Abnormality Analysis of Streamed Log Data

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

***Abstract*—We examine the determination of abnormality of streamed data using the statistical structure of the meta-data associated with it. The vital need for such a subject within a heterogeneous log based environment in real-time comes from the fact that most cloud based applications will use text-based logging as a means of reporting application behavior. The sheer volume of such logs makes retrospective analysis infeasible due to large processing and storage requirements. Our approach is based on conversion of the original data stream into meta-data (graph) and revealing the dominating (normal) statistical patterns within it. Real-time analysis of the stream compared with the meta-data model determines the degree of anomaly of the current data. The resulting graph also reveals the fundamental structure ("behavioral footprint") of the data beyond the sources (physical or virtual devices) and processes.**

***Index Terms*—Cloud and virtualization management, log analysis, big data, events stream, normalcy structure, anomaly and change detection.**

I. Introduction

We treat the problem of statistical structure retrieval from data streams within the "big data" model [1]. The relevance of this kind of analysis is dictated by rapidly growing needs to make run-time decisions on abnormality of cloud based environments from which log data can be streamed. Examples include application logs, user behavior data, social network data, etc.

A fundamental approach to data-agnostic management of data centers applies the dynamic thresholding of monitoring data collected from IT resources instead of out-of-date static or hard thresholding technique. The dynamic thresholds (DT) of a time series metric are the appropriate upper and lower bounds computed using variability, change point, and cyclicality of the data (see [2]-[4]).

In this paper we extend the concept of dynamic thresholding of time series data to any kind of data that is a stream of records and events. In other words, we introduce the notion of normalcy of those streams as an extracted statistical structure and develop a mechanism for their abnormality detection in run-time mode.

In our earlier paper [4] we employed directed graphs to extract patterns from unstructured data. Here we use the same graph model as a tool for capturing the meta-data patterns of streaming log files to determine system anomaly states, based on its historic "normalcy structure", accounting also for change-points occurring in the stream. Prior related work includes the papers by Lin and Siewiorek [5] and Rouillard [6], among others. In particular, [5] is devoted specifically to extraction of intermittent errors in error logs and [6] studies modeling of normal/abnormal events in logs employing rule sets. Another related problem is the workflow model discovery and its transactional improvement from event based logs [7]. In this regard, our analysis is generic in terms of relying on non-contextual and non-rule based pattern extraction in data streams compared to those works (see also references therein). Moreover, it builds on a global correlation structure of the stream to quantify degrees of abnormalities occurring run-time, as well as to localize "responsible anomaly sources" with their contributions to the overall abnormality. No prior research to the best of our knowledge directly targets this setting.

Our approach is based on conversion of the original stream into a meta-data and revealing the dominating (normal) statistical patterns within it. The meta-data is formed via a graph representing different "event types" (nodes) which are detected in the stream along with "event sources" they are associated within the stream. Two nodes in this graph are adjacent if the corresponding event types are in proximity by a time frame and an "event source". The behaviors of those sources in terms of generating different event types create probabilistic correlations between graph nodes computed with application of that proximity measure. The characterized probabilistic graph represents the normal or dominating common behavioral structure that the stream historically follows independently of the event sources. Having this historical normalcy, we are then able to estimate the upcoming stream portions in terms of their deviation from the extracted structure. This is performed through matching the event types in the new data entries with historical graph and checking whether conditionally the most probable event types are realized in the observation window. From this we can quantify the deviation of the current data segment from the most typical (historical) patterns. This quantity actually measures the abnormality degree of the stream.

One of the main problems in processing of data belonging to "big data" field is the limited availability of data for retrospective analysis. By its very nature, limits on storage and I/O can severely restrict algorithms that require the complete data set to be available for analysis. In terms of this restriction, our algorithm is highly scalable. Converting the data stream into a graph structure representing the meta-data, we retain the useful content of the data and build the wanted statistical structure without any need for retrospective analysis.

We apply our method to a set of vCenter (VMware's virtualization management software) logs. Using the

978-1-4799-0913-1/14/$31.00 ©2014 IEEE

information on event types and on fleeting VM's (or hosts) as sources of those event types, we are able to determine the statistical normalcy structure of the stream by the above mentioned graph. Comparison of real-time data to this graph allows us to then determine abnormality patterns. Through that comparison we estimate the degree of abnormality that can be used by an alerting engine within an infrastructure management system. Moreover, the algorithm can be applied to normalcy analysis of virtualized environments at different hierarchical levels (VM, host, cluster, etc.).

The normalcy structure represents the image of pure event type correlations independently of the event sources in the heterogeneous system. In other words, it is the fundamental structure ("behavioral footprint") of the data beyond underlying sources (physical or virtual) and processes. Thus another benefit of this method is that comparison of the current data portions with the meta-data graph may shed light on sources of abnormal situations.

