7-1-2019

Networkmetrics unraveled: MBDA in Action

José Camacho
Dartmouth College, Jose.Camacho@dartmouth.edu

Rasmus Bro

David Kotz
Dartmouth College, David.F.Kotz@Dartmouth.EDU

Networkmetrics unraveled: MBDA in Action

José Camacho

Department of Signal Theory, Telematics and Communications, School of Computer Science and Telecommunications - CITIC, University of Granada, Spain

Rasmus Bro

Chemometrics and Analytical Technology, University of Copenhagen, Denmark

David Kotz

Department of Computer Science, Dartmouth College, Hanover, NH 03755, United States

Abstract

We propose *networkmetrics*, a new data-driven approach for monitoring, troubleshooting and understanding communication networks using multivariate analysis. Networkmetric models are powerful machine-learning tools to interpret and interact with data collected from a network. In this paper, we illustrate the application of Multivariate Big Data Analysis (MBDA), a recently proposed networkmetric method with application to Big Data sets. We use MBDA for the detection and troubleshooting of network problems in a campus-wide Wi-Fi network. Data includes a seven-year trace (from 2012 to 2018) of the network's most recent activity, with approximately 3,000 distinct access points, 40,000 authenticated users, and 600,000 distinct Wi-Fi stations. This is the longest and largest Wi-Fi trace known to date. To analyze this data, we propose learning and visualization procedures that extend MBDA. These procedures result in a methodology that allows network analysts to identify problems and diagnose and troubleshoot them, optimizing the network performance. In the paper, we go through the entire workflow of the approach, illustrating its application in detail and discussing processing times for parallel hardware.

Keywords:
Multivariate Big Data Analysis, Anomaly Detection, Big Data, Parallel Hardware, Dartmouth Campus Wi-Fi, Networkmetrics

1. Introduction

Multivariate exploratory analysis has been recognized as an outstanding approach in several domains, including industrial monitoring [1], network security [2], marketing [3], weather modeling [4], bioinformatics [5], food research [6], and so forth. In this methodology, visualization, interpretation and data interaction are the principal tools for an analyst to interact and understand data and the problem the data reflects. This is an alternative data-driven approach to the one currently dominating machine learning, e.g., deep-learning, in which the model is built as a black box to approximate an output of interest and little interpretation is left to the analyst.

There are advantages and disadvantages in the approach of multivariate exploratory analysis. A main shortcoming is that the analyst has to be proficient in the modeling approach used and knowledgeable about the type of data under analysis, just like a blacksmith needs to be proficient with the hammer and knowledgeable about the properties of metals. However, the benefit is worth the price. Multivariate exploratory analysis is useful to address specific questions, like "what is the best classification of an individual, given a set of classes", or "what is the best prediction of a future outcome", as in other machine-learning approaches. However, a major advantage of the former is that it also provides interpretable information about *why* a model provides a given answer. There are many situations in which an answer is not of practical use, without knowing the "why". Network monitoring is an example: network analysts desire to detect unwanted events during network operation, but they also need to understand their root causes and troubleshoot them as soon as possible. An advantage of multivariate exploratory analysis is that, even if a model is created to respond to a specific question, the interaction of the analyst with the data through the model can bring much more information, like the derivation of new, unexpected findings. This is a useful property that black-box models do not normally provide.

In a recent paper, we introduced the notion of *networkmetrics* [7], the use of multivariate analysis to unravel network problems. The name is a derivation from other areas where multivariate analysis plays a central role in the scientific and professional communities, like in psychometrics, chemometrics or econometrics. In this paper we present an example of what these methods can bring to the network monitoring arena. Specifically, we present a case study of the Multivariate Big Data Analysis (MBDA) tool [8] as applied to monitoring and troubleshooting a campus-wide Wi-Fi network [9]. MBDA

*Corresponding author: J. Camacho (email: josecamacho@ugr.es)

is a complete multivariate anomaly detection and analysis approach, based on a workflow of five steps, that can handle large amounts of data from disparate sources. When an anomaly is identified, the output includes the log entries of raw information associated with it. These, in turn, can be presented to the analyst, so as to elucidate the root causes for the anomaly. In a context where the number of log entries is massive (i.e., Big Data), MBDA works as a magnifier glass, conveniently highlighting anomalous events.

Our contributions in this paper are as follows.

- We illustrate the application of MBDA to a real case study, showing what it can provide to network analysts and presenting the workflow in detail, including the parallelization of the code and processing time results.
- We enhance MBDA with several functionalities useful in real-life practice:
  - We propose an automatic feature-learning procedure, consistent with the MBDA methodology.
  - We incorporate state-of-the-art exploratory analysis visualizations within the central step of MBDA, to make it more data interactive.
  - We use gephi [10] network visualizations to improve the interpretation of diagnosis results, which are supposed to be condensed [8] – but can still be of large size in practical problems.
  - We discuss model updating after anomaly identification and diagnosis, to further refine the anomaly detection.

The rest of the paper is organized as follows. Section 2 introduces the data set under analysis. Section 3 presents the MBDA methodology in brief. Section 4 introduces the learning procedure contributed. Section 5 illustrates the five steps of MBDA in the Wi-Fi data. Section 6 presents the visualization on the anomalies and Section 7 discusses model update. Section 8 provides final conclusions.

2. The Dartmouth Wi-Fi network

Dartmouth College has a compact campus with over 200 buildings on 200 acres. The evolution of the network is documented in the series of papers [11, 12, 9]. The number of students, staff, and academic faculty reached near 6,500, 3,300 and 1,000, respectively, at the end of 2018, and the number of Access Points (APs) is above the 3,000. Researchers at Dartmouth have been capturing data about the usage of the network for many years, providing a perfect case study for tools like MBDA.

The paper analyses a data capture containing the connections of users to the network in a seven year time span: from 2012 to 2018. Data contains Simple Network Management Protocol (SNMP) traps [13] sent from wireless controllers to our collector. The capture reveals the statistics in Table 1. The data set contains a total of 5 Billion traps and 7 TB of data. A total of 38K authenticated users and an undetermined number of non-authenticated users have been connected to the network in the last seven years, using 600K devices. The network infrastructure supports several SSIDs, primarily *Dartmouth Secure*, the WPA2-Enterprise authenticated college network, *Dartmouth Public*, a public-access network, and *eduroam*, the world-wide roaming network for educational institutions [14]. Dartmouth Secure was entirely replaced by eduroam in the final months of the capture.

Table 1: Details of the SNMP trap capture at Dartmouth College.

Statistic	**Number**
Capture period	Jan 1st 2012 - Dec 31st 2018 (2556 days)
log entries (SNMP traps)	5 Billion
Data Size (raw)	7 TB
Access points	3,330
Authenticated Users	38,096
Stations	624,903
SSIDs	20

To collect the trace, the Wi-Fi network controllers forwarded SNMP traps with a record of network activity to the Dartmouth team's servers. Figure 1 shows an example of an SNMP trap as received. Each trap comprises a header, with timestamp and sender and collector information, followed by a variable number of triplets representing SNMP object identifiers (OIDs) with the format '<OID> = <type>: <value>' and separated by hashes (#). OIDs are partly represented in ASN.1 notation, which can be translated into more meaningful OID names using the relevant Management Information Base (MIB). An important OID is the trap type (TT), in which the value is also an OID: '<TT> = OID: <OID>'. More details on the data capture can be found in [9].

3. Multivariate Big Data Analysis

We base our analysis on the Multivariate Big Data Analysis (MBDA) methodology [8], a recently proposed networkmetrics exploratory tool. MBDA makes use of two open software packages available on Github: the MEDA Toolbox [15, 16] and the FCParser [17]. The FCParser is a python tool for the parsing of both structured and unstructured logs. With the MEDA Toolbox, multivariate modeling and data visualization can be performed.

