Statistical Normalcy Determination Based on Data Categorization

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 introduce a statistical learning Normalcy Determination System (NDS) for data-agnostic management of monitoring flows. NDS performs data categorization with analysis tools that identify category-specific normalcy bounds in terms of dynamic thresholds. This information further can be applied for anomaly detection, prediction, capacity planning and root-cause analysis.**

Keywords - monitoring; time series data; statistical process control, normalcy determination, dynamic thresholding, data categorization; parametric and non-parametric statistics.

I. INTRODUCTION

Today's business and IT management face the problem of "big infrastructures" with millions of monitored metrics that need to be efficiently analyzed for gaining valuable insights in terms of underlying system control. The concept of control for ensuring the quality of services of different systems is originating from the ideas of the statistical process control charts which provide a comprehensive tool to determine whether the process is in normalcy state or not. The foundation of these concepts was established by Shewart ([1]) where he developed methods to improve quality and lower costs. Fluctuations and deviations from standards are present everywhere and the problem of constructing a relevant chart is in understanding which variations are normal and which are caused by a problem.

In the classical theory of control the underlying processes have bell-shaped distribution and in those cases control charts are based on strong foundation of parametric statistics. The problems arise when the classical theory of control is applied to processes of other types.

The problem of construction of a relevant control tool is in identification of normalcy bounds of the processes. Since developments of Shewart an explosion in controlling techniques has occurred ([2-4]). Different processes require different measures to be controlled and each one leads to a new control chart with the corresponding normalcy states defined by thresholds ([5-10]).

Modern businesses and infrastructures are dynamic and as a consequence measured metrics are dynamic without any ad-hoc known behavior. These cause extension of the classical ideas to encompass the notion of dynamic normalcy behavior. In some applications a notion of normalcy bound in terms of dynamic threshold (DT) arises naturally ([11-16]). Essentially different approach is determination of normalcy state in terms of correlated events which leads to a directed graph revealing the fundamental structure of a system beyond the sources and processes ([17, 18]).

In this paper, we introduce a fully data-agnostic system ([19, 20]) for determining normalcy bounds in terms of DTs of monitoring time series data without presumed behavior. The system performs data categorization based on some parametric and non-parametric models and applies category-specific procedures for optimized normalcy determination via historical simulation. Although experimental results are obtained based on IT data, the approach is applicable to wider domains since for different applications the data categories can be appropriately defined.

Determined DTs can be further applied for anomaly detection by construction of anomaly events. As soon as the DT's are historically constructed they can be projected into the future as prediction for time-based normalcy ranges. Any data point appearing above or below those thresholds is an abnormality event. An approach described in ([21-24]) employs a directed virtual graph showing relationships between event pairs. An information-theoretic processing of this graph enables reliable prediction of root causes of problems, bottlenecks and black swan events in IT systems.

The NDS described here is realized in VMware's vC Ops ([25]) analytics and the last section presents some results for real customer data.

II. GENERAL DESCRIPTION OF THE SYSTEM

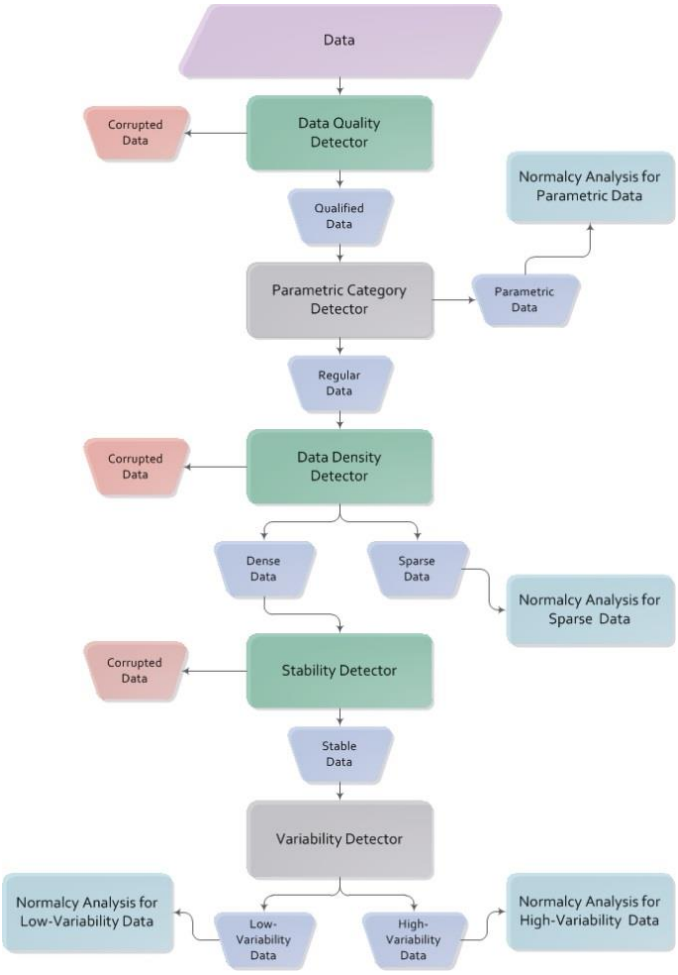
In this section, we present general principles of the NDS which performs fully data-agnostic normalcy determination based on historical simulation. Flowchart 1 illustrates the general concept. The system sequentially utilizes different Data Quality Assurance (DQA) and Data Categorization (DC) routines that allow choosing the right procedure (right category of qualified data) for determination of data normalcy bounds in terms of DTs.

Flowchart 1. General concept of the NDS.

DQA filters data by checking different statistical characteristics defined for data qualification and it further passes through DC. DC performs data identification into categories (e.g. trendy, cyclical, etc.). We repeat this cycle for each time series performing category checking and

identification with hierarchical/priority order until data is identified as belonging to some category or is identified as corrupted. The categorization order or the hierarchy is important as different orders of iterative checking and identification will lead to different final categorization with differently specified normalcy states.

Flowchart 2 shows specific realization of the general concept. Here, NDS consists of three DQA modules – Data Quality Detector, Data Density Detector, Stability Detector, and two DC modules – Parametric Category Detector and Variability Detector.



Flowchart 2. A specific realization of NDS.

As a final result, the initial data is interpreted as Parametric, Sparse, Low-Variability, and High-Variability. In each of those cases the normalcy determination method is different. For instance, Parametric Data can be of different categories (Transient, Multinomial, Semi-constant, and Trendy) with a specific normalcy analysis algorithm in each case.

The functional meanings of the above mentioned detectors are as follows:

Data Quality Detector performs check of sufficient statistics. This block classifies data as Qualified when available data points and length of data are sufficient for further analysis otherwise data is classified as Corrupted.