II. Stream-to-Meta-Data Conversion

We interpret the streamed log data as a flow of text consisting of events with associated event types and event sources (which can be detected via a log parsing procedure). The procedure outlined for the automatic detection of event attributes in [4] is one example of an event detection mechanism. This is the basic assumption behind our algorithm on processing of streamed log data in terms of extracting its fundamental statistical characteristics. Say the stream contains I different types T_i of events and K different sources S_k of events. We aim at investigating how those types are correlated along with emerging stream inputs independently of the sources they are associated with.

Two types T_i and T_j are considered to be related with each other from short term perspectives, if they appear in the stream attached to the same source S_k and within a time window Δt. In other words, to determine the correlations between event types we apply a proximity pair criterion $(S_k, \Delta t)$. Hence, the probability of appearance of T_i under an observed type T_j (both associated with S_k) can be estimated by the following frequency

$$P(T_i \mid T_j, \Delta t) = \frac{1}{N(T_j)} \sum_{k=1}^{K} N(T_i, T_j \mid S_k, \Delta t)$$

as ratio of joint occurrences $N(T_i, T_j \mid S_k, \Delta t)$ of the type pair (T_i, T_j) in the stream over the number $N(T_j)$ of observed type T_j independently of S_k, $k = \overline{1, K}$. Therefore, the conditional probability between two event types provoked by all possible event sources can be determined with hereinafter usage of concise notations $P(i \mid j)$ or $P_{i|j}$.

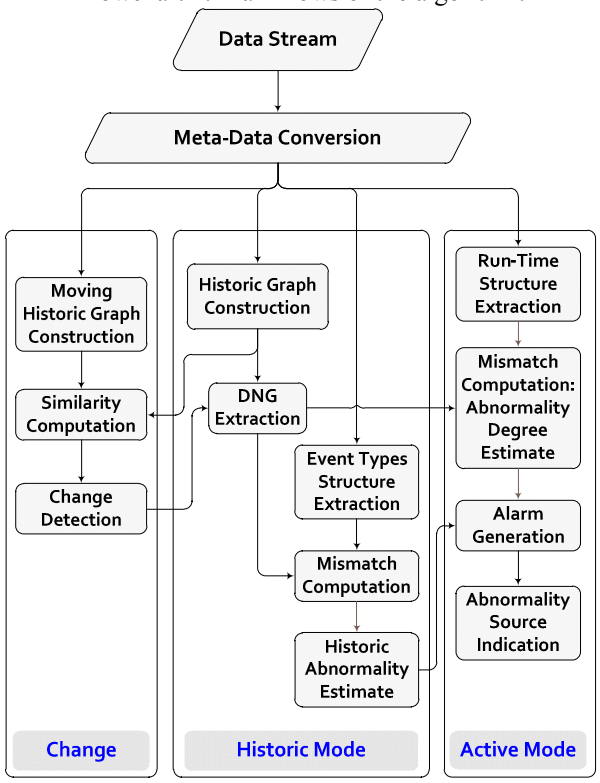
The prior probabilities of event types can be also computed by the frequency $P(T_i) = N(T_i)/L$ where $i = \overline{1, I}$ and L stands for the number of events read from the stream. In view of the above definitions, the conditional probability $P(T_i \mid T_j)$ becomes a tool for measuring the "common behavior" of event types abstracted from underlying event sources. Having computed the prior and conditional probabilities between event types we can build the structure in Fig. 1 showing a weighted graph of event types with those conditionals on edges. Note that the above mentioned frequencies can be updated cumulatively and the corresponding meta-data in form of a directed graph can be updated with the expansion of the stream. Interestingly, this stream-to-meta-data conversion can be implemented by the map/reduce programming model [8].

III. The Normalcy Structure

The constructed meta-data graph of Fig. 1 can be viewed as the dynamically and historically built structure of pairwise correlations of event types. This is the graph used for processing historical normalcy and run-time decision making on abnormality behavior of the stream.

Flowchart 1 describes the main parts of our analysis and their interrelation. The overall algorithm behind this study will be described in detail below.

Flowchart 1. Main flows of the algorithm.



Training of the model. We define the normalcy structure of the stream by its meta-data and in terms of the dominating statistical relations in the graph of Fig. 1. Analogous to outlier removal in time series data, small conditional probabilities are considered to be outliers and thus can be removed from analysis. Eliminating small conditional probabilities from the graph edges (a sensitivity parameter is applied) we reduce it to its dominating correlations subgraph(s). We call this graph Dynamic Normalcy Graph (DNG). It can be easily seen that the DNG is the stream's

historical footprint of common probabilistic behavior of event types that result from all possible event sources.