Intensive parsing requires a parallel processing hardware. We used the Anthill Compute Cluster hosted by the Computer Science Department at Dartmouth. It is a 100 node, 1200 core, 4,288GB distributed ram compute cluster, managed with the grid engine [18] as parallelization software. Matlab and Python scripts using the FCParser and the MEDA Toolbox, respectively, run on top of the parallel hardware as grid jobs.

The MBDA approach consists of 5 steps:

1) Parsing: the raw data coming from structured and unstructured sources are transformed into quantitative features.

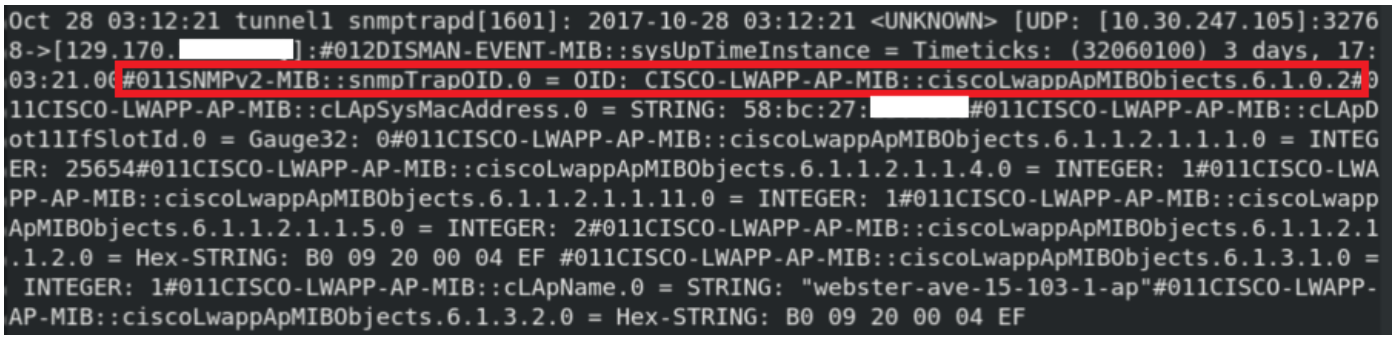


Figure 1: Example of an SNMP trap in the data capture. The second OID, highlighted by a rectangle, represents the trap type. Parts of an IP and a MAC address have been hidden.

2) Fusion: the features of the different sources of data are combined into a single data stream. In the example under analysis there is a single source of data: SNMP traps. Thus, fusion is not required.

3) Detection & Analysis: featured data is visualized and anomalies are identified in time using Principal Component Analysis (PCA) [4, 19] and Multivariate Statistical Network Monitoring (MSNM) [20, 21, 22, 23].

4) Pre-diagnosis: the features associated with an anomaly are found.

5) De-parsing: Using both detection and pre-diagnosis information, the original raw data records related to the anomalies are identified and presented to the analyst.

The first three previous steps are equivalent to what it is commonly done in other machine learning methodologies. However, steps 4 and 5, which perform the diagnosis of the anomalies, are a main advantage of the present proposal. These steps are possible thanks to the white-box, exploratory characteristics of PCA as the core of the MSNM approach in step 3). PCA is easy to interpret in terms of the connection between anomalies and features.

PCA transforms the original features into a lower number of uncorrelated features: the so-called principal components. The principal components are ordered by captured variance. PCA follows the expression:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^t + \mathbf{E}_A, \tag{1}$$

where $\mathbf{X}$ represents the matrix of data, with N rows and M columns, $\mathbf{T}_A$ is the $N \times A$ scores matrix containing the projection of the objects in the principal components sub-space, with A the number of principal components, $\mathbf{P}_A$ is the $M \times A$ loadings matrix containing the linear combination of the variables represented in each of the principal components, and $\mathbf{E}_A$ is the $N \times M$ matrix of residuals.

Scores, loadings and residuals can be visualized using line and scatter plots to gain data understanding. Also, the data can be further compressed in a pair of statistics, the D-statistic and Q-statistic, where anomaly detection can be performed following the MSNM approach. The D-statistic and the Q-statistic for observation n are computed with the following equations:

$$D_n = \mathbf{t}_n \cdot (\Sigma_T)^{-1} \cdot \mathbf{t}_n^t \tag{2}$$

$$Q_n = \mathbf{e}_n \cdot \mathbf{e}_n^t \tag{3}$$

where $\mathbf{t}_n$ is a $1 \times A$ vector with the scores for observation n, $\mathbf{e}_n$ is a $1 \times M$ vector with the residuals, and Σ_T represents the covariance matrix of the scores.

As a summary, table 2 describes the general data pipeline. Steps 1) and 3) perform two steps of compression of the data, first from raw data to features and then from features to components/residuals using PCA and from these to statistics using MSNM. Once anomalies are found, steps 4) and 5) allow to identify the root cause in the raw information associated to them.

4. Learning Counts in Big Data

4.1. Feature-as-a-counter parsing

MBDA makes use of the feature-as-a-counter (FaaC) approach [2] in step 1). Each feature contains the number of times a given event takes place during a pre-defined time interval. Examples of suitable features are the counts of a given word in a log or the number of traffic flows with given destination port in a *Netflow* file. This general definition makes it possible to integrate, in a suitable way, most sources of information.

To implement the FaaC, the FCParser defines *variables* and *features*. Variables represent general entities in the data. In the previous two examples, the variables would be *word* and *destination port*. The Features are defined for a specific value of a variable. Examples of features would be *word='food'* and *destination port='80'*. This representation in variables and features has the relevant advantage that allows for the definition of *default* features, e.g. *word=< ANY >*, useful to count the instances of a variable, regardless its value, in a data record. Variables and features are defined using regular expressions in configuration files. The FCParser applies this configuration to the data, in order to compute feature vectors for each interval of time. This is done using a multi-threading configuration to speed-up computation.

Table 2: Summary of the five steps in MBDA.

STEP	INPUT	OUTPUT	SOFTWARE
1. Parsing	Raw data stream	Stream of features per source	FCParser
2. Fusion	Stream of features per source	Single feature stream	FCParser
3. Detection	Single feature stream	Timestamps for anomalies	MEDA Toolbox
4. Pre-diagnosis	Single feature stream & Timestamps for anomalies	Features for anomalies	MEDA Toolbox
5. De-parsing	Raw data stream & Timestamps & Features for anomalies	Raw log entries for anomalies	FCParser

The default features in step 1) of MBDA play a similar role as residuals do in PCA and MSNM in step 3). Both transformations, steps 1) and 3), work as lossy compression steps, which are fundamental to visualize data when the data volume is massive. However, when doing anomaly detection, it is a good idea to include in the model a summary of what is left out in the compression. This allows the analyst to retain the ability to find uncommon patterns in the residual part, and it is the goal of the residuals in eqs. (1) and (3).

A simplistic example of when default features can be useful follows. Imagine we capture traffic from a network where the main services are *http* (destination port 80) and *smtp* (destination port 53). To monitor the traffic, we define two features: *destination port='80'* and *destination port='53'*, and a default feature *destination port=< ANY >*. During traffic monitoring, there is a sudden burst of Internet Relay Chat (*IRC*) traffic with destination port 6667. The two previous features cannot capture this burst, but the default one can. If something anomalous is detected in a default feature, the analyst can decide whether she needs to redefine features to account for the new situation, or simply look at the raw data for diagnosis.

4.2. Learning procedure

MBDA relies in the manual definition of the features in the configuration files of the FCParser. To write such configuration files, the analyst needs to get familiarized with the data. Unfortunately, in a practical Big Data problem like the one under analysis, the data capture is too massive for direct inspection. If we want to obtain a good description of the content, we need to apply an automatic feature derivation technique. This technique needs to be consistent with the posterior multivariate analysis, so that we minimize loss of information.

There are two basic properties we would like to meet in the learning procedure. First, main sources of variance need to be captured. Second, uncommon characteristics with low variance should also be modeled somehow, in a summary of residual information. The second feature is built-in in the FaaC methodology with the definition of default features, as already discussed.

