Managing Cloud Infrastructures by a Multi-Layer Data Analytics

Arnak V. Poghosyan, Ashot N. Harutyunyan, and Naira M. Grigoryan
VMware
{apoghosyan;aharutyunyan;ngrigoryan}@vmware.com

Abstract—**Today's IT management faces the problem of "virtualized big environments" with hundreds of thousands of objects/resources as virtual machines, hosts, clusters, etc., evolving into cloud services. Admins of those infrastructures heavily rely on smart data-agnostic approaches to get reliable and accurate information regarding any current or upcoming health deterioration, increasingly requesting more proactive solutions. We architected a multi-layer enterprise analytics that employs statistical and machine learning methods to maximally automate the data center operations for an optimal performance management. We share several experience stories on application of the developed approaches and address noise and complexity reduction requirements to increase the operational efficiency of the analytics.**

Keywords–data analytics; data center automation;correlation analysis; outlier detection; thresholding; alarming; complexity reduction; noise reduction; abnormality degree; alteration degree; data categorization; alarm ranking; root cause

I. INTRODUCTION

Effective and actionable recommendations with reasonable resource consumption are the main expectations from the state-of-the-art management systems of very large-scale data centers. Those expectations are hard to achieve because of complex and sophisticated nature of those infrastructures with growing elasticity and transiency in cloud era. In modern environments time is a key factor for success and thus the operational complexity of any toolset must be congruent to the gain in effectiveness it produces.

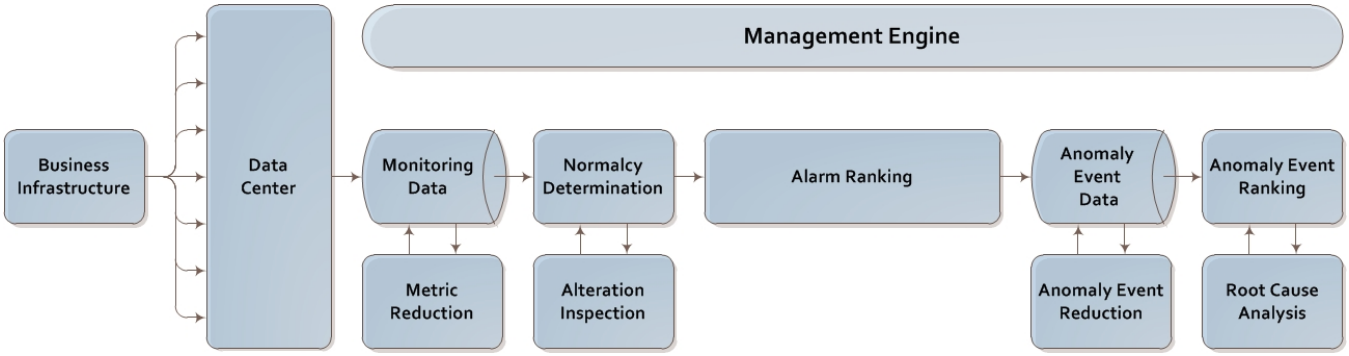
Over the last decade many companies built large suites for IT management. However, the preliminary goal of having a unique product that will manage everything from a "single pane of glass" was not fulfilled. They succeeded in managing specific applications and infrastructure sectors (network, capacity, storage, applications, etc.) where expert knowledge can be converted into helpful recommendations for understanding what to monitor, how to define out-of-the-box alerts and where to search the root causes of the problems. Construction of general frameworks fails due to well-known problems.

While managing environments, all the tools that we have today generate massive volume of metrics and logs. From the customer perspective the fear of not overlooking something important is forcing the collection of everything from everywhere which leads to data monitoring overload. Infrastructure anomalies are typically identified based on those measured metrics when they violate user defined or automatically determined thresholds which in its turn leads to anomaly alert storms which are impossible to handle appropriately. This leads to finger pointing without real understanding the sources of the problems. As a result, admins tend to overprovision resources because experience tells that when you have enough resources, there shouldn't be any problem.

Efforts to identify, isolate those abnormalities and further provide with appropriate recommendations within a reasonable time slot are becoming more and more challenging as infrastructures are complex and dynamic. Moreover, infrastructures become so probabilistic in sophisticated interrelation between constituent objects that growing volume of monitored data doesn't necessarily lead to growth of accuracy of recommendations and their effective management at the troubleshooting level.

