DESTIN: Detecting State Transitions in Network elements

Parisa Foroughi[†‡], Wenqin Shao[‡], Frank Brockners[‡], and Jean-Louis Rougier[†]
[†] Télécom Paris - Department of Networks and Computer Science
[‡] Cisco Systems
Email: parisa.foroughi,rougier@telecom-paris.fr; pforough, wenshao, fbrockne@cisco.com

***Abstract*—Operators are interested in gaining a comprehensive assessment of their network elements and tracking operational changes. Commonly, this assessment is achieved by performing regular checks of different operational counters and defining expert rules from known root causes. The common approach requires the maintenance of a regularly updated set of rules and only goes as far as the operator's pre-gained knowledge of the system. In this paper, a broader set of counters (not limited to the handpicked Key Performance Indicators (KPIs)) is explored with an unsupervised approach. The goal is to leverage the dependencies between the counters in order to discover complex state changes that might have otherwise slipped the operator's view. This paper proposes DESTIN, a multivariate unsupervised change detection for high dimensional time-series data of originally low effective dimension, which provides near real-time state assessment of network device. The efficiency of the method is demonstrated on an experimental test-bed.**

***Index Terms*—Machine learning, Principal angles, Network management, Change detection.**

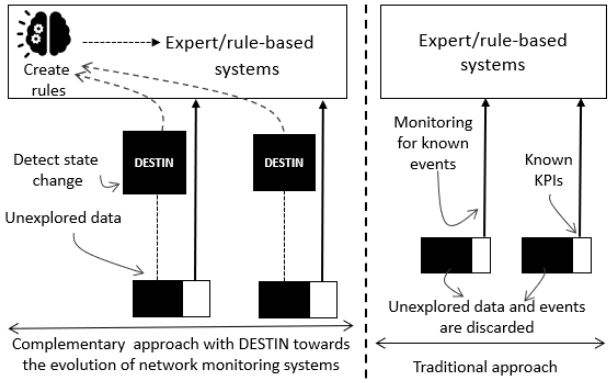

Fig. 1. The complementary scenario of DESTIN in expert and rule-based systems for improving the operators insight of the network element

I. Introduction

Network operators are striving to keep an up to date assessment of the state of their network device and track any operational changes in the device states. A common approach is for an operator to continuously monitor a few known operational counters [1], [2] also known as Key Performance Indicators (KPIs). This approach limits the operator's view to known operational state changes. A network element such as a router or switch can offer hundreds of thousands of operational counters to monitor [3] which can be retrieved by the Simple Network Management Protocol (SNMP) or via streaming telemetry [4]. The set of counters which is available but not monitored could contain additional information, which DESTIN is able to distill.

Fig. 1 depicts the complementary role of DESTIN, the unsupervised multivariate approach proposed in this paper, with expert and rule-based systems in improving the operators knowledge of the state of the network element. In the proposed scheme shown in Fig. 1, DESTIN explores the otherwise ignored counters and analyzes its dependencies in an unsupervised way and sends a notification when a change of state is discovered. The operator can further analyze the flagged timestamp from DESTIN using data-driven methods such as the one proposed by Feltin et al. [5] to distill most representative counters and define new rules and KPIs. The collection of a set of n counters, collected with a regular cadence over m timesteps forms a $n \times m$ matrix. Visualizing this high-dimensional matrix with tools like t-stochastic neighbor embedding (t-SNE) [6] or Uniform Manifold Approximation and Projection (UMAP) [7] shows structure in the data. Fig. 2 visualizes a dataset using t-SNE from a router ($n = 6622$, $m = 1079$) revealing a set of clusters as expected. One can see that the state of the network element, as represented by the n-dimensional vector either does not change much and stays within a *cluster* or shows a significant change. It is also seen that due to the dependencies of the counters, they can be represented in significantly lower dimensions than that of the original. Therefore, to deal with the large amount of unexplored counters while leveraging the hidden dependencies, this paper proposes an unsupervised multivariate approach for data of low effective dimensionality.

This paper presents DESTIN, an online unsupervised change detection method based on principal angles between subspaces, which is well-suited to detect temporal state changes in noisy operational data of a network element. This multivariate method leverages the expected inter-dependencies of counters to flag the potential points in times which can include valuable information for a network operator to expand their knowledge of the KPIs. The performance of DESTIN is evaluated and compared with the well-established subspace-based method, Principal Component Analysis (PCA), in terms of recall, precision and time to detection under some scenarios

978-3-903176-32-4

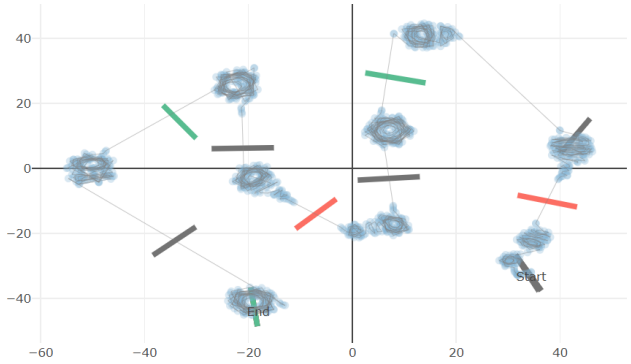

Fig. 2. t-SNE visualization of a dataset [4] of 6622 counters with perplexity=30, the bold bars are intentional inserted events- The x and y axis values are relative

distilled from real world incidents.

The rest of the paper is organized as follows: a comparative overview of the state of the art with DESTIN is given in Section II. The practical architecture for the method application is depicted in Section III. Section IV explains DESTIN's core methodology in terms of parameters, applicability etc. using a geometrical interpretation. Section V includes the data properties and dataset generation. The results are presented in Section VI. Section VII concludes the paper.

