A Novel Technique for Long-Term Anomaly Detection in the Cloud

Owen Vallis, Jordan Hochenbaum, Arun Kejariwal
Twitter Inc.

Abstract

High availability and performance of a web service is key, amongst other factors, to the overall user experience (which in turn directly impacts the bottom-line). Exogenic and/or endogenic factors often give rise to anomalies that make maintaining high availability and delivering high performance very challenging. Although there exists a large body of prior research in anomaly detection, existing techniques are not suitable for detecting long-term anomalies owing to a predominant underlying trend component in the time series data.

To this end, we developed a novel statistical technique to automatically detect long-term anomalies in cloud data. Specifically, the technique employs statistical learning to detect anomalies in both application as well as system metrics. Further, the technique uses robust statistical metrics, viz., median, and median absolute deviation (MAD), and piecewise approximation of the underlying long-term trend to accurately detect anomalies even in the presence of intra-day and/or weekly seasonality. We demonstrate the efficacy of the proposed technique using production data and report Precision, Recall, and F-measure measure. Multiple teams at Twitter are currently using the proposed technique on a daily basis.

1 Introduction

Cloud computing is increasingly becoming ubiquitous. In a recent report, IHS projected that, by 2017, enterprise spending on the cloud will amount to a projected $235.1 billion, triple the $78.2 billion in 2011 [3]. In order to maximally leverage the cloud, high availability and performance are of utmost importance. To this end, startups such as Netuitive [1] and Stackdriver [2] have sprung up recently to facilitate cloud infrastructure monitoring and analytics. At Twitter, one of the problems we face is how to automatically detect long-term anomalies in the cloud. Although there exists a large body of prior research in anomaly detection, we learned, based on experiments using production data, that existing techniques are not suitable for detecting long-term anomalies owing to a predominant underlying trend component in the time series data. To this end, we propose a novel technique for the same.

The main contributions of the paper are as follows:

- First, we propose a novel statistical learning based technique to detect anomalies in long-term cloud data. In particular,
  - We build upon generalized Extreme Studentized Deviate test (ESD) [13, 14] and employ time series decomposition and robust statistics for detecting anomalies.
  - We employ piecewise approximation of the underlying long-term trend to minimize the number of false positives.
  - We account for both intra-day and/or weekly seasonalities to minimize the number of false positives.

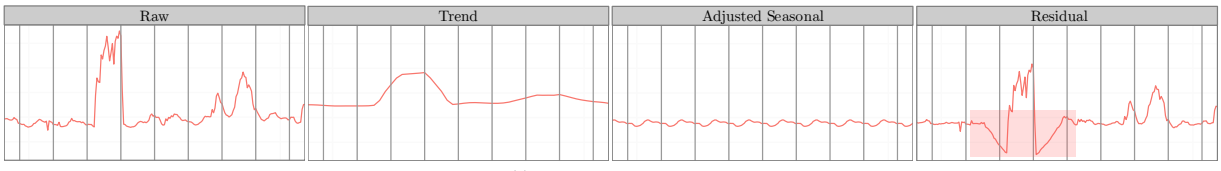
  The proposed technique can be used to automatically detect anomalies in time series data of both application metrics such as Tweets Per Sec (TPS) and system metrics such as CPU utilization etc.
- Second, we present a detailed evaluation of the proposed technique using production data. Specifically, the efficacy with respect to detecting anomalies and the run-time performance is presented.

  Given the velocity, volume, and real-time nature of cloud data, it is not practical to obtain time series data with "true" anomalies labeled. To address this limitation, we injected anomalies in a randomized fashion. We evaluated the efficacy of the proposed techniques with respect to detection of the injected anomalies.

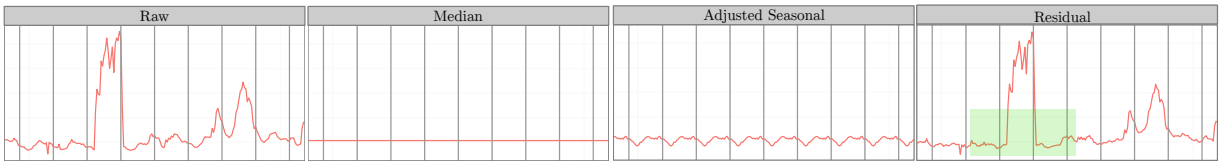
The remainder of the paper is organized as follows: Section 2 presents a brief background. Section 3 details the proposed technique for detecting anomalies in long-term cloud data. Section 4 presents an evaluation of the proposed technique. Lastly, conclusions and future work are presented in Section 5.

