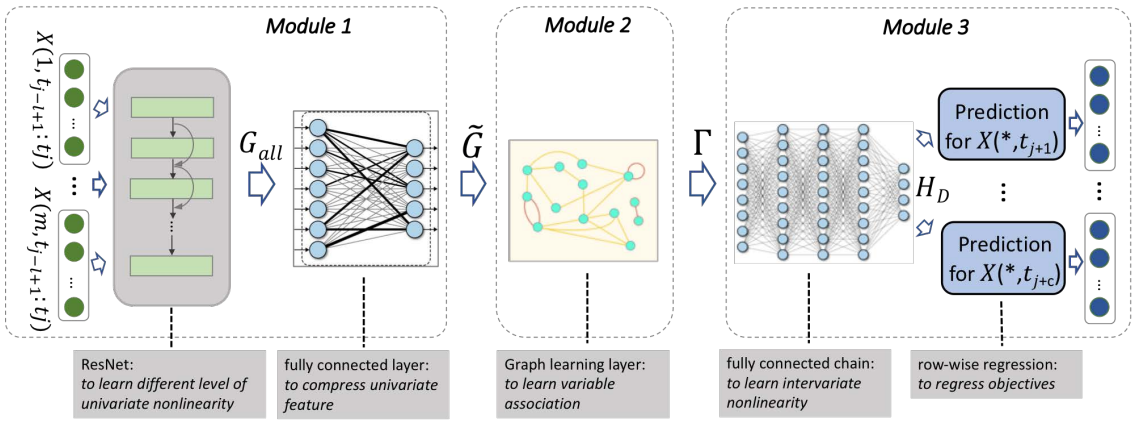


Figure 3: The architecture of DAVAC. The green dots represent the input $X(*, t_{j+l-1} : t_j)$. Module 1 provides the temporal nonlinearity learned on univariate level. Module 2 learns the variable association graph to select Granger causes for each variable regression. Module 3 learns the intervariate nonlinearity and predicts the target $X(*, t_{j+1} : t_{j+c})$ in a row-wise manner.

Granger causes of $X(i, *)$ to predict $X(i, t_{j+1} : t_{j+c})$. For each timestamp k of the prediction target, one weight matrix $R_k \in \mathbb{R}^{m \times d_D}$ is learned through

$$\hat{X}(i, t_{j+k}) = H_D(i, *) \circ R_k(i, *) = \sum_{r=1}^{d_D} H_D(i, r) R_k(i, r) \,, \quad (6)$$

where $i = 1, \cdots, m$ and $k = 1, \cdots, c$. The model learns the optimal combination of each row in H_D to approach $\hat{X}(*, t_{j+1} : t_{j+k})$. By involving more than one timestamp as prediction targets, we are able to learn a more accurate Granger causal graph N with addition information in the target. The prediction window is set as $c \in [2, 4]$.

The loss function is designed to minimize the regression error

$$L_N = \frac{1}{nc} \sum_{\imath=1}^{n} \sum_{k=1}^{c} \|X_\imath(*, t_{j+k}) - \hat{X}_\imath(*, t_{j+k})\|_F + \lambda_N \|N\|_2 \,, \quad (7)$$

where $\imath$ is index of time series, n is the number of training time series, and λ_N is the penalty parameter. The ℓ_2 regularization term is added to avoid over-fitting on the Granger causality learning.

4.4 Anomaly Detection and Learning

The trained normal regression model is used as our anomaly detector. If the regression residual of a new sample is above a threshold, it indicates a high possibility of the appearance of an anomaly. Specifically, the threshold we use is determined by the prediction residuals on a validation set with new normal samples collected from the normal running mode.

Once the anomalous samples are detected, the next step is to create another regression model to learn the anomalous Granger causal graph from these samples. We assume that 1) the anomalous samples arrive consecutively and 2) the anomalous Granger causal graph is stationary. The new model shares the same structure as the normal model. As introduced in Section 3.1, we initialize the new model with the parameters learned from the normal model. The anomalous data is denoted as X'. The loss function L_A is defined as

$$L_A = \frac{1}{n'c} \sum_{\imath=1}^{n'} \sum_{k=1}^{c} \|X'_\imath(*, t_{j+k}) - \hat{X}'_\imath(*, t_{j+k})\|_F + \lambda_A \|A\|_2 + \gamma \|A - N\|_2 \,, \quad (8)$$

where $\imath$ is the index of time series, n' is the number of training time series. We include a new regularization term $\|A - N\|_2$ to control the divergence between normal and anomalous graph. This term prevents the model from over-fitting to the noise given a limited number of anomaly samples.