II. Related Work

This section provides a comparative overview of some of the common methods used in networking context for change detection and analysis which have similar mathematical background.

Lakhina et al. first proposed Principal Component Analysis (PCA) [8] as a technique that is applicable to link-traffic data obtained from SNMP that is capable of detecting network-wide anomalies. The authors refined their approach by proposing a combination of the PCA subspace with a distribution based technique using entropy [9] to detect intrusion attacks for flow data. Ringberg et al. [10] state the limitations of PCA for traffic anomaly detection. The sensitivity of the false-positive rate to the number of principal components, the contamination of normal subspace, the hardship in identifying the right flows, and the right traffic aggregation level for input are the main axis of the paper. Brauckhuff et al. [11] tackle the limitation of the PCA in detecting temporal changes. They propose to use Karhunen-Loeve transformation [12] along with the covariance matrix. Kwitt et al. [13] propose a robust PCA transforming it into an unsupervised approach by using Minimum Covariance Determinant (MCD) estimators. Cardot et al. [14] investigate and provide a comprehensive view of how to use PCA for online change detection in high dimensional data. Vaswani et al. [15] investigate the different approaches to improve the robustness of the PCA and validate their proposed approaches on images and social networks. Ding et al. [16] propose a compressed version of PCA, assuming a spiked covariance matrix for high dimensional data. Their proposed method can track temporal changes and is at the same time an online solution while suitable for high dimensional data of low intrinsic dimensions. Despite all the efforts to improve PCA, combining all the approaches is not trivial if at all possible. Moreover, some of the enhancements use rigid constraints for data that may count as over simplification considering the nature of networking data. The proposed method in this paper can detect changes in an online unsupervised manner with higher precision compared to PCA. It is also robust enough to filter out noise and symmetric spikes in traffic data. The results in Section VI-B depict the improvement over PCA in detection of changes of states.

Another group of techniques are covariance matrix-based approaches. Yeung et al. [17] propose to detect flooding attacks by tracking the covariance matrix and correlations of data. Haung et al. [18], [19] propose a communication-efficient approach to apply PCA to a network. They define a filter based on the covariance matrix and stochastic perturbation theory to only send information from a router to a coordinator when necessary. The same authors propose a new approach to dimensionality reduction based on covariance matrix properties and a distance metric [20]. Compared to the prior work, this paper consumes the original input data rather than computing the covariance matrix. DESTIN is computationally efficient in comparison with the state of the art approaches because of its capability of working over small windows of data-points.

Principle angles which is the core of DESTIN have been originally investigated in the context of auto-regressive-moving-average (ARMA) models [21] to find the hidden relations between the linear systems and their input parameters. In addition to ARMA models, Yuan et al. [22] and Lie et al. [23] use canonical correlations which is an extension of the original principal angles method for information fusion and similarity discovery between groups of variables. Wolf et al. [24] also introduce an improved principal angles that facilitates searching for non-linear patterns by using a kernel. The principal angles technique has been previously used in information fusion and control systems for linear systems but it is not yet used for detecting changes in high dimensional multivariate data. This paper build on the method capabilities and provide guides on how to customize the method for change detection.

III. Architecture and Setup

This section sheds light on practical considerations of the proposed method's deployment on a router. Fig. 3 depicts the building blocks of a working scenario for DESTIN. It also shows how DESTIN can be deployed in conjunction with the existing technology and breaks down its important building blocks. DESTIN can be deployed either on the router or on an analytics management server which receives continuous data streams from the router. The proposed scenario in Fig. 3 has three main building blocks: collector, detector and exporter plus an optional block of explainer that can help the operator in defining the new rules. The collector is responsible for data retrieval from a router and then parsing the data to time-series (Fig. 3). The collection mechanism

can be a simple periodic SNMP data retrieval mechanism or it can use streaming telemetry to form regularly time-spaced collections of a router statistics in forms of time-series. The detector (DESTIN) takes care of pre-processing which consists of first grouping data by choosing the intended time-series targets (*e.g.*, interface level traffics) and then scaling the time-series data to make differently loaded interfaces comparable in value. The explainer/descriptor is an optional component which is triggered by the notification generated by DESTIN. It can provide hints to the operator regarding the most representative counters of the event. The exporting block functionality depends on the overall on-box or off-box approach. In case DESTIN is deployed on the box, the exportation is event-triggered and so there is no need for continuously exporting counters.

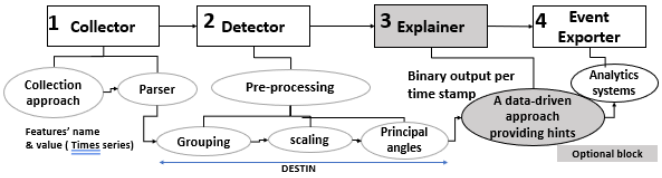

Fig. 3. Building blocks' details for state assessment of network element-optional, existing and proposed blocks

IV. Methodology: Principal angles

In this section, an overview of DESTIN's mathematical and geometrical meaning is presented. Section IV-A includes the definition of a similarity measure τ composed of the cosine of the angles for two matrices A and B which are formed from the streaming telemetry. A geometric interpretation of the principal angles in DESTIN is used for reasoning about the choices made throughout this section. Section IV-B proposes two variations of DESTIN, for change detection using principal angles' concept for groups of operational data. Behind the choices made in Section IV-B lies the observations in Section V. The figures in this section are produced using a sample dataset [25]. For more information see Section V-B.