2 Background

In this section, we present a brief background for completeness and for better understanding of the rest of the paper. Let x_t denote the observation at time t, where $t = 0, 1, 2, \ldots$, and let X denote the set of all the observations constituting the time series. Time series decomposition is a technique that decomposes a time series (X) into three components, e.g., seasonal (S_X), trend (T_X), and residual (R_X). The seasonal component describes the periodic variation of the time series, whereas the trend component describes the "secular variation" of the time series, i.e., the long-term non-periodic variation. The residual component is defined as the remainder of the time series once the seasonality and trend have been removed, or formally, $R_X = X - S_X - T_X$. In the case of



(a) STL with Trend Removal



(b) STL with Median Removal

Figure 1: STL (a) vs. STL Variant (b) Decomposition

long-term anomaly detection, one must take care in determining the trend component; otherwise, the trend may introduce artificial anomalies into the time series. We discuss this issue in detail in Section 3.

It is well known that the sample mean $\bar{x}$ and standard deviation (elemental in anomaly detection tests such as ESD) are inherently sensitive to presence of anomalies in the input data [10, 9]. The distortion of the sample mean increases as $x_t \to \pm\infty$. The proposed technique uses robust statistics, such as the median, which is robust against such anomalies (the sample median can tolerate up to 50% of the data being anomalous [7, 6]). In addition, the proposed technique uses median absolute deviation (MAD), as opposed to standard deviation, as it too is robust in the presence anomalies in the input data [8, 10]. MAD is defined as the median of the absolute deviations from the sample median. Formally, $MAD = median_i(\left|X_i - median_j(X_j)\right|)$.

3 Proposed Technique

This section details the novel statistical learning-based technique to detect anomalies in long-term data in the cloud. The proposed technique is integrated in the Chiffchaff framework [12, 11] and is currently used by a large number of teams at Twitter to automatically detect anomalies in time series data of both application metrics such as Tweets Per Sec (TPS), and system metrics such as CPU utilization.

Twitter data exhibits both seasonality and an underlying trend that adversely affect, in the form of false positives, the efficacy of anomaly detection. In the rest of this section, we walk the reader through how we mitigate the above.

3.1 Underlying Trend

In Twitter production data, we observed that the underlying trend often becomes prominent if the time span of a time series is greater than two weeks. In such cases, the trend induces a change in mean from one (daily/weekly) cycle to another. This limits holistic detection of anomalies in long-term time series.

Time series decomposition facilitates filtering the trend from the raw data. However, deriving the trend, using either the Classical [15] or STL [4] time series decomposition algorithms, is highly susceptible to presence of anomalies in the input data and most likely introduce artificial anomalies in the residual component after decomposition – this is illustrated by the negative anomaly in the residual, highlighted by red box, in Figure 1 (a). Replacing the decomposed trend component with the median of the raw time series data mitigates the above. This eliminates the introduction of phantom anomalies mentioned above, as illustrated by the green box in Figure 1 (b). While the use of the median as a trend substitution works well where the observed trend component is relatively flat, we learned from our experiments that the above performs poorly in the case of long-term time series wherein the underlying trend is very pronounced. To this end, we explored two alternative approaches to extract the trend component of a long-term time series – (1) STL Trend; (2) Quantile Regression. Neither of the above two served the purpose in the current context. Thus, we developed a novel technique, called *Piecewise Median*, for the same. The following subsections walk the reader through our experience with using STL Trend, Quantile Regression, and detail *Piecewise Median*.