Scoring/Ranking. The event types in the current observation window are mapped to the DNG to compare the mismatch between the run-time scenarios to those in historical mode. The degree of mismatch represents the degree of abnormality of the real-time data.

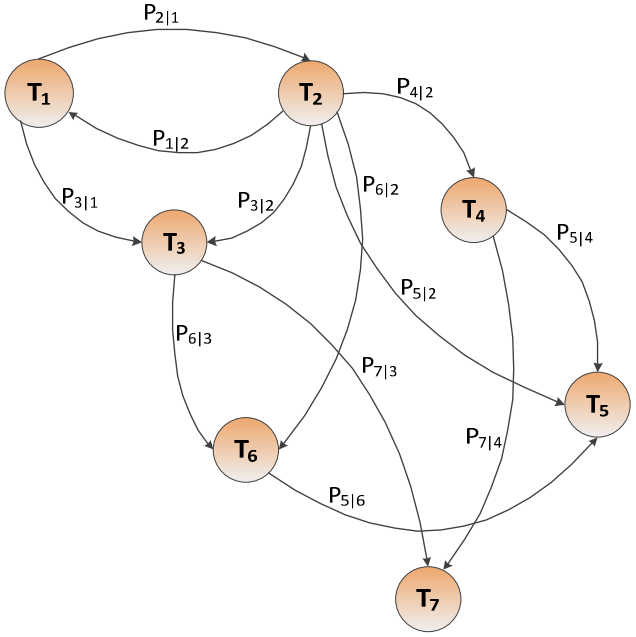


Fig. 1. Meta-data graph of event types, $I = 7$.

Let $T_1, T_2, ..., T_{j(\Delta w)}$ be the event types registered in the current observation window Δw. And let the subset $T_{j_1}, T_{j_2}, ..., T_{j_k(\Delta w)}$ be those event types that pair-wise are in proximity for event source S_k, $k = \overline{1, k(\Delta w)}$. In other words, the following holds:

$$j(\Delta w) = \sum_{k=1}^{k(\Delta w)} j_k(\Delta w) .$$

Those event types can be matched on the DNG and the edges having matched nodes can be highlighted. We can then compute the probabilistic mismatch (i.e. the abnormality degree) between this matched sub-graph and the DNG (relying on the graph isomorphism concept) by the following formula

$$M(\Delta w) = \sum_{k=1}^{k(\Delta w)} M(S_k, \Delta w)$$

where $M(S_k, \Delta w)$ stands for a mismatch assigned to source S_k and calculated by

$$M(S_k, \Delta w) = \frac{\sum_{j=1}^{j_k(\Delta w)} \sum_{i=1, i\neq j}^{\overline{m}(j)} P(i \mid j)}{\sum_{j=1}^{j(\Delta w)} \sum_{i=1, i\neq j}^{m(j)} P(i \mid j)} 100\%$$

and where we assume that nodes adjacent (in DNG) to j nodes are enumerated from 1 to $m(j)$. We also enumerate the absent nodes in Δw from 1 to $\overline{m}(j)$. Note that the summation in $\sum_{i=1, i\neq j}^{m(j)} P(i \mid j)$ as well as in $\sum_{i=1, i\neq j}^{\overline{m}(j)} P(i \mid j)$ is performed over the conditional probabilities on the directed edges arising from the node j in DNG. $M(\Delta w)$ lies in the interval $[0, 100\%]$.

Measuring the mismatch between the run-time flow of events and the historical normalcy allows us to control unacceptable deviations and generate alarms. For that purpose, first we keep track of historic anomalies and estimate their usual (normal) level. That can be performed by simple measures such as whiskers method (that recognizes the concentration of data points from out-of-range values) and high quantile cut of data, or the sophisticated DT computation technology.

In particular, let

$$\overline{M} = \{M_1(\Delta w), M_2(\Delta w), ..., M_h(\Delta w)\}$$

be the series of subsequent mismatches calculated along the historical application of the obtained DNG to the stream with a moving Δw window. And let $Q(q, \overline{M})$ be the $q-$th quantile of $\overline{M}$. Then we define the historical abnormality $D_0(\Delta w)$ according to the method of whiskers as $D_0(\Delta w) = Q(0.75, \overline{M}) + 1.5IQR(\overline{M})$, where $IQR(\overline{M})$ is the interquartile range (the difference between 0.75 and 0.25 quantiles) of $\overline{M}$. In an alternative way $D_0(\Delta w)$ can be defined also as 0.9 or higher quantile of $\overline{M}$ regulated by a sensitivity parameter $s \in [0,1]$, where $s = 1$ corresponds to $Q(0.9, \overline{M})$ and the lowest $s = 0$ to the maximum of $\overline{M}$. The deduced level $D_0(\Delta w)$ is used in run-time estimate of abnormality. So only the abnormalities with their mismatch $M(\Delta w)$ above $D_0(\Delta w)$ are reported. This means that the real and relative abnormality degree $D(\Delta w)$ above $D_0(\Delta w)$ in run-time mode can be obtained with the following difference:

$$D(\Delta w) = M(\Delta w) - D_0(\Delta w).$$

Here the approach can be extended to apply a dynamic thresholding technique (e.g. [3]) for softer abnormality control (see also the study [9] and references therein). In other words, instead of $D_0(\Delta w)$ we can compute its dynamic (time-dependent) version $D_0(\Delta w, t)$.