Principal angles (PA), also known as subspace angles [26], is a subspace-based method that uses angles to present the similarity of two matrices, where a column is an observation vector. The time series collected from a device can form a matrix of observations for any time interval. Let $A^{n\times j}$ and $B^{n\times k}$ be two column matrices with the real-valued column vectors a_i, $i = 1, 2, ..., j$ and b_i, $i = 1, 2, ..., k$, respectively where each column vector is of size n (number of counters/features) and, j and k are the number of observations (data-points). In practice, A and B are the reference and test matrices. Assuming that $\mathcal{A}$ and $\mathcal{B}$ are orthonormal subspaces that span the matrices A and B in $\mathbb{R}^n$ space, and that $p = dim\mathcal{A} \geq q = dim\mathcal{B} \geq 1.$, the angles $\theta_i, i = 1, ..., q$ [27] are defined as follows :

$$cos(\theta_k) = \max_{u\in\mathcal{A}} \max_{v\in\mathcal{B}} u^T v = u_k^T v_k \tag{1}$$

$$s.t \ \ \|u\| = \|v\| = 1, \ u^T u_i = 0, \ v^T v_i = 0, \quad \forall i = 1, ..., k-1$$

Assumption 4-1 : Without loss of generality, it can be assumed that $j = k$. This assumption implies that the comparison is between the same number of data-points.

Assumption 4-2 : For the sake of near-real time detection, the time windows (number of data-points in a matrix) are considered to be less than n. Adding Assumption 4-1, it translates into $\frac{n}{j} > 1$.

A. Principal angles' parametrization

1) Principal angles' parameters : To determine a change indicator from principal angles, the first step is to explore the angles' properties. Principal angles between subspaces $\mathcal{A}$ and $\mathcal{B}$ are the rotations along their orthonormal axes to make the two subspaces overlap [28]. Therefore, the cosine of the angles are the indicators of similarities between the two subspaces [26]. Principal angles are to represent all the information in a given subspace. In practical measurements, the values which form the subspaces can be divided into two groups of signal and noise. In this context, noise and signal terminology is meant as a notation to distinguish intentional triggered changes from the background changes related to the network dynamics. Therefore, principal angles is not applicable to change detection in its original form and requires modifications. Filtering of the noise in principal angles can be controlled by two main parameters: i) the number of orthonormal vectors (n_v), and ii) the number of angles (n_a). As shall be seen, n_v is the number of axes which contain the most important information of a device state and n_a is the number of rotations needed to overlap the subspaces while filtering out as much noise as possible. The reduced subspace from choosing n_v orthonormal vectors from the existing ones is presented as $\mathcal{S}(n_v, A, B)$. This function reduces the subspaces of $\mathcal{A}$ and $\mathcal{B}$ to n_v principal vectors.

Remark 4-1 : The orthonormal bases of a matrix of size $n \times j$ is limited to the rank of the matrix (i.e. $min(n, j)$). Based on Assumptions 4-1 and 4-2, the principal vectors of matrices A and B for the reduced subspaces of $\mathcal{S}(n_v, A, B)$ is upper bounded ($n_v <= j$).

Remark 4-2 : Based on the geometrical interpretation, the angles in PA are presented in an increasing order. This is because the angles are sequentially calculated from the vectors of strongest signal to the least. The more noise in the vectors, the larger the rotation needed for the overlap of subspaces. Therefore, for fixed subspace dimension of n_v, the average value of the cosine of angles is a decreasing function of n_a.

Remark 4-3 : Based on Remark 4-2, the maximum noise is seen in the last angles of a given n_v. Therefore, increasing n_v pushes the noise to higher orders of n_a. Thus, for a fixed n_a, the average value of the cosine of angles is an increasing function of the number of vectors.

The change indicator for PA in this paper is defined to be a product of cosines rather than the average to avoid smoothing the differences in test and reference subspaces. The change indicator is defined as follows:

$$\tau(\mathcal{S}(n_v), n_a) = \prod_{i=1}^{n_a} cos(\theta_i) \tag{2}$$

B. Principal angles' applicability

In this section a gap metric is introduced to test the method's applicability and measure the distance between values in *normal* states and states in which changes occur. Fig. 4, illustrates a change scenario, the data blocks and the corresponding matrices are notated as X, Y, C and x, y, c, respectively. Note that matrix c should contain the change point. XX denotes the comparison of several sets of A and B where both matrices are selected from state X.

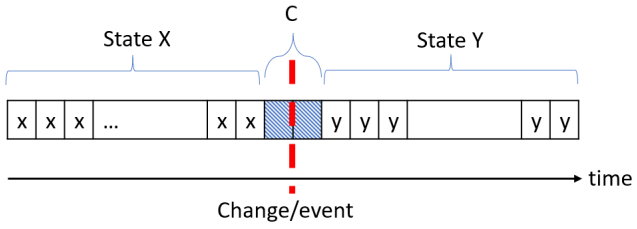

Fig. 4. illustration of a change scenario- x and y are example matrices of features in two different states, C is the interval of change which includes one or more data-points impacted by the change (event)