5 EXPERIMENTAL SETUP

5.1 Flight Data

The real-world dataset for our experiment is collected on a flight-by-flight basis. It includes data from 2000+ flights. Most flights behave normally, but there are a small number of abnormal flights. On each flight, time series are collected from 47 sensors/control signals with a 1 Hz sampling rate, covering all flight phases including takeoff, climbing, cruise, and descend. Each flight has 4000 to 20000 timestamps, depending on the flight distance.

For confidentiality, sensitive descriptions are omitted. To construct the dataset,

- We randomly select 72 normal flights and three abnormal flights. The three abnormal flights are selected because their root causes are known: the first one has a broken correlation between fuel flow and air temperature. The second one shows large differences between left and right duct pressure, of which readings are usually close to each other. The third one has large differences between left and right exit temperature, which normally are in the same range.
- Samples then are generated from the selected flights. Each sample has 12 timestamps and 47 variables. We use the first 10 timestamps to predict the last two timestamps.
- We use 50 normal flights as a training set for learning the normal model and 10 normal flights as the validation set. The test set consists of the rest 12 normal flights and the three abnormal flights.

We tune the parameters according to the minimum loss from validation test. The input format is set as $\ell = 10$ and $c = 2$. *DAVAC*'s structure is set as follows: $Q = 5$, $p = 20$, $s = 50$, $D = 1$ with $d_1 = 50$. λ_N and λ_A are set to be 0.005, and γ is 0.0001.

5.2 Simulated Datasets

Although the real flight dataset can be used to evaluate the accuracy of anomaly detection and diagnosis, it is not straightforward to quantitatively evaluate the learned variable association change due to the lack of ground truth of the underlying variable relationship. Therefore, we also construct two simulated datasets with 40 and 100 variables respectively. Each dataset generation starts with building a variable association graph, and rolls nonlinear association between the associated variables. The graph will be later used as ground truth to evaluate the Granger causality learning.

Graph Generation. The underlying graph follows a three-layer hierarchical structure, as shown in Figure 4. Variables in the first layer are "masters" that serve as Granger causes of the second layer variables. Second and third layer variables are "effectors", of which the causes are from its previous layer. That is, there are directed edges connecting the first to the second layer, and the second to the third layer. Besides, to make the graph more complex, directed edges are added among the second layer variables. The hierarchical structure of each dataset is organized as 5-20-15 and 20-60-20 respectively. For normal graphs N, the sparsity is set as 0.1. The anomalous graphs A are generated by randomly adding/deleting edges on N, as shown in Figure 4(b). In our experiments, we generate A with extra 10% and 20% edges for the two datasets respectively.

Time Series Generation. For the variables in the first layer, the time series are generated as follows:

$$X(i,t_j) = \tanh(\sqrt{3} * d(i) * X(i,t_{j-1}) * \epsilon(i,t_j) + d(i)^2 * cos(X(i,t_{j-5}))) + \epsilon(i,t_j),$$

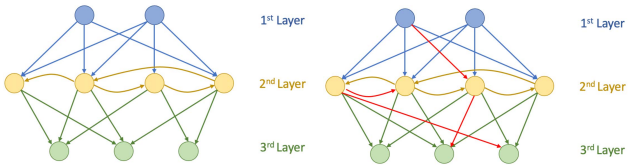
where $d(i)$ is the decay factor of variable i that is randomly selected from uniform distribution $\mathcal{U}(0.95,1)$, and $\epsilon(i,t_j)$ is the random noise sampled from normal distribution $\mathcal{N}(0,1)$. The variables in the second/third layers are generated by:

$$X(i,t_j) = \sum_{l=1}^{m} \mathbb{1}(l \to i) * \sin(r(l \to i) * X(l, t_{j-p(l\to i)})) + \sin(r(l \to i) * X(l, t_{j-p(l\to i)}) * X(i, t_{j-p(l\to i)})) + \epsilon(i,t_j),$$

where $r(l \to i)$ follows $\mathcal{U}(-1,1)$, and $p(l \to i)$ represents the time lag from l to i and is randomly selected from $\{1,2,3\}$. $\mathbb{1}(.)$ returns 1 when condition is true and 0 otherwise.