We developed a learning algorithm to automatically identify a list of common FaaC features in a Big Data set, and included it in the FCParser repository at Github with the name `fclearning.py`. The learning algorithm extracts the features in the data ordered by their percentage of presence, measured as the portion of the log entries where a feature appears. It also defines default features and automatically writes the configuration files.

We used this algorithm in two steps to identify high variance OIDs in the Wi-Fi data. First, the algorithm was parallelized in 2556 processing jobs, one per different day in the capture. The resulting 2556 configuration files contain the FaaC features with a percentage of presence above the 5% during the day. Second, these configuration files were combined in a single configuration file, where we discarded all features with variance below a threshold[1]. This resulted in a total of 90 features, including default features. From their description we can conclude that we can use this data to identify who associated to the network, when the association took place, the APs involved in the connection, and thus the approximate location and movement of the user during it, and the device used.

The whole learning process using the parallel hardware and multi-threading (4 threads per processor) took 12 hours, during which a maximum of 150 jobs were processed in parallel. This means that the processing time could be reduced in 17-folds using a larger computer cluster. The combination of results in a regular server took another 30 minutes.

5. MBDA in action

In this section, we discuss the application of MBDA to the Wi-Fi data set using the learned features. Since there is one single source of data, the second step (fusion) is not necessary.

5.1. Parsing

We use the FCParser to generate the feature vectors with the aforementioned configuration file learned. In agreement with the learning phase, we consider feature vectors for intervals of one day. These contain the number of traps, the total number of OIDs and the number of learned features. This makes a total of 92 features. Each day in the original SNMP capture is transformed into a feature vector, we call 'observation', and each feature in the observation is the number of times the corresponding element is found in the traps of that day. This results in a compression of the data from 7TB to less than 1MB, yielding 2556 observations (days) of 92 features each in matrix $\mathbf{X}$. The compression conveniently transforms a Big Data set into a handleable data set by any analysis package in a common computer.

The parsing was again parallelized in 2556 processing jobs, one per day, and the whole process using the Anthill Computer

[1] We used a threshold of 0,01%, taking the variance in the number of log entries per day as a reference

Cluster and multi-threading (again 4 threads per processor) took 15 hours. Again, a maximum of 150 jobs were processed in parallel, so the same reduction with a larger cluster discussed for the learning procedure applies here.

5.2. Detection & Analysis

Basically, PCA factorizes the data **X** into a part for the observations (the scores) and a part for the features (the loadings), making the visualization simpler and interpretable. The remaining of this section discusses how to obtain and interpret this factorization. PCA has a number of interesting properties in terms of visualizing the data. First, the principal components are linear combinations of the original features and uncorrelated. In particular, distances between points in any scatter plot of the scores, commonly called a score plot, approximate the Mahalanobis distance in the original data. Second, since the principal components are ordered by variance; hence, we can be confident that the main patterns of high variance in the data will be observable in the first components. Therefore, by only inspecting a small set of principal components, we can gain an accurate insight into the data.

The number of principal components to use has to be determined. There are many methods to aid in that decision, and for a general description on which one is best, see [19, 24]. Fortunately, the data exploration is barely affected by selecting different numbers of components, specially because we also visualize a summary of the residuals. If in doubt, it is better to use larger numbers, since we may learn something else from additional principal components.

Still, it is useful to have an initial reference for a suitable number of principal components. The MEDA Toolbox, the software used in this third step of MBDA, provides of a plot with the residual variance per PC combined with the column-wise k-fold cross-validation (ckf) curve [25]. The corresponding plot for the Wi-Fi data is shown in Figure 2. To obtain this plot, we inputted the matrix **X** of parsed data. The residual variance tells us the percentage of information not captured by the model, which should be low enough so that we do not miss much information in the summary of the residuals. The ckf gives information about the predictive ability of components, in terms of how well the structure among the original features is captured. A reasonable choice for the number of principal components should capture a large portion of variance and yield a reasonably good predictive model. In our case, 6 principal components seems like a reasonable choice, since models beyond that number do not improve in prediction ability and add very little variance.

Once we have selected the number of principal components, we can visualize the data using scatter plots of scores and loadings. A toy example of score plot is shown in Figure 3. We can see six points representing observations (rows in the data) scattered around the center of coordinates. We assume that data is mean centered, as data is commonly preprocessed before PCA, and note that interpretation changes for non-centered data. The interpretation is easier if we think of the observations not only as points, but also as vectors connecting the origin of coordinates to the points, as in the illustration. The relationship between observations is revealed by the distance between observations and the angle between vectors. Very close points reflect similarity of the observations in the principal components we are visualizing. However, we are mainly interested in those points far from the origin of coordinates, meaning that they present high scores. Observations with low scores are not well modeled by the inspected components. This is illustrated with observation 1, for which the plot does not provide any useful information other than it has a low score in the first two principal components. Vectors with the same direction indicates similarity, and the closer the points the more similar. This is the case for observations 2 and 6. Thanks to this property, score plots are useful to identify clusters of similar observations and other meaningful patterns, like trends and outliers. Vectors in the opposite direction show an inverse relationship, like observations 4 and 5. Vectors in different directions and distanced reflects they are somehow different (not directly or inversely related). Thus, observation 3, the group of 2 and 6 and the group of 4 and 5 are all different. This property is again useful because if the score plot highlights different groupings of observations, we can conclude that there is an underlaying different information content in the features of these groups. To obtain more information about those differences, we need to look at the cor-

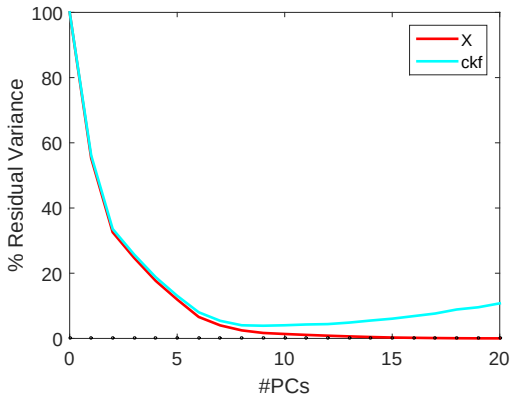


Figure 2: Curves of residual variance and *ckf* [25] for the Wi-Fi data.

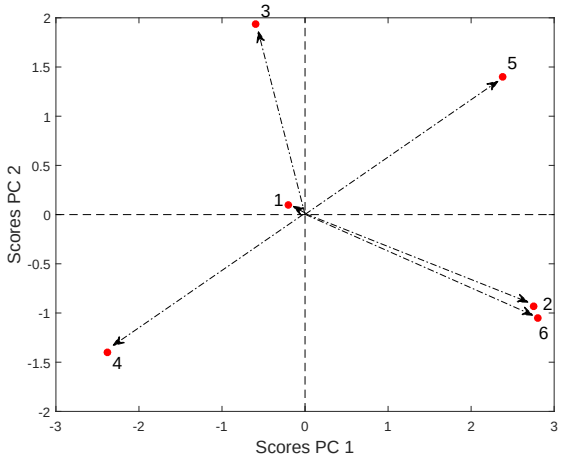


Figure 3: Illustration of interpretation in score and loading plots.

responding loading plot.

Loading plots are similar to score plots, but the former represent the spacial distribution of features (columns in the data) instead of observations. Again, clusters, trends or outliers can be identified in the loading plot, and lead to interesting conclusions on the features distribution. We can also combine scores and loadings in a single plot, commonly called a biplot [26]. Well designed biplots allow similar comparisons in distance and angle between observations and features. Thus, if one observation is located close to a feature in the biplot, we expect this observation to have a high value of that feature. This is useful to draw connections between the patterns of observations and features: e.g. to identify what features make an outlier different from the rest of observations.

