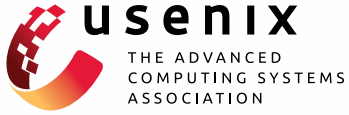


An Enterprise Dynamic Thresholding System

Mazda A. Marvasti, Arnak V. Poghosyan, Ashot N. Harutyunyan, and Naira M. Grigoryan, *VMware, Inc.*

https://www.usenix.org/conference/icac14/technical-sessions/presentation/marvasti

This paper is included in the Proceedings of the 11th International Conference on Autonomic Computing (ICAC '14).

June 18–20, 2014 • Philadelphia, PA

ISBN 978-1-931971-11-9

Open access to the Proceedings of the 11th International Conference on Autonomic Computing (ICAC '14) is sponsored by USENIX.

An Enterprise Dynamic Thresholding System

Mazda A. Marvasti, Arnak V. Poghosyan, Ashot N. Harutyunyan, and Naira M. Grigoryan
Management BU
VMware Inc.
{mazda;apoghosyan;aharutyunyan;ngrigoryan}@vmware.com

***Abstract*— We demonstrate an enterprise Dynamic Thresholding System for data-agnostic management of monitoring flows. The dynamic thresholding based on data historical behavior enables adaptive and more accurate control of business environments compared to static thresholding. We manifest the main blocks of a complex analytical engine that is implemented in VMware vCenter Operations Manager as a principal foundation of the company's data-driven anomaly detection.**

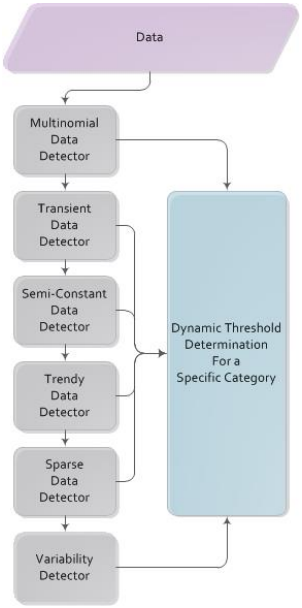
Keywords - monitoring; time series data; dynamic thresholding; data categorization; parametric and non-parametric statistics.

I. INTRODUCTION

Modern enterprise IT management becomes increasingly smart to proactively respond to the performance issues that complex infrastructures necessarily encounter. The management based on expert knowledge utilization is no longer efficient. Monitoring and data measuring of the processes governing the entire IT is a fundamental approach to gain insights from those sophisticated environments with complicated interrelation between the constituent components. The history of this approach goes back to statistical process control [1]. Since contemporary business infrastructures are highly dynamic (and out of classical Gaussian normalcy domain [2]), static thresholding of processes and performance indicators become inadequate. Hence problem diagnostics dictates a soft control of IT environments ([3-14]). In this paper, we follow a path where anomaly detection is based on prediction of upper and lower dynamic thresholds (normalcy range) of categorized data that vary over time [15-19].

VMware's vCenter Operations Manager (vC Ops) [20] is an industry-leading enterprise solution in area of IT management that encodes the bunker of monitoring data from customer IT system into a real knowledge for anomaly detection and problem root cause identification, as well as capacity planning for modern virtualized and cloud computing ecosystems. In this paper we introduce an enterprise data-agnostic dynamic thresholding system (EDTS) that enables vC Ops to act as an atomistic anomaly detection and forecasting of monitoring flows. The data-agnosticism (indicates that data analysis and determination of its normalcy behavior is performed without knowing the essence of the underlying physical and business service processes) enables an universal platform for processing of very large data sets, at the same time, it can lead to a deadlock if the statistical methods are not sufficiently powerful to handle diversity of monitored data types. Our analysis of customer data over several years show that deficiency of data-agnosticism can be compensated by appropriate data categorization, since a specific data category statistically characterizes the underlying process and empowers an efficient construction of relevant normalcy bounds (dominant behavior) thus reliably controlling the flow. This concept leads to an EDTS based on data categorization realized in vC Ops. A simplified and specific realization of EDTS adjusted for IT environments is presented in Flowchart 1. Although selection of the categories is adapted to some IT customer preferences, the overall approach is applicable widely (also out of the IT interests) with appropriate modification of categories and their definition parameters.

Experimental results justify EDTS's potential to effectively handle large infrastructures in terms of both accuracy and complexity. All ideas described in the sequel are filed as a patent [21].



Flowchart 1. A simplified principal scheme of EDTS.

EDTS sequentially utilizes different data categorization detectors that allow choosing the right algorithm for determination of data dynamic thresholds (DT's). The categorization order or the hierarchy is important as different orders of iterative checking and identification will lead to different categories with differently specified normalcy states. The system presented in Flowchart 1 categorizes data as Multinomial, Transient, Semi-constant, Trendy, Sparse, High-Variability or Low-Variability. In each of those cases the normalcy determination method is different. In all categorization scenarios the data additionally is verified against *periodicity* for efficient construction of its normalcy bounds.