In this paper we pursue two goals. The first one is to show how the well-known data science techniques can be compiled into a data-agnostic multi-layer analytics for self-managing of software defined data centers [1]. The second goal is systematic application of complexity reduction techniques on different levels of management hierarchy for increasing the accuracy of recommendations while decreasing the operational complexity.

The paper is organized as follows. Section 2 introduces a specific instantiation of a multi-layer data analytics which includes different modules for statistical analysis and complexity reduction. Section 3 describes an application of the correlation analysis for partially resolving the problem of data monitoring overload. Section 4 illustrates a procedure for data normalcy determination based on the extreme value theory. Alarms are generated when data points are in "out of normal" states and Section 5 presents a procedure of alarm ranking for identifying worthwhile alarms known as anomaly events. Section 6 provides a procedure for alteration inspection in data indicating those time series that need to be re-analyzed for behavioral normalcy pattern detection. Section 7 describes another step for reducing the volume of alert storms by inspecting the correlations between anomaly events. Uncorrelated anomaly events can be further ranked by their importance which leads to better determination of the root causes of the problems (see Section 8). Throughout the paper, we present some experience stories from our experiments on real data. Section 9 gives some concluding remarks.



Flowchart 1. A multi-layer data analytics.

II. Modules of Data Analytics

Flowchart 1 illustrates a specific realization of the multi-layer data analytics based on the above described principles. The system hierarchically applies learning tools on different levels of abstraction, starting from data monitoring up to root cause analysis of performance problems, and on each level performs noise/complexity reduction for increasing the operational efficiency.

The first step is Metric Reduction which identifies constant data and eliminates correlated metrics. Constant metrics don't participate in further analysis and stay under watch for changes in their behavior. Uncorrelated metrics pass through the next Normalcy Determination module that calculates the normalcy bounds (upper and lower dynamic thresholds) based on historical data [2]-[6]. Alarms are generated when metric data violate historical normalcy bounds with unusual degree of abnormality. Alarm Ranking provides tools for creating anomaly events (alerts) based on their abnormality degree calculation [7].

Alteration Inspection keeps track of the measures of data-to-threshold relation [1] based on previously constructed normalcy bounds and newly arrived data to predict if the time has come to re-run the DT analytics for the entire historical data. In other words, by comparison of the normal characteristics with the ones calculated for the current time period, we determine an alteration degree that indicates if the process should undergo threshold recalculation.

In the later stage, the Event Reduction performs next-level reduction [1] by eliminating metrics that have correlated anomaly events. Actually, this module performs further reduction of the anomaly event set. Uncorrelated anomaly events pass through the next Anomaly Ranking module which sorts anomaly events by their importance and impact on the system. Root Cause Analysis module identifies core anomaly events responsible for the performance degradation [8,9].

III. Metric Reduction

Tremendous increase in dimensionality of data in modern cloud infrastructures makes an adequate information reduction a core component in any learning application. By removing redundant metrics, we increase learning accuracy and improve recommendations by decreasing the overall complexity of data analytics. Different approaches for complexity reduction is known in the literature (see [10,11] with references therein) that can be applicable not only for specific applications but also for general frameworks like ours. From our experience, we concluded that classical principal component analysis (PCA) was rather simple and efficient procedure for elimination of the correlated metrics.

We applied a slightly adjusted version of PCA, where instead of constructing the principal components (orthonormal basis), we selected independent metrics (common basis) for further analysis. The foundations of the basic theory of the PCA remained unchanged. We calculated eigenvalues and performed QR decomposition of the correlation matrix [12,13] for numerical rank calculation and for further selection of the independent metrics. Eigenvalues and numerical rank may be calculated also by the Singular Value Decomposition [13].

The Metric Reduction module performs reduction on the metric level by eliminating correlated metrics with the method mentioned above. Some of the monitored metrics are coming from unloaded parts of the infrastructure and behavioral analysis of such constant metrics is worthless. In the sense of complexity reduction such metrics can be eliminated from further analysis (normalcy determination, alert generation, etc.). However, we have to continue monitoring of those metrics as jumps of data points from the baseline bear important information on the underlying processes.

From the other side, complexity reduction in large systems is a complex problem itself and appropriately chosen topology for splitting the system into subsystems for further analysis makes the solution more feasible. In general, monitoring tools naturally regroup time series into metric-groups (such as CPU, network, IOPS, memory groups) for each object/IT resource (such as virtual machines). The metrics within a group have higher chance to be better correlated than the metrics from across groups. Most metric-groups contain a relatively small number of metrics suitable for PCA-like reduction.