The plots corresponding to the first 6 principal components in the Wi-Fi data are depicted in Figure 4. Recall that matrix $\mathbf{X}$ contains 2556 rows, representing days of the capture, and 92 features. We present score plots at the left of the figure and loading plots at the right. Visualizing score plots and loading plots together is useful to take a bird's view of the data. In the score plots, points represent the 2556 days of the data capture and are colored according to the year. In the loading plots, points represent the 92 features and, in order to facilitate interpretation, we also plot a shadow of the scores, like in a biplot.

Figure 4(a) represents the score and loading plot for PC1 vs PC2. These two principal components represent 83% of the variance in the data, and therefore we can learn from them the main patterns of change in $\mathbf{X}$. The score plot at the left shows that the dots with different colors are in different locations. This means there are relevant differences in the contents of the SNMP traps for different years. The loading plot shows that a large majority of the features are located far from the center of coordinates towards the right side. This can be interpreted in connection with the score plot: any day toward the right will have higher value in most of the features. Given that the features represent counts of events in the log entries, as we traverse from left to right in the score plot, the days will have more connection activity. Thus, busy periods are represented towards the far right of the plot, and vacations are clustered to the left, and we could say that the first PC (the horizontal direction in the score and loading plots) represents the general activity in the network. We annotated this in both plots using a horizontal arrow.

The loading plot in Figure 4(a) also shows that the variables are distributed from the bottom to top, and we see a similar distribution for the different years in the score plot: the first two years are in the bottom and the last two in the top, with middle years in between. We also see a separated cluster of days in 2018, highlighted with a circle. A closer look reveals that all the days in the cluster belong to the period from September to November, when eduroam replaced Dartmouth Secure. The vertical pattern in the loading and score plots shows that the distribution of traps has changed across the years: days towards the top have a higher content of traps represented by the features in the top and less of those in the bottom, and vice-versa. Again, we annotated this in the score and loading plots using a vertical arrow. Questioned about this difference, the network staff replied that there was an update in the controllers software, in which the types of traps that were collected changed. This variability in traps for different temporal periods makes the analysis of the data a real challenge.

Figure 4(b) represents PC3 vs PC4. Here we see again some differences among the years and the cluster of 2018. We also see a set of consecutive days from 2017 that depart from the rest of days, which is reflecting that something unusual took place during those days. This is sometimes referred to as an excursion of scores. Another 2-day excursion is shown in 2013. Both excursions are annotated with an arrow. The corresponding loading plot is difficult to interpret in connection with the observed deviations, but there are additional tools [27, 28, 29] that can be used to provide more information. This will be shown in the next step of the approach: the pre-diagnosis.

Figure 4(c), representing PC5 vs PC6, shows again the excursion of 2017 and another in 2012 and 2014, annotated with arrows. The loading plot allows to associate the excursions of 2012-2014 and 2017, also annotated with arrows, to specific features, also annotated with circles. The excursions of 2012-2014 are associated to a high number of failures in the RADIUS server, and the one in 2017 to a high number of restarts of APs. We will gain more detail afterwards.

No additional information was revealed from inspecting the following two principal components (not shown), so we decided to stop the initial analysis here.

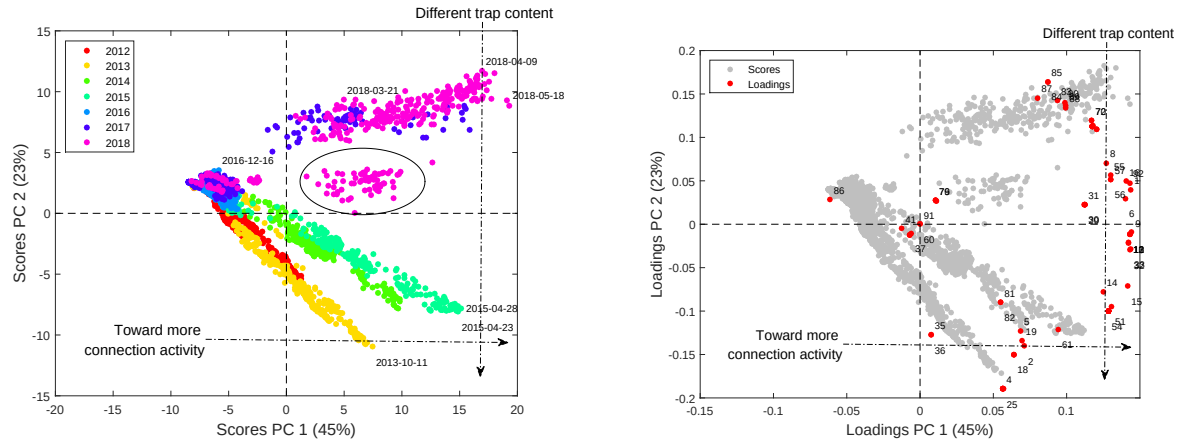
Besides the inspection of score and loading plots, one can visualize a summary of the whole data distribution in one single plot using MSNM: a scatter plot of the observations in terms of the D-statistic and the Q-statistic. In this plot we can also show upper control limits (UCLs) to facilitate the detection of anomalies. UCLs leave below normal observations with a certain confidence level, e.g. 99%. They can be used as hypothesis tests, so that all observations above the limits reject the null hypothesis that they are normal. More information on how to compute these limits can be found in [22, 20].

The MSNM plot for the Wi-Fi data is shown in Figure 5. Anomalies are expected to surpass any of the two control limits. This plot is optimized for anomaly detection, and the excursions mentioned before are clearly observed. However, in the plot we miss other details, like the yearly and seasonal patterns, as well as the difference in traps contents. A main advantage of this plot is that it also includes residuals, which are not accounted for in the 6 principal components. The Q-statistic, which conforms a summary of the residuals, clearly identifies the excursion in 2013 and another anomaly in 2012.

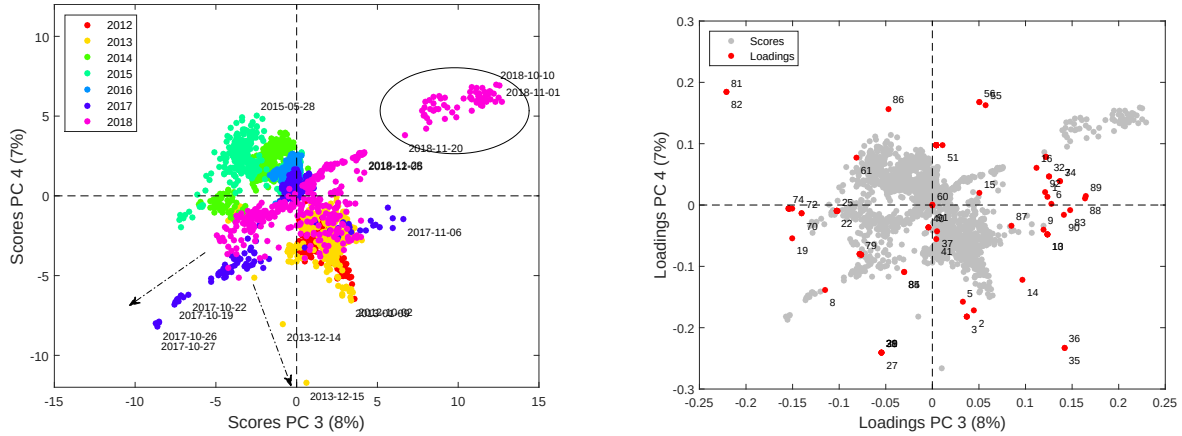
The analysis is completely interactive in a regular computer, meaning that the processing time to obtain each of the plots shown in this section is in the order of seconds.