Moreover, in each of the majority of categorization scenarios data is processed by category-specific *change detection* procedures. The functional meanings of the above mentioned detectors are as follows:

Multinomial Data Detector searches for Multinomial data which takes only integer values after checking errors introduced by the monitoring apparatus. If data is identified as multinomial then DT determination module calculates specific for this category DT's otherwise data transmits to the next detector.

Transient Data Detector looks for Transient Data which can be characterized as multimodal data. If data is identified as transient then DT calculation module performs DT calculation separately for each mode.

Semi-Constant Data Detector checks the data against its "almost constant" behavior. If data is not semi-constant but its latest portion satisfies the category specifications after a global change, then DT construction module performs DT calculation for the latter. Piecewise constant data is from this category.

Trendy Data Detector performs trend identification. If data is trendy then detector classifies trend as linear or non-linear and DT determination module executes a special DT calculation algorithm. If data is not trendy but its latest portion can be selected as trendy (change occurred), then DT determination module performs calculations for that portion.

Sparse Data Category Detector explores data gaps, their amount, and distribution in time. Data goes to the next detector for further analysis if overall gap duration is negligible. Data is classified as Sparse if gaps have uniform distribution in time. If gaps have some accumulation and remaining data is acceptable for further analysis then the selected portion goes to the next detector.

Variability Detector categorizes data either High-Variability or Low-Variability with specific DT calculation procedures. Before final categorization data passes through a change detection procedure for selection of the latest statistically stable portion for final DT determination.

II. Data Categorization

Multinomial Data Detector. This detector calculates some statistical parameters for comparison with the predefined measures. If the check is positive then data is classified as *Multinomial Data*. It is assumed that Multinomial data takes only integer values. Let p_j be the frequency of occurrences of the integer n_j

$$p_j = \frac{n_j}{N}100, \qquad j = 1, \dots, m$$

where N is the total number of integer values and m is the number of different integer values. Data is multinomial if it takes less than m different integer values and at least s of them have frequencies greater than parameter H_1.

Some integer values with small cumulative percentages can be discarded. This can be done by sorting the percentages p_j in descending order and by defining the cumulative sum c_j

$$c_1 = 100,\ c_j = p_j + \cdots + p_m,\ c_m = p_m.$$

Then, if $c_k < H_2(= 0.5\%)$, $c_{k-1} \geq H_2$ the integer values $n_k, n_{k+1}, \dots, n_m$ can be discarded.

Transient Data Detector. *Transient Data* is categorized by multimodality, modal inertia, and randomness of modes appearing along the time axis. *Transient Data* must have at least two modes. Modal inertia means that data points in each mode must have some inertia and they can't oscillate from one mode to the other "quickly". Actually the inertia can be associated with the time duration that data points remain in the selected mode. Categorization is performed by calculation of some transition probabilities. We omit the relevant details from [21]. A similar technique is applied in *Sparse Data Detector* (see below).

Figure 1 shows an example of a *Transient Data*.

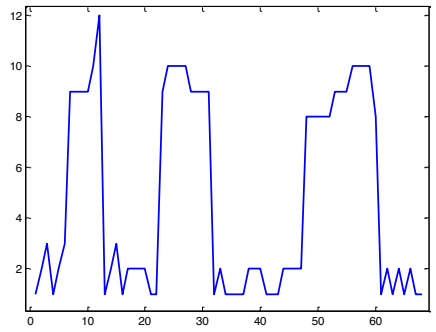


Figure 1. Example of a Transient Data.

Semi-Constant Data Detector. Data is categorized as *Semi-Constant* if

$$iqr(data) = 0$$

where iqr stands for the interquartile range of data. If data is not from the required category but the latest enough long portion satisfies the condition then it is selected for further dynamic threshold determination as *Semi-Constant Data*.

Figure 2 shows an example of *Semi-Constant Data*.

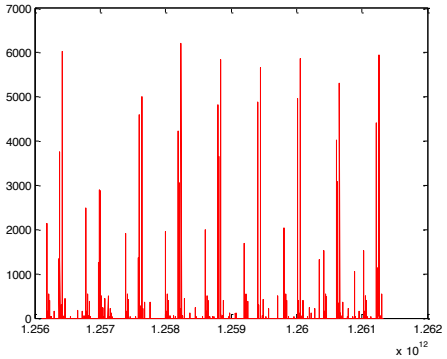


Figure 2. Example of Semi-Constant Data.

Trendy Data Detector. Different classical methods are known for trend determination. Mann-Kendall [22,23] test is appropriate for our purposes although other known tests are also possible to apply. The test categorizes data either *Trendy* or *Non-Trendy*. In case of *Trendy Data* further analysis categorizes the trend into linear and non-linear. Linearity can be checked by the well-known linear regression. If data is *Linear-Trendy* then DT determination module performs a specific DT calculation. If data is not *Linear-Trendy* but the

latest rather long portion of data has linear trend, then we select it for further DT determination.

