Leveraging Cloud Data to Mitigate User Experience from ‘Breaking Bad’

Nicholas A. James† Arun Kejariwal‡ David S. Matteson†
†Cornell University ‡Twitter Inc.

ABSTRACT

Low latency and high availability of an app or a web service are key, amongst other factors, to the overall user experience (which in turn directly impacts the bottomline). Exogenic and/or endogenic factors often give rise to breakouts in cloud data which makes maintaining high availability and delivering high performance very challenging. Although there exists a large body of prior research in breakout detection, existing techniques are not suitable for detecting breakouts in cloud data owing to being not robust in the presence of anomalies.

To this end, we developed a novel statistical technique to automatically detect breakouts in cloud data. In particular, the technique employs Energy Statistics to detect breakouts in both application as well as system metrics. Further, the technique uses robust statistical metrics, viz., median, and estimates the statistical significance of a breakout through a permutation test. To the best of our knowledge, this is the first work which addresses breakout detection in the presence of anomalies.

We demonstrate the efficacy of the proposed technique using production data and report Precision, Recall and F-measure measure. The proposed technique is 3.5x faster than a state-of-the-art technique for breakout detection and is being currently used on a daily basis at Twitter.

1. INTRODUCTION

In a recent report, Mary Meeker from KPCB mentioned that mobile usage continues to rise rapidly (14% Y/Y) and mobile usage now accounts for 25% of the total web usage [1]. In a similar vein, Strategy Analytics reported that mobile data traffic is expected to increase by 300% by 2017 to a peak of 21 Exabytes, from 5 Exabytes in 2012 [2]. Growing traffic and user engagement directly impacts the performance and availability of an app/website. To this end, KISSmetrics reported the following [3]:

- 73% of mobile internet users say that they have encountered a website that was too slow to load.
- 38% of mobile internet users say that they have encountered a website that was not available.
- A 1 second delay in page response can result in a 7% reduction in conversions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2014 ACM X-XXXXX-XX/XX/XX ...$5.00.

Figure 1: Example breakouts observed in production data at Twitter

Likewise, in [4], it was reported that performance has a direct impact on business KPIs (Key Performance Indicators). In [5], Shu and McElroy (now acquired by Hewlett-Packard) reported: If your mobile app fails, 48% of users are less likely to ever use the app again. 34% of users will just switch to a competitor’s app, and 31% of users will tell friends about their poor experience, which eliminates those friends as potential customers.

Amongst a large multitude of factors, breakouts – characterized by either a mean shift or a rampup from one steady state to another in a given time series (exemplified in Figure 1) – in system and/or application metrics can potentially impact performance and availability, thereby adversely impacting the end user experience. A wide variety of factors, some are enumerated below, can induce breakouts in system and/or application metrics.

(a) Continuous code deployment
(b) A/B testing [6, 7, 8]
(c) Launch of new products or new product features
(d) Partial failure of a cluster: Höelzle and Barroso point out that hardware failure in the cloud is more of a norm than exception [9] (also see [10, 11]).

Breakouts can potentially impact latency and availability experienced by the end user. In light of this, it is critical to detect breakouts early (robust breakout detection would also facilitate assessing the efficacy of an A/B test). Although there exists a large body of prior research in breakout detection, existing techniques are not suitable for detecting

1 Breakout, a term commonly used in finance, is referred to as changepoint in statistics.
The main contributions of the paper are as follows:

First, we propose a novel statistical technique, called E-Divisive with Medians (EDM), to automatically detect breakouts in cloud data. Unlike the existing techniques for breakout detection, EDM is robust against the presence of anomalies. To this end, we developed a novel technique to automatically detect breakouts in cloud data owing to not being robust in the presence of anomalies. To this end, we developed a novel technique to automatically detect breakouts in cloud data (which comprises of millions of time series at Twitter [12]).

Second, we present a detailed evaluation of EDM using production data.

- EDM employs E-statistics [14] to detect divergence in mean. Note that, in general, EDM can also be used to detect change in distribution in a given time series (discussed further in Section 3).
- EDM uses robust statistical metrics, viz., median [15] [16], and estimates the statistical significance of a breakout through a permutation test.
- EDM is non-parametric. This is of paramount importance as the cloud data does not follow the commonly assumed normal distribution, as illustrated by Figure 2 or any other widely accepted model. From the figure we note that none of the four segments in Figure 1 follow a common distribution.

To the best of our knowledge, this is the first work which addresses breakout detection in the presence of anomalies.

We also report Precision, Recall and F-measure to assess the efficacy of EDM.

The proposed technique is 3.5× faster than a state-of-the-art technique for breakout detection and is being currently used on a daily basis at Twitter.

The remainder of the paper is organized as follows: Section 2 presents a brief background. Section 3 details the proposed technique for detecting breakouts in cloud data with anomalies. Section 4 presents an evaluation of the proposed technique. Lastly, conclusions and future work are presented in Section 6.