Sample Generation. Samples are generated by splitting the whole time series into multiple overlapping segments by a sliding window with a stride of w. Each time series is in $\mathbb{R}^{m\times(\ell+c)}$, where ℓ is the input window and c is the prediction window. Eventually, there are $\lfloor \frac{n-(\ell+c)}{w} \rfloor$ time series given n is the total timestamps.

Training, Validation, and Testing. The normal model is trained on 10000 time series with 1000 as a validation set, while the test set involves 6000 time series, of which 1000 are anomalous and the rest are normal. We further vary the number of anomalies to test the graph learning sensitivity in Section 6.4. To suppress the noise effect we apply *bootstrapping* by repeatedly training on samples



(a) Normal graph (b) Anomalous graph

Figure 4: Figure (a) and (b) are our simulated normal (N) and anomalous graph (A) , each with three layers. The blue and green edges indicate the association in-between layers, while the yellow indicate those within the second layer. The red edges in anomalous graph represent the change of variable association compared against normal graph.

with replacement for 20 times, and use the average result as the final graph. Intuitively, the edges with large values in most runs tend to stand out, while the edges driven by noise are weakened. During training, we apply early stopping that is triggered by a prediction error increase on 20% holdout samples from the training set.

Parameter Setting. Our model hyperparameters are tuned by performing a grid search to minimize loss in the validation test. For the dataset with 40 variables, we set $\ell = 5$ and $c = 3$. *DAVAC*'s hyperparameters are set as follows: $Q = 5$, $p = 20$, $s = 50$, $D = 1$ with $d_1 = 50$. For the dataset with 100 variables, we set $\ell = 8$ and $c = 4$. *DAVAC*'s hyperparameters are set as follows: $Q = 6$, $p = 20$, $s = 80$, $D = 1$ with $d_1 = 50$. For the regularization coefficients, λ_N and λ_A are set to be 0.005, and γ is 0.0001. Parameter sensitivity test is included in Section 6.4.

5.3 Evaluation Metrics

We evaluate both **anomaly detection performance** and **graph learning quality** for the proposed model and baselines. For anomaly detection, we adopt AUPR (area under precision-recall curve) that is not biased to normal data. Besides, AUROC (area under ROC curve) is also included. Evaluation of variable association graph can be treated as a binary classification evaluation on all the m^2 edges. The connected edges from ground truth are treated as positive edges, while the not connected ones are zeros. We convert all the learned graphs to be with absolute values, and evaluate the performance of graph learning using AUPR and AUROC as metrics. Please note that a classifier that makes random guesses can have AUROC as low as 0.5. However, the AUPR can be less than 0.5.

5.4 Baseline Methods

We select the following baseline methods for anomaly detection:

(1) *MSCRED* [39]. A deep learning model that constructs linear signature matrices to characterize system status, and then feeds them into a convolutional recurrent autoencoder model. The reconstruction error in the signature matrices indicates anomaly degree and the relevant variables.
(2) *cLSTM* [30, 36]. A deep learning model that trains LSTM on normal data to regress future values with prediction error treated as an anomaly degree.

(3) *OmniAD* [35]. A deep learning model using a stochastic recurrent neural network that detects anomalies with interpretations based on the reconstruction probabilities.
(4) *One-Class SVM (OCSVM)* [24]. It converts time series into a phase space then apply RBF kernel to detect anomalies.
(5) *Granger Graphical Models (GGM)* [32]. It applies a linear Granger graphical model and *KL*-divergence to measure correlation anomaly.

The baselines of graph learning include:

(1) *VAR* [1]. It applies vector autoregressive model with regularization to learn Granger causal graph.
(2) *PCKGC* [29]. It evaluates the partial conditioning kernel-based Granger causality with high mutual information.
(3) *Copula* [20]. It transforms the marginal distribution of each variable to a Gaussian domain and then applies VAR.
(4) *cLSTM* [36]. The graph is generated from the absolute sum of all the corresponding parameters from the first layer of the LSTM model.
(5) *pTE* [22]. It calculates the phase transfer entropy by transforming temporal signals of each variable to discrete phases with histogram-based probability.

For all the baselines methods, we tune their hyperparameters according to their *AUPR* to our best effort.