5.3. Pre-diagnosis

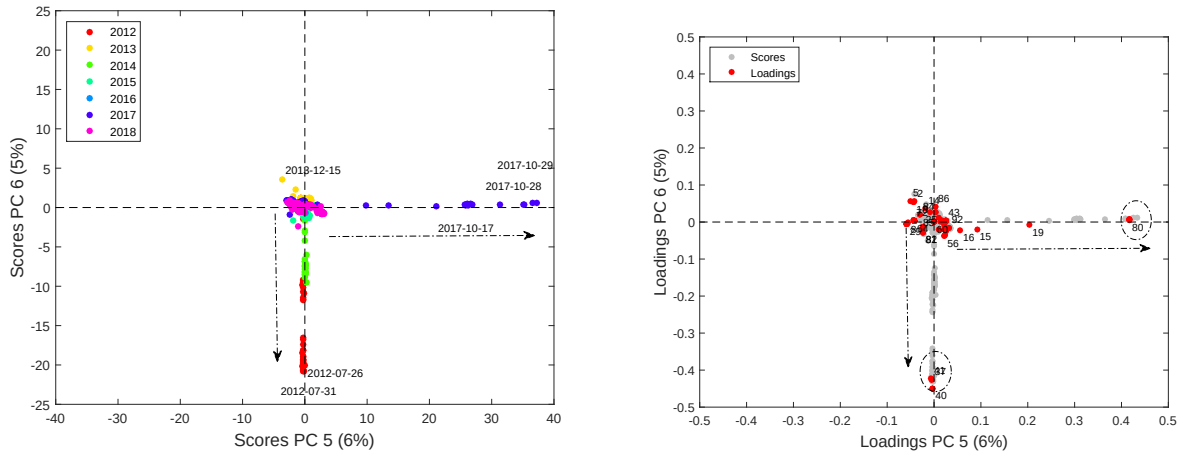
The plots discussed in the previous step provide a general view of the data and can assist in detecting patterns, like trends, outliers, excursions or clusters. However, if we want to have a closer detail on the interaction observations-features, multivariate diagnosis tools are more accurate and easier to interpret.



(a) PC1 vs PC2



(b) PC3 vs PC4



(c) PC5 vs PC6

Figure 4: PCA scores: PC1 vs PC2 (a), PC3 vs PC4 (b) and PC5 vs PC6 (c).

There is a large number of multivariate diagnosis tools, see [30, 31] for a comparison. The MEDA Toolbox includes the oMEDA plot for that purpose. It is a bar plot of the features, built to compare two groups of observations: e.g., observations in an excursion from normal observations. Each bar represents the contribution of the feature to the difference between both groups. A positive bar implies that the first group of observations presents a higher value in the corresponding feature than the second group. A negative bar reflects the opposite. A bar close to zero fro a feature means that both groups of observations have a similar value in that feature.

To illustrate the use of oMEDA, we selected the excursions in 2013 and 2017, which where shown as the main outliers in the Q-statistic and in the D-statistic, respectively. The plots are

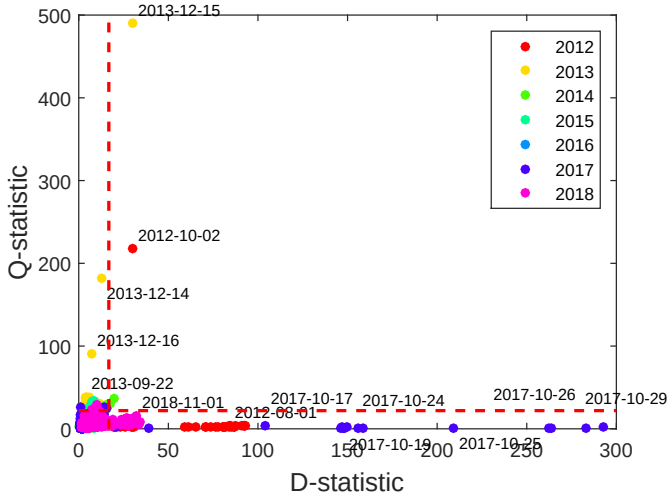
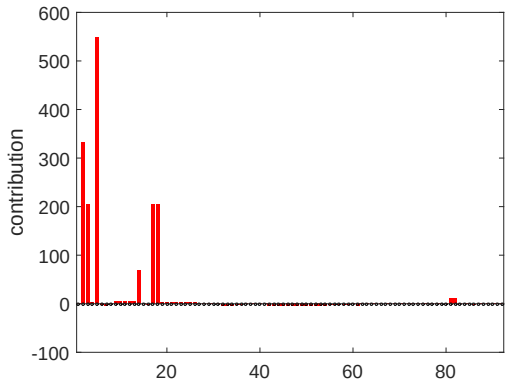
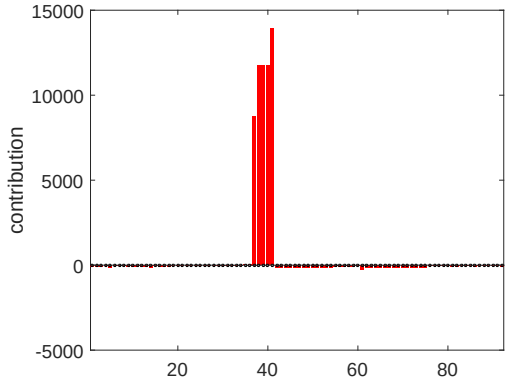


Figure 5: Multivariate Statistical Network Monitoring plot: D-statistic vs Q-statistic.



(a) Excursion in 2013



(b) Excursion in 2017

Figure 6: Pre-diagnosis of the excursions of 2013 and 2017 with oMEDA.

shown in Figure 6. Both of them reflect that each of the excursions are related to a higher value in a different set of features. The specific features are listed in Table 3. We determined that the first excursion is related to a large number of Authentication Fails, one order of magnitude higher than usual. The second excursion is related to an unprecedentedly high number of re-starts of APs, two orders of magnitude higher than usual.

As for 2018, the network staff did not have any registers for these old anomalies, but they suggested that the second one could be related to the installation of a security patch after the publication of a vulnerability. Effectively, October 16th of 2017, the famous KRACK attack against WPA2 [32] and the corresponding patch were released to the public. Even if a restart is necessary after a patch installation, the number and duration (15 days) of the event is remarkable, evidencing that a major management problem took place.

Again, the pre-diagnosis step is in the order of seconds and easily done in a regular computer.

5.4. Deparsing

While the visualizations already provided to the analyst are very informative, it is a good idea to identify the raw log entries related to the patterns found, in order to obtain more detail about them. Examples of findings that may be interesting for us are the specific APs and stations involved in the large number of authentication failures in 2013, or the APs involved in the long period of restarts in 2017. To obtain more detail, the FCParser is employed again in the deparsing step.

The deparsing algorithm [8] takes as inputs the timestamps of a given anomalous pattern, detected in step 3), and the features associated with it, identified in step 4). With this information, the FCParser matches the regular expressions of the features in the specific raw data files. The output is the set of raw log entries that matches at least one of the features, ordered by the number of features they match. With these log entries, the analyst can extract detailed information about the anomalies.

The deparsing algorithm was applied to the excursions in 2013 and 2017 in the Wi-Fi data set. We parallelized the processing using the Anthill Computer Cluster and multi-threading (4 threads per processor), an with as many parallel jobs as days in the excursions. The first excursion took 30 minutes to be processed, and the second one 135 minutes. Some statistics of the deparsing are provided in Table 4. We can see that the number of log entries and nodes involved in the excursion is very large. This means that it is hopeless to expect the analyst to manually inspect this data.

6. Visualization

Given the large number of log entries in the deparsing information, it is hard to extract useful information of the nodes (devices, APs) involved in the anomalies. To improve the detail of the analysis, we combined the output of the deparsing with state-of-the-art visualization techniques. Given that the data represents connections, Gephi [10] seems a suitable tool to visualize the deparsed data. Notice that even if Gephi can be applied to very large sets of connections and nodes, we can only visualize a little portion of the Wi-Fi Big Data set. Thus, the integration of MBDA and Gephi is necessary, and we make the most of the advantages of both approaches.

Table 3: Pre-diagnosis of the excursions of 2013 and 2017 with oMEDA.