2. BACKGROUND

In this section we present a brief background of the concepts used by EDM for detecting breakouts.

2.1 Divergence Measure

To detect breakouts, we employ a metric based on the weighted $L^2$-distance between the characteristic functions of random variables. Let $X$ and $Y$ be independent random variables, $X'$ be an i.i.d copy of $X$ and $Y'$ be an i.i.d. copy of $Y$. Let the cumulative distribution function of $X$ and $Y$ be denoted by $F$ and $G$ respectively.

**Definition 1.** The energy distance between $X$ and $Y$ is defined as follows [14]:

$$E(X,Y) = 2E|X - Y| - E|X - X'| - E|Y - Y'|$$

In [18], Rizzo and Székely showed that the $L^2$-distance between $F$ and $G$ satisfies the following:

$$2 \int_{-\infty}^{\infty} (F(x) - G(x))^2 dx = E(X,Y)$$

For a random variable $X$, its characteristic function $\phi_X(t)$ is defined by $\phi_X(t) = E(\exp(iXt))$. Using this notation, Székely and Rizzo [14] show that the energy distance between $X$ and $Y$ can also be represented in terms of their characteristic functions:

$$E(X,Y) = \int_{-\infty}^{\infty} |\phi_X(t) - \phi_Y(t)|^2 \frac{dt}{\pi t^2}$$

Since the characteristic function, like the cumulative distribution function, uniquely defines a random variable, we define a class of distance measures based on them. Let

$$D(X,Y; \alpha) = \int_{-\infty}^{\infty} |\phi_X(t) - \phi_Y(t)|^2 \omega(t;\alpha) dt$$

where $\omega(t;\alpha)$ is a weight function, parameterized by $\alpha$, such that $D(X,Y; \alpha) < \infty$. The indexing parameter $\alpha$ is used to scale the distance between distributions. For instance, the metric used in Equation 3 is obtained by using $\omega(t;\alpha) = \frac{1}{t^2}$. In [19], Székely and Rizzo suggested the following for $\omega$:

$$\omega(t;\alpha) = \frac{2\pi^3 \Gamma(1-\alpha/2)}{\alpha^2 \Gamma(1+\alpha/2) |t|^{\alpha+1}}$$

where $\Gamma(\cdot)$ is the complete gamma function. Using this weight function allows us to obtain a metric that generalizes...
the one in Equation \[4\] For \(\alpha \in (0, 2]\), the generalized energy distance between \(X\) and \(Y\) is given by:

\[
E(X, Y; \alpha) = 2E|X - Y|^\alpha - E|X - X'|^\alpha - E|Y - Y'|^\alpha
\]

Székely and Rizzo \[18\] also show that with this weight function, and \(\alpha \in (0, 2]\), we have

\[
D(X, Y; \alpha) = E(X, Y; \alpha).
\]

For detecting divergence in mean, \(\alpha\) is set to 2; on the other hand, for detecting arbitrary change in distribution, \(0 < \alpha < 2\) may be a better choice \[19\]. This property is exemplified through the following Lemma.

**Lemma 1.** For any pair of independent random variables \(X\) and \(Y\), and for any \(\alpha \in (0, 2]\), if \(E(|X|^\alpha + |Y|^\alpha) < \infty\), then \(E(X, Y; \alpha) \in [0, \infty)\), and \(E(X, Y; \alpha) = 0\) if and only if \(X\) and \(Y\) are identically distributed. Furthermore, if \(\alpha = 2\), we have that \(E(X, Y; 2) = 0\) if and only if \(EX = EY\).

**Proof.** A proof is given in \[19\]. \(\Box\)

The metric \(E\) allows for a simple and intuitive approximation to \(D\) and doesn’t require any integration. Let \(X_n = \{X_i: i = 1, \ldots, n\}\) and \(Y_m = \{Y_j: j = 1, \ldots, m\}\) be independent iid samples from the distribution of \(X, Y \in \mathbb{R}^d\), respectively, such that \(E|X|^\alpha, E|Y|^\alpha < \infty\) for some \(\alpha \in (0, 2]\). We can then approximate \(E\) by \(\hat{E}\) as follows:

\[
\hat{E}(X_n, Y_m; \alpha) = \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} |x_i - y_j|^\alpha - \frac{n}{2} \sum_{i<j} |x_i - x_j|^\alpha - \frac{m}{2} \sum_{i<j} |y_i - y_j|^\alpha
\]

The first term on the right hand side of Equation \[6\] correspond to the between distance between \(X_n\) and \(Y_m\). The second and third terms on the right side of Equation \[6\] correspond to the within distance of \(X_n\) and \(Y_m\) respectively \[19\].