Experience Story I. We ran experiments on metric groups of virtual machines (VM). We arrived at some

insights from an internal cloud environment of active usage, consisting of 4609 VMs, 987618 VM metrics. Average count of metrics per VM was 276 (maximum is 344).

Figs. 1 and 2 illustrate the volumes of the categories we got for this environment. Fig. 1 shows the percentage of constants, dependents, and independents against all metrics, and Fig. 2 shows the percentages of dependents and independents against variable metrics.

The first interesting fact is the rather big percentage (46%) of the constant metrics. We don't need to include this metrics into our complex management analytics until they become variable.

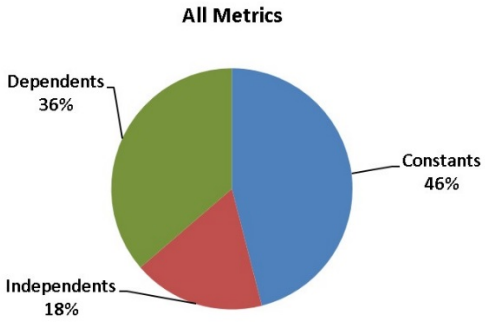


Figure 1. Categories of metrics.

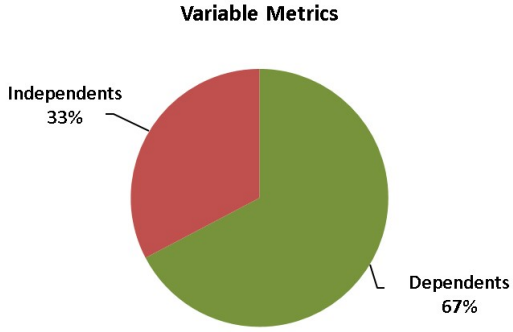


Figure 2. Independents vs dependents.

The second interesting observation is the small percentage of independent metrics among the all or variable metrics. Dependent metrics duplicate the behavior of independent metrics or their linear combinations. In further analysis this can be taken into account resulting in a reduction of representation complexity of the systems under management.

The third important observation is the correlation coefficient between the counts of variable and dependent metrics. It is equal to 0.94 calculated across different VMs. It means that almost no VM is deviating from the law that higher count of variables implies higher count of dependents. This is the first level of dimensionality reduction in virtualized environments.

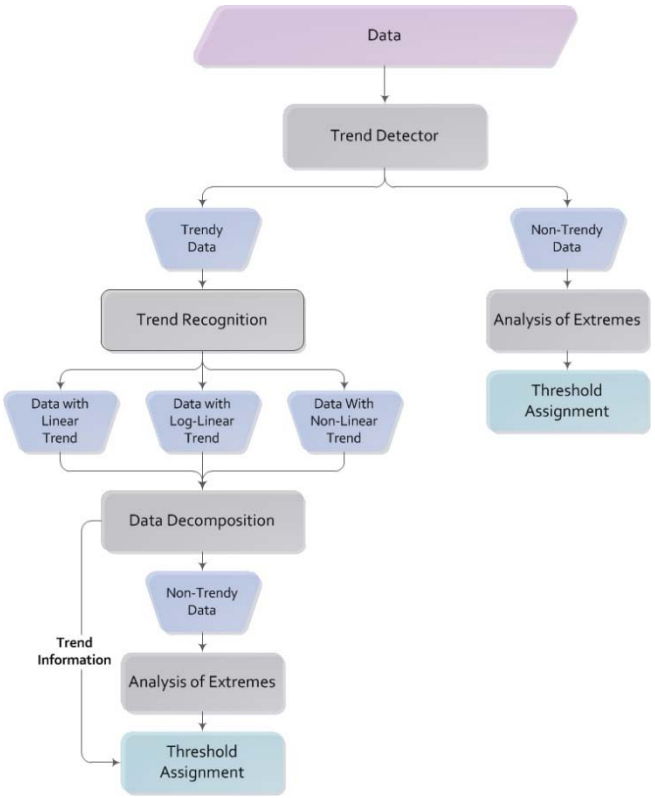
IV. Extreme Value Theory for Normalcy Determination

Normalcy Determination applies behavioral analysis to each metric over time for learning "normal" and "out of normal" patterns. It derives upper and lower control lines (thresholds) of normal behavior for capturing data points which can be candidates for the abnormal behavior of a metric.