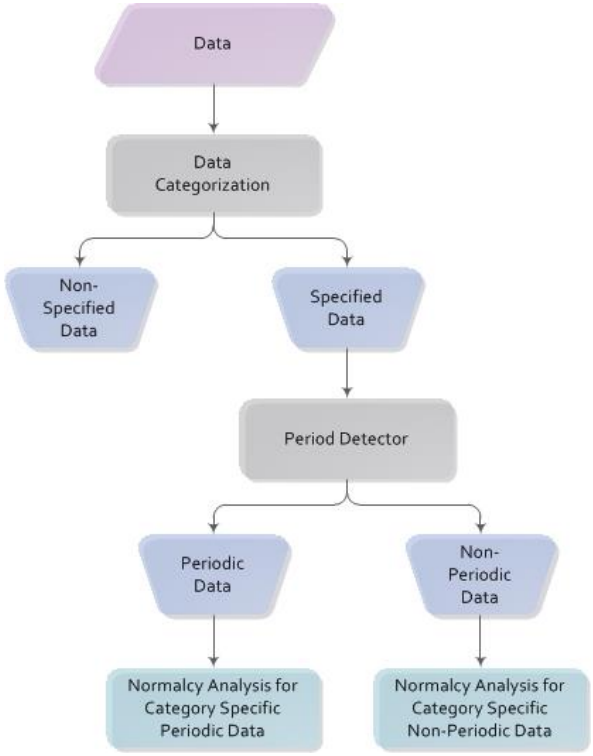
Parametric Category Detector performs data categorization by verifying data against a selected statistical parametric model. If categorization is possible then data is named as Parametric Data otherwise Regular Data.

Data Density Detector filters Regular Data against gaps. Data with extremely high percentage of gaps is Corrupted Data. Data with low percentage of gaps is Dense Data. Data with high percentage of gaps which are uniformly distributed in time is Sparse Data. Data with high percentage of gaps which have localization in time is further processed through a gap filter which output is Dense or Corrupted Data.

Stability Detector analyzes Dense Data in terms of statistical stability. If data is piecewise stable and the latest stable region is enough for further processing then this block performs a data selection, otherwise the data is Corrupted. Stable Data is then passed through Variability Detector.

Variability Detector calculates variability indicators and classifies data into High-Variability or Low-Variability.

In all categorization scenarios the data additionally is verified against periodicity for efficient construction of its normalcy bounds (see Flowchart 3).



Flowchart 3. Categorization in terms of periodicity.

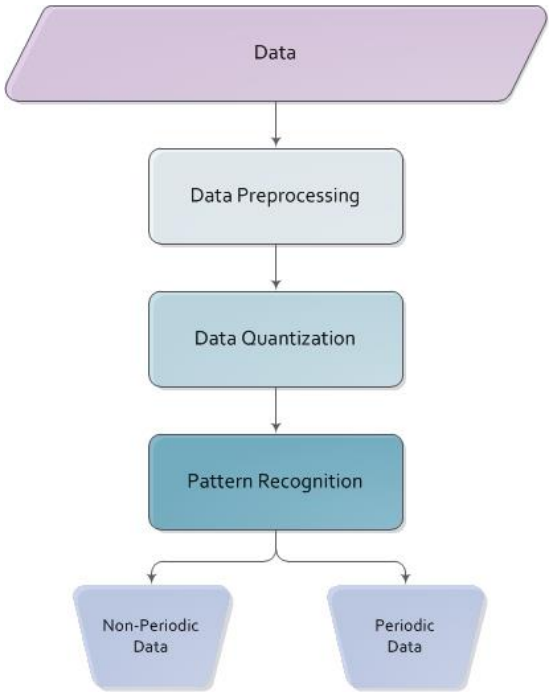
III. Period Detector

Period determination procedure (see Period Detector in Flowchart 3) in NDS is seeking similar patterns in the historical behavior of time series for accurate setting of its normalcy bounds based on the information on cycles.

Some classical techniques known in the literature include seasonality analysis and adjustment [26-30], spectral analysis, Fourier transform, discrete Fourier transform [31-35], data decomposition into cyclical and non-cyclical components and

the Prony method [36,37]. Our procedure is closer to the approach described in [38] that is based on clustering principles.

The main steps of the period determination procedure are presented in Flowchart 4.



Flowchart 4. The main steps of the period determination.

Data preprocessing performs data smoothing and outlier removal by various procedures. We refer to standard classical algorithms [39-42]. The purpose of this step is two-fold: eliminating of extreme high or low outliers that can degrade the range information and smoothing of local fluctuations in data for more robust pattern recognition.

Data Quantization performs construction of the *Footprint* of the historical data for further cyclical patterns recognition. This is a two-step procedure:

1) *Frame construction.*

The range of data (smoothed data) is divided into non-uniform parts by quantiles q_k with $k = k_1, \dots, k_m$, $0 \le k_1 < \cdots < k_m \le 1$, where parameter m and the values of k_j are predefined. Evidently, the grid lines are dense where the 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 "$time\_unit$". Usual setting is $time\_unit = 1\ day$. Parameter "$time\_unit\_parts$" shows the number of subintervals that "$time\_unit$" must be divided. Actually this parameter is the measure of resolution. The bigger the value of "$time\_unit\_part$" then more sensitive is the footprint of the historical data. Figure 1 shows an example of a frame. Gridlines are equidistant along the time axis and non-uniform along the range.

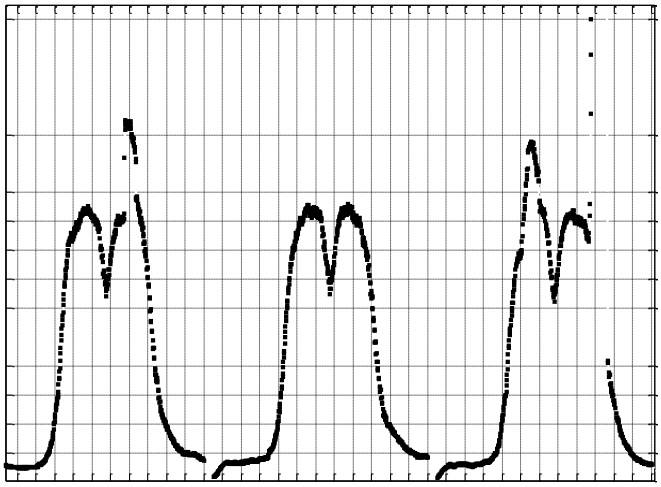


Figure 1. Example of a frame.

2) *Percentage calculation.*

For the given framework, we calculate the percentage of data in every grid-cell and obtain the corresponding column for the given time interval. Collecting all columns, we construct the matrix of percentages for that particular frame. The final matrix is a 2-dimensional classical histogram of historical data. Then for every column we calculate the corresponding cumulative sums getting cumulative distribution of data in each column. We call this matrix as a *Footprint* of historical data.

Pattern Recognition procedure is described as: Let $T = N \times time\_unit$, $N = 1,2, \dots$ We collect the columns of the footprint matrix into subgroups with $L = N \times time\_unit \times time\_unit\_parts$ columns in every subgroup. Overall number of the subgroups equals to $M = length(footprint)/L$ (footprint matrix can be extended by zero columns if needed). For each k-th $k = 1, \dots, L$ column in every subgroup, we check the similarity of columns by the well-known relative L_2-norm:

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, B = \begin{pmatrix} b \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, d(A,B) = \frac{(\sum_{k=1}^{n}(a_k - b_k)^2)^{1/2}}{max(||a||, ||b||)}$$

If

$$d(A,B) \le closeness$$

for some user-defined parameter "closeness" then it is assumed that two columns are similar. Let user defines parameters "$closeness$" $(= 0.2)$ and "$quality$" $(= 75\%)$. In Figure 2 an example is shown where data is divided into T-cycles. For this particular example, we assumed that

$$d(A,E) > closeness, d(A,I) > closeness,$$
$$d(E,M) > closeness.$$

Hence column A is not similar to columns E, I and $M$and we put zero under it. Now, we try column E. We assumed that

$$d(E,I) \le closeness, d(E,M) \le closeness.$$

Hence, column E assumed to be similar to I and M and we put ones under these columns. If the percentage of ones is not less than the value of parameter "quality" then, we declare that the corresponding column of T-cycle is periodic otherwise non-periodic. In our example taking into account that three columns from four compose $75\% \ge$ "$quality$", we declare

that the first column of the T-cycle is periodic and put one in the corresponding column (see Figure 3). We repeat the procedure for all columns (see particular example in Figure 2) and check periodicity of each column (see Figure 3).