By the strong law of large numbers for U-statistics \[20\],

\[
\hat{E} \rightarrow E \text{ as } \min(n, m) \rightarrow \infty.
\]

Furthermore, Székely and Rizzo \[18\] show that under the null hypothesis of equal distributions, i.e., \(E(X, Y; \alpha) = 0\),

\[
\frac{nm}{n+m} \hat{E}(X_n, Y_m; \alpha) \Rightarrow A
\]

as \(\min(n, m) \rightarrow \infty\), where \(A\) is a non-degenerate random variable and \(M \Rightarrow N\) means that \(M\) converges in distribution to \(N\). However, under the alternative hypothesis, \(\frac{nm}{n+m} \hat{E} \rightarrow \infty\) as \(\min(m, n) \rightarrow \infty\). For notational simplicity, we will use the following in the remainder of the paper:

\[
\hat{Q}(X_n, Y_m; \alpha) = \frac{nm}{n+m} \hat{E}(X_n, Y_m; \alpha)
\]

### 2.2 Permutation Test

The convergence of the statistic presented in Equation \[3\] allows us to determine the statistical significance of a proposed breakout. Let the observations of a time series be given by \(Z_1, Z_2, \ldots, Z_n\) and \(1 \leq \tau < \kappa \leq n\) be constants. We define the following sets \(A_{\tau} = \{Z_1, Z_2, \ldots, Z_{\tau}\}\) and \(B_\kappa(\kappa) = \{Z_{\tau+1}, \ldots, Z_\kappa\}\). A breakout location \(\hat{\tau}\) is then estimated as the value that maximizes

\[
\hat{Q}(A_{\tau}, B_\kappa(\kappa); \alpha)
\]

for \(1 \leq \tau < \kappa \leq n\). Along with the estimated breakout location we also have an associated test statistic

\[
\hat{\eta} = \hat{Q}(A_{\tau}, B_\kappa(\kappa); \alpha).
\]

Given \(\alpha = 2\), large values of \(\hat{\eta}\) correspond to a significant change in mean (and a distribution in general). However, calculating a precise critical value requires a knowledge of the underlying distributions, which are generally unknown. Therefore, we propose a permutation test to determine the significance of \(\hat{\eta}\).

Under the null hypothesis that there does not exist a breakout, we conduct a permutation test as follows. First, the observations are permuted to construct a new time series. Then, we re-apply the estimation procedure to the permuted observations. This process is repeated and after the \(r\)th permutation of the observations we record the value of the test statistic \(\hat{q}(r)\).

This permutation test will result in an exact p-value if we consider all possible permutations. However, this is not computationally tractable in general. Therefore, we obtain an approximate p-value by performing a sequence of \(R\) random permutations. The approximate p-value is computer as follows:

\[
\#\{r: \hat{q}(r) \geq \hat{\eta}\}/(R + 1)
\]

The re-sampling risk, the probability of a different decision than the one based on the theoretical p-value, can be uniformly bounded by an arbitrarily small constant using the approach proposed by Gandy \[21\]. In our analysis we test at the 5% significance level and use \(R = 199\) permutations.

### 2.3 Metrics

In order to minimize user impact, it is imperative to detect breakout(s) at the earliest. We qualify the timeliness of breakout detection via EDM using the metric TTD defined below:

**Definition 2.** We define TTD (Time to Detect) as the number of time series observations between the occurrence of a breakout and the breakout estimate reported by a breakout detection algorithm.

**Precision** is the ratio of true positives (tp) over the sum of true positives (tp) and false positives (fp). **Recall** is the ratio of true positives (tp) over the sum of true positives (tp) and false negatives (fn). **F-measure** is defined as follows (refer to \[22\] for a detailed discussion):

\[
F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

### 3. E-DIVISIVE WITH MEDIANS

Suppose that we are given the following time series, \(Z_1, Z_2, \ldots, Z_n\) consisting of independent observations. A breakout is characterized by a value \(\gamma \in (0, 1)\) such that observations
\{Z_1, Z_2, \ldots, Z_{\gamma N}\} have distribution function \(F\), and observations \{Z_{\gamma N+1}, Z_{\gamma N+2}, \ldots, Z_n\} have distribution function \(G\). Furthermore, it is assumed that \(F \neq G\). In order to determine if the observations in the provided time series are identically distributed we perform the following hypothesis test:

\[
H_0 : \gamma = 1 \\
H_A : 0 < \gamma < 1
\]

If the null hypothesis of no breakout is rejected, we must then also return an estimate for the breakout location. Prior work in breakout detection assumes that the time series under consideration is free of anomalies. However, this is not the case in production cloud data. Figure 3 illustrates the impact of the presence of anomalies on the location of a breakout detected. From the figure we note that there are multiple global anomalies, both positive and negative. The impact of the presence of anomalies on the location of a breakout detected. From the figure we note that there are multiple global anomalies, both positive and negative. The presence of anomalies since one single anomaly can greatly change its effect on the final result.

From a TTD perspective, we note that using the proposed algorithms we obtain estimates of the location of breakouts \(\hat{\delta}\). The approximation, \(\tilde{E}\) given in Equation 6 is susceptible to this comes at the expense of needing \(O(n^2 \log(n))\) time to calculate the initial value of our statistic. Since such updates may be done a large number of times, we consider this trade-off to be acceptable.