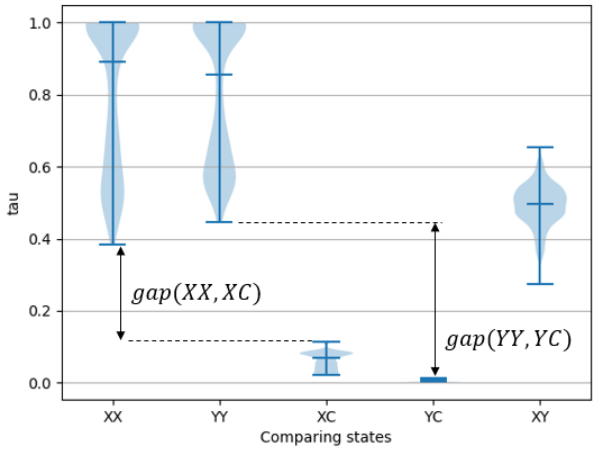

Fig. 5. Violin plot of τ ,(2), values for comparing different states of a device when $n_v = 6$ and $n_a = 3$ - Preprocessing approach is scaling by average values with WSS assumption over all the dataset when $j = k = 10$

Fig. 5 illustrates the violin plot of the defined similarity measure τ (2) when comparing different *states* of a device. The set of values for τ_{XX} shows the dynamics of the network while in state X. It can be seen despite the dynamics, there is a gap between the τ values of the same states (τ_{XX}) and change states (τ_{XC}). It is also seen that the similarity measure τ is larger for similar states compared to when a change happens. It is because in similar states, less rotations are needed to overlap the subspaces than when a change happens. The possibility of detection of a change therefore depends on the existence of the defined gap where it is defined as follows:

$$gap(XX, XC) = min(\tau_{XX}) - max(\tau_{XC}) \tag{3}$$

Based on (3), the following scenarios can exist:

1) Data with $gap(XX, XC) > 0$: a gap exists when the following two conditions are satisfied:

(I) stability: τ_{XX} values maintain a rather stable value meaning that the rotations necessary to overlap the two subspaces while in the same state should be small.

(II) τ_{XC} values should be smaller than the τ_{XX} values.

Each data-point in the data blocks X, Y and C, has an allocated time called timestamp. Therefore, when selecting matrices from data-blocks, they could be close or far away in time. As long as no event happens, the matrices which are closer in time, will require less rotation to overlap (i.e. has high τ values) compared to matrices which are far away in time when in normal states.

Remark 4-4 : Based on Remark 4-2, To avoid including the noise in τ values, not all the angles are considered in the product.

Remark 4-5 : Based on Remarks 4-2, 4-3 and 4-4, the proper number of angles to filter the noise and contain enough signal is distanced from the first few (Remark 4-3) and last few (Remarks 4-2, 4-4) angles. (See Fig. 6 for illustration)

Remark 4-5 decreases the search space and speeds up the search for the right n_a as opposed to the algorithms in [20].

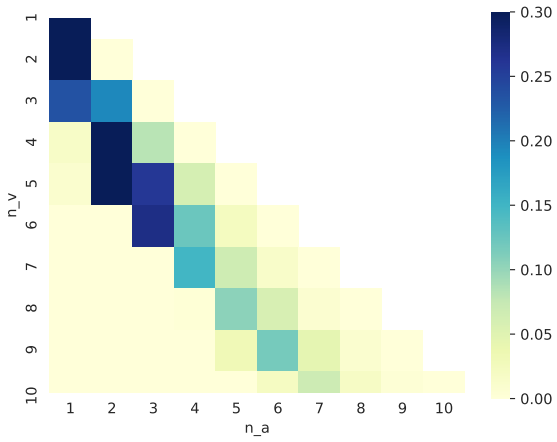

Fig. 6. Heatmap of $gap(XX, XC)$ for when A and B are not overlapping - The first even data interval in dataset available in [25] is used for illustration.

2) Data without initial $gap(XX, XC) > 0$: It is not reasonable to assume that the gap in (3) always exists. Often it is due to the violation of the first condition (stability) that the gap disappears. Overlapping the reference (A) and test (B) matrices by a few data-points can create a dummy stability for τ values. This action results in less rotations when in the same state and thus higher τ values compared to change states.

Remark 4-6 : Overlapping the matrices A and B can generate the stability necessary to satisfy the first condition of maintaining a non-zero and positive gap.

In addition to Remark 4-6, to satisfy the existence of the gap, the second condition should also hold. The next remark explains why the second condition will naturally be satisfied.

Remark 4-7: It can be shown that the normal state has a higher dimension compared to the state which includes a change. In other words, when the two subspaces are reduced by n_v vectors, the change subspace has much more noise in terms of principal angles and thus a lower τ value. Therefore, τ_{XC} is surely less than τ_{XX}. This property is further discussed in Section V and noted in Remark 5-2.

Remarks 4-6 and 4-7 ensure the existence of a gap and the number of angles can be chosen using Remark 4-5. Fig. 7 shows an example of the data behavior after removing the trend and overlapping the same data of Fig. 6. IV-B3 and IV-B4 propose parameters based on the logical analysis of this section to eliminate the need in most cases for a search of the right number of angles and vectors.

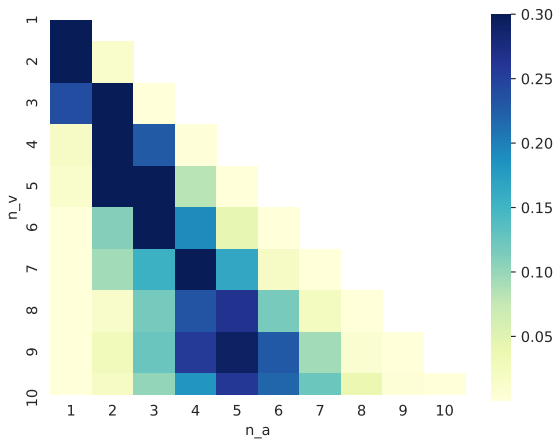

Fig. 7. Heatmap of $gap(XX, XC)$ for when A and B are overlapping by 5 data-points - dataset available in [25]

