Detection and Characterization of Network Anomalies in Large-Scale RTT Time Series

Bingnan Hou, Tongqing Zhou, Zhiping Cai, *Member, IEEE*, Fang Liu

***Abstract*—Network anomalies, such as wide-area congestion or packet loss, could seriously degrade network performance. Detecting and characterizing network anomalies quickly and accurately on end-to-end paths are critical for ensuring network performance. In this work, an unsurprised two-step method for detection and characterization of network anomalies is presented. Our two-step method first detects anomaly from large-scale RTT time series using change-point detection algorithm. Then we characterize the events, i.e. identifying the nodes and links that are most responsible for the detected events, by analyzing the relations between the links with state changes during the anomalous period. Experiments performed on both simulated (artificial time series with ground truth) and real-network (RIPE Atlas traceroute measurements) datasets show that the proposed method achieves better performance w.r.t. accuracy and efficiency than existing solutions. what is more, our method provides valuable insights on anomaly mining in large-scale time series data.**

***Index Terms*—Network performance measurement, network anomaly detection, time series analysis.**

I. Introduction

The possible impact of a network anomaly includes disruption in network connectivity and performance degradation which dissatisfies network users and even causes huge financial losses [1]. It is very significant to develop a method to detect, locate, and assess the impact of network anomalies accurately and effectively when they occur. Many anomaly detection techniques [1], [2], [3] utilize the round-trip time (RTT) data, which can represent network performance timely, to detect the anomalous events. However, these mthods cannot exactly distinguish whether RTT changes are related to anomalous network events or other reasons which also cause RTT fluctuations such as path changes, normal congestions and routing changes. This makes it difficult to pinpoint anomalous events when normal RTT fluctuations cannot be excluded. In addition, most prior works only carry out anomaly detection on a single link or multiple links and do not study the correlation among the abnormal links, which results in the lack of ability to assess the impact of events and locate the root fault point.

In this paper, we propose an unsupervised method to focus on analyzing the amount of RTT changes of the whole monitored network. When a network failure event occurs, a large number of links are affected and many RTT related to these links will be changed. We believe that there is a network anomaly when the number of RTT changes in the network (AS) exceeds a normal reference. What is more, links' RTT time series should share some common characteristics when they pass over the same router or affected by the same accident [1]. Meanwhile, we characterize the events, which finds the the entities (nodes/links) that are most responsible for those changes, by analyzing the hidden relation in links. To be more specific, our method first utilizes a shape-based metric to calculate the distance between links' RTT time series with state change during the anomaly period. In this way, the hidden relation between links is included in the distance matrix. In order to facilitate the visualization of event impact assessment, we reduce the dimension of the distance matrix to a two-dimensional space. Then we calculate the highest density region through the two-dimensional kernel density estimation. Obviously, the links' RTT time series in this region has the most similar shape, i.e., the node in this region are affected by the same event.

Prior works have shown that both supervised and unsupervised learning techniques are quite used in event detection and characterization in different research communities [4]. The limitations of supervised methods are that they require labeled data which is not always available for real-world data, and cannot detect novel events that have never been observed previously [5]. Thus, we utilize an unsupervised method which does not require labeled data, it detects suspicious events if the behavior deviates from the normal behavior in time. Meanwhile, our proposed method does not assume any knowledge about the shape, amplitudes, or size of the time series which shows its high flexibility.

Different from earlier works on network anomaly detection, we experiment with both simulated (artificial time series with ground truth) as well as real-network (RIPE Atlas built-in traceroute measurements [6] without ground truth) datasets to spot events and pinpoint anomalous agents. The artificial time series data has been carefully simulated with the goal of testing the method in controlled environments. A known number of different structures are inserted in noise of various levels and characteristics. We construct twice change-point detection and shape-based similarity measure on these time series. Our proposed approach is able to successfully unearth these artificial events and the event-related individuals that initiated the events with high accuracy. Quantitative evaluation based on ground truth shows that our method can greatly reduce the alarm caused by data noise and accurately locate the event-related links. We also use RIPE Atlas built-in traceroute measurements (May 2015 - June 2015 and November 2015 -

B. Hou, T. Zhou and Z. Cai are with the School of Computer Science, National University of Defense Technology, Changsha 410073, China (e-mail: houbingnan19@nudt.edu.cn; zhoutongqing@nudt.edu.cn; zpcai@nudt.edu.cn).

F. Liu is with the School of Design, Hunan University, Changsha, 410073, China (e-mail: liufang06@gmail.com).

Corresponding authors: Tongqing Zhou and Zhiping Cai.

December 2015) where we construct RTT time series analysis of adjacent hops. Our experiments successfully reveal several big events during the time period of the data, demonstrating that the proposed methods can detect real disruptions and provide valuable insights on anomaly mining in large-scale time series data.

The key contributions of this paper are summarized as follows:

1) We present a network anomaly detection method which utilizes change-point detection algorithm. Compared with the outlier detection based method, our proposed method can greatly reduce the irrelevant alarms caused by RTT fluctuation.
2) We propose a novel unsupervised characterization method which takes advantage of a shape-based similarity measure to analysis the hidden relations between links. Combined with the multidimensional scaling algorithm, we visualized the relations between links, and then distinguish the event-related links from the irrelevant links.
3) Experimental results show that the proposed method achieves better performance w.r.t. accuracy and efficiency compared with existing solutions.

The remainder of the paper is organized as follows. Sec. II discusses some related works on network anomaly detection and characterization. Sec. III describes the design of our method and the algorithm details of anomaly detection and characterization. Sec. IV and Sec. V give our experimental setup and results. We conclude our paper in Sec. VI.

II. Related Work

It is well known that path changes and congestion are the main causes of RTT fluctuations or state changes [7], [8]. In this paper, we focus on the RTT state changes caused by network events which lead to network congestion. As to the noise nature of RTT time series, not all the RTT changes are network event-related. Thus, only monitoring the state of each single link cannot determine whether there is network anomaly. To mine anomalies in large-scale of performance time series data, several methods have been studied.

PCA-based anomaly detection methods, such as [9], [10], [11], [12] have been proposed by researchers to detect and diagnose anomalies on passive measurements. A PCA subspace projection methodology is proposed in [9], [13] where the authors apply PCA on network traffic data and separate of the high-dimensional space occupied by the data into disjoint subspaces corresponding to normal and anomalous network conditions. Hyndman et al. [14] uses PCA to isolate and diagnose the locations of the correlated anomalies in large-scale time series data. However, the PCA-based anomaly detection method only performs well on relatively smooth time series, as for the time series with high normal fluctuations, e.g., RTT time series, it cannot effectively reduce the false alarm rate.

Another common anomaly detection schema utilizes multichannel singular spectrum analysis (MSSA) algorithm for simultaneously denoising and reconstructing time series data [15], [16]. The difference between the predicted value and the real value is then used to determine whether there is anomaly. However, the MSSA algorithm has high computational complexity and is not suitable for the large-scale of time series data.

In this paper, an unsupervised change-point detection method is used to study the correlation of link state changes to eliminate the influence of RTT fluctuation. Prior works on change-point detection are in various fields [17], [18]. Rimondini et al. [8] first applied change detection to network measurement analysis. Their study adjusted the detection sensitivity to make the detected changes most relevant to the BGP changes of the target prefix. However, they ignored the changes of RTT caused by network anomalies. In addition, this study requires some kinds of tuning for each individual RTT time series, so it is difficult to apply this method to large-scale RTT data.

The proposed method uses twice of change-point detection method on large-scale RTT time series data which achieves better performance on accuracy compared to PCA-based and MSSA-based anomaly detection methods. Besides, its time overhead is acceptable.

III. Method

In this section, we first present a conceptual overview of the system design. And then we describe how to detect network event using a robust change detection method. Finally, we introduce the characterization method to identify the event-related links.

A. System overview

Fig. 1 illustrates the work-flow of our proposed method. Our method first extracts RTT data from the monitored network probes at equal time interval to form the RTT time series. Then we evaluate the performance of the entire monitored network, that is, we perceive whether there are network anomalies through the twice of change detection method. The first change detection method detects each RTT time series and records the number of change-points by time, which forms a time series of change-points of the whole network. The second change detection method detects the change-point time series and marks the time of abnormal points. On event characterization, we first measure the shape-based distance (SBD) between the RTT time series marked with change-point during the network anomaly period. The reason for the shape-based distance measure is that we observed that the jitter of links, which is related to events, is similar during the event. Then we use multidimensional scaling (MDS) to project the distance matrix into two-dimensional space. Finally, we locate the highest density region (i.e., the red zone shown in Fig. 1) for characterization of the event.