IV. Indicating Abnormality Sources

During the abnormality detection by the above mentioned mismatch calculation, the event types which contribute to the mismatch can be identified. Then those event types can be checked in terms of the event sources associated with them. An ordered list of recommendations

indicating the most likely sources of abnormality can be generated. To generate those recommendations and indicating the highly "responsible" event sources in an unacceptable abnormality, let $S_1, S_2, ..., S_{k(\Delta w)}$ be the event sources that proximate the event types $T_{j_1}, T_{j_2}, ..., T_{j(\Delta w)}$ in run-time mode. For each event source S_k we compute its mismatch $M(S_k, \Delta w)$ and generate the following series: $M(S_1, \Delta w), M(S_2, \Delta w), ..., M(S_{k(\Delta w)}, \Delta w)$. Those sources are then prioritized with the corresponding mismatches and displayed for the final recommendation to the user as indications of abnormality sources.

V. On Change Point Detection

Global changes in the stream which are able to skew our abnormality analysis can also be detected with a procedure comparing two meta-data portions from the stream in terms of mismatch between their graph-wise representations.

For quantification of similarity between two historical streams we introduce the following measure:

$$Sim(G_1, G_2) = \frac{\|E_1 \cap E_2\| + \|V_1 \cap V_2\|}{\|E_1 \cup E_2\| + \|V_1 \cup V_2\|} \times \times \left(1 - \frac{\sum_{i,j,i \neq j} \left| P_{i|j}(G_1) - P_{i|j}(G_2) \right|}{\|E_1 \cap E_2\|} \right)$$

where $G_1 \equiv (V_1, E_1)$ and $G_2 \equiv (V_2, E_2)$, composed from the set of vertices V_k and edges E_k, $k = 1,2$. This measure ranges from $[0,1]$ with maximum similarity 1.

In the similarity formula above the first fraction is responsible for the geometric similarity of the graphs and the second for the probabilistic closeness. The notation $P_{i|j}(G_k)$ stands for the conditional probabilities on jointly present edges for G_k. We impose an additional constraint for the condition $\|E_1 \cap E_2\| = 0$ by setting $Sim(G_1, G_2) = 0$.

To verify whether the stream has substantially changed during its development, we take advantage of its meta-data structure again, now for recent period of the original stream containing $L/2$ events from total length L, assuming $L/2$ is large enough to provide sufficient statistics. This means that as soon as the sufficient statistics is available we initiate generating a parallel construction of a meta-data graph for $L/2$ –recent size (tail) of the stream. Then we keep updating this graph along with the evolution of the data sliding the tail window with size $\Delta C = L/4$. The latter assumes that we refresh the graph nodes/connections obtained due to the scanning of the stream in the first $L/4$ part of the tail with the new nodes/connections observed in the last $L/4$ portion of the tail observed by the sliding window. At each stage of this moving structure update we have two meta-data graphs, basic one and the moving one, conventionally denoting them $G_1(L)$ and $G_2(L/2)$, respectively. Estimate of their similarity with $Sim(G_1(L), G_2(L/2))$ tells us how close the adaptive data tail to the overall process is. As soon as we detect only a 50% similarity (or another parameterized quantity), then we have to declare a change point and replace the basic historic meta-data $G_1(L)$ with the moving one $G_2(L/2)$ and proceed with the rest of algorithmic blocks as shown in Flowchart 1. With these settings we don't target a sensitive change point detection problem but only a global change identification in the stream occurred in its recent structure versus its overall structure. The tail size and ΔC can be tuned for softer analysis of change.

VI. Results for Virtual Center Events

Here we discuss the application of the abnormality detection algorithm to a parsed log data of vCenter consisting of 200,000 events (a time period spanning one month). In this case, the event sources are VM's or hosts and the event types are the corresponding types from the log, such as VmEmigratingEvent or VmStoppingEvent with additional categories they are attributed to in the stream, for example, "info" (i), "error" (e), or "warning" (w). So for our analysis the combinations such as "VmEmigratingEvent+info" and "VmEmigratingEvent+error" are interpreted as basic event types.