3) Specific Principal Angles (SPA) – for data with trend: When data has a known trend such as an increasing trend, leveraging this special property can introduce a fixed set of parameters when using principal angles. In such cases, if the trend is not removed, the most important signal is the result of the trend shape and the rate by which it changes. Therefore, based on Remark 4-2, only one angle which includes the strongest common signal is enough. Based on Remark 4-3, taking only the first vector is the best option since higher numbers will result in the leakage of signal on to other angles. Thus, in such cases, $n_v = n_a = 1$ can provide a trustworthy change detection. Note that the trend already ensures the stability of τ_{XX} values hence no overlap is required. Traffic counters of streaming telemetry is a good usecase example of such data. Rather it should be avoided since it can dominate the only angle considered. In this paper, to establish an online detection approach, the the two matrices A and B are side by side sliding matrices.

4) Generic Principal Angles (GPA): In cases where no known trend can be assumed or one prefers not to leverage the trend properties, the principal angles' parameters can be determined using the overlapping (common data-points) concept. Often half of the total data-points is a moderate and agreeable option. Though this property depends on the type of the noise and its fluctuations in general. Based on Remark 4-6, the stability of τ_{XX} is then established. In other words, in the subspace of n_v vectors, close to half of the vectors need very little rotation to overlap the two subspaces of test and reference. When overlapping the subspaces, n_v should be set to the number of data-points $n_v = j = k$ (refer to Remark 4-1), to make sure that all the important signal is contained in the subspace. This makes the number of angles(n_a) the only factor remained for filtering noise. Under no statistical assumptions for preprocessing, assuming n_a to be equal to the number of the overlapped data-points can express changes due to inevitable reflection of change when scaling common data-points. In case of Wide Sense Stationary (WSS) assumption for scaling, n_a should be more than the number of overlapped data points. Usually one more than the overlapped data points can reflect the data changes quite well. The performance benchmark results for SPA IV-B3 and GPA IV-B4 using traffic-related counters at interface level are presented in Section VI-B.

5) Binary decision making: In this paper, a simple sigma detector (a detector for which a change is when the values deviate from their average by more than a given δ times their standard deviation) is used on the univariate τ values to flag changes. The sigma detector cools down for a certain amount of time after a detection is made to avoid multiple detections of the same event. The cooling time is advised to be set to a value equal or more than twice the matrices' time interval(i.e $2 \times k$ where k is the number of data-points in matrices A and B). This choice stems from the fact that sliding the time windows side by side in time to form the reference and test matrices results in producing the impact of event until the two matrices are both slide passed the change point.

V. Datasets

This section includes information on streaming data used a in Section IV as well as the simulated events evaluated in Section VI.

A. Datasets' properties

Multivariate change detection is of interest either when the counters (features) have dependencies and correlations or are supposed to represent the parts of one system as a whole [29]. Therefore, an important factor in exploring data of multivariate nature is to investigate the effective dimension of that data. If the effective dimension is significantly less than the original space, it is implied that the counters has strong inter-dependencies. One common practice when investigating the effective dimensions of a subspace is to observe the variance ratios on the principle components.

Fig. 8 shows the scree plot for the variance percentage of principal components for BGP and traffic related counters. It also shows that the effective dimension of the subspace is indeed low and most of the data variance is inside the first few principal components. A simple cumulative sum over the scree

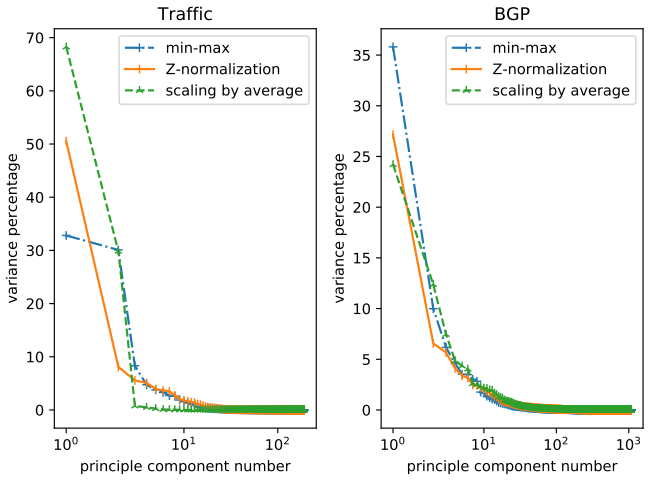

Fig. 8. Scree plots of variance for traffic and BGP counters with different scaling mechanisms- dataset available in [25]

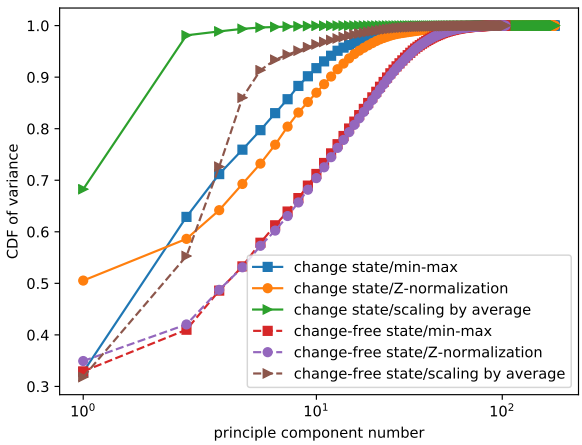

Fig. 9. CDF of variance shown for 192 principal components- comparison of change and change-free- dataset available in [25]

plots in Fig 8 shows that the number of principal elements to contain 99% of data variance in traffic data is 29 and 3 components for min-max and average scaling, respectively. The same numbers for BGP-data are 338 and 150. It also shows that the best suited pre-processing schemes for different groups of data can be different. For instance, scaling by the average for traffic data in addition to preserving most of the variance in the first few elements has the sharpest knee pattern, while for BGP-data the preprocessing schemes do not change the knee position. However, in BGP data the most variance for the first few elements belongs to the min-max scheme. This property is a reminder to consider the nature of data changes which are expected to represent an event. The grouping of data in Fig. 3 is done based on such general intuitions.