Sparse Data Detector explores data in terms of gaps. If the total percentage of gaps is higher than some limit and they have non-uniform distribution in time (it means that gaps have some localization in time) then gap clean up (data selection) procedure will return a regular data for further categorization. If gaps in data have uniform distribution in time then data belongs to a *Sparse Data* category. If gaps in data have extremely high percentage that further analysis is impossible. Data categorization is based on the following measures: 1) percentage of gaps, 2) transition probabilities for gap-to-gap, data-to-data, gap-to-data and data-to-gap. Probability calculation is starting with data monitoring time (Δt) estimation $\Delta t = median(\Delta t_k)$, $\Delta t_k = t_{k+1} - t_k$. Time intervals with $\Delta t_k \le c\,\Delta t$ are normal data intervals while $\Delta t_k > c\,\Delta t$ are gaps. c is a parameter for gap definition. Let T_k be duration (in milliseconds, seconds, minutes, etc., but in the same measures as the monitoring time) of the k-th gapless data portion. For data without gaps we have only one such portion and $T_k = t_N - t_1$. The sum $T = \sum_{k=1}^{N_T} T_k$ is the duration of gapless data where N_T is the count of gapless data portions. Let G_k be duration (in the same measure as T_k) of the k-th gap. The sum $G = \sum_{k=1}^{N_G} G_k$ is the duration of all gaps in data and N_G is the count of gap portions. Obviously $G + T = t_N - t_1$. By ρ we define the percentage of gaps in data

$$\rho = \frac{G}{G+T}100\%.$$

Now, by $p_{11}, p_{10}, p_{00}, p_{01}$ we define the probabilities of data-to-data, data-to-gap, gap-to-gap and gap-to-data transitions, respectively

$$p_{11} = 1 - \frac{N_T}{\frac{T}{\Delta t}},\ p_{10} = 1 - p_{11},$$

$$p_{00} = 1 - \frac{N_G}{G/\Delta t},\ p_{01} = 1 - p_{00}.$$

Data with gaps non-uniformly distributed in time can be specified by the condition

$$\begin{cases} \rho > H_1 \\ \mathrm{p}_{10} < \varepsilon \\ \mathrm{p}_{01} < \varepsilon \end{cases}$$

where the following values of parameters can be reasonably chosen $H_1 = 25\%$ and $\varepsilon = 0.0005$. The main reason for smallness of p_{10} and p_{01} is the smallness of the numbers N_T and N_G while G and T are as big as ρ is assumed. Data from this category can be further processed via data selection procedure that will eliminate (if possible) concentration of gaps. This can be done as follows: calculate the total percentage of gaps in the series of data $\{x_k\}_{k=i}^{j}$, $j = j_1, j_2, \dots, j_s$, $i = i_1, i_2, \dots, i_r$, and select the portion for which $\rho \le H_1$. The selected data is ready for further analysis by sequential detectors.

Data with gaps uniformly distributed in time (*Sparse Data*) can be specified by the condition ($H_2 = 60\%$)

$$\begin{cases} \rho > H_2 \ge H_1 \\ \mathrm{p}_{10} \ge \varepsilon \\ \mathrm{p}_{01} \ge \varepsilon \end{cases}$$

The second and third conditions mean that gaps are uniform in time and technical cleanup is impossible.

Data is useless for further analysis if $\rho > H_3$ ($= 95\%$).

Figure 3 shows an example of *Sparse Data* with the corresponding measures for categorization.

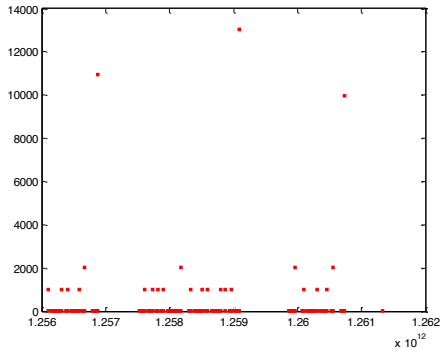


Figure 3. Sparse Data. Here $\rho = 68\%$, $p_{11} = 0.9957$, $p_{10} = 0.0043$, $p_{00} = 0.9979$, $p_{01} = 0.0020$.

Variability Detector calculates variability indicators and categorize data into High-Variability or Low-Variability. Based on the absolute jumps x'_k of data points

$$x'_k = |x_{k+1} - x_k|$$

the following measure R of variability is considered

$$R = \frac{iqr(\{x'_k\}_{k=1}^{N-1})}{iqr(\{x_k\}_{k=1}^{N})}100\%, \quad iqr(\{x_k\}_{k=1}^{N}) \neq 0.$$

Then, if $R \le V$ then data is Low-Variability, otherwise High-Variability. Figure 4 shows an example of Low-Variability data with $R = 0$.

