# Automatic Anomaly Detection in the Cloud Via Statistical Learning

Jordan Hochenbaum    Owen S. Vallis    Arun Kejariwal
Twitter Inc.

## ABSTRACT

*Performance and high availability have become increasingly important drivers, amongst other drivers, for user retention in the context of web services such as social networks, and web search. Exogenic and/or endogenic factors often give rise to anomalies, making it very challenging to maintain high availability, while also delivering high performance. Given that service-oriented architectures (SOA) typically have a large number of services, with each service having a large set of metrics, automatic detection of anomalies is non-trivial. Although there exists a large body of prior research in anomaly detection, existing techniques are not applicable in the context of social network data, owing to the inherent seasonal and trend components in the time series data.*

*To this end, we developed two novel statistical techniques for automatically detecting anomalies in cloud infrastructure data. Specifically, the techniques employ statistical learning to detect anomalies in both application, and system metrics. Seasonal decomposition is employed to filter the trend and seasonal components of the time series, followed by the use of robust statistical metrics – median and median absolute deviation (MAD) – to accurately detect anomalies, even in the presence of seasonal spikes. We demonstrate the efficacy of the proposed techniques from three different perspectives, viz., capacity planning, user behavior, and supervised learning. In particular, we used production data for evaluation, and we report Precision, Recall, and F-measure in each case.*

## 1. INTRODUCTION

Big Data is characterized by the increasing volume (on the order of zetabytes), and the velocity of data generation [1, 2]. It is projected that the market size of Big Data shall climb up from the current market size of $5.1 billion [3] to $53.7 billion by 2017. In a recent report [4], EMC Corporation stated: *"A major factor behind the expansion of the digital universe is the growth of machine generated data, increasing from 11% of the digital universe in 2005 to over 40% in 2020."* In the context of social networks, analysis of User Big Data is key to building an engaging social network; in a similar fashion, analysis of Machine Big Data is key to building an efficient and performant underlying cloud computing platform.

Anomalies in Big Data can potentially result in losses to the business – in both revenue [5], as well as in long term reputation [6]. To this end, several enterprise-wide monitor-ing initiatives [7, 8] have been undertaken. Likewise, there has been an increasing emphasis on developing techniques for detection, and root cause analysis, of performance issues in the cloud [9, 10, 11, 12, 13, 14, 15].

A lot of research has been done in the context of anomaly detection in various domains such as, but not limited to, statistics, signal processing, finance, econometrics, manufacturing, and networking [16, 17, 18, 19]. In a recent survey paper Chandola et al. highlighted that anomalies are contextual in nature [20] and remarked the following:

> *A data instance might be a contextual anomaly in a given context, but an identical data instance (in terms of behavioral attributes) could be considered normal in a different context. This property is key in identifying contextual and behavioral attributes for a contextual anomaly detection technique.*

Detection of anomalies in the presence of seasonality, and an underlying trend – which are both characteristic of the time series data of social networks – is non-trivial. Figure 1 illustrates the presence of both positive and negative anomalies – corresponding to the circled data points – in time series data obtained from production. From the figure we note that the time series has a very conspicuous seasonality, and that there are multiple modes within a seasonal period. Existing techniques for anomaly detection (overviewed in-depth in Section 5) are not amenable for time series data with the aforementioned characteristics. To this end, we developed novel techniques for automated anomaly detection in the cloud via statistical learning. In particular, the main contributions of the paper are as follows:

❒ First, we propose novel statistical learning based techniques to detect anomalies in the cloud. The proposed techniques 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. Specifically, we propose the following:

  ▪ **Seasonal ESD (S-ESD)**: This techniques employs time series decomposition to determine the seasonal component of a given time series. *S-ESD* then applies ESD [21, 22] on the resulting time series to detect the anomalies.