Fig. 9 shows that for a given number of principal components the cumulative variance in change state is more than that of the change-free state. In other words, if a certain normal state(X) is established and presented by a certain number of elements(n_v), a change(C) in that state and transition to another should be noticeable in the number of elements that were already considered for explaining the normal state. This behavior is rather expected since an occurrence of change would result in increasing the variance of the system and thus less elements are needed to represent the same level of variance in data. This property was leveraged in Section IV-A in Remark 4-7. The fundamental points of this section are summarized in the following two remarks.

Remark 5-1 : Despite the high initial dimensionality of streaming telemetry, it can be explained with low dimensions. The low number 3 in our case is due to the stable distribution of traffic over a node's interfaces and sufficient traffic multiplexing.

Remark 5-2 : The number of principal elements that can explain a normal state can reflect the impact of the change. It is because the number of effective dimensions is lower when a change happens compared to the normal state.

B. Lab setup

The datasets used in this paper for benchmarking are retrieved from the routers in a lab and the devices are connected in a Clos-topology, with 4 spines and 8 leaves [30]. All leaves connect to multiple ToR (top of rack) switches, each of which aggregates the traffic from multiple traffic generator nodes. The lab uses BGP as routing protocol. All links have BFD (bi-directional forwarding detection) configured for rapid failure detection. The timeseries datasets are created by retrieving several model driven telemetry (MDT) collections with a interval of 10s over spines 1-3 and leaves 3,4,7 and 8. The types of changes inserted in the lab network include:

(I) Enabling/disabling of interfaces using configuration commands;
(II) Breaking and restoring of BFD sessions (by filtering-out BFD control traffic between devices);
(III) Creation of routing loops (using static routes to make traffic for a particular prefix loop between dr02, dr03 and leaf8);
(IV) Creation of traffic blackholes (removal of Forwarding Information Base (FIB) entries for a prefix using configuration commands to cause the router to silently drop traffic destined to that prefix);
(V) Change of hashing behavior across equal-cost multipath (ECMP) using configuration commands.

In most of the datasets, only one intentional event is inserted. However, there are cases where unintentional changes of state happens. At the time of the insertion of events, a general *event file* is generated that specifies the timestamp of the event. The unintentional changes are not included in the *event file*. The data collections from all devices will share the same event file whether the event is expected to propagate on all of them or not. Therefore, in addition to benchmarking the behavior on an expert-crafted ground-truth, a data-driven scheme is proposed to evaluate the methods' capabilities. It

is noteworthy that the results presented in Sections IV and V use a different dataset compared to evaluation section: The sample dataset consists of 6622 counters while test datasets are composed of different number of features ranging from 300 to 20000 features. The simulated events in this paper are not particularly hard to detect for a known network when using already existing rules and heuristic packages, however it is not trivial when no assumptions are made. The results in Section VI-B compares the performance of DESTIN which considers no assumptions with the expert systems.

VI. Evaluation

This section evaluates DESTIN's performance in terms of recall, precision and average delay to detection for SPA, GPA and PCA methods when considering data-driven filters as well as the expert-defined manual ground truth.

A. Terminology and setup

1) Event versus change definition: Events inserted on a single network device can be reflected on neighboring devices as well. Examples include insertion of a wrong static route that causes traffic to loop, or the failure of an interface which will be reflected in the interface state of the connected router, but likely also on several other devices due to the associated change in routing and forwarding behavior. In this paper, to evaluate the propagation of an event inserted on a single router, in addition to leveraging the concept of event from expert's perspective (already defined rules), some data-driven measures of change are introduced to assess the event's propagation referred in this paper as changes. The data-driven measures introduced in this section also provide an upper-bound on the potentially important time-stamps for an operator.

2) Data-driven measures of change: An event can incur changes on one or more devices with different impact levels on counters (features). The most generic way to approach the evaluation of an event's detectability is to assume the simplest most general time series change scenarios which are mean and variance changes. Therefore, two metrics are defined in this paper to track the ratio of shift level (mean-change) to its noise (standard deviation (std) level) in a univariate and multivariate way.

3) Univariate change to noise ratio (UCNR): Assuming a perfect step-change (mean value change) in time-series, one can calculate the ratio of the step to the standard deviation of that time-series. Using the notations defined in Section IV, x and y are the matrices of shape $\mathbb{R}^{n\times k}$ that belong to the larger data blocks X and Y right before and after the event, respectively. The matrix c is selected from the transient state C. Assuming that the step change value for feature i is noted as l_i, the measure can be expanded to the n available features as bellow:

$$\Delta_{uni} = \sum_{i=1}^{n} \frac{|l_i|}{std(x_i)} \tag{4}$$

Where x_i denotes the row vector of observations for feature i and *std* operator takes the empirical standard deviation of the vector given as input. Equation (4) measures the change on each time-series data in a univariate way and accumulates the effects by summing over the values. This implies that every feature has its own independent impact on the metric. However, this measure is impacted by the number of features in the dataset and cannot be compared between datasets of different sizes. To address this issue, a relativized instance of the values by the average over a fixed number of past values is used. Therefore, in addition to the paled effect of dimensionality, the difference of a change compared to the background noise is better highlighted. The relativized values are denoted as $\widehat{\Delta_{uni}}$ in this paper.