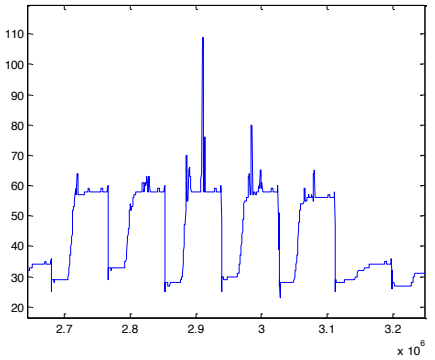


Figure 4. Low-Variability data with $R = 0\%$.

III. Category-Specific DT Determination

As we mentioned above, each data category preliminary passes through some period determination procedure which additionally categorizes data into *Periodic* and *Non-Periodic*. Details of this procedure are presented in [21]. We describe only a high level concept. The period determination is seeking similar patterns in the historical behavior of time series for setting the DT's based on the discovered cyclical information. The algorithm consists of two main steps:

1) Data *Footprint* calculation which provides with two-dimensional distribution of time series based on some predefined frame. First, we calculate percentages of data in each cell of the frame and then we get the corresponding

distribution by taking cumulative sums of those percentages for each column of the frame. More specifically, the range of data (or the range of preliminary smoothed data) is divided into non-uniform parts by quantiles q_k with $k = k_1, \dots, k_m$, $0 \le k_1 < \cdots < k_m \le 1$, with some m and k_j. Evidently, the grid lines are dense where data is dense. For division of data into parts along the time axis two parameters "$time\_unit$" and "$time\_unit\_parts$" are used. "$Time\_unit$" is a basic parameter that defines the minimal length of possible cycle that can be found. Moreover, any cycle can be a factor only of the length of the "$time\_unit$". Usual setting is $time\_unit = 1\,day$. Parameter "$time\_unit\_parts$" shows the number of subintervals (columns in the frame) that "$time\_unit$" must be divided. Actually this parameter is the measure of resolution. The bigger the value of "$time\_unit\_part$", the more sensitive is the footprint of the historical data.

2) *Pattern recognition* procedure which provides with *Cyclochart* of data by comparing different columns of the Footprint in terms of similarity. Figure 5 shows an example of a *Cyclochart* where y-axis shows the measure of confidence that data has some T-days cycle.

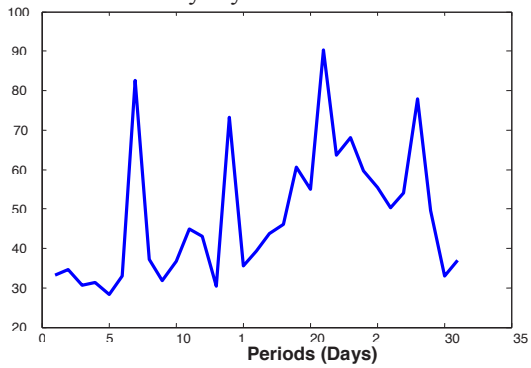


Figure 5. Example of a Cyclochart.

Further investigation of the *Cyclochart* categorizes data into *Periodic* or *Non-Periodic*. If data classifies as *Periodic* then the method provides with information on cycle length and outputs the frame columns in terms of similarity that are finally employed to quantify the time-based DT values. Figure 6 shows an example of a *Periodic Data* (blue curve) with the corresponding upper (red curve) and lower (green curve) DT's.

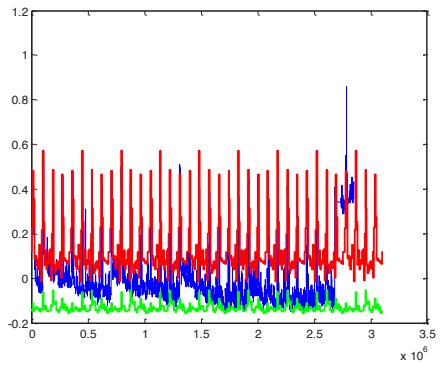


Figure 6. DT's of a Periodic Data.

Now we describe several category-specific DT construction mechanisms:

DT's of Multinomial Data. As mentioned, period determination investigates the cyclicality of data and classifies it into periodic and non-periodic categories. The general scheme for period determination in this case is specialized with the following modification while constructing the *Footprint* of data: instead of the percentages of data in every cell we are taking the values of c_k (see categorization of the *Multinomial Data*) in every column of the frame. If data is claimed *Periodic* then the normalcy set for similar columns are calculated as follows. Data points in similar columns are collected together and corresponding new values of the numbers c_k are calculated. If $c_{k+1} < H$, $c_k \ge H$ then the values $n_1, n_2, \dots, n_k$ constitute the most probable set (normalcy set) of similar columns. If *Multinomial Data* is determined as Non-Periodic then the numbers c_k are calculated for all data points and normalcy set is determined similarly.