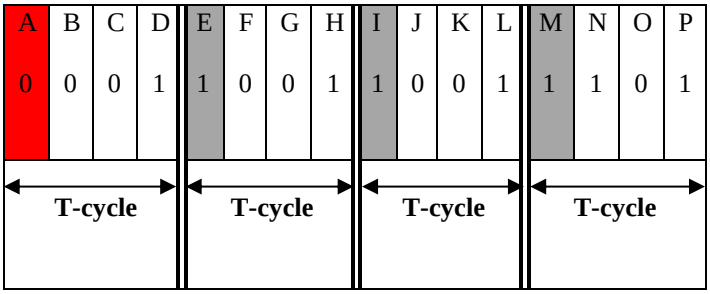


Figure 2. T-cycle checking procedure.

Figure 3 shows that in this particular example, two columns are periodic (A, E, I, M and D, H, L, P) and two columns are non-periodic. As periodic columns compose 50% of all columns then T-cycle has 50% similarity of columns.

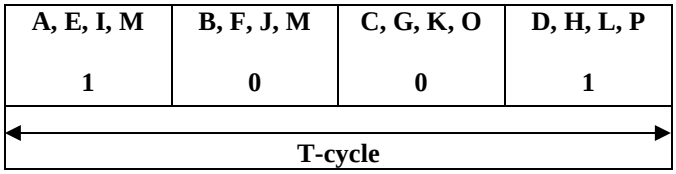


Figure 3. T-cycle similarity calculation.

Repeating this procedure for all possible T-cycles, we compose a *Cyclochart* of data which shows percentage of similarities against T. Figure 4 shows an example of a Cyclochart.

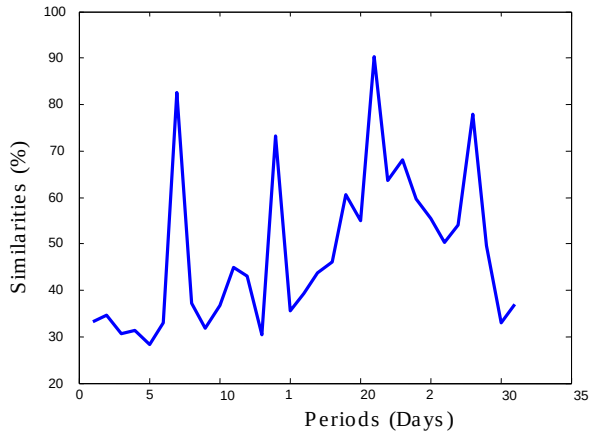


Figure 4. Example of a *Cyclochart.*

Period determination procedure is based on the *Cyclochart* analysis and consists of four steps. Let us consider the particular example of Figure 4.

1) Finding out the local maximums in the *Cyclochart* with their corresponding similarities (see Table 1).

Local maximums	***Similarity***
2	34.7%
4	31.3%
7	82.5%
11	44.9%
14	73.28%
19	60.5%
21	90.3%
23	68.1%
28	78%
31	37%

Table 1. Local maximums (first column) and the corresponding similarities (second column) corresponding to the *Cyclochart* of Figure 4.

2) Construction of the periods that correspond to every local maximum. Data with T-cycle also have kT-cycle for every natural k. Hence for the first row in Table 1 together with the 2-day period, we expect also 4-day, 6-day, … periods. So 2-day period creates the following period series

$$2 \rightarrow 2, 4, 6, 8, 10, \ldots.$$

The peak 4 creates another series

$$4 \rightarrow 4, 8, 12, 16, \ldots$$

and so on.

3) The third step is calculation of the series characteristics for every period series. The following characteristics are assumed to be important: positive factor of the period series is the number of peaks in that series and negative factor of the period series is the number of members in that series which are not the peaks.

Then, we set

$$Strength = Positive\ factor - Negative\ factor.$$

Table 2 shows these characteristics for Table 1.

Local maximum	*Positive factor*	*Negative factor*	*Strength*	***Similarity***
2	**4**	**11**	**-7**	34.7%
4	**2**	**5**	**-3**	31.3%
7	**4**	**0**	**4**	82.5%
11	**1**	**1**	**0**	44.9%
14	**2**	**0**	**2**	73.28%
19	**1**	**0**	**1**	60.5%
21	**1**	**0**	**1**	90.3%
23	**1**	**0**	**1**	68.1%
28	**1**	**0**	**1**	78%
31	**1**	**0**	**1**	37%

Table 2. Positive factors, negative factors, strengths and the corresponding similarities.

4) Period determination based on the defined characteristics by the following procedure. We select the periods with maximum strength. From that list, we choose the periods with minimum negative factor then with maximum similarity. Then, we pick the period with minimum length and check its similarity measure. The similarity of the determined final period must be greater than 20% otherwise data is claimed to be non-periodic. This procedure applied to the results in Table 2 leads to the 7-day period as it has the maximum $Strength = 4$.

We will now discuss the general procedure how the normalcy bounds can be determined based on periodicity taking into account that for specific categories it can be modified appropriately.

In case of non-periodic data normalcy bounds can be determined by the well-known whisker's method. If data is claimed as periodic then normalcy bounds are calculated based on cycle information (see Figure 5 for a specific example).

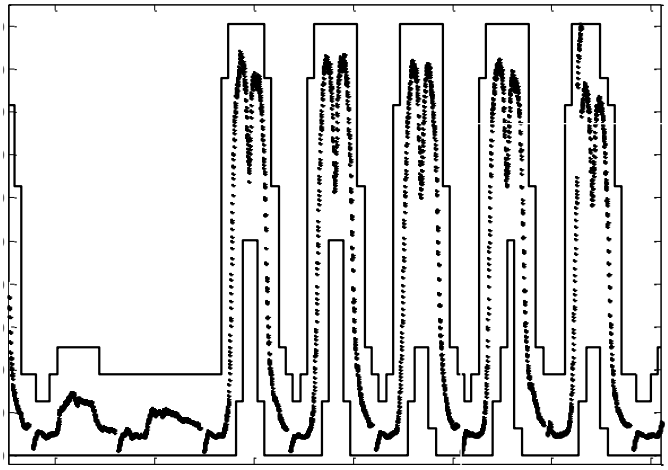


Figure 5. Normalcy bounds for periodic data.

More specifically, consider the case of cyclical data and the following four columns from the *Footprint* which are shifted one from another by the period of data

$$A = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}, \mathrm{B} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}, C = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}, D = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{pmatrix}$$

If

$$d(A,B) \le closeness, d(A,C) \le closeness,$$

and

$$d(A,D) \le closeness,$$

then all columns make a cyclical subgroup and we calculate the bounds based on the four data columns. Then, if

$$d(A,B) \le closeness, d(A,C) \le closeness$$

but

$$d(A,D) > closeness$$

then only the columns A, B, C make a cyclical subgroup. Taking into account that "$quality$" $= 75\%$ and three columns from four compose 75% then we assume that column D is corrupted and consider only the three columns. If

$$d(A,D) \le closeness$$

but

$$d(A,B) > closeness$$

then we discard column A. If less than 75% of these four columns are similar then we have non-cyclical subgroup and take into consideration all columns A, B, C, D. Then, for each group of columns normalcy bounds can be taken as (min, max) values of data (see Figure 6) taking into account preliminary data smoothing factor.