B. Anomaly detection

Acquisition of RTT time series between network probes is the fundamental work for the network anomaly detection. Generally, our method first collects traceroute data of the

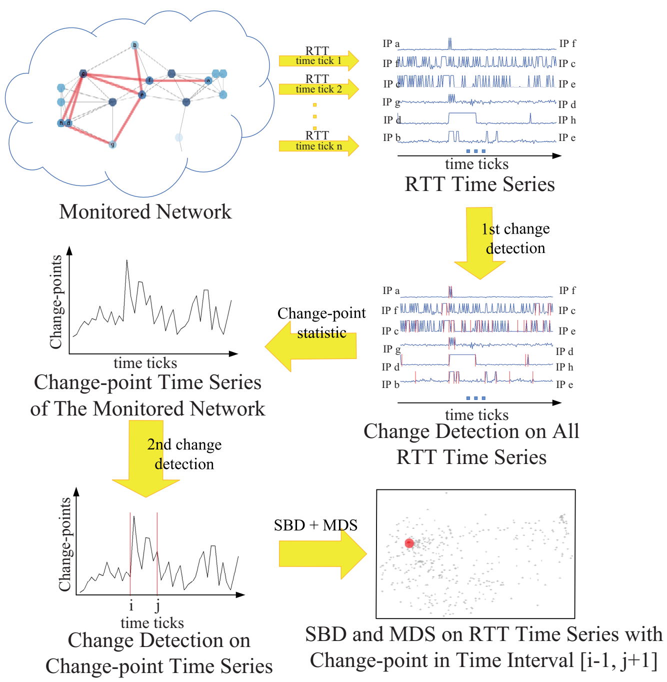

Fig. 1. Work-flow of our proposed method.

monitored network and calculate RTT time series betweens nodes in the trace. Then it performs twice of the change-point detection method:

1) Change detection for RTT time series, that is to detect state changes of every single links. Utilizing the scheme in this step, we can get a change-point time series $X_{1:t} = (X_1, \ldots, X_t)$ of the whole monitored network, where t is the length of the time series and $X_i = \alpha, \ (i \in [1, t])$ if there are change-points in α different links at time tick i, and is zero otherwise.
2) Change detection for change-point time series, that is to detect all the changes of $X_{1:t}$. If $X_{1:t}$ has a change-point at time $j, \ (j \in [1, t])$, it indicates that there is an unusual state change for the whole network at time j.

As with many other time series, end-to-end RTT time series can have sudden changes in level or volatility, often caused by delays or congestion. The points of cutting the time series into fragments with different characteristics is called change-points. The problem of detecting the most appropriate change points is called change-point detection. More formally, suppose we have an ordered sequence of data, $Y_{1:t} = (Y_1, \ldots, Y_t)$. An change-point occurs when there is a time τ, $\tau \in [1, t-1]$, which makes $(Y_1, \ldots, Y_\tau)$ and $(Y_{\tau+1}, \ldots, Y_t)$ showing different properties in some ways. Extending this idea to m ordered changepoints, $\tau_{1:m} = (\tau_1, \tau_2, \ldots \tau_m)$. τ_i is the position of i^{th} changepoints. We define $\tau_0 = 0$ and $\tau_{m+1} = t$. Together with the detected m changepoints, they cut $Y_{1:t}$ into $m+1$ segments, with the i_{th} segment containing $Y_{\tau_{i-1}+1:\tau_i}$. The cost function is calculated for each segment and the detection method strives to minimize the total cost of all segments:

$$\sum_{i=1}^{m+1} \left[\mathcal{C}\left(Y_{\tau_{i-1}:\tau_i - 1}\right)\right] + \beta f(m), \tag{1}$$

where $\mathcal{C}$ is a cost function for a segment and $\beta f(m)$ is a penalty to against over fitting. One commonly used cost function is negative maximum log-likelihood of the segment following a certain distribution [19], [20]:

$$\mathcal{C}\left(Y_{s:t}\right) = -\max_{\theta} \sum_{i=s}^{t} \log f\left(Y_i | \theta\right), \tag{2}$$

Where $f(Y|\theta)$ is a density function with distribution parameter θ. In this case, the choice of cost function is equivalent to the choice of distribution type, such as Normal, Exponential, Gamma and Poisson. When it comes to penalty, $f(m)$ is usually a function linearly related to the number of changepoints m:

$$f(m) = m + (m+1) dim(\theta), \tag{3}$$

where $dim(\theta)$ represents the dimension of the θ distribution (e.g., in the case of Normal distribution, $dim(\theta) = 2$). Common choices of β are information criteria, such as Akaike's Information Criterion (AIC) with $\beta = 2$, Schwarz Information Criterion (SIC, also known as BIC) with

$$\beta = \log t, \tag{4}$$

where t indicates the length of the time series. Hannan-Quinn Information Criterion with

$$\beta = 2 \log \log t. \tag{5}$$

Modified BIC (MBIC) with

$$\beta f(m) = -\frac{1}{2}\left[3 f(m) \log t + \sum_{i=1}^{m+1} \log\left(\tau_i / t - \tau_{i-1}/t\right)\right]. \tag{6}$$

From Eq. (4) - (6), we have MBIC > BIC > Hannan-Quinn. Note that the higher the penalty value, the lower the sensitivity of the detection and the better noise resistance.

As to end-to-end RTT time series change-point detection, the problem now is how to choose the most appropriate penalty and cost function/distribution among the wide variety of existing ones. In this work, we approximate change-point detection with Normal distribution since it has been reported to perform well for RTT time series analysis. According to Shao's work [21], the detection sensitivity of Normal distribution is higher than Poisson and Exponential distribution. This is because the mean and variance of the Normal distribution are independently controlled by two parameters, which increases the chance of finding subtle changes in fitting level or volatility. And the sensitive approach fits our needs, because we don't want to underreport any network exceptions. The remainder of this section details our anomaly detection method using change-point detection algorithm.

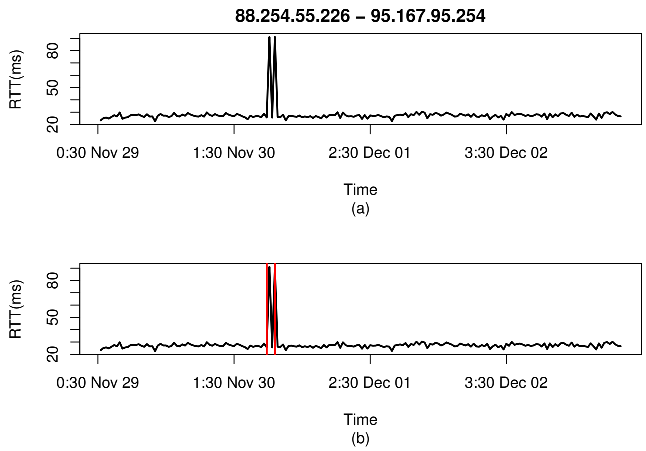

Fig. 2. 192 datapoints of an RTT time series. Red vertical lines correspond to the generated change-points.

*1) **Change-point detection of RTT time series**:* An example of RTT time series change-point detection is shown in Fig. 2, state changes are flagged by our method. Fig. 2(a) shows two real adjacent IP addresses (88.254.55.226 – 95.167.95.254) containing 192 RTT values over a period of 4 days (from November 29^{th}, 2015 to December 2^{nd}, 2015). Red vertical lines shown in Fig. 2(b) correspond to the generated change-points.

Real RTT time series are highly volatile. The causes of fluctuations include periodic congestion, path changes, routing strategy changes and so on. Therefore, network events cannot be identified through the state monitoring of a single link. Thus, we take advantage of twice of change-point detection method to determine network events by considering the correlation between link state changes.

*2) **Change-point detection of change-point time series**:* Traceroute/RTT contains information about hidden relations of links, such as passing through the same router, belonging to the same AS or ISP and experiencing the same network event. When a network failure occurs, it can affect multiple routes or paths. That is to say, there will be a lot of state changes of event-related links. Because of the volatility nature of RTT time series, a part of link state changes when there is no event in the network. However, the number of link state changes at the moment of event occurrence will obviously increase to an abnormal value. In addition, when the network anomaly is eliminated, the number of link state changes will drop significantly and fall back to a normal range.

In fact, our second change-point detection uses the central limit theorem (CLT) for network anomaly detection. According to CLT, regardless of the noise nature of RTT, the change-points (the number of link's state changes) of the whole monitored network will show a Normal distribution as the number of sample increases. If network anomalous events occur, the overall number of change-points in the monitored network will deviate from the Normal distribution, that is, it will be detected as anomaly.

Utilizing the scheme presented in the previous step (Sec. III-B1), we can get change-points of the whole monitored network. Assume the RTT time series of a link is defined as $l^i_{1:t}$ and we define a link-state time series $x^i_{1:t} = (x^i_1, \ldots, x^i_t)$. The RTT time series $l^i_{1:t}$ goes through the change-point detection method to get $x^i_{1:t}$ and $x^i_j = 1,\ (j \in [1, t])$ if the i^{th} link has an change-point on time tick j. We add all the link-state time series, i.e., $X_{1:t} = \sum_{i=1}^{n} x^i_{1:t}$, to get the change-point time series of the whole monitored network. Then change-point detection method is used again to detect the state changes of the whole network, and if there is a state change, it is determined that the network is abnormal. The detailed network anomaly detection algorithm is given in Alg. 1. According to the $\mathcal{N}_{1:t}$ returned by Alg. 1, the start and end time of network events can be detected relatively accurately according to the numerical changes of the overall link-state of the whole network.