Although we can now quickly perform updates we will have to keep track of all \(O(n^2)\) distances. Even for moderately sized time series this may become intractable, even with 24GB of memory. For this reason we make use of interval trees (see the Appendix for further details) in order to obtain an approximate median. Through experimentation we learned that even the \(O(n \log(n))\) update is too slow, and thus we use the following approximation.

Let \(\delta > 1\). We approximate the within distance for the set \(X_n\) as follows:

\[
m_{XX}^{\alpha,\delta} = \text{median} \{|x_i - x_j\alpha : 1 \leq i < j \leq \delta\text{ or } i + 1 = j\}
\]

We similarly define \(m_{YY}^{\alpha,\delta}\). The between distance is approximated by using only \(\delta\) observations from each set. Figure 4 shows two possible ways of selecting the \(\delta\) observations.

**Head** Figure 4 (A) chooses to take the \(\delta\) observations that are at head of both sets \(X\) and \(Y\).

**Tail** Figure 4 (B) chooses to take the \(\delta\) observations at the tail of set \(X\) and the head of \(Y\).

Based on our experiments using production data we learned that using the Tail (as illustrated in Figure 4 (B)) yields better breakout estimates and hence, we use:

\[
m_{XX}^{\alpha,\delta} = \text{median} \{|x_i - y_j\alpha : n - \delta + 1 \leq i \leq n, 1 \leq j \leq \delta\}
\]

In light of the aforementioned approximation, Equation 6 can be written as:

\[
\tilde{Q}(X_n, Y_m; \alpha, \delta) = \frac{nm}{n + m} \tilde{E}(X_n, Y_m; \alpha, \delta) \tag{9}
\]

Figure 4: This figure depicts two different ways of selecting which \(\delta\) observations to use for approximating the between distance.
Typically $\delta$ is chosen such that it is much smaller than $\sqrt{n}$. Therefore, with these approximations we can create the statistic $	ilde{E}(X, \kappa; \alpha, \delta)$, which can be calculated in $O(n \log(n))$ time and updated in $O(\log(n))$ time when using the interval tree approximation.

### 3.2 Algorithm

The EDM algorithm makes use of the $\tilde{E}(\cdot; \alpha, \delta)$ statistic presented in the previous section. Let a time series be given by $Z_1, Z_2, \ldots, Z_n$ and $1 < \delta < \tau$ and $\tau + \delta \leq \kappa \leq n$. We define the following sets: $A_\tau = \{Z_1, Z_2, \ldots, Z_\tau\}$ and $B_\tau(\kappa) = \{Z_{\tau+1}, Z_{\tau+2}, \ldots, Z_\kappa\}$. Thus, both $A_\tau$ and $B_\tau(\kappa)$ have at least $\delta$ observations. Using Equation 9 we obtain the breakoust estimate, $\hat{\tau}$, as follows:

$$\hat{\tau} = \operatorname{argmax}_{\tau, \kappa} \tilde{Q}(A_\tau, B_\tau(\kappa); \alpha, \delta) \quad (10)$$

By solving the maximization problem given in Equation 10 we not only obtain an estimate $\hat{\tau}$, but also its associated test statistic value $\hat{q}$. Given this and a predetermined significance level, we perform a permutation test (detailed in subsection 2.2) to determine whether the reported breakoust is statistically significant.

Algorithm 1 is used to determine $\hat{\tau}$ and $\hat{\kappa}$. We set $D = 10$ in our implementation. However, we suggest selecting $D$ such that $D^2 \approx n$. Then, the algorithm makes use of two key procedures, $\text{ForwardUpdate}$ and $\text{BackwardUpdate}$.

**Parameters:** $Z$, $\delta$, and $D$

Let $T_A$, $T_B$, and $T_{AB}$ be interval trees with $2^D$ leaf nodes

// Initialize within distance trees
for $1 \leq i \leq \delta$
  for $1 \leq j \leq \delta$
    Insert $|Z_i - Z_j|$ to $T_A$
  Insert $|Z_{i+\delta} - Z_{j+\delta}|$ to $T_B$
end

// Initialize between distance tree
for $1 \leq i \leq \delta$
  Insert $|Z_i - Z_{i+\delta}|$ to $T_{AB}$
end

$(m1, m2, m3) = \text{approx. median } \langle T_{AB}, T_A, T_B \rangle$

$\text{bestStat} = \frac{\tau(\kappa - \tau)}{\kappa}(2m1 - m2 - m3)$

$\text{bestLoc} = \delta$

$\text{forwardMove} = 0$

// Update trees
while $\tau \leq n - \delta$
  if $\text{forwardMove} = 1$
    Perform ForwardUpdate
  else
    Perform BackwardUpdate
  end
  $\text{forwardMove} = 1 - \text{forwardMove}$
end

return $\text{bestLoc}$

**Algorithm 1: EDM**

These procedures allow us to efficiently update $\tilde{Q}$ by making use of the current states of the interval trees.