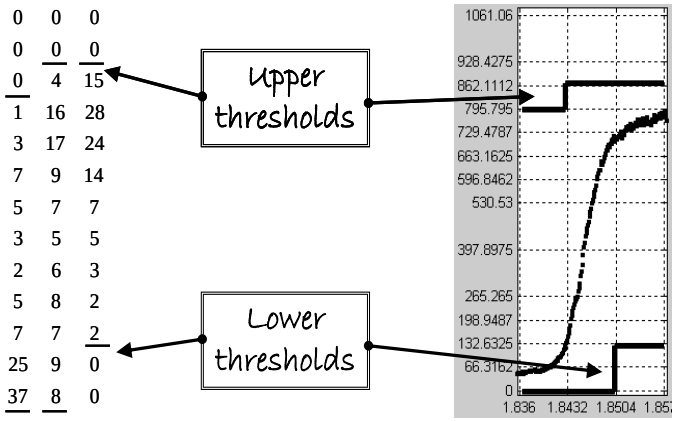
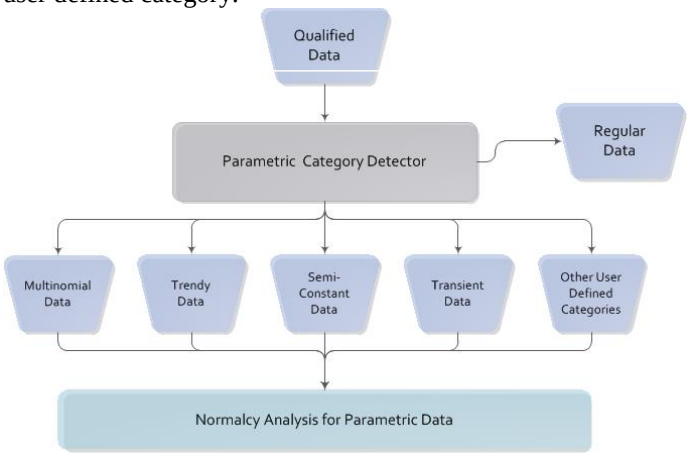


Figure 6. Normalcy bounds construction procedure from *Footprint* taking into account the information on cycles.

IV. Data Quality and Parametric Category Detectors

Data Quality Detector performs check of sufficient statistics. This block classifies data as Qualified when available data points (are greater than 20 points for example) and length of data (is greater than 1 week for example) are enough for further analysis otherwise data is classified as Corrupted.

Flowchart 5 shows the principal scheme of the *Parametric Category Detector*. It specifies data either Parametric or Regular. Parametric Data can belong to different categories: Multinomial, Transient, Semi-Constant, Trendy or any other user defined category.



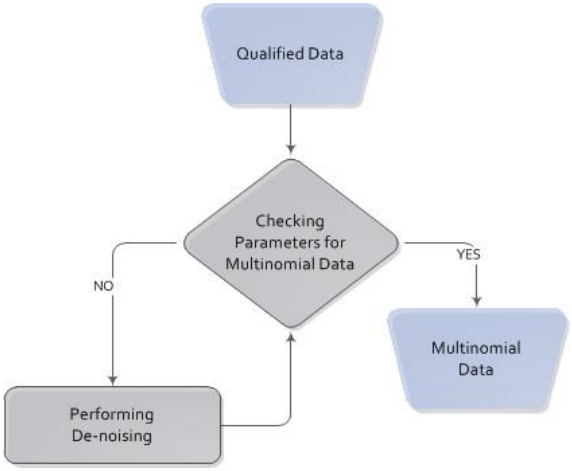
Flowchart 5. Parametric Category Detector.

Multinomial Data. Flowchart 6 describes the process of Multinomial Data (MD) categorization. The module Checking

Parameters for MD calculates statistical parameters for comparison with the predefined measures. If the checking is positive then data is classified as MD otherwise the module Performing De-noising performs data cleaning with sequential checking of predefined parameters. We consider the following predefined statistical measures. 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.



Flowchart 6. Categorization of Multinomial Data.

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.

Two different de-noising procedures can be performed:

The first procedure is filtering against non-integer values with smaller than H_2 percentage ($H_2 < H_1$). If this condition is satisfied then in the remaining analysis the non-integer numbers are discarded.

The second procedure is filtering against integer values with small cumulative percentage. Sorting the percentages p_j in descending order we define the cumulative sum c_j as:

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

Now, if $c_k < H_3$, $c_{k-1} \geq H_3$ then the integer values $n_k, n_{k+1}, \dots, n_m$ can be discarded from further analysis.

Determination of normalcy bounds is started with periodicity analysis. Here, while constructing the *Footprint*, instead of the percentages of data in every cell we are taking the values of c_k in every column. Then, if data is claimed as periodic points in similar columns are collected together and new values of the numbers c_k are calculated. If $c_{k+1} < H$, $c_k \geq H$ then the values $n_1, n_2, \dots, n_k$ constitute the most probable set (normalcy set) of similar columns. If data is determined as non-periodic then the numbers c_k are calculated for all data points and normalcy set is determined similarly.

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. In this context modal inertia means that data points in each mode must have some inertia (they can't oscillate from one mode to the other too quickly). Actually the inertia can be associated with the time duration that data points remain in the selected mode.

Detection of inertial modes is based on calculation of transition probabilities. By the first step we seek for a region/interval of sparse data values and for data with some inertia concentrated in upper and lower regions of this interval. We take two numbers a, b such that

$$x_{min} \leq a < b \leq x_{max},$$

where x_{min}, x_{max} are minimum and maximum values of data, respectively. These numbers divide the interval $[x_{min}, x_{max}]$ into three regions $A \stackrel{\text{def}}{=} [x_{min}, a]$, $B \stackrel{\text{def}}{=} (a, b)$, and $C \stackrel{\text{def}}{=} [b, x_{max}]$. We calculate the following transition probabilities

$$p_{A\to A} = \frac{N_{A\to A}}{N_A}, p_{B\to B} = \frac{N_{B\to B}}{N_B}, p_{C\to C} = \frac{N_{C\to C}}{N_C},$$

where

N_A – is the number of points in $[x_{min}, a)$, N_B – is the number of points in $[a, b]$, N_C – is the number of data points in $(b, x_{max}]$, $N_{A\to A}$ – is the number of points with the property $x(t_i) \in A$ and $x(t_{i+1}) \in A$, $N_{B\to B}$ – is the number of points with the property $x(t_i) \in B$ and $x(t_{i+1}) \in B$, $N_{C\to C}$ – is the number of points with the property $x(t_i) \in C$ and $x(t_{i+1}) \in C$.

Starting from the highest possible position and shifting the region B to the lowest possible we calculate those three transition probabilities and stop the procedure until the following condition is fulfilled

$$p_{A\to A} > H,\ p_{C\to C} > H,\ p_{B\to B} < h, \text{ and } N_A, N_C \gg 1,$$

where the numbers H and h are some predefined parameters. In our experiments below we set $H = 0.75$ and $h = 0.25$. If this procedure ends without finding the needed interval we narrow the region B and repeat the procedure.