  ▪ **Seasonal Hybrid ESD (S-H-ESD)**: In the case of some time series (obtained from production) we observed a relatively high percentage of anomalies. To address such cases, coupled with the fact that mean and standard deviation (used by ESD) are highly sensitive to a large number anomalies [23, 24], we extended *S-ESD* to use the robust statistics *median* [25] and median absolute deviation (MAD) to detect anomalies [26]. Compu-
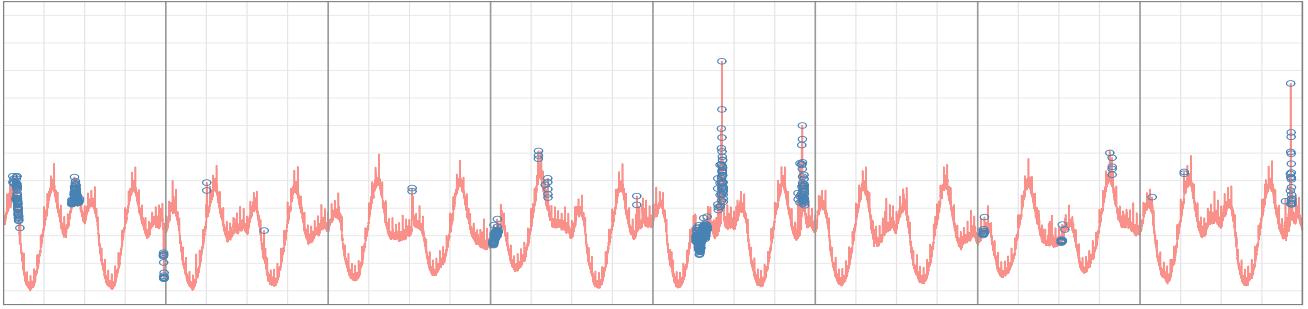
Figure 1: An illustrative time series, obtained from production, with positive and negative anomalies

tationally, *S-H-ESD* is more expensive than *S-ESD*, but is more robust to a higher percentage of anomalies.

❑ Second, we present a detailed evaluation of the proposed techniques using production data. In particular, we present the evaluation from three different perspectives:

▪ *Capacity planning*: Occurrence of anomalies can potentially result in violation of service level agreement (SLA), and/or impact end-user experience. To mitigate the impact of anomalies, timely detection of anomalies is paramount and may require provisioning additional capacity. Given a threshold for a system metric (corresponding to the SLA), we evaluate the efficacy of the proposed techniques to detect anomalies greater than the specified threshold.

▪ *User behavior*: Change in user behavior – which may result due to, but not limited to, events such as the Superbowl – at times manifests itself as anomalies in time series data. Thus, anomaly detection can guide the study of change in user behavior. To this end, we evaluated the efficacy of the proposed techniques with respect to change in user behavior.

▪ *Supervised Learning*: Given the velocity, volume, and real-time nature of cloud infrastructure 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 in smoothed, using B-spline, production data. The randomization was done along three dimensions – time of injection, magnitude, and width of the anomaly. We evaluated the efficacy of the proposed techniques with respect to detection of the injected anomalies.

The rest of the paper is organized as follows: Section 2 lays out the notation used in the rest of the paper, and briefly overviews statistical background for completeness and better understanding of the rest of the paper.

Section 3 details the techniques proposed in this paper, viz., **S-ESD** and **S-H-ESD**. Section 4 presents a detailed evaluation of the aforementioned techniques using production data.

Previous work is discussed in Section 5. Finally, in Section 6 we conclude with directions for future work.

## 2. BACKGROUND

In this section we describe the notation used in the rest of the paper. Additionally, we present a brief statistical background for completeness, and for better understanding of the rest of the paper.

A time series refers to a set of observations collected sequentially in time. Let $x_t$ denote the observation at time $t$, where $t = 0, 1, 2, \ldots$, and let $X$ denote the set of all observations constituting the time series.

### 2.1 Grubbs Test and ESD

In this subsection, we briefly overview the most widely used existing techniques for anomaly detection. In essence, these techniques employ statistical hypothesis testing, for a given significance level [27], to determine whether a datum is anomalous. In other words, the test is used to evaluate the rejection of the *null hypothesis* ($H_0$), for a pre-specified level of significance, in favor of the *alternative hypothesis* ($H_1$).

#### 2.1.1 Grubbs Test

Grubbs test [28, 29] was developed for detecting the largest anomaly within a univariate sample set. The test assumes that the the underlying data distribution is normal. Grubbs' test is defined for the hypothesis:

$$H_0 : \text{There are no outliers in the data set} \quad (1)$$
$$H_1 : \text{There is at least one outlier in the data set} \quad (2)$$

The Grubbs' test statistic is defined as follows:

$$C = \frac{\max_t \mid x_t - \overline{x} \mid}{s} \quad (3)$$

where, $\overline{x}$ and $s$ denote the mean and variance of the time series X. For the two-sided test, the hypothesis of no outliers is rejected at significance level $\alpha$ if

$$C > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{(t_{\alpha/(2N),N-2})^2}{N - 2 + (t_{\alpha/(2N),N-2})^2}} \quad (4)$$

where $t_{\alpha/(2N),N-2}$ denotes the upper critical values of the t-distribution with $N-2$ degrees of freedom and a significance level of $\alpha/(2N)$. For one-sided tests, $\alpha/(2N)$ becomes $\alpha/N$ [30]. The largest data point in the time series that is greater than the test statistic is labeled as an anomaly.