- $\text{ForwardUpdate}$ iterates $\kappa$ from $\tau + \delta + 1$ to $n$ and updates the value of $\tilde{Q}$ after each iteration. Each iteration corresponds to adding values to $B_\tau(\kappa)$.
- $\text{BackwardUpdate}$ iterates $\kappa$ from $n - 1$ to $\tau + \delta + 1$ and updates the value of $\tilde{Q}$ after each iteration. Each iteration corresponds to removing values from $B_\tau(\kappa)$.

- For both procedures $\text{ForwardUpdate}$ and $\text{BackwardUpdate}$, all the parameters are passed by reference. Additionally, both procedures obtain the an approximate medians in $O(D)$ (refer to the Appendix for details). In both cases, all the interval trees are updated. Hence, the statistic value can be computed in logarithmic time.

### 3.2.1 Special Case: $\alpha = 2$

It should be noted that when $\alpha = 2$, it is possible to obtain a much more efficient algorithm. In this case, $\bar{E}(X, Y; 2) = 2(\bar{E}X - \bar{E}Y)^2$; hence, changes in mean can be detected. As mentioned before, a robust location can be estimate by considering the sample median instead of the sample mean. In this case, we define $\bar{E}$ as follows:

$$\bar{E}(A_\tau, B_\tau(\kappa); 2, \delta) = 2[\text{median}(A_\tau) - \text{median}(B_\tau(\kappa))]^2$$

**Parameters:** $Z$, $\delta$, $T_A, T_B, T_{AB}$, $\tau$, $\text{bestStat}$, $\text{bestLoc}$

$n = Z.size()$

$\tau \leftarrow \tau + 1$

Update counts in $T_A, T_B$, and $T_{AB}$ resulting from new $\tau$ value

for $\tau + \delta \leq \kappa \leq n$
  Insert $|Z_n - Z_{\kappa-1}|$ to $T_B$
  $(m1, m2, m3) = \text{approx. median } \langle T_{AB}, T_A, T_B \rangle$
  $\text{stat} = \frac{\tau(\kappa - \tau)}{\kappa}(2m1 - m2 - m3)$
  if $\text{stat} > \text{bestStat}$
    $\text{bestStat} = \text{stat}$
    $\text{bestLoc} = \tau$
  end
end

**Procedure ForwardUpdate**

**Parameters:** $Z$, $\delta$, $T_A, T_B, T_{AB}$, $\tau$, $\text{bestStat}$, $\text{bestLoc}$

$n = Z.size()$

$\tau \leftarrow \tau + 1$

Update counts in $T_A, T_B$, and $T_{AB}$ resulting from new $\tau$ value

while $\kappa \geq \tau + \delta$
  Insert $|Z_n - Z_{\kappa-1}|$ to $T_B$
  $(m1, m2, m3) = \text{approx. median } \langle T_{AB}, T_A, T_B \rangle$
  $\text{stat} = \frac{\tau(\kappa - \tau)}{\kappa}(2m1 - m2 - m3)$
  if $\text{stat} > \text{bestStat}$
    $\text{bestStat} = \text{stat}$
    $\text{bestLoc} = \tau$
  end
  $\kappa \leftarrow \kappa - 1$
end

**Procedure BackwardUpdate**
4. EVALUATION

In this section we detail the evaluation methodology and present results demonstrating the efficacy, measured in terms of TTD (refer to subsection 2.3) of the algorithms presented in the previous section. Our experiments show that the presence of anomalies can significantly skew the TTD of breakout algorithm.

4.1 Methodology

The efficacy of EDM and EDM-X was evaluated using a wide corpus of time series data obtained from production. The time series corresponded to both system and application metrics. For example, but not limited to, the following metrics were used:

- System Metrics
  - CPU utilization, Heap usage, Disk writes

- Application Metrics
  - Latency

In addition to the time series of the metrics mentioned above, we also used minutely time series of the stock price of a publicly traded company. Overall, more than 20 data sets were used for evaluation. Given the velocity, volume, and real-time nature of cloud infrastructure data, it is not practical to obtain time series data with “true” breakouts labeled. However, to determine TTD, location of a “true” breakout is needed. To this end, for the data sets (obtained from production) we used for evaluation, we determined the “true” breakouts manually and then computed the TTD.

4.2 PELT and E-Divisive

Visual analysis serves as the starting point for deriving insights from Big Data [24,25,26]. With the increase in volume in Big Data, there has been increasing impetus being given to extreme scale visual analytics [27]. The May 2013 edition of IEEE Computer covered the challenges in the realm of Big Data visual analytics [25,29]. However, as mentioned earlier, due to the velocity and volume of cloud data, visual detection of breakouts is not practical. Furthermore, sometimes a breakout isn’t always obvious due to the range of the observed values. This is exemplified by Figure 7. From Figure 3b we note that there is an anomaly on the left hand side due to which even a 21% change in mean is
cannot be detected via visual inspection. However, on zooming in (in other words, limiting the range of the y-axis), see Figure 5b we observe the aforementioned breakout.