Fig. 2 shows the normalcy structure of the log processed on the above mentioned events data and Table 1 details its node description (A defines the attribute column). This DNG represents the pure event type correlations where the outlier relations are filtered out. Additionally, to compress the structure only strong correlations (higher 0.8) are illustrated. Similar graphs are obtained for a series of experiments that confirm that the vCenter has its inherent statistical and fundamental structure of event type behaviors independent of the applications that run on the VM's.

Those experiments were performed for different portions of the vCenter log containing more than 1,000,000 event records as well as for the whole data set. In all cases it was possible to derive a DNG with high probabilistic connections between a subset of defined event types.

Several observations from the obtained DNG can be made:

I. DNG contains an unconnected fragment (nodes 32 and 33), i.e. a sub-graph, which means that the virtual center imprints isolatable event types. In case of the nodes 32 and 33, one can make a conclusion that most of the time (94%) "VmBeingRelocatedEvent+info" results in "VmRelocatedEvent+info" with 6% failure that would result in an abnormality situations.

II. There are event types with only outgoing connections (like node 29) and event types with only incoming connections (node 4). In other words, the composite event type "VmRegisteredEvent+info" inevitably leads a collection of event types (23,51,49, etc.), meanwhile a series of event types (2,5,16, etc.) ultimately lead to "VmResourceReallocatedEvent+info".

III. An important class of correlations is related to deterministic connections. For example, “VmInstanceUuidConflictEvent+error” (49) generates “VmInstanceUuidChangedEvent+info” (51) without any alternative. The same happens with “VmRenamedEvent+warning” (35) and “VmReconfiguredEvent+info” (11), however these event types have no impact on other types and are of no influence to the rest of the system.

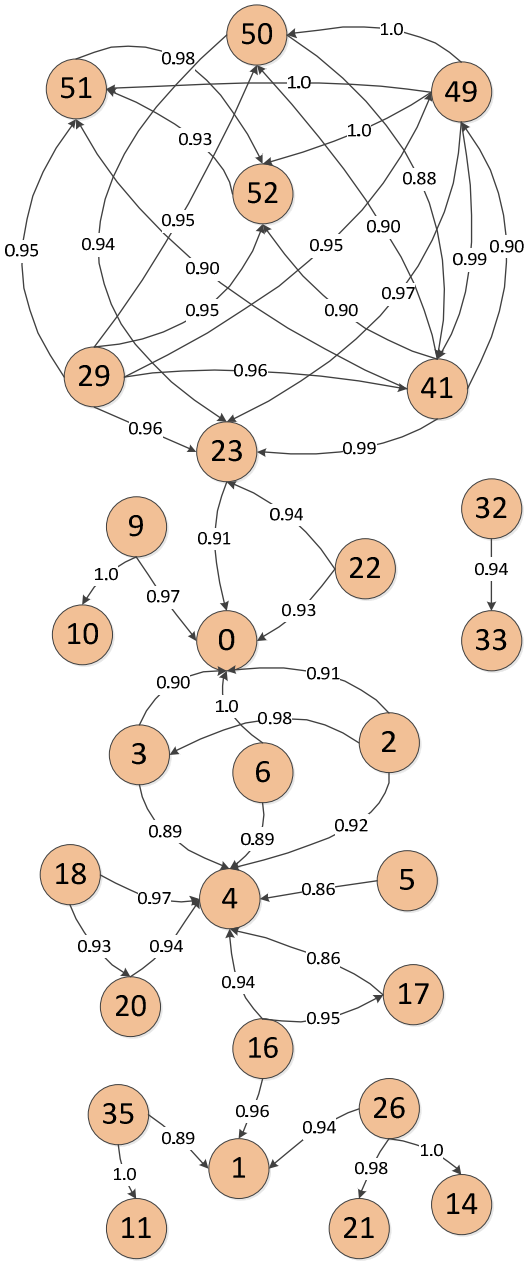


Fig. 2. Normalcy structure (DNG) of vCenter events.

The extracted DNG becomes the “behavioral footprint” of the virtual center in terms of VMs behaviors. It means that the lifecycle of any VM should follow the structure that the DNG dictates. Any deviation introduced by VMs in the current stream becomes the abnormalities. It can be either an evolving critical abnormality or an out-of-normal state that can also be inherent to the stream in terms of its historical behavior. That is why we evaluate typical out-of-normalcy (so called historical abnormality) of the stream in order to differentiate it from the abnormality that needs to become an alert.

Table 1. Nodes of Fig. 2.