In practice, we observe that there is more than one anomaly in the time series data obtained from production. Conceivably, one can iteratively apply Grubbs' test to detect mul-

tiple anomalies. Removal of the largest anomaly at each iteration reduces the value of $N$; however, Grubbs' test does not update the value obtained from the t-distribution tables. Consequently, Grubbs' test is not suited for detecting multiple outliers in a given time series data.

Several other approaches, such as the Tietjen-Moore test[1] [31], and the extreme Studentized deviate (ESD) test [21, 22] have been proposed to address the aforementioned issue. Next, we briefly overview ESD.

### 2.1.2 Extreme Studentized Deviate (ESD)

The Extreme Studentized Deviate test (ESD) [21] (and its generalized version [22]) can also be used to detect multiple anomalies in the given time series. Unlike the Tietjen-Moore test, it only requires an upper bound on the number of anomalies ($k$) to be specified. In the worst case, the number of anomalies can be at most 49.9% of the total number of data points in the given time series. In practice, our observation, based on production data, has been that the number of anomalies is typically less than 1% in the context of application metrics and less than 5% in the context of system metrics.

ESD computes the following test statistic for the $k$ most extreme values in the data set.

$$C_k = \frac{max_k \mid x_k - \overline{x} \mid}{s} \qquad (5)$$

The test statistic is then compared with a critical value, computed using Equation 6, to determine whether a value is anomalous. If the value is indeed anomalous, it is removed from the data set, and the critical value is recalculated from the remaining data.

$$\lambda_k = \frac{(n-k)t_{p,n-k-1}}{\sqrt{(n-k-1+t_{p,n-k-1}^2)(n-k+1)}} \qquad (6)$$

ESD repeats this process $k$ times, with the number of anomalies equal to the largest $k$ such that $C_k > \lambda_k$. In practice, $C_k$ may swing above and below $\lambda_k$ multiple times before permanently becoming less then $\lambda_k$.

In the case of Grubbs' test, the above would cause the test to prematurely exit; however, ESD will continue until the test has run for $k$ outliers.

## 2.2 Median and Median Absolute Deviation

The techniques discussed in the previous subsection use mean and standard deviation. It is well known that these metrics are sensitive to anomalous data [25, 32]. As such, the use of the statistically robust median, and the median absolute deviation (MAD), has been proposed to address these issues.

The sample mean $\bar{x}$ can be distorted by a single anomaly, with the distortion increasing as $x_t \to \pm\infty$. By contradistinction, the sample median is robust against such distortions, and can tolerate up to 50% of the data being anomalous. Thus, the sample mean is said to have a *breakdown point* [33, 34] of 0, while the sample median is said to have a *breakdown point* of 0.5.

For a univariate data set $X_1, X_2, ..., X_n$, MAD is defined as the median of the absolute deviations from the sample median. Formally,

---

[1]Although the Tietjen-Moore test can be used to detect multiple anomalies, it requires the number of anomalies to detect to be pre-specified. This is not practical in the current context.

$$MAD = median_i(|X_i - median_j(X_j)|) \qquad (7)$$

Unlike standard deviation, MAD is robust against anomalies in the input data [26, 25]. Furthermore, MAD can be used to estimate standard deviation by scaling MAD by a constant factor $b$.

$$\hat{\sigma} = b \cdot MAD \qquad (8)$$

where $b = 1.4826$ is used for normally distributed data (irrespective of non-normality introduced by outliers) [35]. When another underlying distribution is assumed, Leyes et al. suggest $b = \frac{1}{Q(0.75)}$, where Q(0.75) is the 0.75 quantile of the underlying distribution [23].

## 2.3 Precision, Recall, and F-Measure

As mentioned earlier in Section 1, the efficacy of the proposed techniques were evaluated from three different perspectives. In each case we report the following metrics – *Precision, Recall*, and *F-measure* (these metrics are commonly used to report the efficacy of an algorithm in data mining, information retrieval, et cetera).

In the context of anomaly detection, *Precision* is defined as follows:

$$\text{Precision} = \frac{|\{S\} \cap \{G\}|}{|\{S\}|} = \frac{tp}{tp + fp} \qquad (9)$$

where $S$ is the set of detected anomalies, and $G$ is the set of ground-truth anomalies. In other words, *Precision* is the ratio of true positives (tp) over the sum of true positives (tp) and false positives (fp) that have actually been detected.

In the context of anomaly detection, *Recall* is defined as follows:

$$\text{Recall} = \frac{|\{S\} \cap \{G\}|}{|\{G\}|} = \frac{tp}{tp + fn} \qquad (10)$$

where *fn* denotes false negatives. Lastly, *F-measure* is defined as follows:

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \qquad (11)$$

Since Precision and Recall are weighted equally in Equation 11, the measure is also referred as the $F_1$-measure, or the balanced F-score. Given that there is a trade-off between *Precision* and *Recall*, the *F-measure* is often generalized as follows:

$$F_\beta = (1 + \beta^2) \times \frac{\text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}} \quad \text{where } \beta \geq 0$$

If $\beta > 1$, F is said to become more recall-oriented and if $\beta < 1$, F is said to become more precision-oriented [36].