Algorithm 1 Network anomaly detection

Require: Links' RTT time series $\mathcal{L} = \{l^1_{1:t}, \ldots, l^n_{1:t}\}$;
Ensure: Change-points of the total link-state time series $\mathcal{N}_{1:t}$;

$\mathcal{N}_{1:t} \leftarrow (0, \ldots, 0)_{1\times t}$;
$X_{1:t} \leftarrow (0, \ldots, 0)_{1\times t}$;
for each $l^i_{1:t} \in \mathcal{L}$ **do**
  $x^i_{1:t} \leftarrow \mathit{Changepoint\ Detection\ Procedure}(l^i_{1:t})$;
  $X_{1:t} \leftarrow X_{1:t} + x^i_{1:t}$;
end for
$\mathcal{N}_{1:t} \leftarrow \mathit{Changepoint\ Detection\ Procedure}(X_{1:t})$;
return $\mathcal{N}_{1:t}$;

Efficient computation of anomaly detection method: To minimize Eq. (1), we utilize the pruned exact linear time (PELT) algorithm [22] which can result in a time complexity of $O(t)$, where t indicates the length of the time series. And this is more computationally efficient compared with other algorithms due to the use of dynamic programming and pruning. Thus the time complexity of our anomaly detection method (i.e., twice change-point detection processes) is $O((n+1)\cdot t) \approx O(nt)$, where n and t indicate the number of time series and the length of time series, respectively.

C. Anomaly characterization

In the previous section, we described the process of using change-point detection method to detect network anomalies. For each time period found to be anomalous for our method, we also identify the nodes and links which are responsible for it. On the characterization of events, our proposed method first extracts the links which has change-point at or near the anomalous moments and then distinguishes between event-related and irrelevant links. Considering that the links related to the same event will have similar violent fluctuations during the occurrence of event, we first (1) *adopt a shape-based distance (SBD) measure* to extract hidden relations between links. Then we (2) *take advantage of multidimensional scaling (MDS)* to project the relations of links into a two-dimensional space for visualization and observing the relations between links. Finally, we will (3) *use kernel density estimate (KDE)* to obtain a 'relational densest region'. And this densest region will be used to locate the anomalous nodes/links and assess the impact of the event.

*1) **Shape-based distance**:* The shape-based similarity measure of time series needs to be able to handle the distance calculation of amplitude and phase distortion. One of the most commonly used measurement algorithms, Dynamic Time Warping (DTW) [23], is not suitable for our massive RTT time series because of its high computational complexity. Besides, cross-correlation is widely used as similarity measure in signal processing and KPI anomaly detection due to its high computational efficiency [24], [25]. Based on cross-correlation, Paparrizos et al. [26] proposed shape-based distance (SBD), which was applied to time series data and achieved good results. In this paper, we use SBD to measure the similarity of our RTT time series in order to distinguish links affected by events from links which are not related to events.

For two time series $Y_{1:t} = (Y_1, \ldots, Y_t)$ and $Z_{1:t} = (Z_1, \ldots, Z_t)$, cross-correlation keeps $Z_{1:t}$ unchanged and slides $Y_{1:t}$ over $Z_{1:t}$ to calculate the inner-product for each shift s of $Y_{1:t}$. We denote a shift of a sequence as follows:

$$Y_{(s)} = \begin{cases} (\overbrace{0, \ldots, 0}^{|s|}, Y_1, Y_2, \ldots, Y_{t-s}), & s \geq 0 \\ (Y_{1-s}, \ldots, Y_{t-1}, Y_t, \underbrace{0, \ldots, 0}_{|s|}), & s < 0. \end{cases} \tag{7}$$

For all possible shifts $s \in [-t+1, t-1]$, we can compute the inner-product $CC_s(Y_{1:t}, Z_{1:t})$ as the similarity between time series $Y_{1:t}$ and $Z_{1:t}$ with a phase shift s. It is defined as:

$$CC_s(Y_{1:t}, Z_{1:t}) = \begin{cases} \sum_{i=1}^{t-s} Y_{s+i} \cdot Z_i, & s \geq 0 \\ \sum_{i=1}^{t+s} Y_i \cdot Z_{i-s}, & s < 0. \end{cases} \tag{8}$$

The cross-correlation solves for the maximum value of Eq. (8), representing the similarity between $Y_{1:t}$ and $Z_{1:t}$ at an optimal phase shift s. Intuitively, at the best offset, similar patterns in $Y_{1:t}$ and $Z_{1:t}$ align to maximize the inner-product. Therefore, the cross-correlation measure can overcome the influence of phase shift and represent the shape similarity between two time series. In practice, a normalized cross-correlation (NCC) is widely used to limit the value to $[-1, 1]$ according to Eq. (9)

$$NCC(Y_{1:t}, Z_{1:t}) = \max_s \left(\frac{CC_s(Y_{1:t}, Z_{1:t})}{\|Y_{1:t}|_2 \cdot \|Z_{1:t}\|_2} \right). \tag{9}$$

Then we define SBD according to NCC [26]:

$$SBD(Y_{1:t}, Z_{1:t}) = 1 - NCC(Y_{1:t}, Z_{1:t}). \tag{10}$$

SBD ranges from 0 to 2, where 0 means two time series have exactly the same shape. A smaller SBD means higher shape similarity, conversely, a larger SBD means lower shape similarity. In this work, we use SBD to calculate the distances of RTT time series between links to measure the similarity of fluctuations.

*2) **Multidimensional scaling**:* Utilizing the schema present in the previous step, We extract the links with change-point during the network anomaly period as the suspected event-related link, and obtain a distance matrix by calculating the SBD between these links. The SBD matrix describes the RTTs' fluctuation similarity between the suspected links. It is known that the fluctuations of RTT time series of normal links are random and have different shapes. Nevertheless, during the event, the RTT fluctuations of the event-related link tends to have morphological similarity. Thus, we hope to find a morphologically similar region with the densest links using the distance matrix. In order to facilitate the subsequent density calculation and visualization, we first take advantage of multidimensional scaling (MDS) to project the distance matrix into two-dimensional space.

MDS is a dimensionality reduction algorithm which seeks a configuration, usually in a lower dimension, such that distances between the objects best match those in the original distance matrix [27]. Suppose there are n suspicious links, and the SBD of RTT time series between link i and link j is d_{ij}, $(i, j \in [1, n])$. In this work, we use a non-metric MDS [28] to find a configuration of points representing the links' RTT time series in two-dimensional space, where the approximate distances $\hat{d}_{ij}$ match closely as possible the original distances d_{ij} in some meaningful sense. To do this, we define $\hat{d}_{ij}$ as a function of the original distance d_{ij}, by $\hat{d}_{ij} = f(d_{ij})$, where f is a monotonic function such that $\hat{d}_{ij} \leq \hat{d}_{xy}$, $x, y \in [1, n]$ whenever $d_{ij} \leq d_{xy}$. For a particular configuration of points, MDS lets the standardized sum of squares of the differences between d_{ij} and $\hat{d}_{ij}$, also termed as $STRESS^2$, be defined as

$$STRESS^2 = \frac{\sum_{i,j} \left(d_{ij} - f(d_{ij})\right)^2}{\sum_{i,j} d_{ij}^2}. \tag{11}$$

The value of $STRESS$ is an indication as to how well the configuration represents the original distances. The objective is to find a configuration that has minimum $STRESS$ [29]. Usually, we minimize $STRESS$ over f by a gradient descent algorithm, for then f can be found by isotonic regression.

*3) **Kernel density estimation**:* In previous step, we use MDS to convert SBD to Euclidean distances in the form of points in two-dimensional space. Next, we apply two-dimensional kernel density estimation (KDE) to seek for the region with the highest density of links' relations for characterization of event-related links and nodes. The two-dimensional KDE is most straightforward for the normal kernel aligned with axes. The kernel estimate is

$$f(x, y) = \frac{\sum_i^n \phi\left(\left(x - x_i\right)/b_x\right) \phi\left(\left(y - y_i\right)/b_y\right)}{n b_x b_y}, \tag{12}$$

for a sample of points $(x_1, y_1), \ldots, (x_n, y_n)$, a fixed kernel ϕ and the bandwidth on axes b_x and b_y. Eq. 12 can be evaluated on a grid as XY^T where $X_{ji} = \phi\left(\left(gx_j - x_i\right)/\sqrt{n} b_x\right)$ and gx_j is the j^{th} grid point, and similarly for Y [27].

Now we can get the densest region of relations between links (i.e., the grid point with maximum density value). Then we find the closest point P_i, which represents a link actually, to this grid point. The k points most closest to P_i are obtained according to the SBD matrix (Sec. III-C1). The links and nodes represented by these k points are the event-related links and nodes we located.