3.2 STL Trend

STL [4] removes an estimated trend component from the time series, and then splits the data into *sub-cycle series* defined as:

Definition 1 *A sub-cycle series comprises of values at each position of a seasonal cycle. For example, if the series is monthly with a yearly periodicity, then the first sub-cycle series comprised of the January values, the second sub-cycle series comprised of the February values, and so forth.*

LOESS smooths each sub-cycle series [5] in order to derive the seasonality. This use of sub-cycle series allows the decomposition to fit more complex functions than the classical additive or the multiplicative approaches. The

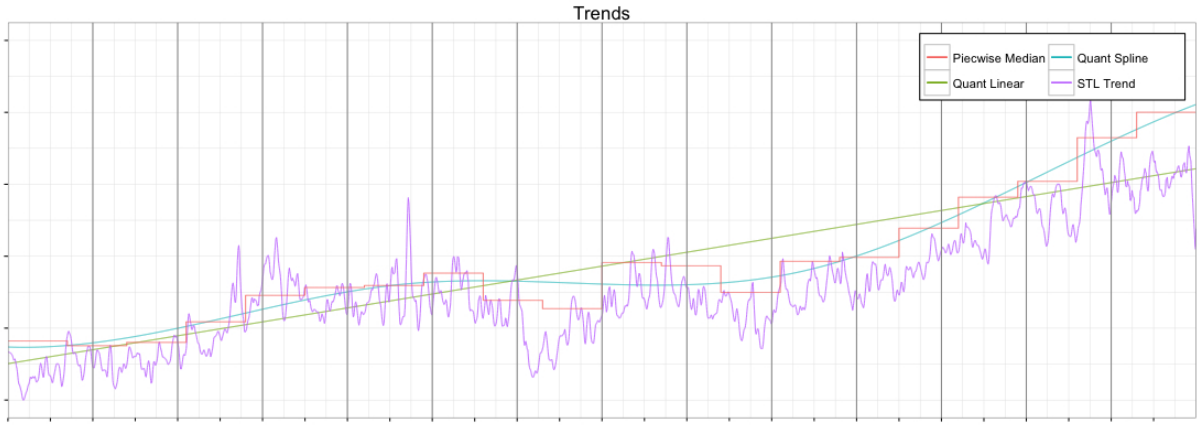


Figure 2: An illustration of the trends obtained using different approaches

estimated seasonal component is then subtracted from the original time series data; the remaining difference is smoothed using LOESS, resulting in the estimated trend component. STL repeats this process, using the most recent estimated trend, until it converges on the decomposition, or until the difference in iterations is smaller then some specified threshold. The purple line in Figure 2 exemplifies the trend obtained using STL for a data set obtained from production.

3.3 Quantile Regression

While least squares regression attempts to fit the mean of the response variable, quantile regression attempts to fit the median or other quantiles. This provides statistical robustness against the presence of anomalies in the data. Marrying this with a B-spline proved to be an effective method for estimating non-linear trends in Twitter production data that are longer then two weeks. This extracts the underlying trend well for long-term time series data. However, we observed that it would overfit the trend once the data was two weeks or less in length. This meant that large blocks of anomalies would distort the spline and yield a large number of both False Negatives (FN) and False Positives (FP). The aforementioned pitfall (meaning, overfitting) on two-week windows means that Quantile B-spline is not applicable in a piecewise manner, and as such, provided further motivation for a piecewise median approach. Additionally, from the figure we note that Quantile Linear, the line in green, also poorly fits the long-term trend.

3.4 Piecewise Median

In order to alleviate the limitations mentioned in previous subsections, we propose to approximate the underlying trend in a long-term time series using a piecewise method. Specifically, the trend computed as a piecewise combination of short-term medians. Based on experimentation using production data, we observed that stable metrics typically exhibit little change in the median over 2 week windows. These two week medians provide enough data points to decompose the seasonality, while also acting as a usable baseline from which to test for anomalies. The red line in Figure 2 exemplifies the trend obtained using the piecewise approach.

We now present a formal description the proposed technique. Algorithm 1 has two inputs: the time series X and maximum number of anomalies k.