To this end, we first evaluated the PELT (Pruned Exact Linear Time) method by Killick and Haynes [30]. This is a parametric method that can be used to detect single as well as multiple breakout analysis. In the current context, we focus only on its properties for estimating a single breakout. This method is usually applied by using a log-likelihood function to measure fit, but as shown in [17] the underlying concepts can be extended to a number of different measure of fit. One the major benefits of this algorithm is its speed, which has been shown to have an expected linear running time.

We also evaluated the E-Divisive method [31]. This is a non-parametric breakout detection algorithm that is based upon the statistic presented in Equation 6. Akin to PELT, this method can also be used to estimate multiple breakouts, but we will once again only examine its performance at identifying a single breakout. However, unlike PELT, E-Divisive is a non-parametric algorithm and makes weak distributional assumptions. Hence, E-Divisive can be applied in a wider range of settings, such as those where one is not certain that PELT’s assumptions necessarily hold. On the other hand, E-Divisive has a quadratic running time, which is much slower than that of PELT.

4.2.1 Data Without Anomalies

First, we applied the PELT procedure to the datasets mentioned earlier in this section. Figure 6a exemplifies a case wherein the PELT method is efficient in detecting a breakout. This is further supported by the TTD values in column 3 of Table 1. However, since PELT makes distributional assumptions through its use of likelihood functions, PELT’s performance suffer - large TTD value - when these assumptions do not hold. This is illustrated by Figure 6b and column 3 of Table I

To address this problem, we used E-Divisive to compute breakout location. Figure 6c and column 2 of Table 1 show that in almost all cases E-Divisive results in a smaller TTD. Furthermore, since E-Divisive is a non-parametric method it can be applied to a wider array of settings, especially those where PELT’s assumptions are not guaranteed to hold. However, although E-Divisive is significantly slower than PELT we find this an acceptable trade off because of the decreased TTD and greater range of applications.

| Dataset | Raw Data TTD | Rolling Median TTD | Anomalies Removed TTD |
|---------|--------------|-------------------|----------------------|
| 1       | 0 0 0 0      | 71 74             |                      |
| 2       | 0 0 0 1 18 42|                   |                      |
| 3       | 2 1 1 1     |                   |                      |
| 4       | 0 6 38      |                   |                      |
| 5       | 0 65 65     |                   |                      |
| 6       | 2 5 2 2 5   |                   |                      |
| 7       | 6 7 7 1 7   |                   |                      |
| 8       | 3 4 2 4 3   |                   |                      |
| 9       | 9 8 6 8 15  |                   |                      |
| 10      | 14113 15 14114 16 14113 15  |                      |
| 11      | 0 0 4     |                   |                      |
| 12      | 0 1 1 1     |                   |                      |
| 13      | 45 2 45 2  |                   |                      |
| 14      | 0 1 2 0 1 0 |                   |                      |
| 15      | 1 1590 0 1590 |               |                      |
| 16      | 0 1 0 1     |                   |                      |
| 17      | 2 263 263 263 1681 1733 |                |                      |
| 18      | 1 0 2 0 2 0  |                   |                      |
| 19      | 1 61 61 105 108  |                 |                      |
| 20      | 4479 5607 4476 5607 4479 5607 |         |                      |
| 21      | 27 349 41 13  -  |                  |                      |
| 22      | 0 0 3 19 4 4  |                   |                      |
| 23      | 4 1 15 15  -  |                   |                      |
| 24      | 32 44 17 0 1 1 |                 |                      |
| 25      | 0 6 18 89 5 4  |                   |                      |

Table 1: TTD for the E-Divisive and PELT methods when applied to raw and rolling median time series

Figure 6: Efficacy of PELT and E-Divisive
4.2.2 Data with anomalies

In the previous section we showed that when a dataset doesn’t contain any anomalies that both PELT and E-Divisive can be used to compute robust estimates locations of a breakout. However, this is not the case in the presence of anomalies as illustrated by Figure 6c. A common approach to mitigate the effect of anomalies is local smoothing. The smoothers we considered were the rolling mean and rolling median. For these smoothers, each observation is replaced by either the mean or median of its neighboring values. As anomalies can still effect the smoothed values when calculating the rolling mean, we used the rolling median. Although these methods can reduce the impact of anomalies, it can result in an increased TTD as seen from columns 4 and 5 of Table 1. Another drawback to this approach is that one must choose the size of the neighborhood to use to calculate the smoothed values. A neighborhood that is too small will limit the mitigation of the effect of an anomaly; on the other hand, a neighborhood one that is too big can potentially smooth the mean changes (a breakout) in a time series.