Efficient computation of anomaly characterization method: From Eq. (8), the computation of CC_s for all values of s requires $O(\tilde{t}^2)$ time, where $\tilde{t}$ is the time series length i.e., the number of data points contained in a time series at the time of anomalous period. However, utilizing the convolution theorem and fast Fourier transform can reduce the computational complexity to $O(\tilde{t}log(\tilde{t}))$ [26]. Thus, the time complexity of SBD is $O(\frac{\tilde{n}(\tilde{n}-1)}{2} \cdot \tilde{t}log(\tilde{t}))$, where $\tilde{n}$ is the number of time series i.e., the number of suspicious links at the anomalous period. As to the MDS, An iterative algorithm is used in Eq. (11), which will usually converge in around 10 iterations. And this is necessarily an $O(\tilde{n}^2)$ calculation, where $\tilde{n}$ indicates the number of suspicious links. As to the KDE, The computational complexity is $O(g\tilde{n})$ given g grid points and $\tilde{n}$ sample points. In this paper, we set the number of grid points in each direction is 20, which means g = 400. Therefore, the total computational complexity of the proposed anomaly characterization method is $\Omega(\tilde{n}^2)$, making it prohibitively expensive for large data sets. However, the anomaly characterization only occurs when an anomaly is detected, and $\tilde{n} \ll n$, where n is the total number of time series in original data. Thus, the computational complexity of anomaly characterization is acceptable.

IV. Experiment Setup

In this section, we first describe the two experimental datasets (artificial data and real data), and then introduce the parameter settings of our proposed method.

A. Artificial time series and events

*1) **Data description**:* The simulated dataset which contains artificial time series and events are generated with the goal of testing the method in controlled enviroments. Considering the high noise nature of end-to-end RTT time series, we apply auto-regressive moving average (ARMA) model to simulate the RTT time series, which proved to do well in the end-to-end delay prediction [30].

*2) **Data generation**:* The $ARMA(p,q)$ model can be expressed as:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}. \tag{13}$$

On the basis of ensuring stationary of the time series, we randomly generate parameters p, q, φ_i, θ_j. Without loss of generality, we set $p+q \leq 3$, $\varphi_i, \theta_j \in [-1, 1]$. This model is then used to simulate the RTT time series of monitored links in this work.

As to artificial events, the four basic shapes (i.e., box, ramp-cliff, cliff-ramp and sine) from the classic Cylinder-Bell-Funnel dataset [31] are used. The *box* is characterized by a plateau from time tick a to b, the *ramp-cliff* by a gradual increase from time tick a to b followed by a sudden decline, and the *cliff-ramp* by a sudden increase at time tick a and a gradual decrease until b. The *sine* shape is usually used to represent a typical wave signal. These four shapes represent the typical morphology of events found in time series in many fields [32]. Fig. 3 shows an instance of each of the four shapes with some Gaussian noise added.

In the simulated dataset, the length of the event period (i.e., from time tick a to b) containing a shape is kept fixed to 128

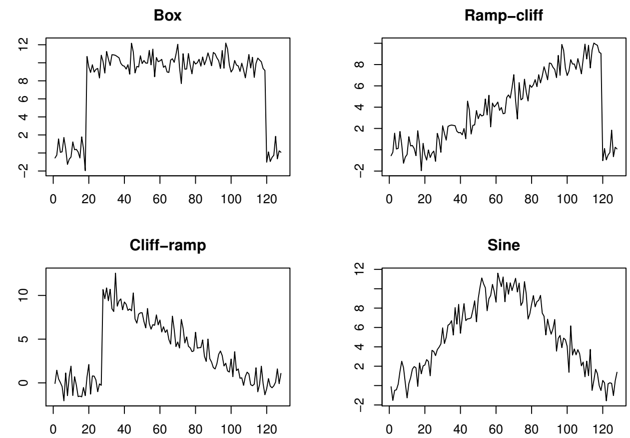

Fig. 3. Examples of box, ramp-cliff, cliff-ramp, and sine shapes.

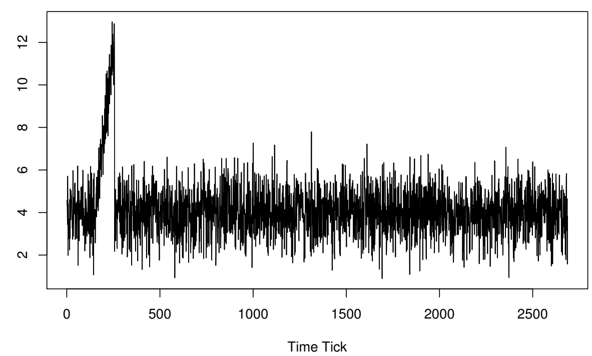

Fig. 4. An instance of generated RTT time series with artificial event.

time ticks. Thus, using the four basic shapes, the artificial event is generated. However, we were not able to directly evaluate the methods detection accuracy working only on time series with artificial events. To evaluate the impact of event scope on detection accuracy more directly, we need controlled experiments involving anomalies at varying intensions. To do this, the anomalous time series was mixed with different amounts of normal background time series. We simulated 10 network events, each of which is 128 time ticks apart. For each artificial event, we randomly selected 50 time series from the overall n (varying from 50 to 3200) simulated RTT time series and embedded the artificial shape with different levels of amplitude in a fixed time period as event-related links. An instance of generated RTT time series with the shape of ramp-cliff is shown in Fig. 4. And this link is the event-related link for the first simulated event with a start and end time tick from 129 to 256.

B. Real network measurement data

*1) **Data description**:* Our real dataset collection is done by downloading RIPE Atlas built-in measurements [6] with the API it provides. The RIPE Atlas built-in traceroute measurements are made up of traceroutes from all built-in probes to 13 DNS root servers every 30 minutes. Due to the widely

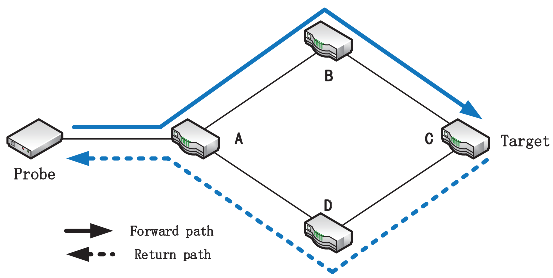

Fig. 5. An example of traceroute traffic asymmetry.

distribution of probes and anycast DNS root server deployment, this is actually traceroute data collected from more than 500 root servers. In this paper, We analyzed the built-in traceroute measurements from May 1^{st} to June 30^{th}, 2015 and November 1^{st} to December 31^{st}, 2015. Corresponding to a total of 1.01 billion IPv4 traceroutes. According to some of our exclusions, tens of thousands of RTT time series generated from the traceroutes per day.

*2) **Data preprocessing**:* However, using traceroutes to calculate RTTs of adjacent hops presents the challenge of traceroute traffic asymmetry due to the diversity of routing [33], [34]. Fig. 5 illustrates an example of round-trip traffic asymmetry. The solid line and dotted line represent the forward and return path, respectively.

To deal with this problem, we utilize the solution proposed by [3], which takes advantage of the path diversity of multiple probes to the same destination to precisely monitor the delay fluctuations of adjacent links. Let us revisit the example shown in Fig. 5. Suppose RTT_{PX} represents the RTT from probe P to a target X. The difference between the RTT from P to the adjacent routers B and C is noted as differential RTT Δ_{PBC} which is decomposed as follows:

$$\begin{aligned} \Delta_{PBC} &= RTT_{PC} - RTT_{PB} \\ &= \delta_{BC} + \delta_{CD} + \delta_{DA} - \delta_{BA} \\ &= \delta_{BC} + \varepsilon_{PBC} \end{aligned} \tag{14}$$

Where δ_{BC} is the delay of link l_{BC} and ε_{PBC} is the time difference between the two return paths. The value of δ_{BC} only depends on routers B and C, and is unrelated to the probe P. In contrast, ε_{PBC} is tied to P. Suppose we have n probes P_i, $i \in [1, n]$, all of the probes go forward through B and C, but each returns in a different path. Thus the differential RTTs Δ_{P_iBC} for all probe results has the same δ_{BC} and independent ε_{P_iBC}. The independence of ε_{P_iBC} also means the distribution of Δ_{P_iBC} will remain stable as the sample grows given that δ_{BC} is a constant. In contrast, a significant change in δ_{BC} affects all the different RTT values, and the distribution of Δ_{P_iBC} varies with δ_{BC} changes. Monitoring these changes allows us to discard the uncertainty in the return path (ε_{P_iBC}) and focus only on the delay changes of adjacent links (δ_{BC}).