Algorithm 1 Piecewise Median Anomaly Detection

1. Determine periodicity/seasonality
2. Split X into non-overlapping windows $W_X(t)$ containing at least 2 weeks

for all $W_X(t)$ **do**
  Require:
  n_W = number of observations in $W_X(t)$
  $k \leq (n_W \times .49)$
  3. Extract seasonal S_X component using STL
  4. Compute median $\tilde{X}$
  5. Compute residual $R_X = X - S_X - \tilde{X}$
  /* Run ESD / detect anomalies vector X_A with $\tilde{X}$ and MAD in the calculation of the test statistic */
  6. $X_A = ESD(R_X, k)$
  7. $v = v + X_A$
end for
return v

It is important to note that the length of the windows in the piecewise approach should be chosen such that the windows encompasses at least 2 periods of any larger seasonality (e.g., weekly seasonality owing to, for example, weekend effects). This is discussed further in Section 4.1

4 Evaluation

The use of different trend approximations yields different sets of long-term anomalies. First, we compare the efficacy of the proposed *Piecewise Median* approach with

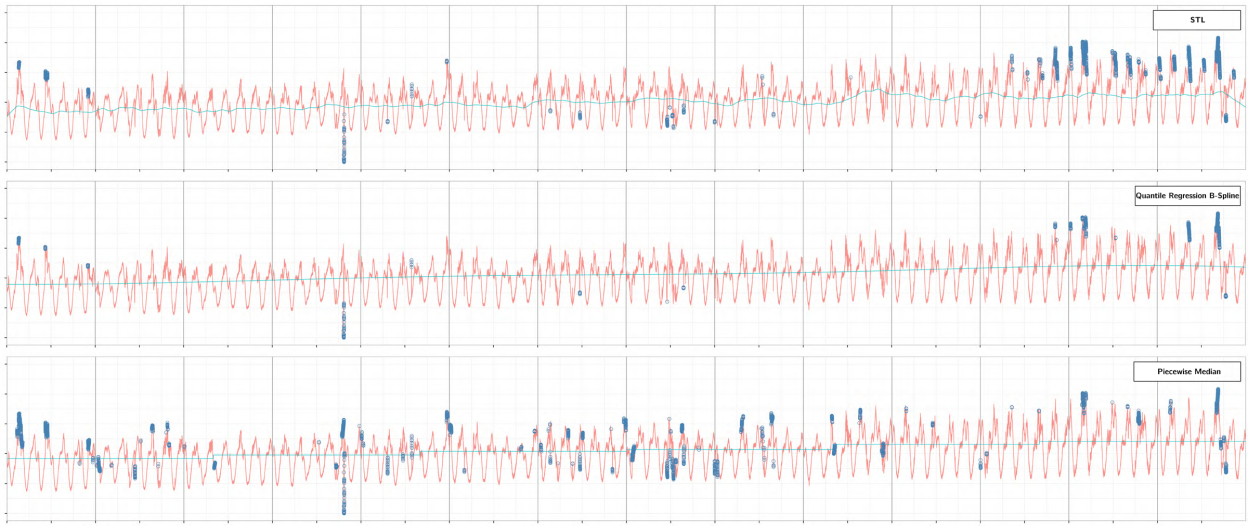


Figure 3: Anomalies found using STL (top), Quantile B-spline (middle), and Piecewise Median (bottom)

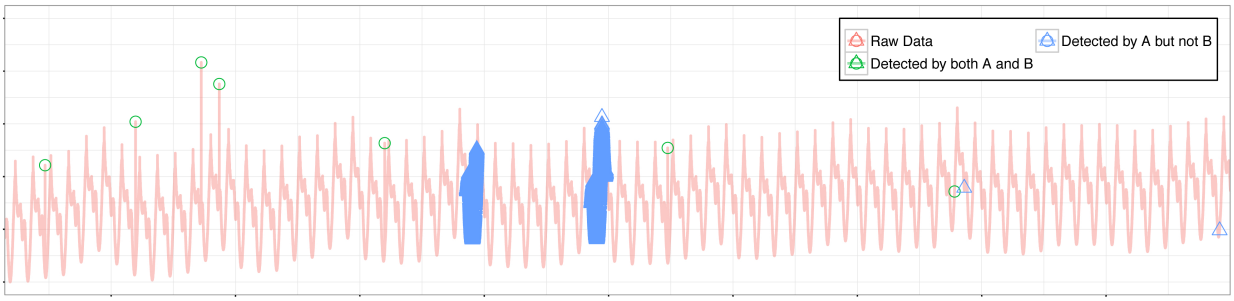


Figure 5: Intersection and Difference between anomalies detected using two-week piecewise windows containing partial weekend periods (A) and full weekend periods (B)

STL and Quantile Regression B-Spline. Second, we report the run-time performance of each. Lastly, we compare how the proposed technique fared against ground truth (via anomaly injection).