## 3. TECHNIQUES

In this section we detail our approaches **Seasonal ESD (S-ESD)** and **Seasonal Hybrid ESD (S-H-ESD)**. Note that both the approaches are currently deployed to automatically detect anomalies in production data on a daily basis. We employed an incremental approach in developing the two approaches. In particular, we started with evaluating the "rule of thumb" (the *Three Sigma Rule*) using production
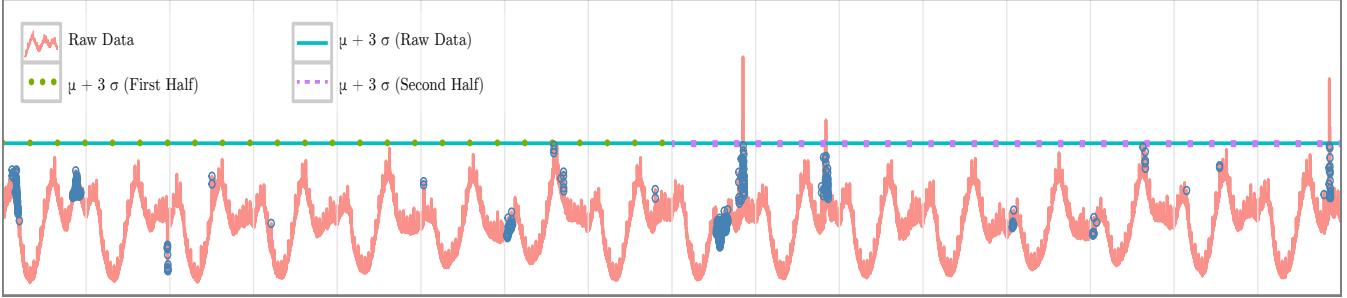
Figure 2: Application of $3 \cdot \sigma$ to the time series corresponding to Figure 1

data and progressively learned the limitations of using the existing techniques for automatically detecting anomalies in production data. In the rest of this section we walk the reader through the core steps of our aforementioned incremental approach.

## 3.1 Three-Sigma Rule

As a first cut, the $3 \cdot \sigma$ "rule" is commonly used to detect anomalies in a given data set. Specifically, data points with values more than 3 times the sample standard deviation are deemed anomalous. The rule can potentially be used for capturing large global anomalies, but is ill-suited for detecting seasonal anomalies. This is exemplified by Figure 2, wherein the seasonal anomalies that could not be captured using the $3 \cdot \sigma$ "rule" are annotated with circles.

Conceivably, one can potentially segment the input time series into multiple windows and apply the aforementioned rule on each window using their respective $\sigma$. The intuition behind segmentation being that the input time series is non-stationary and consequently $\sigma$ varies over time. The variation in $\sigma$ for different window lengths is shown in Figure 3.
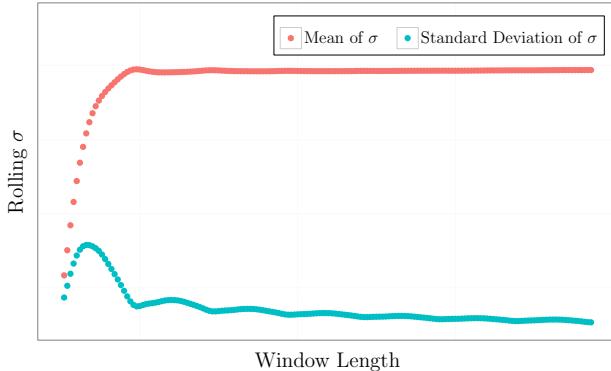


Figure 3: Distribution of $\sigma$ for different window lengths

From Figure 2 we note that applying the rule on a per-window basis – the time series was segmented into two windows – does not facilitate capturing of the seasonal anomalies.

The $3 \cdot \sigma$ "rule" assumes that the underlying data distribution is normal. However, our experience has been that the time series data obtained from production is seldom, if at all, normal (see Figure 4 for an example).