Timestamps	Features selected
2013-12-14 – 2013-12-16	bsnDot11StationAuthenticateFail, bsnAuthenticationFailure, bsnDot11StationAssociateFail, bsnStationReasonCode, bsnAuthFailureUserType, bsnAuthFailureUserName
2017-10-16 – 2017-10-30	ciscoLwappApIfUpNotify, ciscoLwappApIfDownNotify cLApAdminStatus, cLApSysMacAddress, cLApPortNumber

Table 4: Deparsing of the excursions of 2013 and 2017 with oMEDA.

Timestamps	log entries/tot	#APs	#Stations	#Users
2013-12-14 – 2013-12-16	5.4M/8.4M (64%)	824	595	103
2017-10-16 – 2017-10-30	19.0M/64.1M (30%)	1,376	0	0

Figure 7 shows the anomaly in 2013, where according to our previous findings, the number of authentication failures increased in one order of magnitude in comparison to a regular day. The anomaly took place during one entire weekend, and as a reference we present the log entries of the previous weekend in Figure 7(a). Colored nodes in the graph represent user devices, colored and labeled by the manufacturer. We use the manufacturer as a layer of anonymization for the devices that maintains information to identify device-specific patterns. Grey nodes represent APs. Sizes in nodes and edges represent the number of log entries they appear in in the capture, and are consistent in the two plots. The reference day shows a trend we also identified in [9]: there is a Hewlett Packard station generating a massive amount of log entries. The station is continuously associating to a high number of APs. The MAC of the station was identified and reported to the network staff, but it is noteworthy that this behavior went unnoticed during years. Aside from that, we can identify a very popular AP: rockefeller-3-1-ap.

Let us focus now on the anomaly in Figure 7(b). We can see that the authentication problem can be considered a general network failure, since it affects a large number of APs and stations. We also see a large amount of authentication failures in rockefeller-3-1-ap and the Hewlett Packard station, but this is expectable during a general network failure, given the large number of associations of these nodes in a regular day. The visualization also shows a curious pattern in Wi-Fi cards made by Zebra Technologies and Shenzhen Reecam Tech. We found log entries of some of these machines in the reference day, but the number was almost negligible. However, during the network failure, all stations with these cards presented a very large number of association attempts, probably reflecting a very aggressive connection approach. We also see several Apple stations with a similar pattern, and to a lesser extent some Intel stations. For stations in blue, the manufacturer is not identified and it is labeled as IEEE Registration Authority.

The second anomaly under consideration, representing a very large number of AP re-starts from 2017-10-16 to 2017-10-30, is shown in Figure 8. Recall that this behavior was related to the installation of a major security patch in Wi-Fi. However, we can see that the number of re-starts is too high: the number of traps go above the 1M for some APs, and span the whole 15 days period.

The previous two plots improve the interpretation of the anomalies to a large extent, and point out to specific actors (stations, APs) in the problem, which simplifies the troubleshooting.

7. Model update

Once we have identified, diagnosed and troubleshot a number of anomalies, we would like to correct the model to remove their effect. This will give us the opportunity to focus on additional, less distinct, anomalies, and also to use the model for the monitoring of future days (e.g. with MSNM).

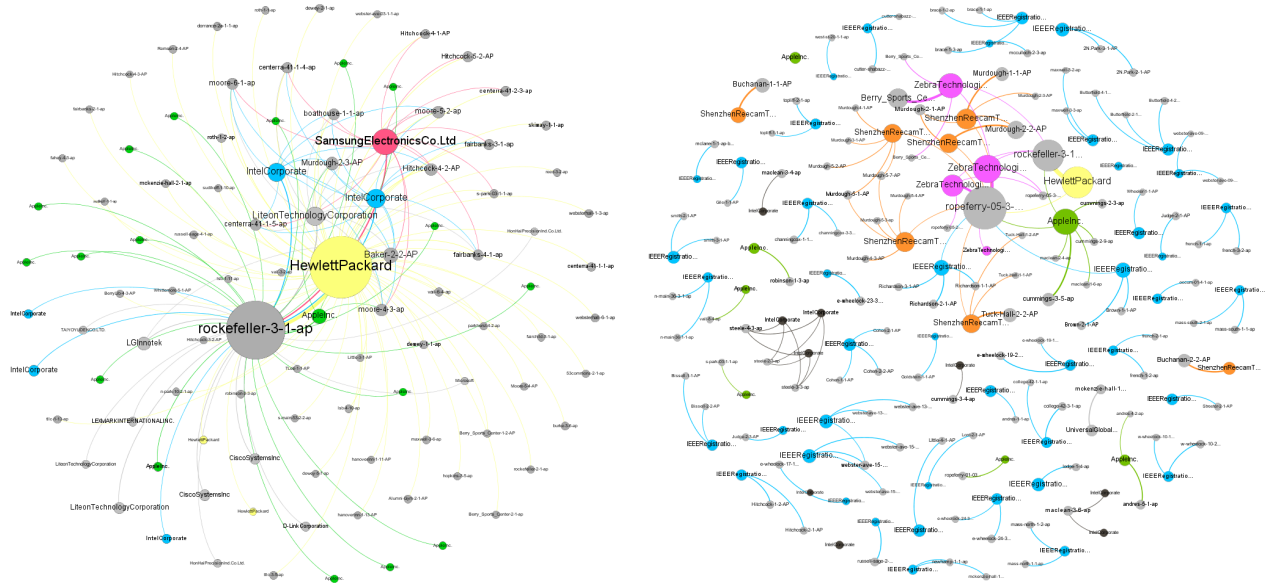
We can think of two approaches to extract the already analyzed anomalies from the model. One choice is to discard the raw log entries de-parsed for the anomaly. This requires little processing time. We will call this approach log-wise extraction. Another possibility, which requires even less processing, is discarding the complete feature vector corresponding to the anomalous observation. This has the implication that some non-anomalous raw log entries will also be discarded in the process. We will call this approach observation-wise extraction.

In Fig. 9, the score plot for PC1 vs PC2 in the original model is compared to the models after log-wise and observation-wise extraction of the two anomalies analyzed in previous section. We can see that the log-wise method can have unexpected consequences. In our case, the two anomalies reflect a general network failure that prevented common connections from taking place. After extracting the anomalous log entries, the remaining activity is lower than in other regular days, and for this reason the observations deviate from the rest. If we extract the complete days, this problem is obviously avoided.

Fig. 10 compares the MSNM plots after both forms of extraction. The log-wise extraction leads to detect the same anomalies as before, which interfere with the performance of the method to detect new anomalies. Contrarily, the observation-wise extraction cleans the model an allows the identification of new anomalies, which can be subsequently diagnosed, deparsed and visualized.

Clearly, this example points to the use of observation-wise extraction as the model update rule. However, it should be noted that if the anomaly has not such leverage on the log entries of a day, log-wise extraction may be suitable and even preferred.

Table 5 lists anomalies found in two iterations of MBDA. We can see that AP re-starts and RADIUS server problems are recurrent. Some of these may anomalies may point to maintenance operations, in which case the analyst may directly discard the day in the monitoring model. More iterations lead to more anomalies.



(a) Reference (b) Authentication failures

Figure 7: Gephi: Reference plot for the log entries of the weekend 2013-12-07 – 2013-12-09 (a) and authentication failures during weekend 2013-12-14 – 2013-12-16 (a) and d. Visualizations are filtered for nodes below 5,000 log entries and edges below 1,000 log entries, which are not shown.

Table 5: Anomalies detected in two iterations of MBDA.