Another approach is to remove anomalies before performing breakout analysis. To this end, we used the S-H-ESD algorithm to automatically detect anomalies. Subsequently, the anomalies were removed and breakout was detected using both PELT and E-Divisive – see columns 6 and 7 of Table 1. However, we do not consider this an ideal approach as anomaly and breakout detection are tightly intertwined. This stems from the fact that breakouts can cause normal observations to appear as anomalies, whereas anomalies can cause the data to appear to have a different mean. Unlike the local smoothing approach preemptive anomaly removal effects both E-Divisive and PELT. Both algorithms become less able to identify a change, as is expected because of the relationship between breakout and anomaly detection.

4.3 EDM

We next evaluated the efficacy of EDM. The TTD values for E-Divisive, EDM-X and EDM are reported in Table 2. Recall that EDM is designed to detect breakouts in an anomaly “aware” fashion. From the table we note that in most cases that TTD values are in the same ballpark as in the case of E-Divisive. In a couple of cases – Datasets 10 and 20 – both EDM-X and EDM outperform E-Divisive significantly, see Figures 7a and 7b. From Figure 7a we note that, unlike E-Divisive, EDM-X was able to detect the true location of the change in mean. This is due to fact that EDM-X was not susceptible to the anomalies at the left hand side of the time series. Likewise, from Figure 7b we note that EDM is robust against the anomalies on the right hand side of the true location of mean change; hence, EDM returned a very accurate estimate of the breakout.

Amongst EDM-Head and EDM-Tail, the latter seem to perform better in most cases. This is desirable from a recency perspective. Only in the case of Dataset 13 EDM-Tail performs significantly worse than E-Divisive.

The Precision, Recall and F-measure for both EDM-X and EDM is reported in Table 3. From the table we note EDM-X has a higher F-measure than EDM-Head and EDM-Tail for the data sets we used. The approximate p-values obtained using the permutation test (detailed in subsection 2.2) for each run are tabulated in Table 4. From the table we see that in some cases the p-value is higher than our threshold of 5%.

Based on our experimental results, we argue for the use of EDM when it is suspected that anomalies might be present in a given time series. In addition, the run time of EDM-X and EDM is much smaller to that of E-Divisive, see Table 5.
In our analysis, when performing the permutation test for EDM and EDM-X, the maximum number of permutations were always performed. However, the implementation of E-Divisive in the ecp package allows for early termination of the permutation test. Inspite of this, Figure 8 shows that EDM and EDM-X are at least 2× as fast as E-Divisive in almost all cases, and sometimes 6× faster.

Even though the EDM and EDM-X algorithms have been shown to be competitive with E-Divisive in the absence of anomalies, and better in the presence of anomalies, these methods do have their own limitations. For instance, see Figure 9. From the figure we note that EDM reports an inaccurate breakout estimate. This is attributed to the large number of anomalies as well as the fact that the anomalies are closely intertwined with the normal observations.

Another limitation of EDM and EDM-X is that they are both only able to detect a single breakout. Thus, if more than one breakout exists, it is unclear which (if any) will be found by EDM-X and EDM. Furthermore, depending on the size and nature of the breakouts, it is possible for performance to degrade, i.e., TTD may increase. This results from the fact that both EDM and EDM-X attempt to partition the time series into two homogeneous segments.

5. RELATED WORK

Breakout detection has been research in a wide variety of fields owing to the different applications. In this section we present a brief overview of prior work in breakout detection in statistics, finance, medicine and signal processing.

As mentioned earlier, breakout is referred to as a changepoint in statistics. Changepoint detection has been researched in statistics for over five decades [32, 33, 34, 35]. These come in two flavors: parametric and non-parametric. Many of the existing parametric methods assume that the underlying distribution belongs to the exponential family [35]. There has
been recent research in detecting changes with heavy tailed distributions [36]. Many of these approaches make use of limiting distributions obtained from Extreme Value Theory [37]. In cases where it is difficult or impossible to prove that the data adheres to parametric assumptions non-parametric approaches provide an alternative solution. These methods place less restrictive assumptions on the data and can thus be used more widely in general; however, due to the weaker assumptions, these methods are less powerful than their parametric counterparts [38]. Although most of the prior researched centered around detecting changes in mean, detecting changes in variance (with known/unknown mean value) has garnered some attention [39, 40, 41].

Tsay [42] presents an approach to detect changes in mean of an ARMA model in the presence of anomalies. Unlike EDM, the approach employs a two staged process that first removes the anomalies and then carry out breakout analysis. Another approach to handle anomalies during breakout detection is to assume that the data follows a heavy tailed distribution [43] and thus large values become less uncommon [44].

5.0.1 Parametric Analysis
The parametric algorithms used to perform breakout analysis assume that the observed distributions belong to a family of distributions \( \mathcal{F}_\theta = \{ F_\theta : \theta \in \Theta \} \), such that each member of the family can be uniquely identified by its parameter value. Once the class of distributions has been specified, parametric methods attempt to detect changes in the parameter value. Specifically, these approaches usually attempt to maximize a likelihood. For example, Carlin et al., [45], Lavielle and Teyssière [46] employ this approach. These papers however, assume a Gaussian distribution. An extension of this to select methods of the exponential family [47] is supported in the changepoint R package [48].