In light of the aforementioned limitations, we find that the $3 \cdot \sigma$ "rule" is not applicable in the current context. Next, we explored the efficacy of using moving average for anomaly detection.
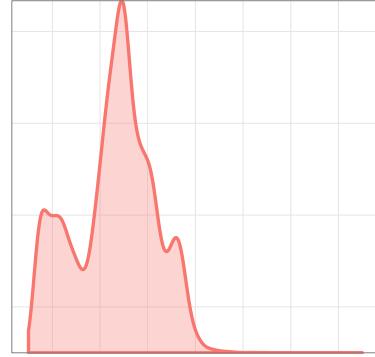


Figure 4: Data distribution of the time series corresponding to Figure 1

## 3.2 Moving Averages

One of the key aspects associated with anomaly detection is to mitigate the impact of the presence of white noise. To this end, the use of moving averages has been proposed to filter (/smooth) out white noise.

The most common moving average is the *simple moving average* (SMA), which is defined as:

$$\text{SMA}_t = \frac{x_t + x_{t-1} + \ldots + x_{t-(n-1)}}{n} \tag{12}$$

Note that SMA weighs each of the previous $n$ data points equally. This may not desirable given the dynamic nature of the data stream observed in production and from a recency perspective. To this end, the use of the *exponentially weighted moving average* (EWMA), defined by Equation 13, has been proposed [37].

$$\text{EWMA}_T \equiv \begin{cases} y_t = x_t, & t = 1 \\ y_t = \alpha(x_t) + (1 - \alpha)y_{t-1}, & t > 1 \end{cases} \tag{13}$$

In [38], Carter and Streilein argue that in the context of streaming time series data EWMA can potentially be "volatile to abrupt transient changes, losing utility for appropriately detecting anomalies". To address this, they proposed the *Probabilistic Exponentially Weighted Moving Average* (PEWMA), wherein the weighting parameter $\alpha$ is adapted by $(1 - \beta P_t)$, $P_t$ is the probability of $x_t$ evaluated under some modeled distribution, and $\beta$ is the weight placed on $P_t$. The SMA, EWMA and PEWMA for the time series of Figure 1 is shown in Figure 5a. From the figure we note that the SMA and EWMA do not fare well with respect to capturing the daily seasonality. Although PEWMA traces the raw time series

(a) SMA, EWMA and PEWMA for the time series of Figure 1



(b) Scatter plot of "true" anomalies and anomalies detected by applying ESD on PEWMA
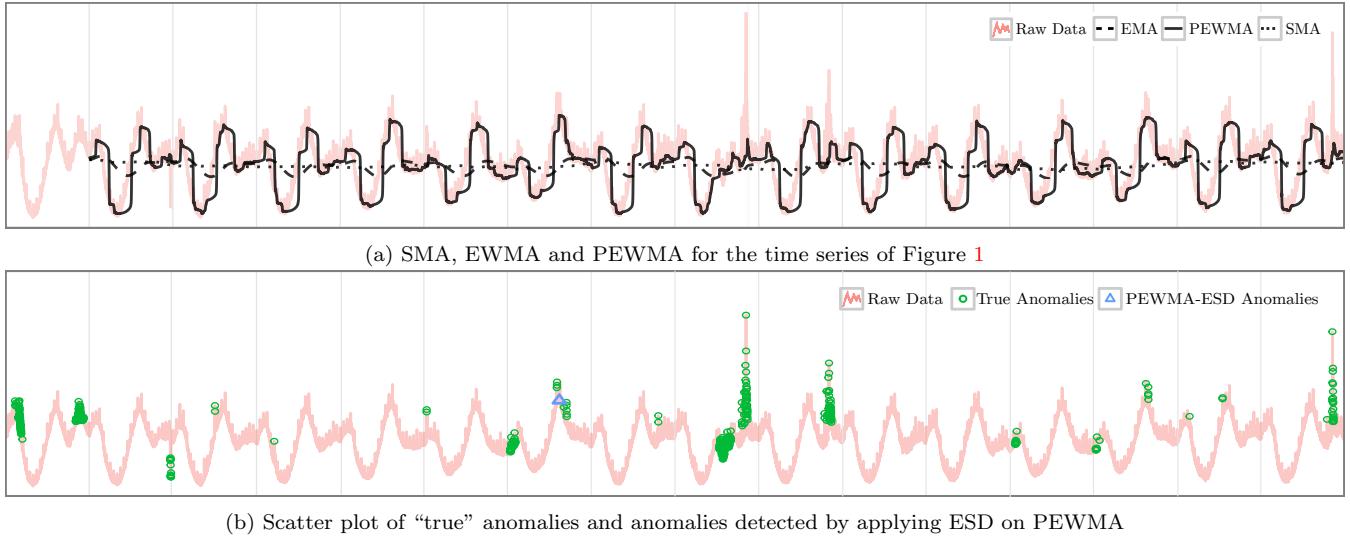
Figure 5: Illustration of limitations of using moving averages for anomaly detection

better (than SMA and EWMA), it fails to capture both global as well as seasonal anomalies.