DT's of Semi-Constant Data. For Semi-constant data every data point greater than $q_{0.75}$ (quantile) or less than $q_{0.25}$ is an outlier. If the percentage of outliers is greater than $p\%$ ($p = 15\%$), then we check for periodicity in outlier data by the procedure described above. For periodicity analysis data points equal to the median are excluded from the analysis.

DT calculation for *Non-Periodic Semi-Constant Data* is performed separately for upper (for data points that are greater or equal to median) and lower (for data points that are less than or equal to median) parts of data. Since the process of obtaining of both upper and lower bounds are similar, we'll explain the method only for the upper DT. The main principle is maximization of an objective function

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

where $a > 0$ is a sensitivity parameter, for example $a = 0.9$; P is the percentage of data points within the median of data and any upper line higher than the median (see Figure 7);

$S_{max} = (t_{max} - t_{min})(Upper\ Line - Data\ Median)$

and S is the square of the area within data points and data median.

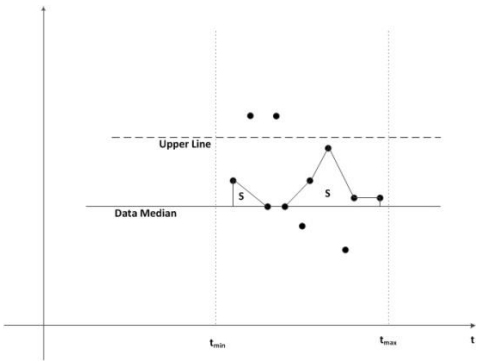


Figure 7. Auxiliary drawing for the objective function.

We consider two different approaches for determination of DT's via maximization of the objective function: data range and data variability based. In the data-range-based analysis, we divide the range within median and maximum of data

(while determining the upper bound) into m parts and for each level calculate the values $g_k, k = 1,2, \dots m$ of the objective function. Then, the level that corresponds to $\max(g_k)$ gives the appropriate upper bound. Instead of dividing the range into equal parts it is reasonable also to divide the range of data by corresponding quantiles that will give unequal division according to the density of data points along the range. Here preliminary abnormality cleaning of data can be performed. For this, removal of data points with abnormal concentration in the given time window is performed. Abnormal concentration can be detected by the following procedure. For the given time window (for example 10% of data length), we calculate the percentage of data points with values higher than 0.75-quantile. Then by moving this window along data, we calculate the corresponding percentages. Any percentage higher than the upper whisker indicates abnormal concentration and must be discarded from further calculation. We repeat the same abnormality cleaning procedure for data points lower than 0.25-quantile.

In the data-variability-based approach, we calculate the variability of data points x_k against median of data μ

$$v = \left(\frac{1}{N-1} \sum_{k=1}^{N} (x_k - \mu)^2 \right)^{1/2}$$

and consider the following set of upper lines

$$[\mu + z_j \, v], \quad j = 1,2, \dots$$

For each level, we calculate the corresponding values g_j of the objective function as described above and, we take the level that corresponds to $\max(g_j)$ as the appropriate upper DT. The following values can be used for z_j

$$z_1 = 1, z_2 = 1.5, z_3 = 2, z_4 = 3, z_5 = 4.$$

In case of periodic data the same procedure is applicable for each periodic column of the *Footprint* of data.

DT's of Transient Data can be obtained by similar procedure for each mode separately based on maximization of the objective function as we do it for $f(t)$ below.

DT's of Linear-Trendy Data. In case of *Linear-Trendy Data,* we perform decomposition of the original data $f_0(t)$ into the following form

$$f_0(t) = f(t) + kt + b$$

and perform DT calculation for $f(t)$based on the following objective function

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

where S is the square of the area limited by t_{min}, t_{max} and some lower and upper lines (see Figure 8),

$$S_{max} = h(t_{max} - t_{min})$$

and P is the fraction of data within upper and lower lines and a is a user defined parameter. Then we calculate standard deviation $\sigma$of $f(t)$ and consider the following set of lower and upper lines

$$[kt + b - z_j \, \sigma, kt + b + z_j \, \sigma], \quad j = 1,2, \dots$$

Next we calculate g_j and take the level corresponding to $\max(g_j)$. We use the following values for z_j

$$z_1 = 1, z_2 = 1.5, z_3 = 2, z_4 = 3, z_5 = 4.$$

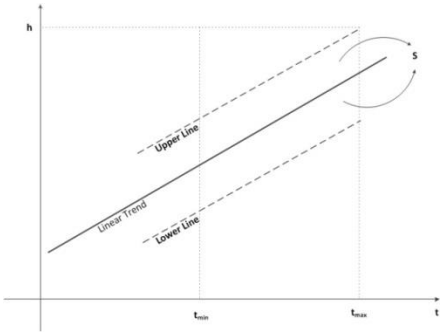


Figure 8. Auxiliary drawing for definition of the objective function.

Figures 9 shows an example of *Linear-Trendy Data* with the corresponding DT's.