5.0.2 Non-parametric Analysis
A very common approach is to perform density estimation [49]. Although density estimation seems like a natural approach, other ideas have been shown to yield satisfactory results. For example, Lung-Yut-Fong et al. [49] perform analysis by working with rank statistics; Matteson and James [50] present an approach based upon Euclidean distances.

5.1 Finance
One of the more popular application areas of breakout detection is finance [50, 51, 52]. In this regard, models are regularly used to analyze return and stock price data. It is often assumed that the model parameters remain constant over the observed period. However, if the parameters are mostly time varying, the obtained results are likely to become out-of-date and consequently may not be robust [51]. Explicit examples of trading strategies that make use of breakout detection can be found in [51] which rely on historical analysis, charts and familiarity with the market.

The ARCH model of Engle [53] and its various generalizations are very often used to model the returns for a number of financial instruments. Franses and Ghijols [54] present a method for fitting GARCH models to financial data that may have additive outliers. In a similar vein, in [55], Matteson and James presented an approach that only requires a few mild statistical conditions to hold and doesn’t rely on any back testing. Regardless of the strategy, both works show that acknowledging the existence of breakouts can increase profits, or better yet, change would be losses into gains.

5.2 Medical Applications
Breakout detection also has applications in medicine. For example, Grigg et al. describe the use of the cumulative sum (CUSUM) chart, RSPRT (resetting sequential probability ratio test), and FIR (fast initial response) CUSUM to detect improvements in a process as well as detecting deterioration in a medical setting. In genetics, array comparative genomic hybridization is used to record DNA copy numbers. Changes in the DNA copy number can indicate a portion of a gene that may be effected by cancer or some other abnormal feature. Thus, detecting breakout in this setting can provide insights about future medical research.

Breakout analysis also finds application in segmentation of electroencephalogram (EEG). An EEG is a measure of the brain’s electrical activity which is recorded by electrodes on the subject’s scalp. EEGs can be used in the process of diagnosing disorders such as epilepsy and insomnia, since such disorders cause clear changes in the EEG readings. Breakout procedures have been suggested as a way to remove the human bias in the analysis of such data [58, 59]. Other application areas include studying breast cancer survival rates [60], analysis of MRI data [61], and many more [35].

5.3 Signal Processing
Breakout detection has been researched in the field of signal processing (and others such as, but not limited to, computer vision, image processing) but is usually referred to edge detection or jump detection [62, 63, 64, 65]. In [66], Basseville presented a survey of techniques to detect changes in signals and systems; Ziou and Tabbone present an overview of edge detection techniques in [67]. In the context of dynamic systems, Tugnait presented techniques to detect changes in [68].

In [69], Jackson et al. presented an algorithm for optimal partitioning of data on an interval. The algorithm was subsequently enhanced by Killick et al. [17] to detect breakouts with an expected linear running time.

6. CONCLUSIONS
In this paper, we proposed a novel statistical technique, called E-Divisive with Medians (EDM), to automatically detect breakouts in cloud data. Unlike the existing techniques for breakout detection, EDM is robust against the presence of anomalies. EDM employs E-statistics [14] to detect divergence in mean. Note that, in general, EDM can also be used detect change in distribution in a given time series. Further, EDM uses robust statistical metrics, viz., median, and estimates the statistical significance of a breakout through a permutation test. We used production data and to evaluate the efficacy of EDM and reported Precision, Recall and F-measure to demonstrate the same.

EDM is 3.5× faster than the state-of-the-art technique for breakout detection and is being currently used on a daily basis at Twitter.

As future work, we intend to extend EDM to support breakout detection in the presence of seasonality. Further, we plan to explore data transformation techniques to address the limitations mentioned in Section [4].
a half-open interval. Each internal node corresponds to the union of the intervals of its children. Thus, the root represents the interval $[0, 1]$. In this data structure each node will contain a count of the number of observations that lie within its interval.

Owing to the nature of the tree, one can find an approximate median in $O(D)$ time. One can find a value $m$, such that $K = \lceil \frac{n}{2} \rceil$ of our observations are less than or equal to $m$ in the following manner: Starting at the root node compare the value of its left child with $K$. If its value is larger than $K$, move to that node. On the other hand, if $K$ is larger, subtract the value of the left node from $K$ and move to the right child. This procedure is continued until a leaf node is reached; then, the midpoint of the leaf’s corresponding interval is returned. However, if at some point an internal node is reached who’s value is equal to $K$, the following is carried out: Let $a$ and $b$ be the values of the left and right children respectively, and $x, y$ the midpoints of their corresponding intervals. The following is returned:

$$\frac{1}{a+b} (a \times x + b \times y)$$

The major benefit of using an interval tree to obtain an approximate median instead of finding the true median is that the data structure can be updated efficiently and does not require sorting. Furthermore, from our experiments we have found that the relative difference between the true median and the approximation to be below 10%. Figure 10 illustrates how to update the tree as well as how to an approximate median.