6 EXPERIMENTAL RESULT AND DISCUSSION

Given normal multivariate time series as a training set and unlabeled time series as testing input collected from different flights, our purpose is to see if the algorithms can ***detect the abnormal flights***, and ***diagnose the association change*** between sensors. Our model is implemented in *PyTorch* and trained by *Adam* optimization with the learning rate set as 1*e*-3.

6.1 Anomaly Detection Result Analysis

Figure 5 shows anomaly detection performance on simulated data. Apparently, our *DAVAC* is superior to all the baselines. Especially, AUPR of *DAVAC* is almost 13% better than the second-best method (*OmniAD*). *MSCRED*, *cLSTM* and *GGM* take the temporal dependency into account, therefore, they outperform *OCSVM*.

For the experiments on the real flight dataset, we evaluate if the algorithms can identify all the three anomalous flights in the testing set. Figure 6 shows the results of the top-four methods according to our simulated experiments. The anomaly score of each flight is measured by 95% percentile of all the time series scores from that flight. We can clearly see that the proposed *DAVAC* assigns significantly higher scores to the three verified anomalous flights compared against those of normal flights. On the other hand, *MSCRED* and *OmniAD* fail to identify all the three anomalous flights.

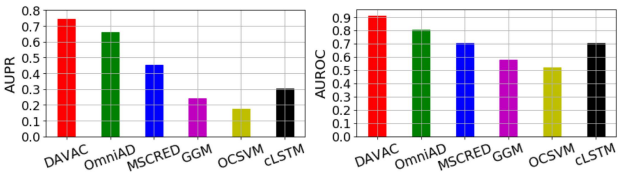


Figure 5: Simulated anomaly detection comparison.

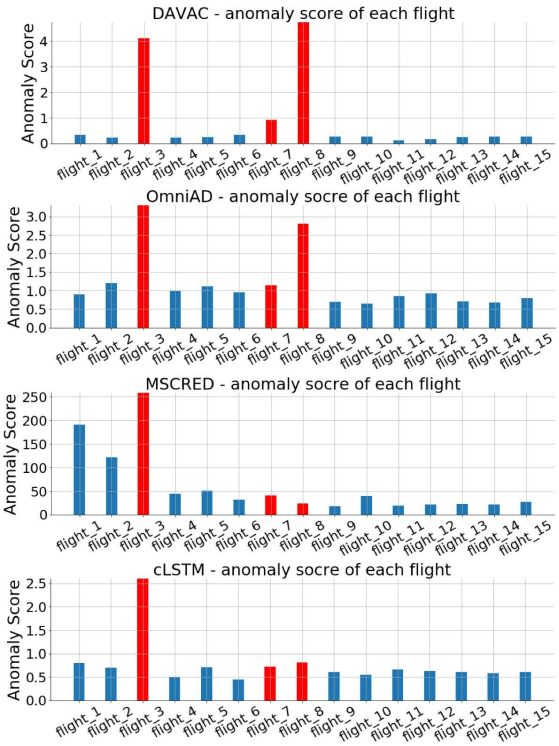


Figure 6: Anomaly scores of real flights. X-axis is the ID of each flight and y-axis is anomaly score. The red (blue) bars represent the anomalous (normal) flights by ground truth. *DAVAC* assigns high anomaly scores to all three anomalous flights, while the other methods fail to detect all the three.

6.2 Diagnosis and Graph Learning

We examine if the algorithms can diagnose and detect the root causes of the anomalous flights. We test all anomaly detection methods on the three anomaly flights separately. Figure 7 shows the

Flight ID	*flight_3*	*flight_7*	*flight_8*
OmniAD	Fuel Flow ✓	Left Duct Pressure ✗	Left Exit Temperature ✓
MSCRED	Fuel Flow ✓	Left Exit Temperature ✗	Left Exit Temperature ✓
cLSTM	Fuel Flow ✓	Left Duct Pressure ✗	Left Exit Temperature ✓
DAVAC	Air Temperature -> Fuel Flow ✓	Right Duct Pressure -> Left Duct Pressure Right Exit Temperature -> Right Duct Pressure ✓	Left Exit Temperature -> Right Exit Temperature ✓

Figure 7: Learning the root cause of anomalous flights. A green tick indicates correct finding while a red cross shows irrelevant finding. Our *DAVAC* successfully detects the variable association change for all the three anomalous flights, while the other deep learning methods can only detect the most relevant variables of two , but missed the other one.