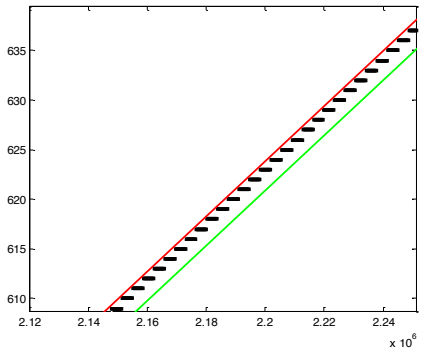


Figure 9. *Linear-Trendy Data* with the corresponding DT's**.**

DT's of Sparse Data. For period determination procedure, we put "*time_unit_parts*" $= \left[\frac{\text{"time\_unit"}}{median(T_k)+median(G_k)} \right]$. If data is classified as *Periodic* then DT calculation is performed according to the found cycles otherwise DT's can be determined based on the utilization of the objective function.

DT's of High- and Low-Variability Data. First data is checked for periodicity by setting different preliminary parameters while calculating the *Footprint* of data – less sensitive for High-Variability data, then DT determination is performed based on cycles or objective function utilization.

Figure 10 shows an example of *Low-Variability Data* with the corresponding normalcy bounds.

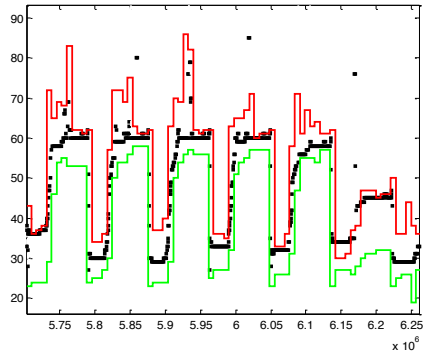


Figure 10. Data from Figure 4 with upper and lower DT's.

IV. System Validation

The results obtained for a specific customer data by EDTS are presented below. Note that Figures 6, 9, and 10 are also real enterprise examples. We selected some 3215 monitoring metrics with one month length and applied the categorization procedures. Table 1 shows the distribution along different data categories and Table 2 indicates the count of periodic and non-periodic data. In particular, for semi-constant data we observe the following picture. From 532 metrics (see Table 1) 267 have percentage of outliers less than 15% and they are claimed as non-periodic without any further checking. The remaining 235 metrics are checked for periodicity and in 212 cases periods are found. It is worth noting that results obtained for the specific customer data can't be in any manner generalized to other cases. The graphs below demonstrate several snapshots from the product with monitored time series data and their adaptively updated DT's by EDTS. We observe reliably detected DT's, data changes, periods, and relevant out-of-normal areas (yellow) reported as alarms.

Table 1. Distribution along the categories.

Data Category	Count (Percentage) of Metrics in the Category
Multinomial	724 (22.5%)
Trendy	165 (5.1%)
Semi-Constant	532 (16.5%)
Transient	102 (3.2%)
Sparse	88 (2.7%)
Low-Variability	826 (25.7%)
High-variability	669 (20.8%)
Corrupted	109 (3.4%)

Table 2. Count of periodic and non-period data.

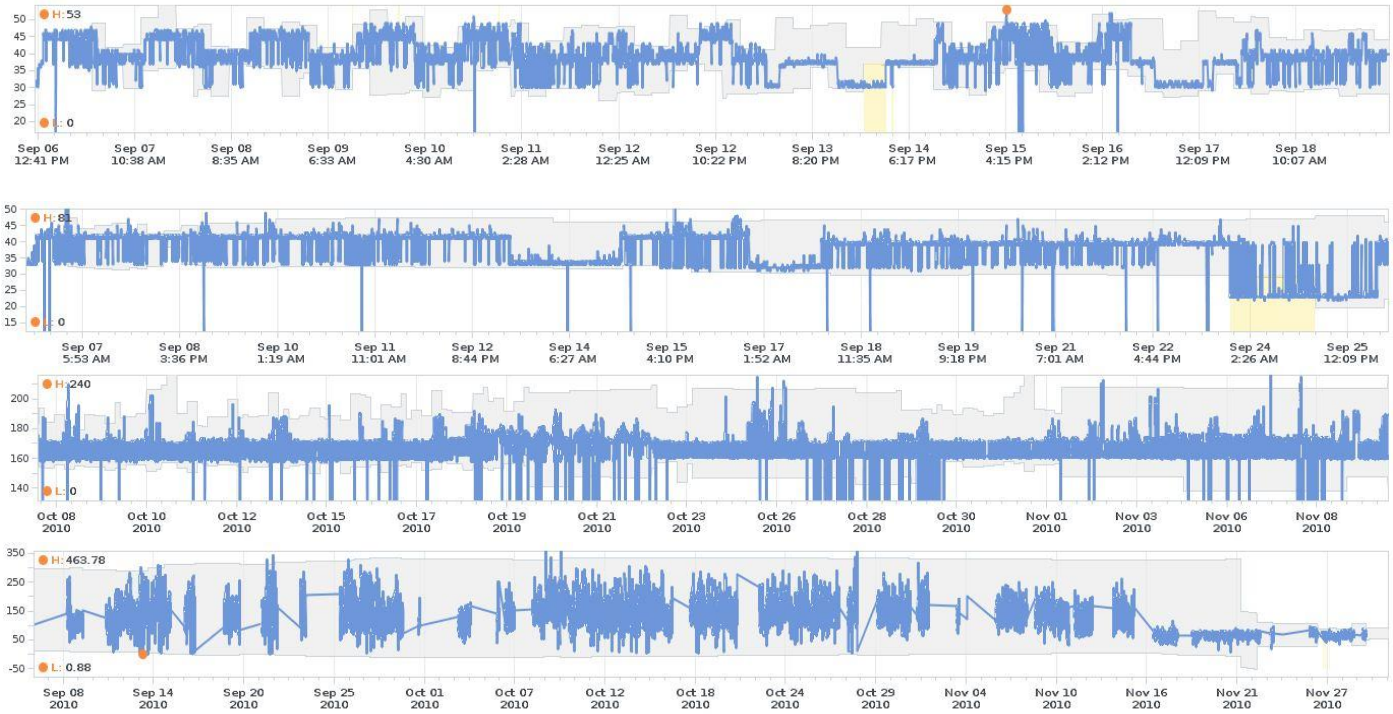
Periodic	Non-Periodic	Corrupted	Overall
1511	1595	109	3512