Timestamps	Description	Level (compared to normal)	Iteration
2013-12-14 – 2013-12-16	Authentication Failures	10x	1
2017-10-16 – 2017-10-30	APs Re-starts	100x	1
2012-06-08 – 2012-08-01	Failures in RADIUS server	40x	1
2012-10-02	Failures in RADIUS server	100x	1
2014-05-06 – 2014-05-28	Failures in RADIUS server	40x-100x	1
2013-01-23, 2015-05-11, 2015-09-22, 2015-10-22, 2015-12-09–2015-12-10, 2018-03-13, 2018-11-22–2018-11-25, 2018-12-19 – 2018-12-20	APs Re-starts	10x-20x	2
2012-01-09, 2012-01-16, 2012-01-24,2012-02-08, 2012-02-13, 2012-04-12, 2012-11-01, 2013-06-11, 2014-05-05, 2014-08-16, 2015-03-22, 2016-07-28, 2017-12-19, 2018-04-12	Failures in RADIUS server	20x-50x	2

8. Conclusion

In this paper, we show a case study of the application of Multivariate Big Data Analysis (MBDA) tool, a networkmetric tool optimized to analyze Big Data streams. The application is concerned with the detection, diagnosis and troubleshooting of communication failures in a Wi-Fi campus network. The results illustrate that multivariate methods can bring light into complex massive data sets.

MBDA can work on top of parallel hardware in order to speed up computation. We analyzed 7TB of data in a little more than a day, and this can be reduced to a couple of hours in a high-throughput cluster. While parallel computations are required to analyze Big Data, interaction with the data can be done in a regular computer, combining the advantages of interactive models with the power of parallel processing.

The core of MBDA in this paper is Principal Component Analysis. However, the MBDA framework can be easily extended to other exploratory models, more powerful to analyze specific data sources, like sparse methods, regression & classification methods, n-way models, constrained (e.g. non-negative) methods and many others.

Acknowledgement

This work was supported by Dartmouth College, and in particular by the many network and IT staff who assisted us in configuring the Wi-Fi network infrastructure to collect data, and who patiently answered our many questions about the network and its operation. We furthermore appreciate the support of research colleagues and staff who have contributed to our data-collection and data-analytics infrastructure over the years: most notably Wayne Cripps, Tristan Henderson, Patrick Proctor, Anna Shubina, and Jihwang Yeo. Some of the Dartmouth effort was funded through support from ACM SIGMOBILE and by an early grant from the US National Science Foundation un-

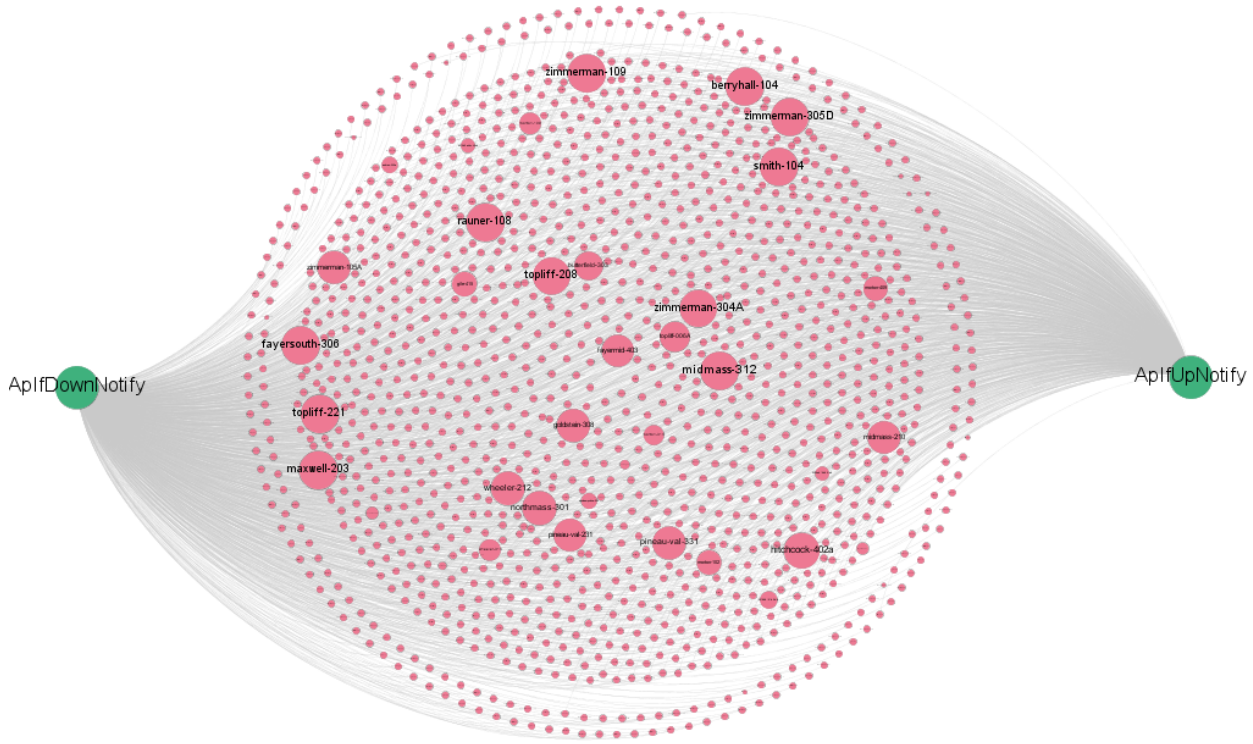


Figure 8: Gephi: AP re-starts in the period 2017-10-16 – 2017-10-30. For reference, the regular amount of re-starts is two orders of magnitude lower and would not be perceptible in this plot.

der award number 0454062. This work was also supported by the Ministerio de Educación, Cultura y Deporte under the Programa Estatal de Promoción de Talento y su Empleabilidad en I+D+i, Subprograma Estatal de Movilidad, del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013-2016, grant number PRX17/00320 (Associated to a Fulbright Scholarship), and the Plan Propio de la Universidad de Granada, grant number PP2017.VS.02. Jose Manuel García-Giménez is acknowledged for his enthusiastic work on the FCParser.

References

[1] A. Ferrer, Latent structures-based multivariate statistical process control: A paradigm shift, Quality Engineering 26 (1) (2014) 72–91.

[2] J. Camacho, G. Maciá-Fernández, J. Díaz-Verdejo, P. García-Teodoro, Tackling the big data 4 vs for anomaly detection, in: Proceedings of IEEE INFOCOM, 2014, pp. 500–505. doi:10.1109/INFCOMW.2014.6849282.

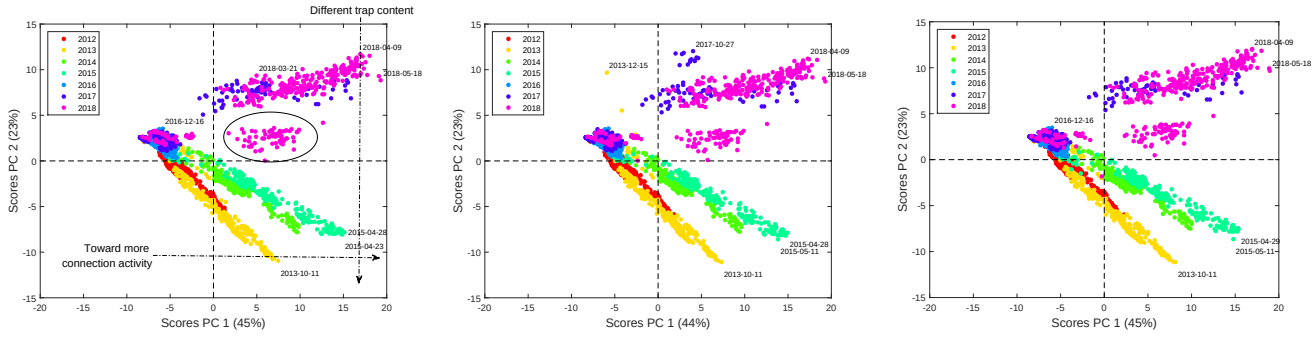
[3] J. Hernández-Méndez, F. Muñoz Leiva, J. Sánchez-Fernández, The influence of e-word-of-mouth on travel decision-making: consumer profiles, Current Issues in Tourism 1-14 (2013) 1–21. doi:10.1080/13683500.2013.802764.

[4] I. Jolliffe, Principal component analysis, Springer Verlag Inc., EEUU, 2002.