In our experiments we divide the interval $[x_{min}, x_{max}]$ into $N + 1$ equal parts

$$x_{min} < x_1 < x_2 < \cdots < x_N < x_{max}$$

and check sequentially the following intervals (a, b):

$$(x_{min}, x_N), (x_1, x_{max}), (x_{min}, x_{N-1}), (x_1, x_N),$$
$$(x_2, x_{max}), \dots, (x_{min}, x_1), (x_1, x_2), \dots, (x_N, x_{max}).$$

When the first needed interval is found then the procedure stops. If it is needed the procedure can be repeated for the lowest (A) region and the highest (C) region for finding new inertial modes if data is supposed to be multi-modal. If the needed interval was not found then actually the data is without inertial modes in terms of the given resolution.

Now suppose that we found M inertial modes and the corresponding regions are

$$A_1 = [a_1, b_1], \dots, A_M = [a_M, b_M].$$

The next step is checking the transiency of each inertial mode. We select one of the found inertial modes, delete all other data points which are outside of this region and by $x(t_k)$ denote data points of this mode. The first step is estimation of the monitoring time by the following formula

$$\Delta t = median(\Delta t_k), \quad \Delta t_k = t_{k+1} - t_k.$$

We suppose that time intervals with $\Delta t_k \leq c\,\Delta t$ are normal data intervals while $\Delta t_k > c\,\Delta t$ are holes (gaps). $\boldsymbol{c}$ is a

predefined parameter for the hole determination. It is assumed that for transient data the holes must be "uniformly" distributed along time axis. This can be checked by transition probabilities.

Let T_k be the 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 holes 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. N_T is the number of gapless data portions. Let G_k be duration (in the same measures as T_k) of the k-th hole. The sum

$$G = \sum_{k=1}^{N_G} G_k$$

is the duration of all holes in data. N_G is the number of hole portions. Obviously, $G + T = t_N - t_1$.

By ρ we define the percentage of holes in data

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

Calculation of Probabilities. By $p_{11}, p_{10}, p_{00}, p_{01}$ define the probabilities of data-to-data, data-to-gap, gap-to-gap and gap-to-data transitions, respectively

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

and

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

We seek for an inertial mode for which

$$\rho > P,\ p_{10} > \varepsilon,\ p_{01} > \varepsilon.$$

Where P and ε are user defined parameters.

If at least two inertial modes satisfy these conditions then data is transient.

Normalcy determination is performed separately for each mode.

Semi-Constant Data. Data is Semi-Constant if

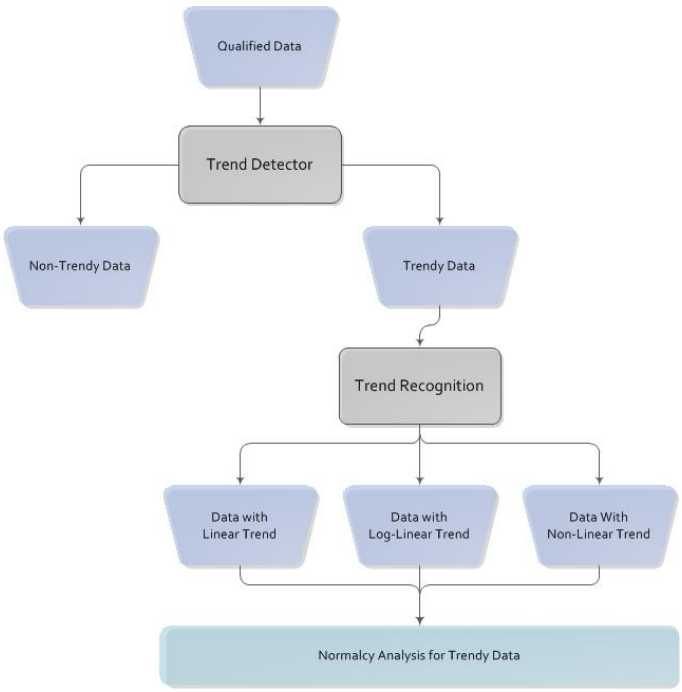
$$iqr(\{x_k\}_{k=1}^{N}) = 0$$

where N corresponds to data length and iqr stands for interquartile range of $x_k = x(t_k)$.

If data is not semi-constant but its latest enough long time period satisfies the condition then, we select it for further normalcy bounds determination.

Normalcy determination of the Semi-Constant Data can be performed as follows. 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 that, data points equal to the median are excluded from the analysis. In case of non-periodic data the normalcy bounds are calculated by whisker's method. In case of periodic data the same procedure is applicable for each periodic column of the *Footprint* of data.

Trendy Data. Different classical methods are known for trend determination [43-48]. In our analysis Trendy Data recognition and related determination of its normalcy bounds consists of three main steps (see Flowchart 7).



Flowchart 7. Trend determination and normalcy analysis.

1) Trend identification by Trend Detector which separates Qualified Data into Trendy and Non-Trendy Data.

2) Trendy Data goes through Trend Recognition module that classifies the trend into linear, log-linear and non-linear categories. The main purpose of this step is decomposition of the original time series $f_0(t)$, consisting of N points, into sum of non-trendy time series $f(t)$ and trend component $trend(t)$

$$f_0(t) = f(t) + trend(t)$$

that allows more accurate normalcy analysis based on $f(t)$.

3) Specific normalcy bounds calculation for each category. Trend Detector performs different classical tests for trend detection. *Mann-Kendall* (MK) test is appropriate for our purposes although other known tests are also possible to apply. MK statistic (S_0) can be computed by the formula

$$S_0 = \sum_{k=1}^{N-1} \sum_{j=k+1}^{N} sign(x_j - x_k)$$

In general, the procedure consists of the following steps: data smoothing, calculation of the MK statistic S_0 for the smoothed data. If S_0>0 then trend can be increasing, otherwise (if S_0 <0) decreasing. Then, calculation of the trend measure

$$p = \left| \frac{S_0}{S_{max}} \right| 100\%$$

where

$$S_{max} = \sum_{k=1}^{N-1} \sum_{j=k+1}^{N} 1$$

Data is trendy if, for example $p > 40\%$.

Trend Recognition reveals the nature (linear, log-linear or non-linear) of the trend. We are checking linear and log-linear trends by the linear regression analysis. Goodness of fit is checked by the following formula

$$R = 1 - \frac{R_{regression}}{R_0}$$

where $R_{regression}$ is the sum of squares of the vertical distances of the points from the regression line and R_0 is similar quantity for the line with zero slop and passing through the mean of data (null hypothesis).

If R is, for example, greater than 0.6 then it is assumed that trend is linear otherwise the log-linearity is checked by the same procedure for $f(e^{c\,t})$, where c is some constant. If the corresponding goodness of fit is greater than 0.6 then data is assumed to be log-linear otherwise data is non-linear trendy.

Normalcy is performed as follows:

Data with Linear Trend. We decompose original data $f_0(t)$ into form

$$f_0(t) = f(t) + linear\_trend(t)$$

where

$$linear\_trend(t) = kt + b$$

with coefficients k and b determined by linear regression analysis and perform periodicity analysis for $f(t)$ as we described above.