V. Related Work

In terms of our application, the performance of EDTS is estimated according to users experience on indicative and missed alarms, as well as the generated noise level that the useful information is embedded in. In this context, our categorization techniques allow achieving essentially better trade-off between the produced recommendation (alarm) noise and its accuracy in problem indication. That would not be possible with classical parametric approaches including Fourier transform, discrete Fourier transform [24-28], Prony's method [29,30]) as well as with other common purpose enterprise algorithms (including our algorithm of Section III that produces DT's based on data footprint even when cyclical patterns are not discovered). Moreover, the categorization in terms of those specific classes enables an efficient root cause analysis [31,32] based on the abnormality events (DT violation alarms) space that our system outputs. Furthermore, [19] reports about reliably predicted root causes of suddenly occurring influential outages at large enterprise infrastructures. This method relies on historically analyzed mutual impact factors of out-of-DT events.

Note that EDTS handles only structured monitoring data. For the unstructured data sets (like log files) we have developed a graph-based approach [33,34] that extracts the dominating correlation pattern between the main event types in data as dynamic normalcy structure and applies it to identification of "large"/abnormal deviations from that structure to determine performance anomalies.

Finally, we refer the reader to the papers [35,36] which outline the approaches and trends of the area of anomaly detection up to the recent days.



REFERENCES

[1] D.J. Wheeler, and D.S. Chambers, *Understanding Statistical Process Control*, Knoxville, TN: SPC Press, 1986.

[2] M.A. Marvasti, "How normal is your data?," *VMware technical white paper*, http://www.vmware.com/files/pdf/vcenter/-VMware-vCenter-Operations-How-Normal-Is-Your-Data-WP-EN.pdf, 2011.

[3] J.P. Buzen and A.W. Shum, "MASF: multivariate adaptive statistical filtering," in *Int. Computer Measurement Group (CMG) Conf.*, Nashville, TN, USA, Dec. 4-8, pp. 1-10, 1995.

[4] H. Kang, H. Chen, and G. Jiang, "PeerWatch: a fault detection and diagnosis tool for virtualized consolidation systems," in *ACM Int. Conf. on Automatic Computing (ICAC)*, Washington, DC, USA, June 7-11, pp. 119-128, 2010.

[5] C. Wang, V. Talwar, K. Schwan, and P. Ranganathan, "Online detection of utility cloud anomalies using metric distributions", in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Osaka, Japan, April 19-23, pp. 96-103, 2010.

[6] C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterelid, and K. Schwan, "Statistical techniques for online anomaly detection in data centers," in *IFIP/IEEE Int. Symp. Integrated Network Management (IM)*, Dublin, Ireland, May 23-27, pp. 385-392, 2011.

[7] M. Jiang, M. A. Munawar, T. Reidemeister, and P. A. Ward, "Automatic fault detection and diagnosis in complex software systems by information-theoretic monitoring," in *IEEE Conf. Dependable Systems and Networks (DSN)*, Lisbon, Portugal, June 29 – July 2, pp. 285-294, 2009.

[8] K. Ozonat, "An information-theoretic approach to detecting performance anomalies and changes for large-scale distributed web services," in *IEEE Conf. Dependable Systems and Networks (DSN)*, Anchorage, Alaska, June 24-27, pp. 522-531, 2008.