[5] H. Zou, T. Hastie, R. Tibshirani, Sparse Principal Component Analysis, Journal of Computational and Graphical Statistics 15 (2) (2006) 265–286. arXiv:1205.0121v2, doi:10.1198/106186006X113430.

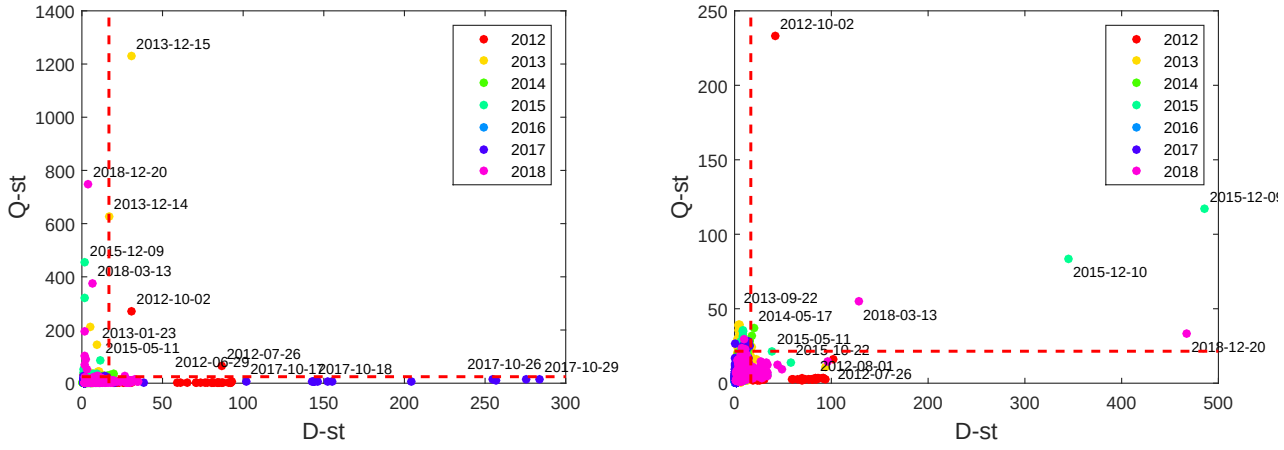
[6] R. Bro, Multi-way analysis in the food industry - models, algorithms, and applications, Tech. rep., MRI, EPG and EMA, Proc ICSLP 2000 (1998).

[7] J. Camacho, R. Magán-Carrión, P. García-Teodoro, J. J. Treinen, Networkmetrics: Multivariate big data analysis in the context of the Internet, Submitted to Journal of Chemometrics (Wiley) (2016) 45.



(a) (b) (c)

Figure 9: PCA scores PC1 vs PC2: original model (a), after log-wise extraction of outliers (b) and after observation-wise extraction (c).



(a) (b)

Figure 10: MSNM plot after extraction of outliers: log-wise extraction (a) and observation-wise extraction (b).

[8] J. Camacho, J. M. García-Giménez, N. M. Fuentes-García, G. Maciá-Fernández, Multivariate big data analysis for intrusion detection: 5 steps from the haystack to the needle, Submitted to Computer & Security.arXiv:1906.11976.

[9] J. Camacho, C. McDonald, R. Peterson, X. Zhou, D. Kotz, Longitudinal analysis of a campus Wi-Fi network, Submitted to Computer Networks.

[10] M. Bastian, S. Heymann, M. Jacomy, Gephi: An open source software for exploring and manipulating networks, in: International AAAI Conference on Weblogs and Social Media, 2009.
URL `http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154`

[11] D. Kotz, K. Essien, Analysis of a campus-wide wireless network, Wireless Networks 11 (1–2) (2005) 115–133. doi:10.1007/s11276-004-4750-0.

[12] T. Henderson, D. Kotz, I. Abyzov, The changing usage of a mature campus-wide wireless network, Computer Networks 52 (14) (2008) 2690–2712. doi:10.1016/j.comnet.2008.05.003.

[13] J. Case, M. Fedor, M. Schoffstall, J. Davin, A Simple Network Management Protocol (SNMP), RFC 1157, RFC Editor (May 1990).
URL `https://www.rfc-editor.org/rfc/rfc1157.txt`

[14] Eduroam: World wide education roaming for research & education, `https://www.eduroam.org/`, accessed: 2018-09-30.

[15] J. Camacho, A. Pérez-Villegas, R. A. Rodríguez-Gómez, E. Jiménez, Multivariate exploratory data analysis (MEDA) toolbox for Matlab, Chemometrics and Intelligent Laboratory Systems 143 (0) (2015) 49–57. doi:10.1016/j.chemolab.2015.02.016.

[16] GitHub repository for the MEDA Toolbox, `https://github.com/josecamachop/MEDA-Toolbox`, accessed: 2018-09-30.

[17] GitHub repository for the FCParser, `https://github.com/josecamachop/FCParser`, accessed: 2018-09-30.

[18] Open grid scheduler, `http://gridscheduler.sourceforge.net`, accessed: 2018-09-30.

[19] J. Jackson, A User's Guide to Principal Components, Wiley-Interscience, England, 2003.

[20] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, G. Maciá-Fernández, PCA-based multivariate statistical network monitoring for anomaly detection, Computers & Security 59 (2016) 118–137. doi:10.1016/j.cose.2016.02.008.
URL `http://www.sciencedirect.com/science/article/pii/S0167404816300116`

[21] J. V. Kresta, J. F. Macgregor, T. E. Marlin, Multivariate statistical monitoring of process operating performance, The Canadian Journal of Chemical Engineering 69 (1) (1991) 35–47. doi:10.1002/cjce.5450690105.

[22] P. Nomikos, J. F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE Journal 40 (8) (1994) 1361–1375. doi:10.1002/aic.690400809.

[23] A. Ferrer, Multivariate statistical process control based on principal component analysis (MSPC-PCA): Some reflections and a case study in an autobody assembly process, Quality Engineering 19 (4) (2007) 311–325. doi:10.1080/08982110701621304.

[24] E. Saccenti, J. Camacho, Determining the number of components in principal components analysis: A comparison of statistical, crossvalidation and approximated methods, Chemometrics and Intelligent Laboratory Systems 149, Part A (2015) 99–116. doi:10.1016/j.chemolab.2015.10.006.
URL `http://www.sciencedirect.com/science/article/pii/`

S0169743915002579

[25] E. Saccenti, J. Camacho, On the use of the observation-wise k-fold operation in PCA cross-validation, Journal of Chemometrics 29 (8) (2015) 467–478. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.2726, doi:10.1002/cem.2726.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.2726

[26] K. Gabriel, The biplot graphic display of matrices with application to principal component analysis, Biometrika 58 (1971) 453–467.

[27] T. Kourti, P. Nomikos, J. F. MacGregor, Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS, Journal of Process Control 5 (4) (1995) 277–284. doi:10.1016/0959-1524(95)00019-M.

[28] J. A. Westerhuis, S. P. Gurden, A. K. Smilde, Generalized contribution plots in multivariate statistical process monitoring, Chemometrics and Intelligent Laboratory Systems 51 (2000) 95–114.

[29] J. Camacho, Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models, Journal of Chemometrics 25 (11) (2011) 592–600. doi:10.1002/cem.1405.

[30] C. F. Alcala, S. J. Qin, Reconstruction-based contribution for process monitoring, Automatica 45 (7) (2009) 1593–1600.

[31] M. Fuentes-García, G. Maciá-Fernández, J. Camacho, Evaluation of diagnosis methods in PCA-based multivariate statistical process control, Chemometrics and Intelligent Laboratory Systems 172 (2018) 194–210. doi:10.1016/j.chemolab.2017.12.008.
URL http://www.sciencedirect.com/science/article/pii/S0169743917302046

[32] M. Vanhoef, F. Piessens, Key reinstallation attacks: Forcing nonce reuse in WPA2, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS '17, ACM, 2017, pp. 1313–1328. doi:10.1145/3133956.3134027.
URL http://doi.acm.org/10.1145/3133956.3134027