If $f(t)$ is non-periodic then normalcy bounds of $f_0(t)$ are straight lines (upper and lower dynamic thresholds) which we set up by maximization of the objective function. As an objective function we consider the following expression

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

where S is the square of the area limited by t_{min}, t_{max} and some lower and upper lines (see Figure 7),

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

and P is the fraction of data within upper and lower lines and a is a user defined parameter.

Then we calculate variability (standard deviation) of $f(t)$

$$\sigma = std(f(t))$$

and consider the following set of lower and upper lines

$$\left[kt + b - z_j\,\sigma, kt + b + z_j\,\sigma\right],\ \ j = 1,2,\ldots$$

calculating each time the corresponding value g_j of the objective function. Lines that correspond to $\max(g_j)$ we take as appropriate normalcy bounds.

In our experiments we use the following values for z_j

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

If $f(t)$ is periodic then the procedure described above can be performed for each set of similar columns by calculating variability (σ_m) of the m-th set and considering the following normalcy bounds

$$\left[kt + b - z_j\,\sigma_m, kt + b + z_j\,\sigma_m\right],\ \ j = 1,2,\ldots$$

Then maximum of the objective function will give the normalcy bounds of the m-th set.

Data with Log-Linear Trend. Taking into account that $f(e^{c\,t})$ is data with linear trend the above described procedure is valid for this as well.

Data with Non-Linear trend. For this case we select the last reasonable portion of data and calculate the normalcy bounds according to the above described procedure for non-periodic case.

Figure 8 shows an example of trendy periodic data with the corresponding normalcy bounds.

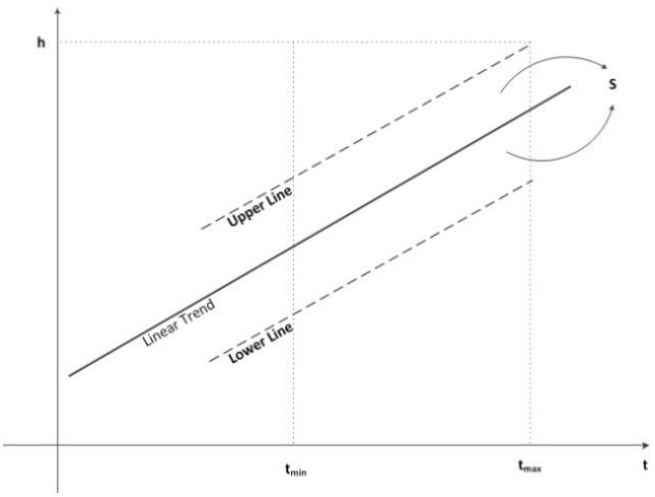


Figure 7. Auxiliary drawing for definition of the objective function.

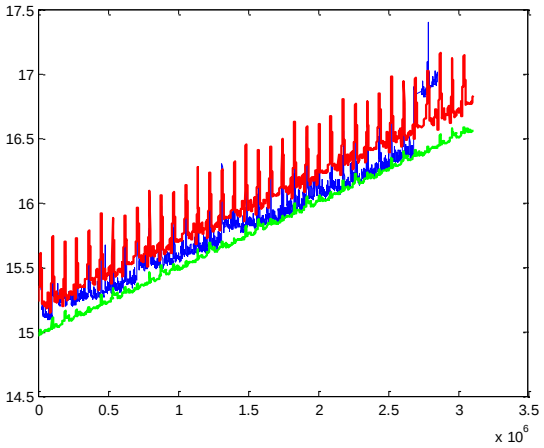


Figure 8. Normalcy bounds for trendy periodic data. Red curve is upper threshold, green curve lower threshold and blue curve is the original data.

V. Data Density Detector

Data density recognition is based on probability calculation that reveals distribution of gap. According to our analysis we differentiate the following categories: Dense Data (relative to estimated monitoring time), Sparse Data (relative to estimated monitoring time) and data with technical gap (localized gap due to malfunction of device) that after data selection will belong to Dense Data cluster, and finally, Corrupted Data.

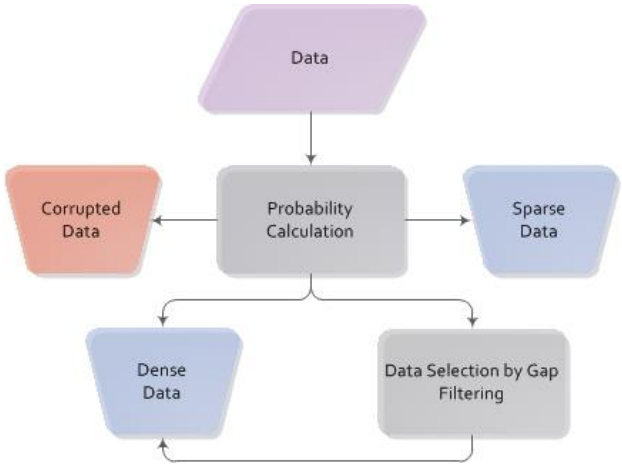
The principle scheme of density recognition and recovering (data selection) procedures is presented in Flowchart 8. For categorization purposes we deal with the following measures that characterize the nature of gap presence in data:

1) percentage of gaps,

2) probabilities for gap-to-gap, data-to-data, gap-to-data and data-to-gap transitions.

If the total percentage of gaps is acceptable then data is categorized as Dense Data.

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 give a Dense Data.
If gaps have uniform distribution in time then data belongs to a Sparse Data cluster. If gaps have extremely high percentage that further analysis is impossible then data belongs to Corrupted Data cluster.
We omit technical details as calculation of the transition probabilities can be performed as for Transient Data.



Flowchart 8. Data Density Detector.

For normalcy determination data is preliminary checked for periodicity. In case of Sparse Data duration of gaps is reasonable to take into account.

VI. Stability Detector

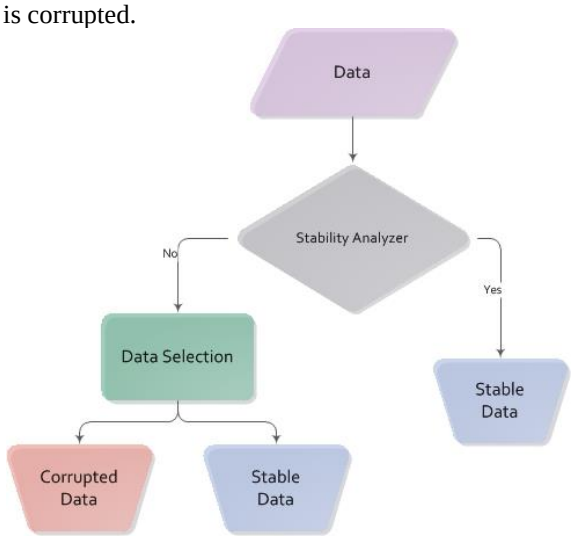
The problem of change detection in time series ([49-55]) is a well-known statistical problem. Stability Detector (see Flowchart 9) performs data processing for statistical stability recognition. If data is stable or its stable portion can be selected then the data (or selected portion) is defined as Stable Data otherwise Corrupted.
Stability identification is accomplished by construction of data *StabiloChart* that shows the stability intervals of time series and allows selection of the recent enough long data region for further analysis.
For every given m we calculate the quantity

$$s_m = \frac{|iqr(\{x_k\}_{k=m-n}^{m}) - iqr(\{x_k\}_{k=m}^{m+n})|}{iqr(\{x_k\}_{k=1}^{N})} 100\%$$

that shows the relative change (left-right) attached to the point x_m in terms of the iqr measure where n is some parameter (for example $n = \left[\frac{T}{4}\right]$, where T is the length of data).
If

$$s_m < S$$

then we set $s_m = 0$ showing the stability against the point x_m with the given sensitivity $S\ (= 70\%)$, otherwise we put $s_m = 1$ showing instability against the point x_m.
The graph of s_m's obtained along the moving (by a preset data points) x_m is the *Stabilochart* of the data. *Stabilochart* shows if data is stable, the latest stable portion can be selected, or data is corrupted.