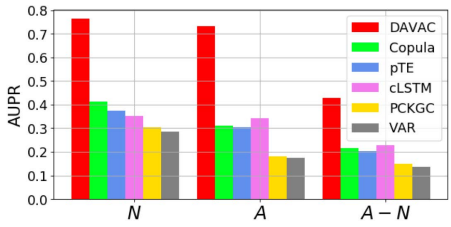
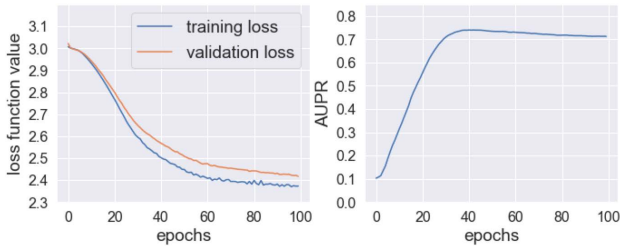


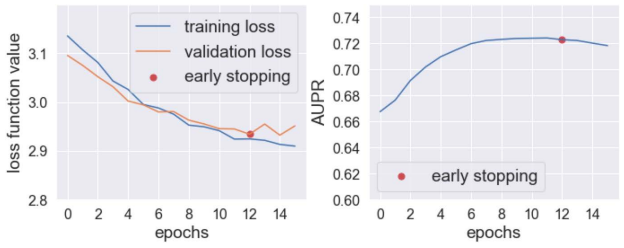
Figure 8: Graph learning comparison of the normal graph N, the anomalous graph A, and the association change $|A - N|$. It shows that our *DAVAC* has the highest accuracy on all the three learning targets.

output from the top-performing methods. Our *DAVAC* successfully detects the changed association that relate to the root causes, while *MSCRED*, *OmniAD* and *cLSTM* can only detect the most relevant variables of two flights. One possible explanation is that *MSCRED* assumes that normal/anomalous patterns can be captured by the change in linear correlations among variables, which is not always true. On the other hand, *OmniAD* and *cLSTM* rank variables based on univariate reconstruction error, instead of variable associations.

To quantitatively evaluate the performance of graph learning, we run experiments on the simulated data and show results in Figure 8. The input normal data for learning has 5000 time series with 40 variables, while there are 700 anomalous time series with 20% variable association (edge) change on the normal graph. To have



(a) Training/validation loss and AUPR during learning N



(b) Training/validation loss and AUPR during learning A

Figure 9: Loss convergence and graph learning quality by *DAVAC*. Our model converges on both training and validation phases, when AUPR rises almost simultaneously, which indicates that the quality of the learned graph improves as our learning loss reduces. Besides, early stopping helps the model avoid overfitting given limited amount of anomalies.

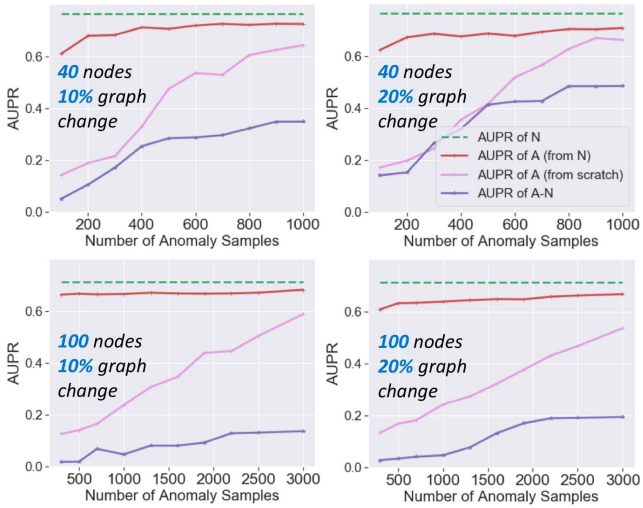


Figure 10: Sensitivity test on different number of variables, anomaly and graph change. The green constant line shows the normal graph N learning quality. The red and pink lines illustrate the learning quality of anomalous graph A, with (red) or without initialization (pink), respectively. The purple line shows the learning quality of $|A - N|$.

a fair comparison, we assume that the labels of the anomalies are given in this test, therefore all the algorithms can learn abnormal graph from the same data.