4) Multivariate change to noise ratio (MCNR): A different way to measure the impact of a change is to consider the shift change in the feature space of R^n in a multivariate manner and calculate the metric using vectors as follows:

$$\Delta_{mult} = \frac{||\overline{y} - \overline{x}||}{std((\overline{y} - \overline{x})^T \times [(x - \overline{x}) \times 1_k^T])} \tag{5}$$

Where$\overline{x}$ and 1_k denote the empirical mean of matrix x and a column matrix of all ones of size k, respectively. For the same reasons mentioned in VI-A3 the values are divided by their average and denoted as $\widehat{\Delta_{mult}}$.

5) Performance assessment parameters: To facilitate a fair comparison of methods, three different approaches are used to define what is considered *ground truth* for detecting a change on a particular network device based on the logic and metrics defined in this section so far.

- **manual -** devices which are expected to be impacted by a change, as defined by an expert based on topology and type of event. These devices are referred to, in this paper, as the *local devices* of the particular event. The rest of the devices are referred to as *remote devices*. The following terms are then defined for performance evaluation:
  - Recall: The ratio of the number of correct detection notifications on at least one of the local devices to the total number of intentional events inserted on the devices.
  - detection relevance: The ratio of number of the correct detections on all devices, either remote or local, to the number of total notifications made by the detector.

  Manual ground-truth represents the lower bound in the number of events on the very least number of devices expected to reflect the event.
- **data-driven using UCNR / MCNR -** all devices exceeding a certain threshold of UCNR / MCNR around the time of the change insertion will be selected as a valuable change point to be benchmarked. To assess the performance in these cases the following terms are used:
  - Recall: The number of correct detections on all devices (either local or remote) divided by the sum of number of total devices expected from the data-driven measure to detect change.
  - Precision: The ratio of correct detections to the total number of detections made.
  - f1 score: The harmonic mean of recall and precision.

TABLE I
RECALL, PRECISION AND TIME TO DETECTION

methods	**Ground-truth defined**									
	manual		**data-driven UCNR**				**data-driven MCNR**			
	Recall	Detection relevance	Recall	Precision	f1 score	Delay	Recall	Precision	f1 score	Delay
PCA	0.73	0.40	0.60	0.50	0.54	**9.65**	0.56	0.49	0.52	**11.36**
SPA	**0.79**	**0.78**	0.76	**0.87**	**0.81**	24.5	0.69	**0.84**	**0.75**	30.31
GPA	**0.79**	0.64	**0.78**	0.77	0.77	22.29	**0.72**	0.76	0.74	28.85
Total events	**82**		**242**				**357**			

TABLE II
RESULTS BREAK-DOWN BY EVENT TYPE FOR UCNR FILTER

network event	**Detected**			**Total**
	PCA	SPA	GPA	
Shutdown interface	53	62	**63**	73
ECMP hash changes	36	51	**52**	57
Enable interface	25	**28**	**28**	50
Blackhole	26	36	**37**	44
BFD restore/break	3	**5**	**5**	13
Routing loop	3	3	**4**	5

B. Numerical results

This section presents results when the matrices A and B have $j = k = 7$. The n_v and n_a parameters are chosen based on Sections IV-B3 and IV-B4 for SPA and GPA, respectively. The overlapped data-points and n_a values for GPA are considered both to be 3 and no assumptions such as WSS property are made. For the sake of a fair comparison with PCA, traffic counters are used for performance evaluation.

Tab. I, compares different methods in terms of recall, precision, and the average delay for the detection of a change. PCA is a method that compares a number of data-points to one data-point while principal angles compare multiple data-points to each other. It is expected that principal angles (PA) have more delay in detection compared to PCA due to its input format. Tab. I shows that the delay of PA is almost twice that of PCA subspace which in terms of data-points amounts to only 1 or 2 more data-points (i.e. 10-20 seconds).

The benchmarking results from the manual ground-truth show that in the total of 82 manually inserted events, 73%, 79% and 79% of the events were detected on at least one of the local devices for PCA subspace, SPA and GPA, respectively. Evaluation of detection relevance show that the false alarms are lower in SPA by 38% and 24% compared to PCA subspace. This shows that leveraging a known trend for data while using PA can significantly improve the noise filtering. However, PA method even in its most generic sense, GPA, improves the precision by 24%.

In both data-driven cases, the threshold is set to 3 for UCNR and MCNR metrics. The justification behind is that the metrics are calculated only around the event over small windows of data, it can be assumed that only in that small interval the data is expected to exhibit WSS properties. It is known that for a WSS process, a deviation of 3 times the noise value is the indicator of surpassing the normal behavior. Setting this threshold results in a total of 242 and 357 datasets, for UCNR and MCNR, respectively. The difference in the sets supports the fact that ideally one expects to detect more events in a multivariate approach rather than in a univariate one. In the data-driven approach in terms of precision, SPA and GPA are 37% and 27% better than PCA subspace when using UCNR as the filter. It is noticeable that leveraging existing assumptions such as known trends can improve the accuracy in PA. MCNR also performs similarly. In terms of recall, PA on average has 17% improvement compared to PCA.

Tab. II depicts the results per event type. It is seen that all methods have almost the same potential in detecting enable interface, BFD break/restore and routing loop events. However, there is a noticeable difference in detecting changes of ECMP hashing between PA and PCA. The reason behind the fair level of improvement is the impact of ECMP-hashing related changes on the inter-dependencies of the counters when the event happens. For changes like shutting down an interface or traffic blackholing which result in a drop in traffic, PA also shows the best performance due to the method's robustness with respect to the assumptions on prepossessing.