In order to limit the impact of ε_{P_iBC}, we try to increase the diversity of the return paths by avoiding all the probes from the same AS. We designed two strategies to ensure the diversity of probes. The first strategy, which aims to ensure the diversity of the return paths, is that the probes which used to calculate the RTTs of adjacent hops must be from at least three different ASs. The second strategy uses normalized entropy to ensure a balanced number of probes per AS. Let $A = \{a_i | i \in [1, m]\}$ be the number of probes for each of the m ASs monitoring a certain link, then the entropy $H(A)$ is defined as:

$$H(A) = -\frac{1}{\ln m} \sum_{i=1}^{m} P(a_i) \ln P(a_i) \tag{15}$$

Low entropy, $H(A) \simeq 0$, means most probes are concentrated in one AS, while high entropy, $H(A) \simeq 1$, means probes are evenly distributed in all ASs. our second strategy ensures that $H(A) > 0.5$. If this is not met (i.e., $H(A) <= 0.5$), we will randomly select the probe from the AS which has the most probes (i.e., $a_i = max(A)$) and discarding it until the second strategy is satisfied.

Note that there are a lot of measurements for the end-to-end RTT from multiple probes of different ASs at one time tick. We use the median RTT which accounts for it does not fluctuate greatly due to significant changes of individual probes. Meanwhile, as to missing value in the measurement, we set it to a relatively large value (e.g., 3 times the maximum measured value of links' RTT). This is because the missing value may be caused by routing changes due to network congestion. For this, we want to be able to detect a link state change where there is a missing value.

C. Baseline methods

We compare our proposed method with the following baselines:

1) PCA-based method [9], [14]: The anomaly detection method is based on a separation of the high-dimensional space occupied by the set of RTT time series data into disjoint subspaces corresponding to normal and anomalous network conditions. For the characterization of the event, it computes a vector of features (e.g., lag correlation, strength of seasonality, spectral entropy, etc.) on each time series. Then it uses principal component decomposition on the features, and uses various bivariate outlier detection methods to locate the anomalous time series.
2) MSSA-based method [15]: It utilizes multivariate singular spectrum analysis (MSSA) to build a generative model for detection of changes in the characteristics of a random process. The model builds up a sliding window online anomaly detector which gives an anomalous score for a time tick in multiple time series data. For the characterization of the event, the links with the maximum deviation between the predicted value and the real value are regarded as the anomalous links.

For all these baseline methods, we have modified their base version to compatible with large-scale of time series data.

D. Evaluation metrics

We evaluate the methods performance in terms of two tasks i.e., accuracy and efficiency. We evaluate the precision, recall and time overhead between the proposed detection method and the baselines. As for the characterization of the event, we evaluate the Jaccard similarity and time cost. All these metrics are defined as follows:

1) Precision, which gives the fraction of true events reported over all reported events.
2) Recall, which gives the fraction of true events reported over all true events.
3) Time cost, which compares the anomaly detection and characterization time cost between all the methods.
4) Jaccard similarity, defined as $J(\hat{\mathcal{L}}, \mathcal{L}) = \frac{|\hat{\mathcal{L}} \cap \mathcal{L}|}{|\hat{\mathcal{L}} \cup \mathcal{L}|}$ for the located anomalous node/link set $\hat{\mathcal{L}}$ and the real anomalous node/link set $\mathcal{L}$.

E. Parameter configuration

In the process of the anomaly detection, we detected whether there is an anomaly using twice of the change-point detection algorithm. For cost function $\mathcal{C}$, we use Normal distribution. And we apply MBIC which has the strongest noise resistance, as a penalty, for the first change-point detection of RTT time series. This is because the RTT time series has strong noise and is easy to cause false postive. Meanwhile, for the second change-point detection of change-point time series, we apply BIC as the penalty which has a higher sensitivity so as to reduce the false negative. The size of detection time window W_1 was set to 3 time ticks for the change-point detection of single link's RTT time series and we use a larger window size W_2 of 5 time ticks for the change-point time series of the whole monitored network. In general, the larger the window gets, the more aggregated the alarms become. Note that the interval of each adjacent time tick in our real dataset is half an hour, this is because RIPE built-in measurements initiates traceroute every 30 minutes. In KDE, we simply apply Normal distribution as the kernal function.

V. Experiment results

In this section we present our experimental results of event detection and characterization both in the simulated and real network datasets using our proposed method.

All experiments are performed on a Linux platform with an AMD OPTERON X3216 (3.0GHz) and 32 GB DRAM memory, running Ubuntu. Our proposed method and the baseline methods are all implemented in R with publicly available [1].

A. Results on simulated dataset

*1) **Event detection**:* As described in Sec. IV-A2, our simulated dataset has n ($n \in [50, 3200]$) generated RTT time series and is embedded with 10 anomalous events. Each anomaly has 50 event-related links and each event lasts for 128 time ticks with spaced 128 time ticks apart.

Fig. 6 shows the ground-truth and our anomaly detection results on the simulated dataset where anomaly intensity is 12.5% i.e., event-related links account for 12.5% of all the monitored links ($n = 400$). The blue and green vertical lines

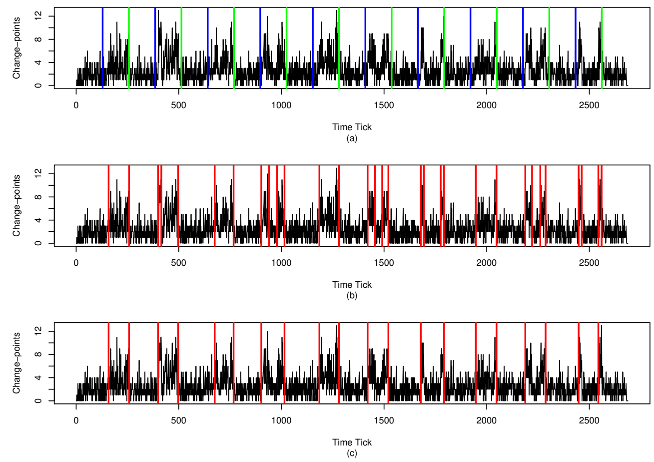

Fig. 6. Change-point detection of the change-point time series of the whole simulated network, where anomalous links comprise 12.5% of all the links.

shown in Fig. 6(a) indicate the start and end time of the event, respectively. The result shows that all the simulated events were detected. Note that change-points may also exist during the artificial events due to the large state changes of links during the event. Therefore, in order to determine the duration of the event, we aggregate the change-points which are close in time (e.g., the 2^{nd}, 6^{th} and 10^{th} artificial events shown in Fig. 6(b)). Fig. 6(c) shows the aggregate results which are consistent with the actual time of artificial events shown in Fig. 6(a).

*2) **Event characterization**:* After detecting anomalous events, we identify the nodes and links which are responsible for those events. In this section, we take the 1^{st}, 5^{th} and 10^{th} artificial events as examples to illustrate in detail, where event-related links account for 12.5% of all the monitored links (i.e., $n = 400$).

As shown in Fig. 6(c), the starting and ending time tick of the 1^{st} event is 158 and 258. In order to incorporate all event-related links into the subsequent analysis, we move the starting and ending time of the event forward and backward by $w(=10)$ time windows. That is, we extract all links with change-point during the time tick $148(=158-w)$ and $268(=258+w)$. Then we calculate the SBD between these links and apply MDS to convert SBD to Euclidean distances as shown in Fig. 7(a). The red hollow points indicate the 50 event-related links (i.e., the ground truth), while the blue cross points represent event-independent links (these links also have change-point in the anomaly period). we can see that the relations (distances) between event-ralated links are more intensive.

Next, we apply two-dimensional KDE to find the densest region and the grid point g_{max} with the largest density value shown in Fig. 7(b). Then we find the nearest point to g_{max} as a center p_c and obtain the $k(=50)$ nearest points to p_c according to the SBD matrix. The k nearest points $p_i,\ i \in [1, 50]$ are the event-related links we located. Among the $k(=50)$ suspicious links located by our charaterization method in the 1^{st} artificial event, the real event-related links is 49 which shows a high Jaccard similarity (i.e., 96.1%).

Fig. 8 and Fig. 9 show the MDS and KDE on SBD matrix

[1] https://github.com/hbn1987/Artt

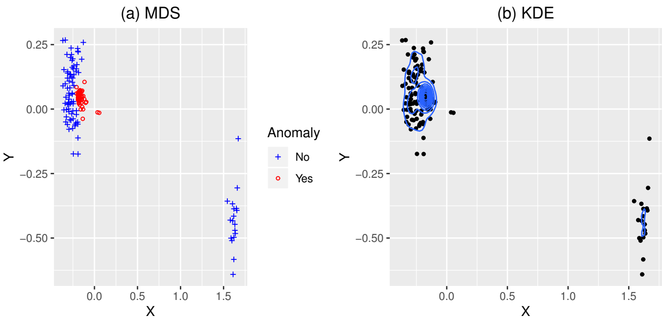

Fig. 7. MDS and KDE on SBD matrix of the 1^{st} artificial event, where anomalous links comprise 12.5% of all the links.