[9] K. Viswanathan, L. Choudur, V. Talwar, C. Wang, G. MacDonald, and W. Satterelid, "Ranking anomalies in data centers," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Maui, HI, April 16-20, pp. 79-87, 2012.

[10] G. Jiang, H. Chen, K. Yoshihira, and A. Saxena, "Ranking the importance of alerts for problem determination in large computer systems," in *ACM Int. Conf. Automatic Computing (ICAC)*, Barcelona, Spain, June 15-19, pp. 3-12, 2009.

[11] Y. Tan and X. Gu, "On predictability of system anomalies in real world," in *IEEE Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Miami Beach, FL, Aug. 17-19, pp. 133-140, 2010.

[12] X. Gu and H. Wang, "Online anomaly prediction for robust cluster systems," in *IEEE Int. Conf. Data Engineering (ICDE)*, Shanghai, March 29-April 2, pp. 1000-1011, 2009.

[13] Y. Tan, X. Gu, and H. Wang, "Adaptive system anomaly prediction for large-scale hosting infrastructures," in *ACM SIGACT-SIGOPS Symp. Principles of Distributed Computing (PODC)*, Zurich, Switzerland, July 25-28, pp. 173-182, 2010.

[14] L. Cherkasova, K. M. Ozonat, N. Mi, J. Symons, and E. Smirni, "Anomaly? Application change? or workload change? Towards automated detection of application performance anomaly and change," in *IEEE Conf. Dependable Systems and Networks (DSN)*, Anchorage, AK, June 24-27, pp. 452-461, 2008.

[15] D. Dang, A. Lefaive, J. Scarpelli, and S. Sodem, "Automatic determination of dynamic threshold for accurate detection of abnormalities," US Patent 20110238376, Published 2011.

[16] J. C. Jubin, V. Rajasimman, N. Thadasina, "Hard handoff dynamic threshold determination", US Patent 20100056149, published 2010.

[17] Ying-Ru Chen, "Method of motion detection using adaptive threshold," US Patent 8077926 B2, 2011.

[18] H.M. Sun, S.P. Shieh, "A construction of dynamic threshold schemes," *Electronic Letters*, pp 2023-2025, NSC 84-2213-E-009-081, Nov. 1994.

[19] H.M. Sun, S.P. Shieh, "On dynamic threshold schemes," *Information Processing Letters* 52, pp 201-206, NSC 84-2213-E-009-081, 1994.

[20] VMware vCenter Operations Manager. http://www.vmware.com/products/vcenter-operations-manager.

[21] A.V. Poghosyan, A.N. Harutyunyan, N.M. Grigoryan, and M.A. Marvasti, "Data-agnostic anomaly detection," applied US patent 13/853,321, Filed March 29, 2013.

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

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

[24] R.N. Bracewell, *The Fourier transform and its applications* (3rd ed.), Boston: McGraw-Hill, ISBN 0-07-116043-4, 2000.

[25] B. Boashash, ed., *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*, Oxford: Elsevier Science, ISBN 0-08-044335-4, 2003.

[26] A.V. Oppenheim, R.W. Schafer, and J.R. Buck, *Discrete-Time Signal Processing*, Upper Saddle River, N.J., Prentice Hall, ISBN 0-13-754920-2, 1999.

[27] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, NJ, 2005.

[28] S.M.G. Kendall, *Time Series*, Second Edition, Charles Griffin & Co., ISBN 0-85264-241-5, 1976.

[29] D.W. Tufts and R. Kumaresan, "Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood," *Proc. IEEE*, vol. 70, pp. 975-989, 1982.

[30] R. de Prony, "Essai Experimentale et Analytique," *J. Ecole Polytechnique (Paris)*, pp. 24-76, 1795.

[31] 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", *VMware Technical Journal*, vol. 2, issue 1, pp. 35-45, 2013.

[32] M.A. Marvasti, A.V. Poghosyan, A.N. Harutyunyan, and N.M. Grigoryan, "Method and apparatus for root cause and crytical pattern prediction using virtual directed graphs", US Patent 20130097463, published 2013.

[33] A.N. Harutyunyan, A.V. Poghosyan, N.M. Grigoryan, and M.A. Marvasti, "Abnormality analysis of streamed log data", in *IEEE/IFIP Netwotk Operations and Management Symposium (NOMS)*, 5-9 May, Krakow, Poland, 2014.

[34] M.A. Marvasti, A.V. Poghosyan, A.N. Harutyunyan, and N.M. Grigoryan, "Methods and systems for for abnormality analysis of streamed log data", US Patent 20140053025, published 2014.

[35] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, pp. 1-72, Sept. 2009.

[36] C. Wang, S.P. Kavulya, J. Tan, L. Hu, M. Kutare, M. Kasick, K. Schwan, P. Narasimhan, and R. Gandhi, "Performance troubleshooting in data centers: an annotated bibliography," *ACM SIGOPS Operating Systems Review*, vol. 47, issue 3, pp. 50-62, 2013.