In the classical theory of control charts [14] it is assumed that time series have "bell-shaped" normal distribution and "out of normal" states can be determined using the mean and standard deviation of the corresponding processes. If, for example, a data point of the process is away from the mean by 3 standard deviations, then the point is assumed as out of control. In IT environments, underlying distributions of data metrics are diverse and "bell-shaped" ones are critically rare. Hence, it is unreasonable to rely on Gaussian models in our anomaly detection problems.

We propose a fully data-agnostic normalcy determination method based on Extreme Value Theory (EVT) [15]. The approach that we discussed in [2]-[5] was rather resource-consuming due to time-dependent behavioral analysis of the metrics. Current approach [6] is less expensive due to time-independent analysis of extremes.

Flowchart 2 illustrates the general concept behind the data normalcy analysis.



Flowchart 2. Principal algorithm of threshold assignment.

Trend Detector identifies those metrics that are trendy. It performs different classical tests for trend determination [16,17]. If data is classified as non-trendy, then further analysis of extremes determines the threshold values. If data is classified as trendy, Trend Recognition identifies its nature. The latter can be linear, log-linear, or non-linear. We check linearity of trend by the common least square linear regression analysis. If the overall trend is non-linear, but the latest enough long data has a linear trend, then we select it for further analysis of extremes and threshold determination. If the trend of data or its latest period is not linear then log-linearity is checked by the same procedure for $x(e^t)$, where $x(t)$ is the original time series. Finally, if data (or its latest portion) is neither linear nor non-log-linear, then trend is declared as non-linear and the latest period of data is selected for threshold assignment.

Data Decomposition module decomposes original time series $x_0(t)$ (or its latest portion) into a sum of non-trendy time series $x(t)$ and trend component $trend(t)$

$$x_0(t) = x(t) + trend(t).$$

The Analysis of Extremes performs threshold determination for $x(t)$ based on the analysis of the "tail" of non-trendy data. The final threshold for the original data $x_0(t)$ can be reconstructed by adding the trend. We apply EVT methods for automated threshold assignments to time series data. The algorithms are based on data "tail" determination according to different concepts.

The first concept leads to a parametric procedure and defines data "tail" based on the measure of decay of the distances of data points over threshold (POT). The line above which the points give the best fit (if possible) to a parametric family of predefined distributions (that measures the corresponding decay) is set as a control line. The second concept leads to a non-parametric procedure and defines the "tail" based on the measure of uncertainty of data distribution above threshold. The level for which its POT's indicate the maximum uncertainty is set as a control line. If the tail of time series can't be fit by neither parametric nor non-parametric procedures, then we postpone threshold determination until additional data points will be available for tail analysis.

Parametric Approach. We fit data tail by standard distributions provided by EVT [15]. In case of power law, the Cauchy distribution can be verified. In case of a sub-exponential decay, the Generalized Pareto (GP), Generalized Extreme Value (GEV), Weibull, and other distributions can be verified. If the tail of data over given threshold can be appropriately fitted by the selected parametric model, then those POT's behave as "outliers" and the lowest threshold can be used for calculation of the values at risk [18].

Non-Parametric Approach. We calculate the well-known entropy measure [19]

$$H(X_c) = -\sum_{k=1}^{n} p_k \, log_n(p_k)$$

for the tail of data X_c, where

$$X_c = \{x_k - q_c\}_{k=1}^{M}$$

for different quantiles q_c with $0.9 \le c \le 0.99$ and data points $x_k, k = 1, \dots, M$ exceeding the quantile. Probabilities $0 \le p_k \le 1, k = 1, \dots, n$ can be measured by the corresponding histograms with n number of bins ($n = 10$ in our experiments). Entropy values measure the uncertainty of X_c. The quantile q_c that corresponds to the maximum entropy indicates the upper control line. It means that we detect the data portion that is maximally away from its dominating concentrations. If there is no maximum value or the maximum value is less than 0.5 then the data is intractable in terms of this non-parametric procedure. Non-parametric entropy maximization method is much faster and simpler in terms of implementation than the parameter fitting method described above.

Experience Story II. Consider time series in Figure 3 describing an IT process. Green line in Figure 3 indicates upper control line derived by fitting data tail according to parametric approach. As a predefined distribution we considered the GP. The theory behind parameter determination is well developed (see [20] with references therein). Red line in Figure 3 indicates upper control line derived by non-parametric approach. Figure 4 shows the entropy values for different quantiles of data and the maximum entropy corresponds to $c = 0.961$.