VII. CONCLUSION

This paper proposes DESTIN which is an online multivariate change detection methodology based on principal angles to assess the state changes of a network element. The results show that the method has a potential to timely detect state changes with 31% and 16% average improvement separately in precision and recall compared to PCA for traffic data. It is seen that DESTIN performs better for almost all types of changes but specially well with events such as ECMP hash changes since the changes incorporate the change in the inter-dependencies of counters. In addition, due to the robustness in preprocessing assumptions, it performs better for traffic blackholing and interface shutdown changes. The results as presented in benchmarking framework also show that a change in a network can propagate beyond what is naturally expected by an expert. Future work will include a mathematical framework for tuning the sensitivity of DESTIN based on the operator's preference. It will also provide an alternative method for distilling the important features from data when DESTIN fires an alarm.

REFERENCES

[1] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proceedings of the 1999 IEEE*

Symposium on Security and Privacy (Cat. No. 99CB36344). IEEE, 1999, pp. 120–132.

[2] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?" in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2017, pp. 25–36.

[3] J. Cordova-Garcia, "Sparse control and data plane telemetry features for bgp anomaly detection," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 240–245.

[4] Cisco Innovation Edge, "Network anomaly telemetry datasets," https://github.com/cisco-ie/telemetry/tree/master/11, 2019.

[5] T. Feltin, P. Foroughi, W. Shao, F. Brockners, and T. H. Clausen, "Semantic feature selection for network telemetry event description," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–6.

[6] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[7] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[8] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM computer communication review*, vol. 34, no. 4, pp. 219–230, 2004.

[9] ——, "Mining anomalies using traffic feature distributions," *ACM SIGCOMM computer communication review*, vol. 35, no. 4, pp. 217–228, 2005.

[10] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2007, pp. 109–120.

[11] D. Brauckhoff, K. Salamatian, and M. May, "Applying pca for traffic anomaly detection: Problems and solutions," in *IEEE INFOCOM 2009*, 2009, pp. 2866–2870.

[12] Y. Hua and W. Liu, "Generalized karhunen-loeve transform," *IEEE Signal Processing Letters*, vol. 5, no. 6, pp. 141–142, 1998.

[13] R. Kwitt and U. Hofmann, "Unsupervised anomaly detection in network traffic by means of robust pca," in *2007 International Multi-Conference on Computing in the Global Information Technology (ICCGI'07)*, 2007, pp. 37–37.

[14] H. Cardot and D. Degras, "Online principal component analysis in high dimension: Which algorithm to choose?" *International Statistical Review*, vol. 86, no. 1, pp. 29–50, 2018.

[15] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery," *IEEE signal processing magazine*, vol. 35, no. 4, pp. 32–55, 2018.

[16] Q. Ding and E. D. Kolaczyk, "A compressed pca subspace method for anomaly detection in high-dimensional data," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7419–7433, 2013.

[17] D. S. Yeung, S. Jin, and X. Wang, "Covariance-matrix modeling and detecting various flooding attacks," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 2, pp. 157–169, 2007.

[18] L. Huang, X. Nguyen, M. Garofalakis, J. M. Hellerstein, M. I. Jordan, A. D. Joseph, and N. Taft, "Communication-efficient online detection of network-wide anomalies," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 134–142.

[19] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft, "In-network pca and anomaly detection," in *Advances in Neural Information Processing Systems*, 2007, pp. 617–624.

[20] T. Huang, H. Sethu, and N. Kandasamy, "A new approach to dimensionality reduction for anomaly detection in data traffic," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 651–665, 2016.

[21] K. De Cock and B. De Moor, "Subspace angles between arma models," *Systems & Control Letters*, vol. 46, no. 4, pp. 265–270, 2002.

[22] Y.-H. Yuan, Q.-S. Sun, Q. Zhou, and D.-S. Xia, "A novel multiset integrated canonical correlation analysis framework and its application in feature fusion," *Pattern Recognition*, vol. 44, no. 5, pp. 1031–1040, 2011.

[23] W. Liu, X. Yang, D. Tao, J. Cheng, and Y. Tang, "Multiview dimension reduction via hessian multiset canonical correlations," *Information Fusion*, vol. 41, pp. 119–128, 2018.

[24] L. Wolf and A. Shashua, "Learning over sets using kernel principal angles," *Journal of Machine Learning Research*, vol. 4, no. Oct, pp. 913–931, 2003.

[25] Cisco, "Telemetry dataset," 2019. [Online]. Available: https://github.com/cisco-ie/telemetry/tree/master/11

[26] A. V. Knyazev and P. Zhu, "Principal angles between subspaces and their tangents," *arXiv preprint arXiv:1209.0523*, 2012.

[27] Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of computation*, vol. 27, no. 123, pp. 579–594, 1973.

[28] C. Jordan, "Essai sur la géométrie à n dimensions," *Bulletin de la Société mathématique de France*, vol. 3, pp. 103–174, 1875.

[29] M. Lavielle and G. Teyssiere, "Detection of multiple change-points in multivariate time series," *Lithuanian Mathematical Journal*, vol. 46, no. 3, pp. 287–306, 2006.

[30] "telemetry/telemetry_topology_maps.pdf at master · cisco-ie/telemetry · github," https://github.com/cisco-ie/telemetry/blob/master/topology-description-docs/telemetry-topology-maps.pdf, (Accessed on 09/17/2020).