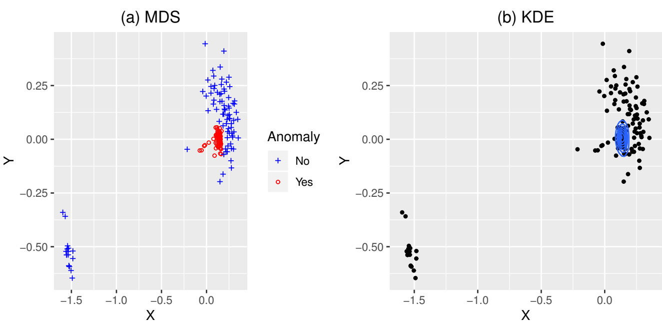

Fig. 8. MDS and KDE on SBD matrix of the 5^{th} artificial event, where anomalous links comprise 12.5% of all the links.

of the 5^{th} and 10^{th} artificial events, respectively. In the two artificial events, we located $k(=50)$ links with a Jaccard similarity of 92.3% and 69.5%, respectively.

3) ***Quantitative results:*** The detection precision and recall of the proposed and baseline methods at varying anomaly intensions are shown in Fig.10. First, all anomalies are easily detected when the proportion of event-related link is high. Second, we note that with the reduction of the proportion of event-related links, the detection accuracy of the proposed method gradually declined. This is because the event-related links must be greater than a certain threshold (i.e., the event has a certain influence on the network) in order for change-point detection algorithm to feel the state change of the overall monitored network. Third, the precision of our proposed method is higher than that of baselines in most cases of anomaly intensions,

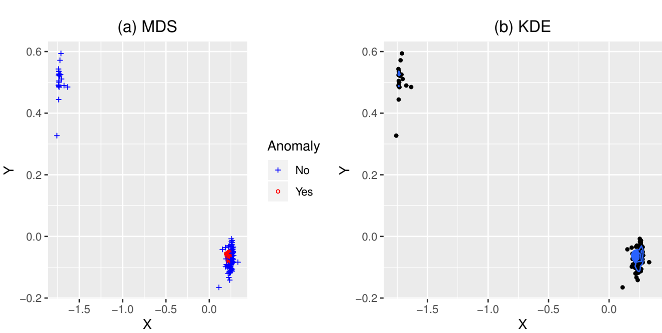

Fig. 9. MDS and KDE on SBD matrix of the 10^{th} artificial event, where anomalous links comprise 12.5% of all the links.

indicating that the twice change-point detection method can effectively resist the false positives caused by noises.

Fig. 11 shows the average Jaccard similarity of all the method at varying anomaly intensions. The average Jaccard similarity is calculated by the average of 10 artificial events and it measures the accuracy of the event characterization. Obviously, the accuracy of the proposed characterization method is much higher than that of baselines.

Fig. 12 shows the average detection and characterization time cost of all the methods at varying anomaly intensions. With the increase of data volume, the detection time cost of MSSA-based method increases rapidly and is several orders of magnitude higher than other methods as shown in Fig. 12(a). Thus, the MSSA-based detection method is not applicable to large-scale of time series data. As to the characterization time cost shown in Fig. 12(b). The time overhead of PCA-based method is orders of magnitude higher than other methods. This is because it requires multi-dimensional feature extraction for all links at the anomalous period, while our proposed method only analyzes links with state changes at anomalous period, which greatly reduces the amount of data required for analysis and improves the efficiency of event characterization.

B. Results on real network mearsurent dataset

In this section, we present three cases using the RIPE Atlas dataset [6] where traceroute data is preprocessed according to Sec. IV-B2.

1) ***Case 1: DDoS attack on DNS root servers:*** Our first case study shows the impact of a large distributed DDoS attacks on the performance of the network we monitored. According to the records of related researches [35], [36], there were two DDoS attacks against DNS root server during this event, which caused a large area of network anomalies. The first attack took place between 06:50 and 09:30 UTC on November 30^{th}, 2015 and the second between 05:10 and 06:10 UTC on December 1^{st}, 2015.

Event detection: Monitoring the change-points magnitude for the traceroutes show the two attacks in Fig. 13. The two peaks on November 30^{th}, 2015 and December 1^{st}, 2015 detected by our change-point detection method indicate that there are link state changes beyond the normal range in the network.

Event characterization: Fig. 14(a) and Fig. 14(b) show the MDS and KDE on the SBD matrix of different attack periods, i.e., the first attack on November 30^{th}, 2015 and the second on December 1^{st}, 2015.

We carry out the IP addresses we located which map for these event-related links on the first and second attacks shown in Fig. 15 and Fig. 16, respectively. The red node indicates the nodes corresponding to the 50 nearest links to the relation central link p_c. These links are the most suspicious links related to the event that we located. The nodes in purple represent the nodes corresponding to the 75 nearest links to link p_c. The grey nodes represent the corresponding nodes of the 100 links closest to link p_c. The links between nodes are represented by lines. Our method of troubleshooting based on the correlation level of nodes will help to locate the fault node quickly, thus greatly reducing the troubleshooting time.

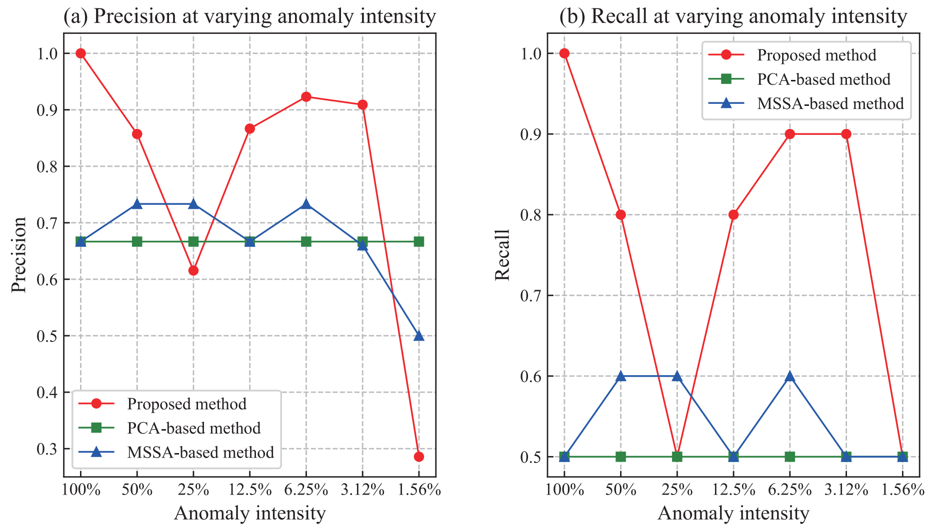

Fig. 10. The detection precision and recall of the proposed and baseline methods at varying anomaly intensions.

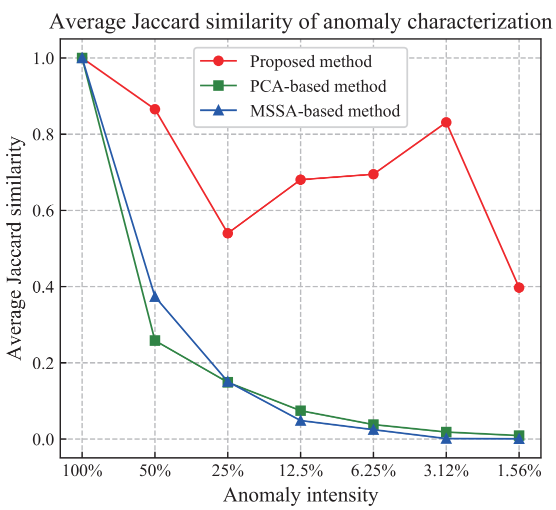

Fig. 11. The average Jaccard similarity of all the method at varying anomaly intensions.

*2) **Case 2: Telekom Malaysia BGP route leak**:* The second case study reveals a different type of network outage from the first one, a network event caused by exceptional routing traffic. On 12^{th} June 2015, Telekom Malaysia (AS4788) mistakenly sent BGP notices to its provider (Level(3) Global Crossing) at 08:43 UTC. The resulting traffic attraction to Telekom Malaysia had led to increase delays for Internet users around the world. The incident was acknowledged by Telekom Malaysia and reported by the BGP monitoring project [37], [38].

Event detection: Fig. 17 depicts the magnitude in terms of the change-points on the whole monitored network. The peak detected by the change-point detection method in Fig. 17 is 08:30 - 11:30 UTC on June 12^{th}, in good agreement with time reported in the Telekom Malaysia report [37].

Event characterization: We conducted MDS and KDE for SBD matrix of the links with state changes between $6:00(= 8:30-w)$ and $13:00(11:30+w)$ on June 12^{th} shown in Fig. 18, where $w = 5$ time ticks (i.e., 2.5 hours). Fig. 19 shows the IP addresses we located. The reverse DNS queries for these addresses show congestion in many European countries and this is consistent with the facts.