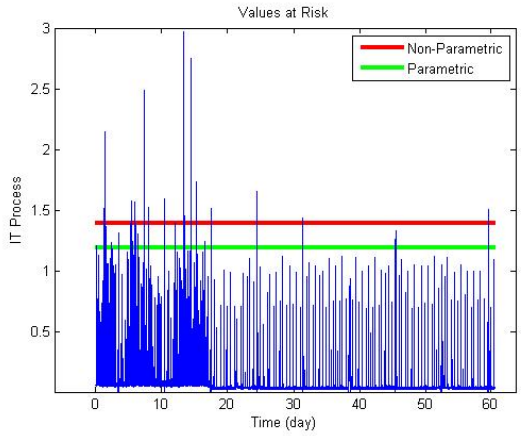


Figure 3. Time series with control lines derived by parametric (green line) and non-parametric methods (red line).

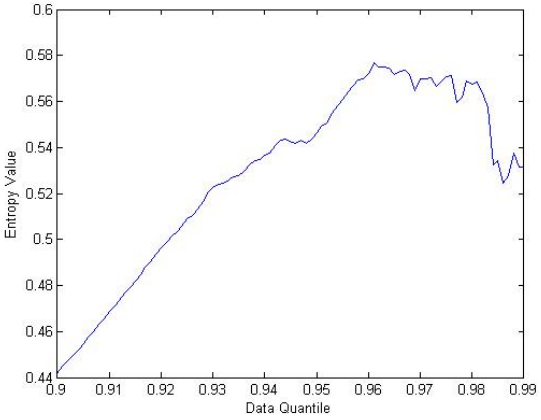


Figure 4. Entropy values for the metric of Figure 3.

V. ALARM RANKING

A proper "out of normal" state declaration is important for minimizing the number of false positive and false negative recommendations. Alarm Ranking module provides an abnormality degree estimation for threshold violations (alarms that may be converted to worthwhile alerts).

The premise behind the analysis is that metrics often violate their control thresholds for certain periods of time and not all violations are worthy for attention. By looking at the historical behaviors of those violations, current-time violations can then be appropriately compared to degree of deviation from the metrics historical behavior; the larger the deviation, the greater the probability that the alarm is noteworthy.

We consider two dimensions of abnormality [7]; the magnitude of data points over the threshold, and the duration of "out-of-threshold" state. Thus, the historical abnormality degree is a vector $G_0 = (\tau_0, d_0)$ composed correspondingly of the averages of historical alarm durations and amplitudes of violations.

When a threshold violation occurs, its duration and averaged amplitude of violations may be measured for a run-time abnormality degree estimate $G_{run} = (\tau_{run}, d_{run})$ (blue points in Fig. 5). Then, the components of G_{run} are compared against the components of G_0. If the both components are higher than the historical ones, then an anomaly event is triggered (red zone in Fig. 5). If both components are less than the corresponding components of the historical abnormality degree then the alarm is in the green zone (safe zone). The rose zone is optional for alert declaration.

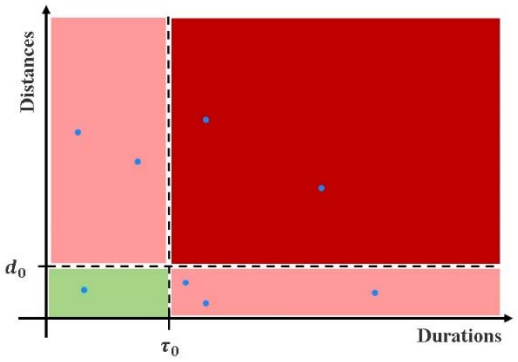


Figure 5. Anomaly generation quadrant.

Two dimensions of the abnormality can be merged into a single measure of alarm (alert) ranking and user can be informed only on the worthwhile events.

VI. ALTERATION INSPECTION

Recalculation of normalcy bounds is a complex procedure. Determination of the exact time of recalculation in case of global or local changes and postponing recalculation for conservative data decreases the complexity and resource consumption in the data analytics. Frequent (mostly daily) computation of normalcy bounds (upper and lower thresholds) for monitoring time series data requires high resource allocation. This problem needs to be addressed especially for very large infrastructures. If the data sets of the infrastructure show little variation, then there is no need of re-computation of thresholds. Therefore, having an engine that keeps track of those variations and identifies the metrics that truly need the behavioral re-compute would substantially reduce the analytics overhead/complexity.