Flowchart 9. Stability Detector.

VII. Variability Detector

Variability Detector performs data processing for variability recognition. Two different categories can be recognized: Low-Variability and High-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

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

Data clustering is performed by the following comparison with parameter $V\ (= 20\%)$: if $R \leq V$ then data is from Low-Variability cluster otherwise from High-Variability cluster.
Normalcy determination for both categories is performed by different setup of preliminary parameters – less sensitive for High-Variability Data.

VIII. Experimental Results and Discussion

We present some results of experiments on an actual customer data set. First we performed experiments for short-term data with almost one month duration. NDS was applied to 3215 time series metrics. Table 3 shows the distribution along different data categories. Table 4 shows the count of periodic and non-periodic data.

Data Category	*Count (Percentage) of Metrics in a Specific 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 3. Distribution along the categories for short-term data set.

Periodic	*Non-Periodic*	*Corrupted*	*Overall*
1511	1595	109	3512

Table 4. Count of periodic and non-period data for short-term data set.

We also examined the distribution of periodic data in some categories. For 532 Semi-Constant Data (see Table 3), 267 have percentage of outliers less than 15% and they are claimed as non-periodic without any further checking. The remaining 235 metrics are investigated in sense of periodic structure and in 212 of them periods are found. In case of High-Variability and Low-Variability categories periods are found for 378 and 165 metrics, respectively.

Second, we performed experiments for long-term data with almost three month length. We obtained 3956 metrics and Table 5 shows distribution along different categories. Table 6 shows distribution of metrics along periodicity. For 586 Semi-Constant metrics 324 have outliers less than 15% and they are categorized as non-periodic, 262 checked for periodicity and for 221 periods are found. Then, for High-Variability and Low-Variability categories periods are found for 457 and 165 metrics, respectively.

It is worth noting that results obtained for the specific customer can't be in any manner generalized to other cases. The results can vary widely from one customer to another without any intersection.

Data Category	*Count (Percentage) of Metrics in a Specific Category*
Multinomial	877 (22.2%)
Trendy	406 (10.3%)
Semi-Constant	586 (14.8%)
Transient	89 (2.2%)
Sparse	129 (3.3%)
Low-Variability	1130 (28.6%)
High-variability	683 (1.6%)
Corrupted	56 (1.4%)

Table 5. Distribution along the categories for long-term data set.

Periodic	*Non-Periodic*	*Corrupted*	*Overall*
1742	2158	56	3956

Table 6. Count of periodic and non-period data for long-term data set.

However, results obtained for a specific customer can provide useful information about the customers' environment. In terms of our approach this can also lead to some optimizations by excluding procedures for a specific category that is not common for the customer. For example, in Table 5, we see that Sparse and Transient categories cover only 5.5% of the overall data set and the system can be applied without specifying them.

Another important insight is that data category is not an invariant property. Change in length of data in general changes the category. Moreover, data selection module is picking up the last stable portion and categorization is performed only on this portion. So visually data can be corrupted but its latest stable portion belongs to some of the predefined categories.

Figures 10, 11, and 12 present reliably predicted normalcy bounds obtained by the NDS.

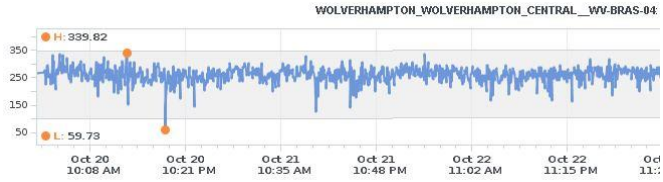


Figure 9. An example of a non-periodic data with the corresponding normalcy bounds.

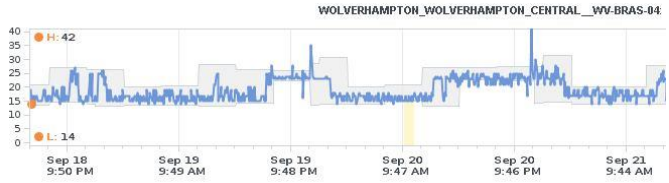


Figure 10. An example of a periodic data with the corresponding normalcy bounds. Yellow area is alarm.

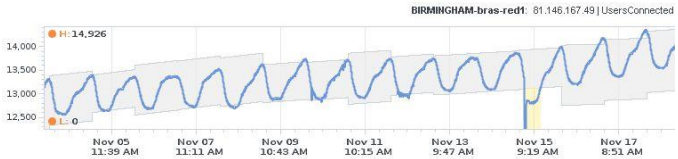


Figure 11. An example of a Trendy Data with the corresponding normalcy bounds and an alarm.

REFERENCES

[1] W.A.Shewhart (1931), "Economic control of quality manufactured product", New York: D. Van Nostrand Company, Republished in 1980 by the American Society of Quality Control.

[2] D.J. Wheeler, and D.S. Chambers (1986), "Understanding statistical process control", Knoxville, TN: SPC Press.

[3] D.J. Wheeler (1991), "Shewhart's chart: myths, facts, and competitors," 45th Annual Quality Congress Transactions.

[4] F. Alt (1985), "Multivariate Quality Control", Encyclopedia of Statistical Sciences, Volume 6.

[5] C.A. Parris, "Hard disk drive infant mortality tets", Patent application number: US 09/387,677. Filing date: Aug 31, 1999. Publication number: US6408406B1. Publication date: Jun18, 2002. Publication type: grant.

[6] F.L. Paulson, "System and method for determining optimal sweep threshold parameters for demand deposit accounts", Patent application number: US 08/825,012. Filing date: Mar 26, 1997. Publication number: US5893078 A. Publication date: Apr 6, 1999. Publication type: grant.

[7] J.D. Luan, "Threshold alarms for processing errors in a multiplex communications system", Patent application number: EP19880112793. Filing date: Aug 5, 1988. Publication number: EP0310781B1. Publication date: Mar 10, 1993. Publication type: grant.

[8] P. Celka, "A method and system for determining the state of a person", Patent application number: PCT/EP2013/064678. Filing date: Jul 11, 2013. Publication number: WO2014012839 A1. Publication date: Jan 23, 2014.

[9] B.M. Jakobson, "Systems and methods for authenticating a user and device", Patent application number: US 13/523, 425. Filing date: Jun 14, 2012. Publication number: US20130340052 A1. Publication date: Dec 19, 2013.

[10] T. Blumensath, M.E. Davies (2009), "Iterative hard thresholding for compressed sensing", Applied and Computational Harmonic Analysis, Vol. 27, no. 3, pp. 265-274.

[11] H. Huang, H. Al-Azzawi, and H. Brani (2014), "Network traffic anomaly detection", ArXiv:1402.0856v1.