Three learning targets are evaluated respectively: normal (N) and anomalous (A) graphs, and the difference between them ($A-N$). For all targets, our *DAVAC* is significantly better than all the baselines. Specifically, *DAVAC* not only learns better normal graph (N, 86%+ better than the second-best), but also learns better anomalous graph (A, 110%+ better than the second-best) given the limited amount of anomaly. Furthermore, *DAVAC* discovers the variable association change with the best quality (85%+ better than the second-best). As for the other methods, *Copula* and *pTE* perform better with a sufficient amount of training samples. However, *cLSTM* adapts better to learn from anomalies. It is because we let *cLSTM* start from normal parameters to quickly adapt to anomalies. This once again proves the effectiveness of our strategy of information leverage.

6.3 Convergence Analysis

To show the convergence of our *DAVAC* on learning N and A, Figure 9 displays the learning loss and AUPR score across training epochs. We have three main observations: (1) It shows that both training and validation converge well, proving that our *DAVAC* model is reasonably stable. (2) AUPR rises almost simultaneously as loss converges, which illustrates that the quality of the learned graph improves as the prediction error converges. (3) It is worth noticing that we use early stopping while training on anomaly data, in order to avoid overfitting given a limited amount of training samples.

6.4 Sensitivity Analysis

We are interested to see how *DAVAC* performs with a various number of variables and anomaly severity (graph change) with a various

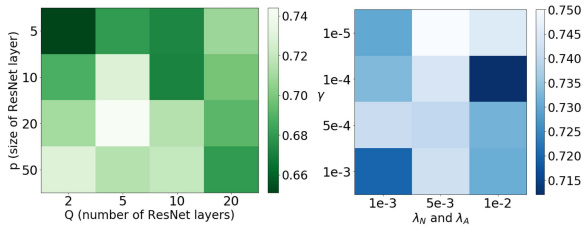


Figure 11: Parameter sensitivity test in *AUPR*.

number of anomalies. Besides, we also want to show the effectiveness of our information leverage from normal data into learning anomalous graph. Figure 10 provides a systematic test. First, we can see that, compared to learning from scratch, learning A from N has a much higher *AUPR*, especially with fewer anomalies. This again verifies our previous intuition in Section 3.1. As more anomalies are observed, better graph change ($A - N$) is learned. Comparatively, *DAVAC* is more adaptive when the number of variables and anomalies is small.

On the other hand, *DAVAC* is also robust against hyperparameter tuning. Figure 11 shows our parameter sensitivity test in *AUPR* on simulated anomaly detection. We test with several key hyperparameters like the size and number of ResNet layers in Module 1, and regularization parameters in Equation (7) and (8). The performance in *AUPR* using simulated data validates our model robustness.

7 CONCLUSION

We propose a model that can jointly detect anomalous flights and the variable association change given multivariate time series collected from aircrafts. Our model is capable to learn the underlying association change with a limited amount of anomalies. Experimental results on both simulated and real-world datasets indicate that *DAVAC* outperforms other baselines by a significant margin.

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*. 66–75.

[2] Steven L Bressler and Anil K Seth. 2011. Wiener–Granger causality: a well established methodology. *Neuroimage* 58, 2 (2011), 323–329.

[3] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[4] Dehua Cheng, Mohammad Taha Bahadori, and Yan Liu. 2014. FBLG: a simple and effective approach for temporal dependence discovery from time series data. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*. ACM, 382–391.

[5] Wei Cheng, Kai Zhang, Haifeng Chen, Guofei Jiang, Zhengzhang Chen, and Wei Wang. 2016. Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 805–814.

[6] Belkacem Chikhaoui, Mauricio Chiazzaro, and Shengrui Wang. 2015. A new Granger causal model for influence evolution in dynamic social networks: the case of DBLP. In *Proceedings of the 29-th AAAI Conference on Artificial Intelligence*. AAAI Press, 51–57.

[7] 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*. ACM, Dallas, TX, USA, 1285–1298.

[8] Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. 2020. RobustTAD: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv preprint arXiv:2002.09545* (2020).

[9] Stefan Haufe, Klaus-Robert Müller, Guido Nolte, and Nicole Krämer. 2010. Sparse causal discovery in multivariate time series. In *Causality: Objectives and Assessment*. 97–106.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition,CVPR*. IEEE Computer Society, Las Vegas, NV, USA, 770–778.