ID	Event Type	A
0	vim.event.AlarmStatusChangedEvent	i
1	vim.event.TaskEvent	i
2	vim.event.VmBeingHotMigratedEvent	i
3	vim.event.VmEmigratingEvent	i
4	vim.event.VmResourceReallocatedEvent	i
5	vim.event.VmMessageEvent	i
6	vim.event.DrsVmMigratedEvent	i
9	vim.event.AlarmActionTriggeredEvent	i
10	vim.event.AlarmSnmpCompletedEvent	i
11	vim.event.VmReconfiguredEvent	i
14	vim.event.VmAcquiredMksTicketEvent	i
16	vim.event.VmStoppingEvent	i
17	vim.event.VmPoweredOffEvent	i
18	vim.event.VmStartingEvent	i
20	vim.event.DrsVmPoweredOnEvent	i
21	vim.event.VmPoweredOnEvent	i
22	vim.event.VmDisconnectedEvent	i
23	vim.event.VmConnectedEvent	i
26	vim.event.VmResettingEvent	i
29	vim.event.VmRegisteredEvent	i
32	vim.event.VmBeingRelocatedEvent	i
33	vim.event.VmRelocatedEvent	i
35	vim.event.VmRenamedEvent	w
41	vim.event.VmDiscoveredEvent	i
49	vim.event.VmInstanceUuidConflictEvent	e
50	vim.event.VmMacConflictEvent	e
51	vim.event.VmInstanceUuidChangedEvent	i
52	vim.event.VmMacChangedEvent	w

For the example of Fig. 2 we compute the historical abnormality estimate $D_0(\Delta w)$ and show it in the plot of Fig. 3. This figure depicts the mismatches $M(\Delta w)$ along the historical log for the extracted DNG with $\Delta w = 30$ minute sliding by 5 minute intervals. Here the computed value for $D_0(\Delta w)$ is 25.55% (for sensitivity $s = 0.7$), therefore, abnormalities are indicated at run-time for values above this level.

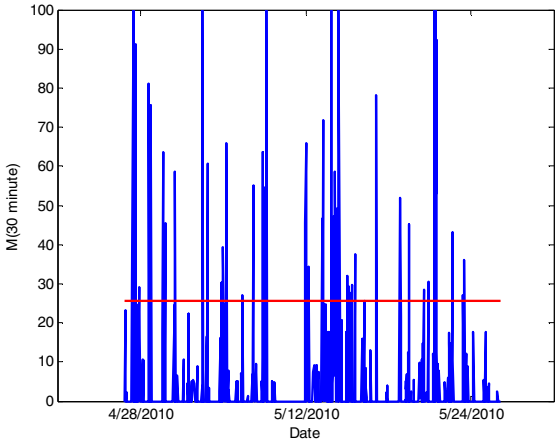


Fig. 3. Historical abnormality estimate $D_0(\Delta w) = 25.55\%$

Fig. 4 shows abnormality jumps in run-time mode detected for the same log after its DNG extraction. For the two abnormality peaks in Fig. 4 we were then able to generate a list indicating the highly probable misbehaving VMs (Table 2). The first abnormality process occurred on 5/24/2010 at 04:17PM (time (T) point A in Fig. 4) and the second one on 5/24/2010 at 05:18PM (point B in Fig. 4). One VM was detected in each case that generated events, however, failed to generate the highly correlated events associated with them. Table 2 shows these highly culpable VMs (with mismatch scores (MS) of 40.9% and 100%, respectively) with columns of generated event (GE) ID and missing event (ME) ID.

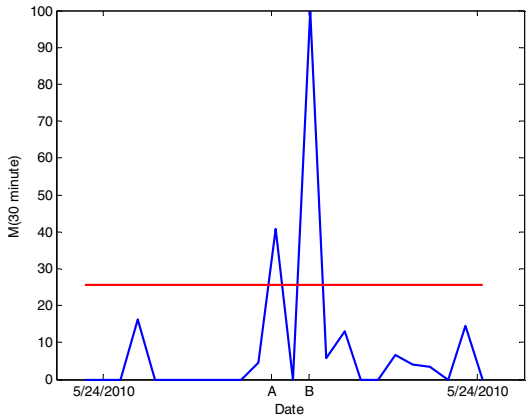


Fig. 4. Run-time abnormality above $D_0(\Delta w)$

Table 2. Recommended VMs at abnormal state of vC log.

T	VM Name	GE ID	ME ID	MS
A	communities-lt-db-1	0, 1, 10	-	40.9%
		9	0	
		2, 3	4	
B	ora-dev2-ksdmmk-d1	23	0	100%

This essentially shows that breakage of expected correlations become abnormality events that can indicate when it's appropriate to look into the log data.

To verify the robustness of our analysis against Δw, we computed the function $D_0(\Delta w)$ for different window sizes, getting a near constant behavior.