4.1 Efficacy

We evaluated the efficacy of the proposed *Piecewise Median* approach using production data. In the absence of classification labels, i.e., whether a data point is anomalous or not, we worked closely with the service teams at Twitter to assess false positives and false negatives. Figure 3 exemplifies how STL, Quantile Regression B-Spline, and *Piecewise Median* fare in the context of anomaly detection in quarterly data. The solid blue lines show the underlying trend derived by each approach. From the figure, we note that all the three methods are able to detect the most egregious anomalies; however, we observe that the overall anomaly sets detected by the three techniques are very different. Based on our conversation with the corresponding service teams, we learned that STL and Quantile Regression B-Spline yield many false positives. The Quantile Regression B-Spline appears to be more conservative, but misses anomalies in the intra-day troughs. By comparison, STL detects some of these intra-day anomalies, but it becomes overly aggressive towards the end of the time series, yielding many false positives. In contrast, *Piecewise Median* detects the majority of the intra-day anomalies, and has a much lower false positive rate than STL. These insights also mirror Figure 4, wherein we note that roughly 45% of the anomalies found by *Piecewise Median* are the same as STL and B-spline, with B-spline being the more conservative.

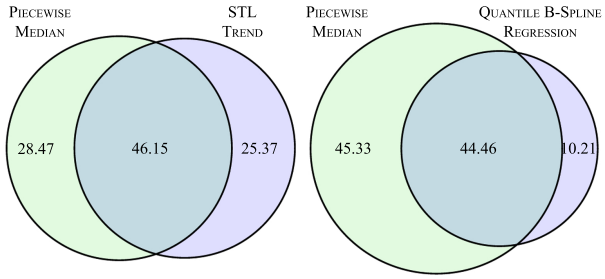


Figure 4: Intersection and set difference of anomalies found in STL and Quantile B-spline vs. Piecewise Median

In production data, we observed that there might be multiple seasonal components at play, for example daily and weekly (e.g., increased weekend usage). In light of this, the *Piecewise Median* assumes that each window

captures at least one period of each seasonal pattern. The implication of this is illustrated in Figure 5 which shows intersection and differences between the set of anomalies obtained using the *Piecewise Median* approach using the same data set but "phase-shifted" to contain partial (set A) and full (set B) weekend periods. From the figure we note that anomalies detected using set A contains all the anomalies detected using set B; however, the former had a high percentage of false positives owing to the fact that set A contained only the partial weekend period.

4.2 Run-time Performance

Real-time detection of anomalies is key to minimize impact on user experience. To this end, we evaluated the runtime performance of the three techniques. *Piecewise Median* took roughly four minutes to analyze three months of minutely production data, while STL and Quantile B-spline took over thirteen minutes. Table 1 summarizes the slowdown factors. From the table we note a slow down of $> 3\times$ which becomes prohibitive when analyzing annual data.

	Avg Minutes	Slowdown	Percentage Anomalies
STL Trend	13.48	3.19x	8.6%
Quantile B-Spline	13.62	3.22x	5.7%
Piecewise Median	4.23	1x	9.5%

Table 1: Run time performance

4.3 Injection based analysis

Given the velocity, volume, and real-time nature of cloud infrastructure data, it is not practical to obtain time series data with the "true" anomalies labeled. Consequently, we employed an injection-based approach to assess the efficacy of the proposed technique in a supervised fashion. We first smoothed production data using time series decomposition, resulting in a filtered approximation of the time series. Anomaly injection was then randomized along two dimensions – time of injection, and magnitude of the anomaly. Each injected data set was used to create nine different test sets (time series), with 30, 40, and 50% of the days in the time series injected with an anomaly at 1.5, 3, and 4.5σ (standard deviation). The 1σ value was derived from the smoothed times series data.