We evaluated the efficacy of using SMA, EWMA, and PEWMA as an input for anomaly detection. Our experience, using production data, was that the respective moving averages filter out most of the seasonal anomalies and consequently are ill-suited for the current context. This is illustrated by Figure 5b from which we note that application of ESD on PEWMA fails to detect "true" anomalies (annotated on the raw time series with circles).

Further, given that a moving average is a lagging indicator, it is not suited for real time anomaly detection.

Arguably, one can use shorter window length, when computing a moving average, to better trace the input time series. However, as shown in [39], the standard error of $\sigma$ ($\propto \frac{1}{\sqrt{n-1}}$), increases as the window length decreases. This is illustrated in Figure 3 (window length decreases from right to left).

## 3.3 Seasonality and STL

As discussed in the previous section, Twitter data (obtained from production) exhibits heavy seasonality. Further, in most cases, the underlying distribution exhibits a multi-modal distribution, as exemplified by Figure 4. This limits the applicability of existing anomaly detection techniques such as Grubbs and ESD as the existing techniques assume a normal data distribution. Specifically, we learned based on data analysis that the presence of multiple modes yields a higher value of standard deviation (the increase can be as much as up to 5%) which in turn leads to masking of detection of some of the "true" anomalies.

To this end, we employ time series decomposition wherein a given time series $(X)$ is decomposed into three – seasonal $(S_X)$, trend $(T_X)$, and residual $(R_X)$ – components. The residual has a unimodal distribution that is amenable to the application of anomaly detection techniques such as ESD. Before moving on, let us first define *sub-cycle series* [40]:

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 is the January values, the second is the February values, and so forth.*

Time series decomposition approaches can be either additive or multiplicative [41], with additive decomposition being appropriate for seasonal data that has a constant magnitude as is the case in the current context. The algorithm first derives $T_X$ using a moving average filter, and subsequently subtracts it from the $X$. Once the trend is removed, $S_X$ is then estimated as the average of the data points within the corresponding sub-cycle series. Note that the number of data points per period is an input to the algorithm in order to create the sub-cycle series. In the current context, we set the number of data points per period to be a function of the data granularity. The residual $R_X$ is estimated as follows:

$$R_X = X - T_X - S_X \tag{14}$$

The residual component computed above can be potentially corrupted by extreme anomalies in the input data.

This issue is addressed by STL [40], a robust approach to decomposition that uses LOESS[42] to estimate the seasonal component. The algorithm consists of an inner loop that derives the trend, seasonal, and residual components and an outer loop that increases the robustness of the algorithm with respect to anomalies. Akin to classical seasonal decomposition, the inner loop derives the trend component using a moving average filter, removes the trend from the data and then smoothes the sub-cycle series to derive the seasonal component; however, STL uses LOESS to derive the seasonality, allowing the decomposition to fit more complex functions than either the additive or multiplicative approaches. Additionally, STL iteratively converges on the decomposition, repeating the process several times, or until the difference in iterations is smaller then some specified threshold ($\epsilon$). Optionally, STL can be made more robust against the influence of anomalies by weighting data points in an outer loop. The outer loop uses the decomposed components to assign a robustness weight to each datum as:

$$Weight_t = B\left(\frac{|R_{X_t}|}{6 \times median|R_{X_t}|}\right)$$

where B is the bisquare function defined as:

$$B(u) \equiv \begin{cases} (1-u^2)^2 & for \quad u \le 0 < 1 \\ 0 & for \quad u > 1 \end{cases}$$