Alteration Inspection module makes a forecast on alteration in data-to-thresholds relation while applying the last computed normalcy bounds and the run-time data (or the data arrived after the last threshold compute). In this way it identifies outdated thresholds and forwards the changed metrics to the normalcy pattern determination.

The data-to-thresholds alteration degree may be computed using the following function

$$g(P,S) = e^{-a(1-P)}\frac{S}{S_{max}}$$

that measures the data-to-thresholds relation, where $a > 0$ is a sensitivity parameter (e.g., $a = 10$), P is the percentage or fraction of data that lie between upper (u) and lower (l) thresholds over a time interval $[t_{min}, t_{max}]$, S_{max} is the rectangle defined by the upper and lower thresholds for the same time interval

$$S_{max} = (t_{max} - t_{min})(u - l),$$

and S is the area between metric values and the lower threshold:

$$S = \frac{1}{2}\sum_{k=1}^{M-1}(x_{k+1} + x_k - 2l)\,(t_{k+1} - t_k),$$

where $x_k = x(t_k)$ is data point at time stamp t_k and M is the number of metric values with time stamps in the time interval $[t_{min}, t_{max}]$.

The data-to-threshold relation $g^{cur}(P,S)$ is computed for a current time interval and compared with a previously computed relation $g^{pre}(P,S)$ for the same metric but for an earlier time interval. When their difference is high the thresholds for the metric are re-computed, otherwise the prior thresholds are still the relevant patterns.

Experience Story III. Our experience shows that most of the metrics of 2-3 months of data are stable and their thresholds don't change frequently. The latter means that those metrics don't introduce a run-time actuality to spend resources for daily behavioral pattern extraction. Since during the computation of thresholds for a given metric its last several months' data is read from the disk and heavily analyzed, it results in a significant consumption of machine's Disk IO, CPU, and Memory resources. Therefore, the alteration inspection is able to totally omit the redundant threshold calculations for a metric. Some details of our experiment on a real data set are as follows: reduction in time complexity of an everyday threshold calculation is 45% and reduction in metric data requests from the database is 70%.

VII. Anomaly Event Reduction

The next important step is event based noise reduction. Investigation of the alerts' set and their interdependence can lead to interesting and important conclusions on infrastructure behavior.

One important subcategory for reduction is the set of metrics without any events. Information about such metrics can provide valuable insights leading to some proactive actions towards optimizations, infrastructure reorganizations, etc. Another important subcategory is the set of metrics with small amount of events. We may rank them by giving high rates to metrics with impactful alarms.

Metrics with acceptable number of anomaly events may be analyzed for revealing event correlations. For each metric $x_k(t)$ we construct its dual metric $x_k^*(t)$ which takes value $x_k^*(t) = 1$ if at time t metric $x_k(t)$ has an active alarm otherwise $x_k^*(t) = 0$. Then, we perform correlation analysis for the dual-metric space finding those metrics that have uncorrelated anomaly events.

Experience Story IV. Experiments were performed for determining the effectiveness of event reduction on real datasets with 3377 metrics, 400 resources and 709 metric groups (with almost two-month data). Event Reduction demonstrates the following outcomes: from 3377 metrics, 1332 (39%) are eliminated as they have less than 5 alarms, 128 (3.8%) are eliminated based on alarm correlations. Together, we get reduction of 1460 (43%) metrics.

VIII. Anomaly Ranking and Root Cause Analysis

The final set of uncorrelated events pass through a further anomaly ranking and root cause identification [8,9] based on historical analysis of events with associated probabilistic correlations. Applying information measures between random variables, which embody those events origins of problems may be detected and used to generate real-time recommendations for locations in a hierarchical system. It also includes a framework for identification of bottlenecks and black swan type issues.

The effectiveness of produced recommendations may be further optimized by feedback analysis [21]-[23].

IX. Conclusion

We introduce and investigate a fully automated data-agnostic management system which performs sequential application of analyses tools for recommendations on different levels of abstraction. On each level, the complexity/noise reduction modules perform optimizations for improving the operational efficiency of the data analytics. Systematic implementation of such dimensionality reduction techniques significantly improves the effectiveness and efficiency of enterprise data analytics with an enhanced user experience while reducing its overhead.