[12] D. Dang, A. Lefaive, J. Scarpelli, and S. Sodem, "Automatic determination of dynamic threshold for accurate detection of abnormalities", Patent application number: US 12/847,391. Filing date: Jul 30, 2010. Publication number: US20110238376 A1. Publication date: Sep 29, 2011.

[13] J. C. Jubin, V. Rajasimman, N. Thadasina, "Hard handoff dynamic threshold determination", Patent application number: US 1/459,317. Filing date: Jun 30, 2009. Publication number: US20100056149 A1. Publication date: Mar 4, 2010.

[14] Ying-ru chen, "Method of motion detection using adaptive threshold", United States, Patent application number: US 12/353,679. Filing date: Jan 14, 2009. Publication number: US 8077926 B2. Publication date: Dec 13, 2011.

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

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

[17] A.V. Poghosyan, A.N. Harutyunyan, N.M. Grigoryan, M.A. Marvasti, "Methods and systems for abnormality analysis of streamed log data". Patent application number: US 13/960,611. Filing date: Aug 6, 2013. Publication number: US20140053025 A1. Publication date: Feb 20, 2014.

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

[19] A.V. Poghosyan, A.N. Harutyunyan, N.M. Grigoryan, M.A. Marvasti. "Data-agnostic anomaly detection". Patent application number: US 13/853,321. Filed March 29, 2013.

[20] M.A. Marvasti, A.V. Poghosyan, A.N. Harutyunyan, and N.M. Grigoryan, "An Enterprise Dynamic Thresholding System". Accepted for presentation in the 11th International Conference on Autonomic Computing (ICAC'14), June 18-20, 2014, Philadelhia, US.

[21] M.A. Marvasti, A.V. Poghosyan, N.M. Grigoryan, and A.N. Harutyunyan, "Method and apparatus for root cause and critical pattern prediction using virtual directed graphs". Patent application number: US 13/271,554. Filing date: Oct 12, 2011. Publication number: US20130097463 A1. Publication date: Apr 18, 2013.

[22] M.A. Marvasti, A.V. Poghosyan, N.M. Grigoryan, and A.N. Harutyunyan, "Automated analysis of unstructured data". Patent application number US 13/417,933. Filing date: Mar 12, 2012. Publication number: US20130097125 A1. Publication date: Apr 18, 2013.

[23] M.A. Marvasti, A.V. Poghosyan, A.N. Harutyunyan, N.M. Grigoryan (2013), "An anomaly event correlation engine: identifying root causes, bottlenecks, and black swans in IT environments". VMware Technical Journal, Vol. 2, Issue 1, 35-45.

[24] M.A. Marvasti, A.V. Poghosyan, A.N. Harutyunyan, and N.M. Grigoryan, "Pattern detection in unstructured data: An experience for avirtual IT infrastructure", Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, Belgium, May 27--31, 2013, pp. 1048-1053.

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

[26] A.G. Barnett, and A.J. Dobson (2010), "Analysing seasonal health data". Springer. ISBN 978-3-642-10747-4.

[27] W.P. Cleveland (1972), "Analysis and forecasting of seasonal time series". Ph.D. dissertation, University of Wisconsin.

[28] G.E.P. Box, and G.M. Jenkins (1970), "Time series analysis: forecasting and control". San Francisco: holden-Day, Inc.

[29] C.I. Plosser (1976), "Time series analysis and seasonality in econometric models with an applicaion to a monetary model. Ph.D. dissertation, University of chicago, Graduate School of business.

[30] K.F. Wallis (1974), "Seasonal adjustment and the relations between variables". Journal of the American Statistical Association 69, pp.18-31.

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

[32] B. Boashash, ed. (2003), "Time-frequency signal analysis and processing: A Comprehensive Reference", Oxford: Elsevier Science, ISBN 0-08-044335-4.

[33] A.V. Oppenheim, R.W. Schafer, and J.R. Buck (1999), "Discrete-time signal processing". Upper Saddle River, N.J.: Prentice Hall. ISBN 0-13-754920-2.

[34] P. Stoica, and R. Moses (2005), "Spectral analysis of signals". Prentice Hall, NJ.

[35] S.M.G. Kendall (1976), "Time series", Second Edition, Charles Griffin & Co.. ISBN 0-85264-241-5.

[36] D.W. Tufts, and R. Kumaresan (1982), "Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood," Proc. IEEE, Vol. 70, pp. 975-989.

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

[38] M.A. Marvasti, A.V. Grigoryan, A.V. Poghosyan, N.M. Grigoryan, and A.N. Harutyunyan, "Methods for the cyclical pattern determination of time-series data using a clustering approach", Patent application number: US 12/186,496. Filing date: Aug 5, 2008. Publication number: US20100036857 A1. Publication date: Feb 11, 2010.

[39] J.S. Simonoff (1998), "Smoothing Methods in Statistics", 2nd edition. Springer ISBN 978-0387947167.

[40] A.W. Bowman, and A. Azzalini (1997), "Applied smoothing techniques for data analysis", Oxford University Press, London.

[41] C.R. Goodall (1991), "A survey of smoothing techniques", in Modern Methods of Data Analysis, Chapter 3, J. Fox & J.S. Long, eds, Sage, Beverly Hills, pp. 126–176.

[42] C.L. Mallows (1980), "Some theory of nonlinear smoothers", The Annals of Statistics 8, pp. 695–715.

[43] H.B. Mann (1945), "Nonparametric tests against trend". Econometrica 13, pp. 245–259.

[44] M.G. Kendall (1975), "Rank Correlation Methods". Griffin, London, UK.

[45] D.A. Ratkowsky (1989), "Handbook of Nonlinear Regression Models". Marcel Dekker, New York, USA.

[46] S. Yue, P. Pilon, and G. Cavadias (2002), "Power of the Mann-Kendall and Spearman's rho tests for detecting monotonic trends in hydrological series". J. Hydrol. 259, pp. 254–271.

[47] A.C. Davison, and D.V. Hinkley (1997), Bootstrap Methods and Their Applications. Cambridge University Press, Cambridge, UK.

[48] N.N. Abdelmalek (1974). "On the discrete linear L1 approximation and L1 solution of over-determined linear equations", Journal of Approximation Theory, Vol. 11, pp. 35 – 53.

[49] M. Basseville, and I. Nikiforov (1993), "Detection of abrupt changes: theory and application". Information and System Science Series. Prentice Hall, Englewood Cliffs, NJ.

[50] R.P. Adams, and D.J.C. MacKay (2007), "Bayesian online changepoint detection". ArXiv:0710.3742v1.

[51] B. Brodsky and B. Darkhovsky (1993), "Nonparametric methods in change-point problems". Kluwer Academic Publishers, Dordrecht, the Netherlands.

[52] Krishnaiah and C.R. Rao, eds. (1988), Handbook of Statistics, Vol. 7, pp. 403–425. Elsevier, Amsterdam, the Netherlands.

[53] F. Gustafsson (1996), "The marginalized likelihood ratio test for detecting abrupt changes". IEEE Transactions on Automatic Control, Vol. 41, no 1, pp. 66–78.

[54] Y. Kawahara, T. Yairi, and K. Machida (2007), "Change-point detection in time-series data based on subspace identification". In Proceedings of the 7th IEEE International Conference on Data Mining, pp. 559–564.

[55] U. Paquet (2007), "Empirical Bayesian change point detection". Graphical Models, Vol. 1995, pp. 1–20.