VII. Appendix: Analysis of Economic Data

The methodology was applied to an economics data set to show its agnostic nature. This data set consisted of key short-term economic indicators (EI) of OECD countries (available at http://stats.oecd.org/). OECD is an international economic organization of 34 countries founded in 1961 to stimulate economic progress and world trade. Also data of 8 non-member countries with large economies are included.

We aimed at extracting the normalcy structure of EIs for above mentioned countries within 1995-2007 and detecting the abnormality or change in that structure provoked by the recent economic recession. In this scenario, the event sources are the member countries and the event types are their main short-term (quarterly) EIs relative to the preceding period. Table 3 shows the event types participating in the DNG of OECD+8 economic normalcy structure of Fig. 5. Overall, 20 key indicators (with theoretically possible 60 event types – positive, negative, and zero changes in indicators being different event types) are included in the analysis. However, only 17 positive change indicators constitute this DNG. The parameters used for this analysis were $\Delta t = 1$ quarter, $\Delta w = 1$ quarter.

Table 3. EIs in DNG relative to previous period.

ID	Event Type/Key Economic Indicators
1	DomesticProducerPrices-Manufacturing_positive
2	GDP-PrivateFinalConsumptionExpenditure_positive
3	GrossDomesticProd_positive
4	Employment_positive
5	Financial-SharePrices_positive
6	GDP-Exports_positive
7	GDP-ImportsOfGoodsAndServices_positive
8	GDP-GovernmentConsumptionExpenditure_positive
9	GDP-GrossFixedCapitalFormation_positive
10	IntProduction_positive
11	IntTrade-ExportInGoods_positive
12	IntTrade-ImportInGoods_positive
13	LaborCompensation-HourlyEarnings_positive
14	Sales-RetailTrade_positive
15	ServiceExports_positive
16	ServiceImport_positive
17	UnitLaborCost-BusinessSector_positive

Fig. 6 shows the historic and runtime processing of the abnormality indicator (mismatch), illustrating the largest abnormality in Q4 of 2008. This, of course, is the "Great Recession" event of 2008. Table 4 shows the abnormality sources and the missing events leading to the triggering of abnormality.

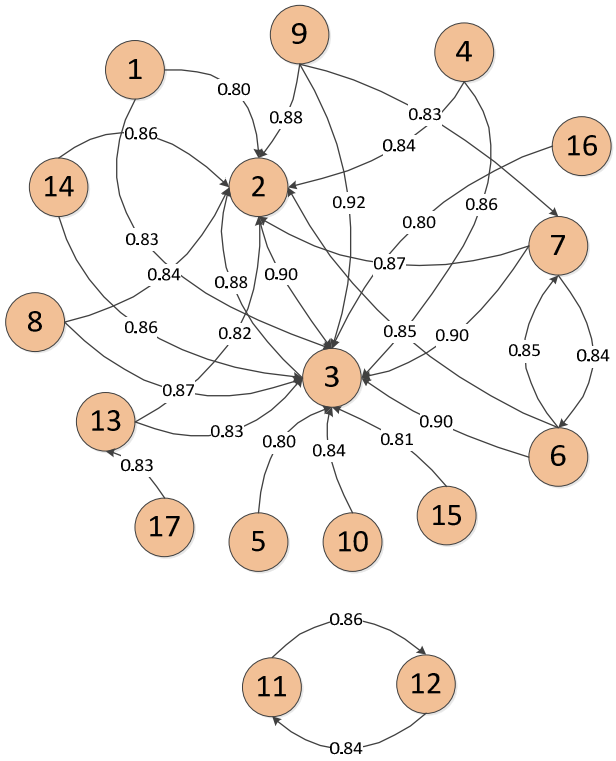


Fig. 5. DNG of economic data.

The mismatch score column in this table indicates the fraction with which each source (or country) contributed to the abnormality indication. What is interesting to note from this list is that Greece is in first place in terms of having had experienced the greatest deviation from its normal economic activity (which since then has led to many rounds of austerity measures). Note that the United States is 23rd on this list indicating that the 2008 event was not as impactful to the US as it was on other countries.

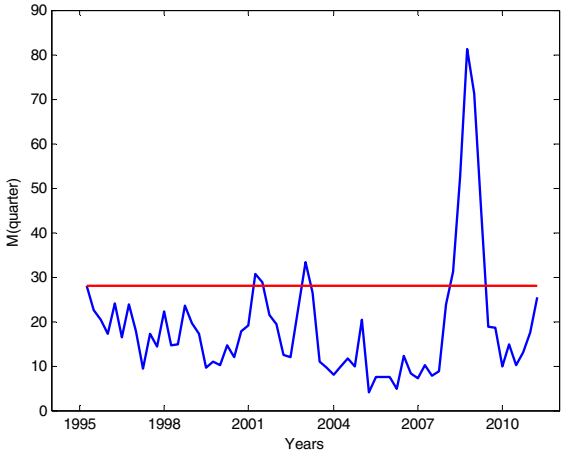


Fig. 6. Historic (1995-2007) and runtime (2008-2011) abnormalities of economic data.