As with the actual production data, STL and Quantile B-Spline exhibit a $4\times$ slowdown (see Table 2). The faster

	Avg Minutes	Slowdown
STL Trend	21.59	4.49x
Quantile B-Spline	21.48	4.78x
Piecewise Median	4.49	1x

Table 2: Run time performance in case of anomaly injection

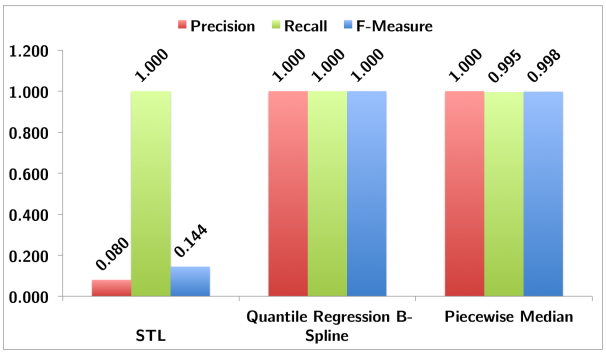


Figure 6: Precision, Recall, and F-Measure

run-time of Piecewise Median analyzes larger time series in a shorter period. This might prove useful where the detection of anomalies in historical production data can provide insight into a time sensitive issue. Additionally, from Figure 6 we note that Piecewise Median performs almost as well as Quantile B-spline, while STL has very low precision.

From the results presented in this section, it is evident that *Piecewise Median* is a robust way (has high F-measure) for detecting anomalies in long term cloud data. Further, *Piecewise Median* is $> 4\times$ faster!

5 Conclusion

We proposed a novel approach, which builds on ESD, for the detection of anomalies in long-term time series data. This approach requires the detection of the trend component, with this paper presenting three different methods. Using production data, we reported Precision, Recall, and F-measure, and demonstrated the efficacy of using Piecewise Median versus STL, and Quantile Regression B-Spline. Additionally, we reported a significantly faster run-time for the piecewise approach. In both instances (efficacy and run-time performance), the anomaly detection resulting from Piecewise Median trend performs as well, or better then, STL and Quantile Regression B-Spline. The technique is currently used on a daily basis. As future work, we plan to use the proposed approach to mitigate the affect of mean shifts in time series on anomaly detection.

References

[1] Netuitive. http://www.netuitive.com.

[2] Stackdriver. http://www.stackdriver.com/.

[3] Cloud- related spending by businesses to triple from 2011 to 2017, Feb. 2014.

[4] Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. STL: a seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics 6*, 1 (1990), 373.

[5] Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association 74*, 368 (1979), 829–836.

[6] Donoho, D. L., and Huber, P. J. The notion of breakdown point. *A Festschrift for Erich L. Lehmann* (1983), 157184.

[7] Hampel, F. R. *Contributions to the theory of robust estimation*. University of California, 1968.

[8] Hampel, F. R. The influence curve and its role in robust estimation. *Journal of the American Statistical Association 69*, 346 (1974), 383–393.

[9] Hampel, F. R., Ronchetti, E., Rousseeuw, P. J., and Stahel, W. A. *Robust statistics: the approach based on influence functions*. Wiley, New York, 1986.

[10] HUBER, P. J., AND RONCHETTI, E. *Robust statistics*. Wiley, Hoboken, N.J., 1981.
[11] KEJARIWAL, A., LEE, W., VALLIS, O., HOCHENBAUM, J., AND YAN, B. Visual analytics framework for cloud infrastructure data. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on* (Dec 2013), pp. 886–893.
[12] LEE, W., KEJARIWAL, A., AND YAN, B. Chiffchaff: Observability and analytics to achieve high availability. In *Large-Scale Data Analysis and Visualization (LDAV), 2013 IEEE Symposium on* (Oct 2013), pp. 119–120.
[13] ROSNER, B. On the detection of many outliers. *Technometrics 17*, 2 (1975), 221227.
[14] ROSNER, B. Percentage points for a generalized ESD many-outlier procedure. *Technometrics 25*, 2 (1983), 165172.
[15] STUART, A., KENDALL, M., AND ORD, J. K. *The advanced theory of statistics. Vol. 3: Design and analysis and time-series*. Griffin, 1983.