*3) **Case 3: Amsterdam Internet exchange outage**:* In this case, the network event was caused by a misconfiguration of an Internet switching device, which resulted in widespread connection problems in the Amsterdam Internet exchange (AMS-IX) around 10:20 UTC on May 13^{th}, 2015. This event prevents many networks from exchanging traffic via the AMS-IX platform, which in turn makes many Internet services unavailable [39]. AMS-IX reported that the problem was resolved at 10:30 UTC, but some reports indicate that network traffic and performance did not return to normal until 12:00 UTC [40].

Event detection: As shown in Fig. 20, there is a significant peak on May 13^{th} from 9:30 UTC to 11:30 UTC using our change-point detection method. This period coincides with the time of the event.

In this case, the detection of network anomaly is not through the change of raw RTT data. Packet loss or path changes during the event caused a lot of missing values of RTT. As described in Sec. IV-B2, we artificially set a large value for the missing ones. That is to say, when the change-point detection algorithm encounters a missing value, it is likely that this point will be regard as a state change point. When there is a large amount of missing data in the whole network, the state change of the network will be detected.

Event characterization: The SBD and KDE on SBD matrix is shown in Fig. 21 and Fig. 22 shows the event-related links and nodes we located according to the level of correlation.

VI. CONCLUSION

In this paper, we proposed an unsupervised approach for detecting and characterizing events in large-scale RTT time series. Our proposed twice change-point detection algorithm

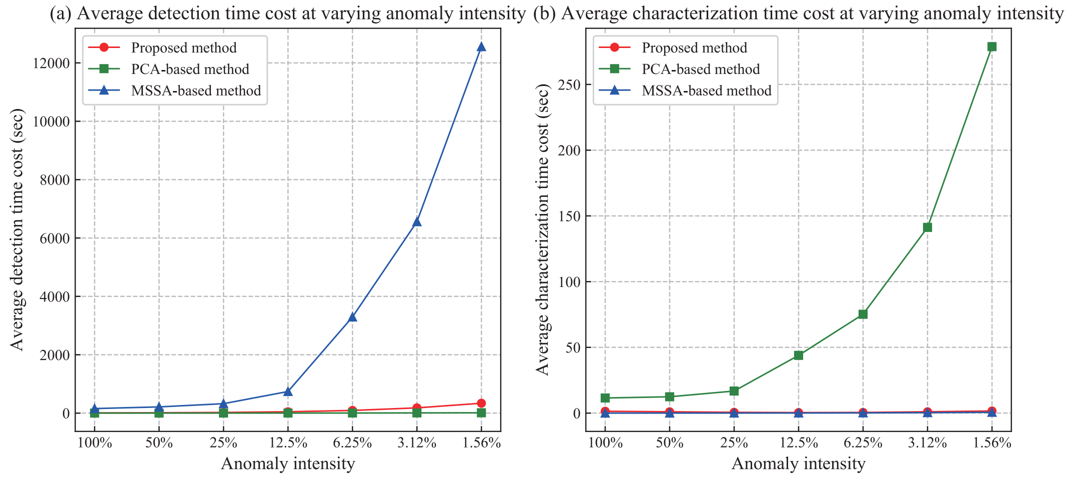

Fig. 12. The average detection and characterization time cost of all the methods at varying anomaly intensions.

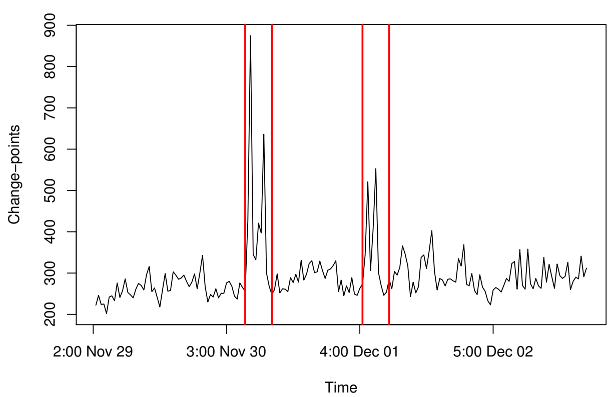

Fig. 13. The change-point detection on the change-point time series of the monitored network from November 29^{th}, 2015 to December 2^{nd}, 2015.

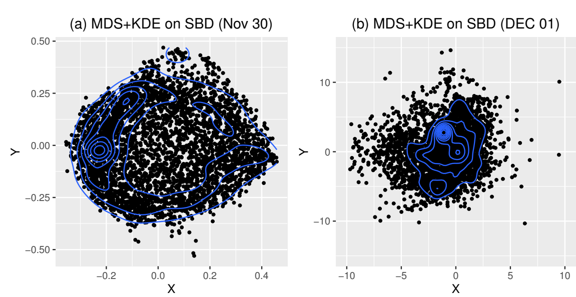

Fig. 14. MDS and KDE on the SBD matrix of different attack periods.

Fig. 15. A part of the visualization result on the located event-related nodes on November 30^{th}, 2015.

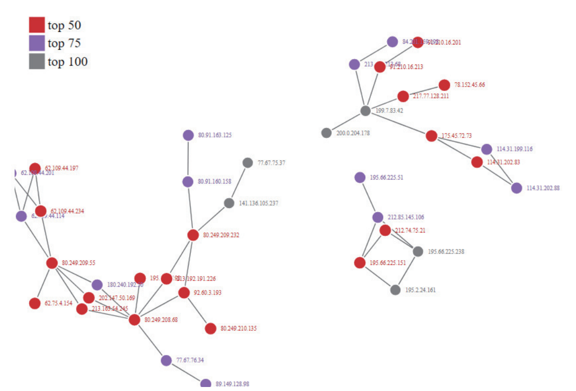

Fig. 16. A part of the visualization result on the located event-related nodes on December 1^{st}, 2015.

which greatly compresses the error alarms caused by the noise nature of RTT time series and improve the detection accuracy. Another key aspect of our method is its focus on characterization which incorporates three different techniques: a shape-based distance measure, a multidimensional scaling and the kernel density estimation, in addition to spotting suspicious event-related links, we also pinpoint the specific nodes according to correlation level that are most responsible for the anomaly.

We validated our proposed method on a simulated dataset of artificial time series and events. Our approach has successfully detected the anomalies, as well as unearthing the links and nodes responsible for those events with high accuracy. Additional experiments on a real network measurement dataset identified three major events with the suspicious nodes/links involved in those events which agree with the facts.

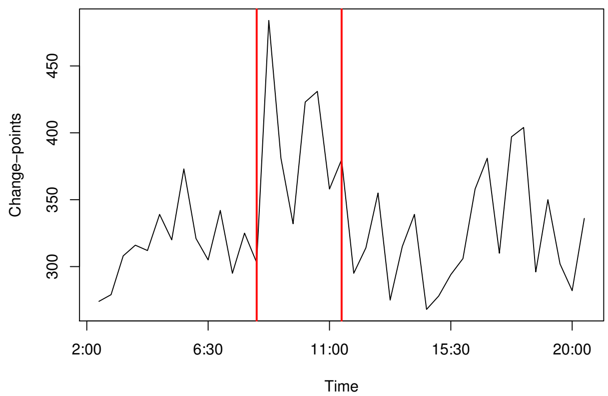

Fig. 17. The change-point detection on the change-point time series of the monitored network on June 12^{th}, 2015.

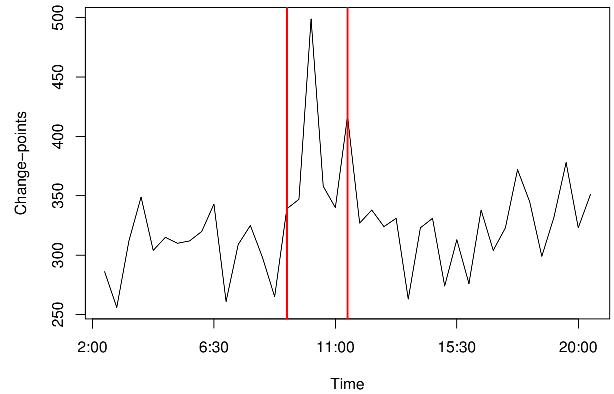

Fig. 20. The change-point detection on the change-point time series of the monitored network on May $13th$, 2015.

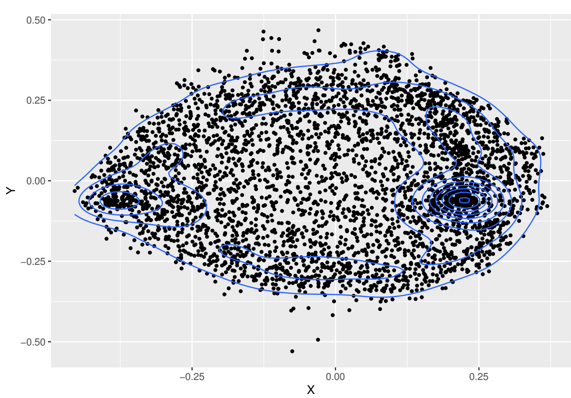

Fig. 18. MDS and KDE on the SBD matrix of Telekom Malaysia BGP route leak on June 12^{th}, 2015.