Table 4. Abnormality sources for the peak at Q4-2008, $M(\Delta w) = 81.2\%$, see Fig. 6.

Event Source	GE ID	ME ID	MS
Greece	7,4,14,17,9	13,2,3,6	4.65
N. Zealand	13,1,4,8,17,16	2,3	4.01
Iceland	13,1,10,11,8	12,2,3	3.74
Australia	13,12,4,14,11,8,17	2,3	3.71
Japan	7,8,17,16	13,2,3,6	3.36
Poland	4,14,2,8,17,9	13,7,3	3.31
S. Africa	4,8,9	7,2,3	3.20
Israel	4,8,9	7,2,3	3.20
Austria	4,14,8,17	13,2,3	3.19
Germany	13,14,8,17	2,3	2.88
UK	13,14,8,17	2,3	2.88
Finland	13,4,8,17	2,3	2.79
Netherlands	13,4,8,17	2,3	2.79
Sweden	13,4,8,17	2,3	2.78
Mexico	13,4,8,17	2,3	2.78
Czech Rep.	4,8,17	13,2,3	2.26
S. Korea	4,8,17	13,2,3	2.26
France	13,4,2,8,17	3	1.99
Belgium	13,8,17	2,3	1.96
Canada	13,8,17	2,3	1.96
Denmark	13,8,17	2,3	1.96
Italy	13,8,17	2,3	1.96
US	13,8,17	2,3	1.96

VIII. Conclusions

We introduced a new decision making framework for information retrieval from data streams under special constraints on complexity and scalability. By a convolutional extraction of a representative meta-data from the stream we create the stream's "behavioral footprint" and use this in run-time evaluation of the underlying system on the current data flows. The meta-data can be stored in form of a graph based on an "event source"/"event type" principle. Application of our approach to IT data sets results in extension of the traditional normalcy analysis of time series data obtained from monitoring of IT infrastructures to include modern cloud infrastructures with high order of transiency. Furthermore, the method is applied to processing of log data generated by virtual centers. The new insight that we get here is that the dynamic environment produced by fleeting VMs can be projected via some probabilistic laws on behavioral event types. Matching this image (graph) with a run-time image one can come up with abnormality estimate of the transient infrastructure with additional capability to localize the underlying sources of misbehaviors. Due to universality of the proposed algorithm, the meta-data construction model and related analysis is applicable to data sets from various applications with similar anticipated results. This was demonstrated by applying the algorithm to economics data showing the 2008 "Great Recession" event.

Determining "event source" and "event type" would allow a "behavioral footprint" of the meta-data to be created which can then be used in real-time to determine brakeage of correlation as an abnormality indicator.

References

[1] T. White, *Hadoop: The Definitive Guide*, 1st Edition, O'Reilly Media, 2009.

[2] 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, no. 1, pp. 35-45, 2013.

[3] M.A. Marvasti, A.V. Grigoryan, A.V. Poghosyan, N.M. Grigoryan, and A.N. Harutyunyan, "Methods for the cyclical pattern determination using a clustering approach", US patent 20100036643.

[4] M.A. Marvasti, A.V. Poghosyan, A.N. Harutyunyan, and N.M. Grigoryan, "Pattern detection in unstructured data: An experience for a virtualized IT Infrastructure", *Proc. IFIP/IEEE Int. Symp. Integrated Network Management,* May 27-31, Ghent, Belgium, pp. 1048-1053, 2013.

[5] T. Y. Lin and D. P. Siewiorek "Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis", *IEEE Trans. Reliability*, vol. 39, no 4, Oct. 1990.

[6] J. P. Rouillard, "Real-time log file analysis using the simple event correlator (SEC)", *Proc. XVIII Large Installation System Administration Conf.*, Nov. 14-19, Atlanta, GA, 2004.

[7] W. Gaaloul and C. Godart, "Mining workflow recovery from event based logs", *Lecture Notes in Computer Science,* vol. 3649, pp. 169-185, 2005.

[8] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters", *Proc. 6th Symp. Operating System Design and Implementation*, San Francisco, CA, Dec. 6-8, pp. 137-149, 2004.

[9] D. Breitgand, M. Goldstein, E. Henis, and O. Shehory, "Efficient control of false negatives and false positive errors with separate adaptive thresholds", *IEEE Trans. Network and Service Management*, vol. 8, no. 2, pp. 128-140, June 2011.