References

[1] A. V. Poghosyan, A. N. Harutyunyan, N. M. Grigoryan, and M. A. Marvasti, Methods and systems to manage cloud computing infrastructures, US patent appl. 14/701066, 2015.

[2] M. A. Marvasti, A. Grigoryan, A. V. Poghosyan, N. M. Grigoryan, and A. N. Harutyunyan, Methods for the cyclical pattern determination of time-series data using clustering approach, US patent US9245000 B2, 2016.

[3] A. V. Poghosyan, A. N. Harutyunyan, N. M. Grigoryan, and M. A. Marvasti, Data-agnostic anomaly detection, US patent US20140298098 A1, 2014.

[4] M. A. Marvasti, A. V. Poghosyan, A. N. Harutyunyan, and N. M. Grigoryan, "An enterprise dynamic thresholding system", Proc. USENIX 11th Int. Conf. ICAC, June 18-20, Philadelphia, PA, pp. 129-135, 2014.

[5] M. A. Marvasti, A. V. Poghosyan, A. N. Harutyunyan, and N. M. Grigoryan, "Statistical normalcy determination based on data categorization", VMTJ, vol. 3, no. 1, 43-55, 2014.

[6] M. A. Marvasti, A. N. Harutyunyan, N. M. Grigoryan, and A. V. Poghosyan, Automated methods and systems for calculating hard thresholds, US patent appl. 14/314490, 2015.

[7] M. A. Marvasti, A. N. Harutyunyan, N. M. Grigoryan, and A. V. Poghosyan, Methods and systems that estimate a degree of abnormality of a complex system, US patent appl. 14/701217, 2015.

[8] M. A. Marvasti, A. V. Poghosyan, A. N. Harutyunyan, and N. M. Grigoryan, Method and apparatus for root cause and critical pattern prediction using virtual directed graphs, US patent 8751867 B2, 2014.

[9] M. A. Marvasti, A. V. Poghosyan, A. N. Harutyunyan, and N. M. Grigoryan, "An anomaly event correlation engine: identifying root causes, bottlenecks, and black swans in IT environments", VMTJ, vol. 2, no. 1, 35-45, 2013

[10] Y. J. Shin, and C. H. Park, "Analysis of correlation based dimension reduction methods", IJAMCS, vol. 21, no. 3, 549–558, 2011.

[11] M. B. Reddy, and Dr. L. S. S. Reddy, "Dimensionality reduction: an empirical study on the usability of IFE-CF measures", IJCSI, vol. 7, issue 1, no. 1, pp. 74-81, 2010.

[12] J. E. Gubernatis, and T. E. Booth, "Multiple extremal eigenpairs by the power method", JCPhys, vol. 227, issue 19, pp. 8508-8522, 2008.

[13] G. H. Golub, and C. F. Van Loan, Matrix Computations, Johns Hopkins, 1996.

[14] W. A. Shewhart, Economic control of quality manufactured product, New York: D. Van Nostrand Company, 1931.

[15] M. Falk, J. Huesler, and R.-D. Reiss, Laws of Small Numbers: Extremes and Rare Events, Springer Basel AG 2011.

[16] H.B. Mann, "Nonparametric tests against trend", Econometrica, 13, 245–259, 1945.

[17] M. G. Kendall, Rank Correlation Methods, Griffin, London, UK, 1975.

[18] G. A. Holton, Value at Risk: Theory and Practice, AP, 2003.

[19] C. E. Shannon, "A mathematical theory of communication", Bell System Technical Journal, 27 (3), 379–423, 1948.

[20] M. L. Rizzo, "New goodness-of-fit tests for Pareto distributions", Astin bulletin, 39(2), pp. 691-715, 2009.

[21] M. A. Marvasti, A. V. Poghosyan, A. N. Harutyunyan, and N. M. Grigoryan, "Ranking and Updating Beliefs based on User Feedback: Industrial Use Cases", Proc. IEEE 12th ICAC, Grenoble, France, July 7-10, pp. 227-230, 2015.

[22] N. M. Grigoryan, M. A. Marvasti, A. V. Poghosyan, A. N. Harutyunyan, and Y. Yankov, Data-agnostic adjustment of hard thresholds based on user feedback, US patent 20150370682A1, 2015.

[23] A. N. Harutyunyan, N. M. Grigoryan, M. A. Marvasti, A. V. Poghosyan, and Y. Yankov, Data-agnostic methods and systems for ranking and updating beliefs. US patent appl. 14/104351, 2013.