[11] Tsuyoshi Idé, Spiros Papadimitriou, and Michail Vlachos. 2007. Computing correlation anomaly scores using stochastic nearest neighbors. In *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 523–528.

[12] Dominik Janzing, David Balduzzi, Moritz Grosse-Wentrup, Bernhard Schölkopf, et al. 2013. Quantifying causal influences. *The Annals of Statistics* 41, 5 (2013).

[13] Guofei Jiang, Haifeng Chen, and Kenji Yoshihira. 2006. Discovering likely invariants of distributed transaction systems for autonomic system management. In *2006 IEEE International Conference on Autonomic Computing*. IEEE, 199–208.

[14] Guofei Jiang, Haifeng Chen, and Kenji Yoshihira. 2006. Modeling and tracking of transaction flow dynamics for fault detection in complex systems. *IEEE Transactions on Dependable and Secure Computing* (2006).

[15] Maciej Kamiński, Mingzhou Ding, Wilson A Truccolo, and Steven L Bressler. 2001. Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. *Biological cybernetics* 85, 2 (2001), 145–157.

[16] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J Kim. 2017. A survey of deep learning-based network anomaly detection. *Cluster Computing* (2017), 1–13.

[17] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. 2015. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1939–1947.

[18] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*. Springer, 703–716.

[19] Jinbo Li, Witold Pedrycz, and Iqbal Jamal. 2017. Multivariate time series anomaly detection: a framework of Hidden Markov Models. *Applied Soft Computing* 60 (2017), 229–240.

[20] Yan Liu and Mohammad Taha Bahadori. 2013. An examination of practical Granger causality inference. In *Proceedings of the 13th SIAM International Conference on Data Mining*. SIAM, 467–475.

[21] Yan Liu, Alexandru Niculescu-Mizil, Aurelie C Lozano, and Yong Lu. 2010. Learning temporal causal graphs for relational time-series analysis. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 687–694.

[22] Muriel Lobier, Felix Siebenhühner, Satu Palva, and J Matias Palva. 2014. Phase transfer entropy: a novel phase-based measure for directed connectivity in networks coupled by oscillatory interactions. *Neuroimage* (2014).

[23] Helmut Lütkepohl. 2005. *New introduction to multiple time series analysis*. Springer Science & Business Media.

[24] Junshui Ma and Simon Perkins. 2003. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, Vol. 3. IEEE, 1741–1745.

[25] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. 2010. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE.

[26] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv:1607.00148* (2016).

[27] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. In *Proceedings*, Vol. 89. Presses universitaires de Louvain, 89–94.

[28] Daniele Marinazzo, Wei Liao, Huafu Chen, and Sebastiano Stramaglia. 2011. Nonlinear connectivity by Granger causality. *Neuroimage* 58, 2 (2011), 330–338.

[29] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. 2008. Kernel-Granger causality and the analysis of dynamical networks. *Physical Review E* 77, 5 (2008), 056215. https://doi.org/10.1103/physreve.77.056215

[30] Anvardh Nanduri and Lance Sherry. 2016. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In *2016 Integrated Communications Navigation and Surveillance (ICNS)*. Ieee, 5C2–1.

[31] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 353–362.

[32] Huida Qiu, Yan Liu, Niranjan A Subrahmanya, and Weichang Li. 2012. Granger causality for time-series anomaly detection. In *2012 IEEE 12th international conference on data mining*. IEEE, 1074–1079.

[33] Thomas Schreiber. 2000. Measuring information transfer. *Phys. Rev. Lett.* 85 (Jul 2000), 461–464. Issue 2. https://doi.org/10.1103/PhysRevLett.85.461

[34] Linda Sommerlade, Marco Thiel, Bettina Platt, Andrea Plano, Gernot Riedel, Celso Grebogi, Jens Timmer, and Björn Schelter. 2012. Inference of Granger causal time-dependent influences in noisy multivariate time series. *Journal of neuroscience methods* 203, 1 (2012), 173–185.

[35] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2828–2837.

[36] 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).

[37] Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. 2005. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences* 360, 1457 (2005), 969–981.

[38] Shun Yao, Shinjae Yoo, and Dantong Yu. 2015. Prior knowledge driven Granger causality analysis on gene regulatory network discovery. *BMC Bioinformatics* 16, 1 (28 Aug 2015), 273. https://doi.org/10.1186/s12859-015-0710-1

[39] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1409–1416.

[40] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[41] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net.