(a) STL with Trend Removal
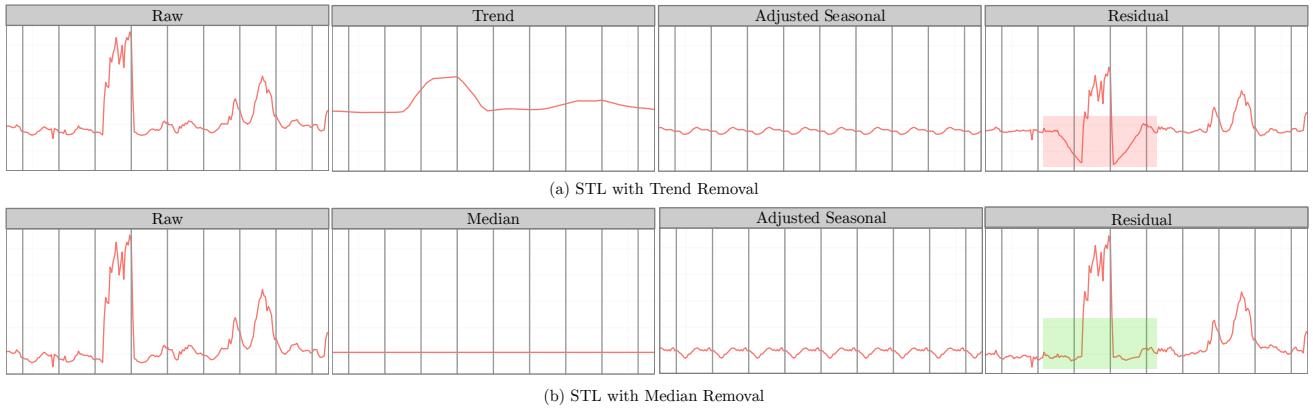


(b) STL with Median Removal

Figure 6: STL (a) vs. STL Variant (b) Decomposition

These weights are then used to converge closer to the "true" decomposed components in the next iteration of the inner loop.

## 3.4 Seasonal ESD (S-ESD)

To recap, the applicability of the existing techniques (overviewed in Section 2) is limited by the following:

- Presence of seasonality in the Twitter time series data (exemplified by Figure 1).

- Multimodal data distribution of Twitter time series data (illustrated by Figure 4).

To this end, we propose a novel algorithm, referred to as *Seasonal-ESD* (S-ESD), to automatically detect anomalies in Twitter's production data. The algorithm uses a modified STL decomposition (discussed in subsubsection 3.4.1) to extract the residual component of the input time series and then applies ESD to detect anomalies.

This two step process allows S-ESD to detect <u>both</u> global anomalies that extend beyond the expected seasonal minimum and maximum and local anomalies that would otherwise be masked by the seasonality.

A formal description of the algorithm is presented in Algorithm 1.

In the rest of this subsection, we detail the modified STL algorithm, elaborate S-ESD's ability to detect global and local anomalies, as well as the limitations of the algorithm.

Section 4 presents the efficacy of S-ESD using production data – both core drivers (or business metrics) and system metrics.

### 3.4.1 STL Variant

Applying STL decomposition to time series of system metrics yielded, in some cases, spurious anomalies (i.e., anomalies not present in the original time series) in the residual component. For example, let us consider the time series shown in Figure 6a wherein we observe a region of continuous anomalies in the raw data. On applying STL decomposition, we observed a breakout (ala a pulse) in the trend component. On deriving the residual component using Equation 14, we observed an inverse breakout, highlighted with a red rectangle in Figure 6a, which in turn yielded spurious anomalies.

To address the above, we use the median of the time series to represent the "stable" trend value which is in turn used to compute the residual component as follows:

---

**Algorithm 1** S-ESD Algorithm

**Input**:

$X = $ A time series

$n = $ number of observations in $X$

$k = $ max anomalies (iterations in ESD)

**Output:**

$X_A = $ An anomaly vector wherein each element is a tuple $(timestamp, observed\ value)$

**Require:**

$k \leq (n \times .49)$

1. Extract seasonal component $S_X$ using STL Variant
2. Compute median $\tilde{X}$

/* Compute residual */

3. $R_X = X - S_X - \tilde{X}$

/* Detect anomalies vector $X_A$ using ESD */

4. $X_A = \text{ESD}(R, k)$

**return** $X_A$

---

$$R_X = X - S_X - \tilde{X} \qquad (15)$$

where $X$ is the raw time series, $S_X$ is the seasonal component as determined by STL, and $\tilde{X}$ is the median of the raw time series. Replacing the trend with the median eliminates the spurious anomalies in the residual component as exemplified by Figure 6b. From the figure we note that the region highlighted with a green rectangle does not have any spurious anomalies, unlike the corresponding region in Figure 6a.

### 3.4.2 Global and Local Anomalies

Unlike the techniques overviewed in Section 2, S-ESD can detect local anomalies that would otherwise be masked by seasonal data. These local anomalies are bound between the seasonal minimum and maximum and may not not appear to be anomalous from a global perspective. However, they are indeed anomalous and it is important to detect these as they represent a deviation from the historical pattern. For instance, local anomalies found in Twitter data
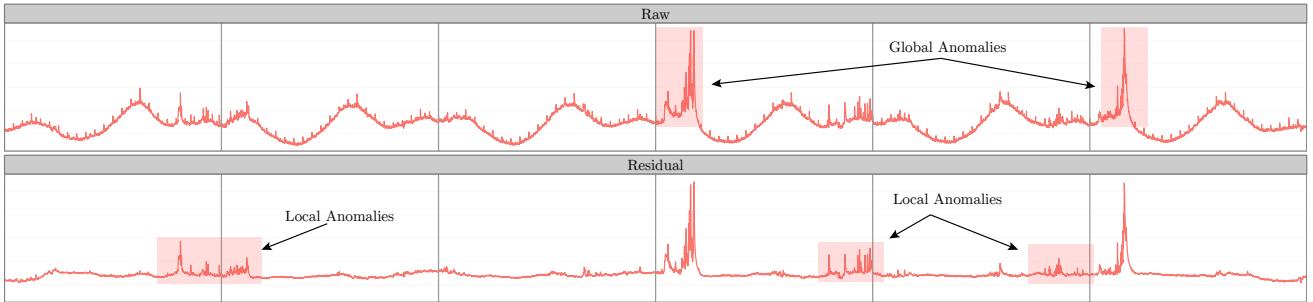
Figure 7: Global and Local Anomalies Exposed using S-ESD