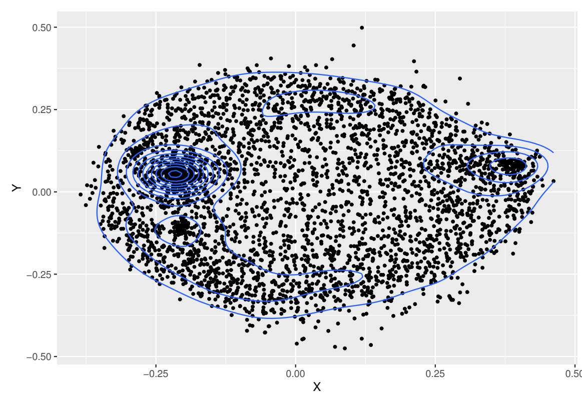

Fig. 21. MDS and KDE on the SBD matrix of Amsterdam Internet exchange outage on May $13th$, 2015.

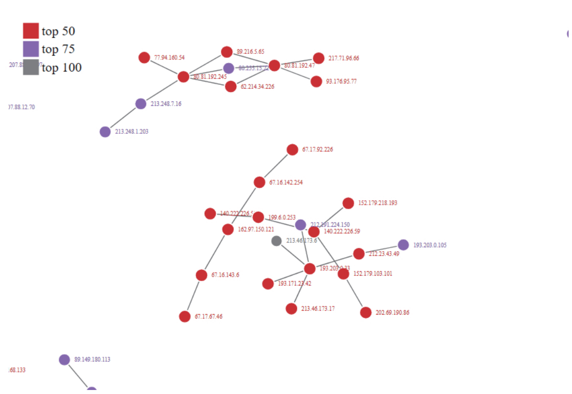

Fig. 19. A part of the visualization result on the located event-related nodes on June 12^{th}, 2015.

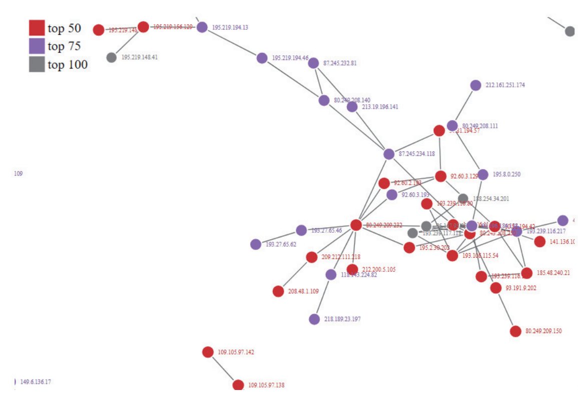

Fig. 22. A part of the visualization result on the located event-related nodes on May $13th$, 2015.

In short, our experimental results have provided evidence that our proposed approach is successful for event detection and characterization with high performance both in simulated dataset with ground truth and real dataset with real events. And its relatively accurate positioning will greatly reduce network troubleshooting time.

References

[1] J. Wei, Q. Zhang, and X. Li, "Network anomaly detection and localization," in *IEEE International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2016, pp. 8–13.

[2] W. W. T. Fok, X. Luo, R. Mok, and W. Li, "Monoscope: Automating network faults diagnosis based on active measurements," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2013, pp. 210–217.

[3] R. Fontugne, E. Aben, C. Pelsser, and R. Bush, "Pinpointing delay and forwarding anomalies using large-scale traceroute measurements," in *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2017, pp. 15–28.

[4] S. Rayana and L. Akoglu, "An ensemble approach for event detection and characterization in dynamic graphs," in *ACM SIGKDD Workshop on Outlier Detection and Description*, 2014.

[5] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *VLDB*, 2004.

[6] RIPE Atlas Built-in measurements. [Online]. Available: https://atlas.ripe.net/docs/built-in

[7] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu, "Understanding network delay changes caused by routing events," in *ACM SIGMETRICS performance evaluation review*, vol. 35, no. 1, 2007, pp. 73–84.

[8] M. Rimondini, C. Squarcella, and G. D. Battista, "Towards an automated investigation of the impact of bgp routing changes on network delay variations," in *International Conference on Passive and Active Network Measurement*. Springer, 2014, pp. 193–203.

[9] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM computer communication review*, 2004, pp. 219–230.

[10] A. Abdelkefi, Y. Jiang, W. Wang, A. Aslebo, and O. Kvittem, "Robust traffic anomaly detection with principal component pursuit," in *Proceedings of the ACM CoNEXT Student Workshop*, 2010, p. 10.

[11] M. Misra, H. H. Yue, S. J. Qin, and C. Ling, "Multivariate process monitoring and fault diagnosis by multi-scale pca," *Computers and Chemical Engineering*, vol. 26, no. 9, pp. 1281–1293, 2002.

[12] Z. Chen, K. Y. Chai, S. L. F. Bu, and C. T. Lau, "Combining mic feature selection and feature-based mspca for network traffic anomaly detection," in *IEEE International Conference on Digital Information Processing, Data Mining, and Wireless Communications*, 2016, pp. 176–181.

[13] Y. Zhang, S. Debroy, and P. Calyam, "Network-Wide Anomaly Event Detection and Diagnosis With perfSONAR," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 666–680, 2016.

[14] R. J. Hyndman, E. Wang, and N. Laptev, "Large-Scale Unusual Time Series Detection," in *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 1616–1619.

[15] Q. Dong, Z. Yang, Y. Chen, X. Li, and K. Zeng, "Exploration of Singular Spectrum Analysis for Online Anomaly Detection in CRNs," *ICST Transactions on Security and Safety*, vol. 4, no. 12, 2017.

[16] H. Hassani and R. Mahmoudvand, "Multivariate Singular Spectrum Analysis: A General View and New Vector Forecasting Approach," *International Journal of Energy and Statistics*, vol. 1, no. 1, pp. 55–83, 2013.

[17] C. Jie and A. K. Gupta, "Parametric statistical change point analysis: with applications to genetics," *Medicine and Finance*, 2012.

[18] K. Haynes, P. Fearnhead, and I. A. Eckley, "A computationally efficient nonparametric approach for changepoint detection," *Statistics and Computing*, vol. 27, no. 5, pp. 1293–1305, 2017.

[19] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2011.

[20] L. Horvath, "The maximum likelihood method for testing changes in the parameters of normal observations," *Annals of Statistics*, vol. 21, no. 2, pp. 671–680, 1993.

[21] W. Shao, J. L. Rougier, A. Paris, F. Devienne, and M. Viste, "One-to-one matching of rtt and path changes," in *IEEE International Teletraffic Congress (ITC)*, 2017, pp. 196–240.

[22] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal Detection of Changepoints With a Linear Computational Cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[23] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 2003.

[24] T. W. Liao, "Clustering of time series data - a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[25] Z. Li, Y. Zhao, R. Liu, and D. Pei, "Robust and rapid clustering of kpis for large-scale anomaly detection," in *IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–10.

[26] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2015, pp. 69–76.

[27] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. Springer, 2002.

[28] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[29] T. F. Cox, "Multidimensional scaling in process control," in *Handbook of Statistics*, 2003, vol. 22, pp. 609–623.

[30] M. Yang, A. Bashi, J. Ru, X. R. Li, H. Chen, and N. S. Rao, "Predicting internet end-to-end delay: A statistical study," *Annual Review of Communications*, vol. 58, pp. 665–677, 2005.

[31] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," in *ACM SIGKDD*, 2002.

[32] Y. Kang, D. Belušić, and K. Smith-Miles, "Detecting and classifying events in noisy time series," *Journal of the Atmospheric Sciences*, vol. 71, no. 3, pp. 1090–1104, 2014.

[33] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "In search of path diversity in isp networks," in *ACM Internet Measurement Conference (IMC)*, 2003, pp. 313–38.

[34] Z. Han, E. K. Lua, M. Pias, and T. G. Griffin, "Internet Routing Policies and Round-Trip-Times," in *International Workshop on Passive and Active Network Measurement (PAM)*. Springer, 2005, pp. 236–250.

[35] Root server operators events of 2015-11-30. [Online]. Available: http://www.root-servers.org/ news/events-of-20151130.txt.

[36] M. Weinberg and D. Wessels, "Review and analysis of attack traffic against a-root and j-root on november 30 and december 1, 2015," OARC 24, Tech. Rep., 2016.

[37] Telekom malaysia: Internet services disruption. [Online]. Available: https://www.tm.com.my/OnlineHelp/Announcement/Pages/internet-services-disruption-12-June-2015.aspx

[38] Route leak causes global outage in level 3 network. [Online]. Available: https://blog.thousandeyes.com/route-leak-causes-global-outage-level-3-network

[39] Follow-up on previous incident at ams-ix platform. [Online]. Available: https://ams-ix.net/newsitems/195

[40] Does the internet route around damage? a case study using ripe atlas. [Online]. Available: https://labs.ripe.net/Members/emileaben/does-the-internet-route-around-damage