(a) Anomalies detected via S-ESD: 1.11% Anomalies ($\alpha$ =0.05)



(b) Anomalies detected via S-H-ESD: 29.68% Anomalies ($\alpha$ =0.05)
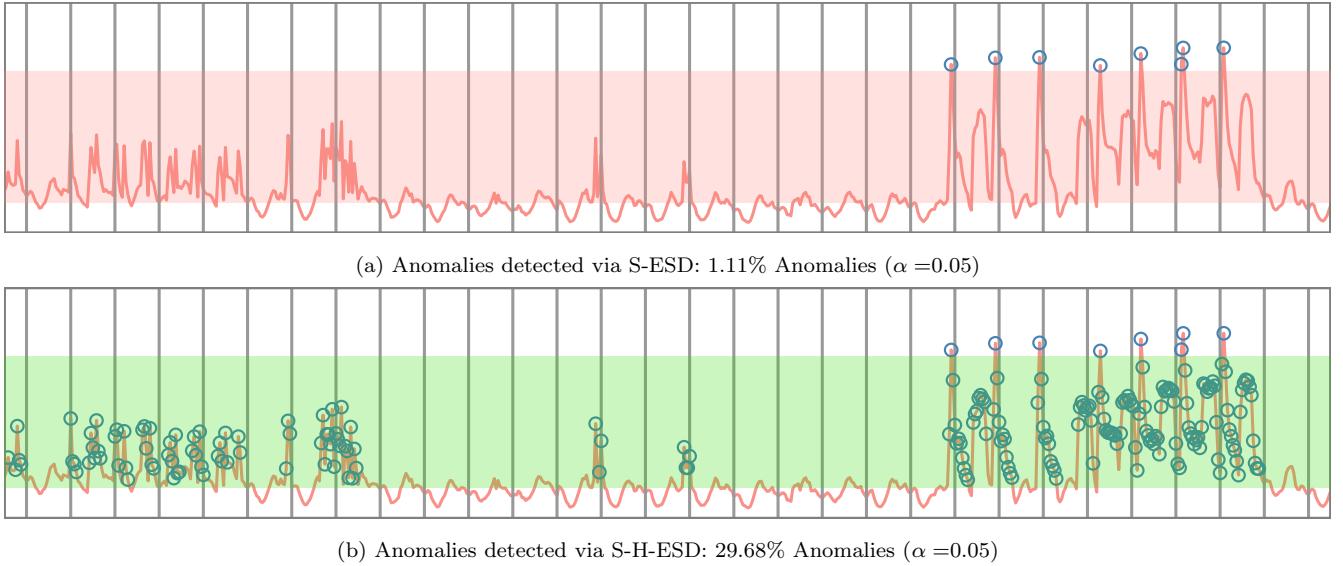
Figure 8: S-ESD (a) vs. S-H-ESD performance with highly anomalous data

and other social network data, may reflect changes in user behavior, or the systems in the data center/cloud. Figure 7 illustrates the use of STL variant to expose both global and local anomalies.

### 3.4.3 S-ESD Limitations

Although S-ESD can be used for detection of both global and local anomalies, S-ESD does not fare well when applied to data sets that have a high percentage of anomalies. This is exemplified by Figure 8a wherein S-ESD does not capture the anomalies corresponding to the region highlighted by the red rectangle.

As discussed earlier in subsection 2.2, a single large value can inflate both the mean and the standard deviation. This makes ESD conservative in tagging anomalies and results in a large number of false negatives. In the following section we detail the application of robust statistics as a further refinement of S-ESD.

## 3.5 Seasonal Hybrid ESD (S-H-ESD)

Seasonal Hybrid ESD (S-H-ESD) builds upon the S-ESD algorithm described in the previous subsection. In particular, S-H-ESD uses the robust statistical techniques and metrics discussed in subsection 2.2 to enable a more consistent measure of central tendency of a time series with a high percentage of anomalies. For example, let us consider the time series shown in Figure 8a. From the graph we observe that the seasonal component is apparent in the middle region of the time series; however, a significantly large portion of the time series is anomalous. This can inflate the mean and standard deviation, resulting in true anomalies being mislabeled as not anomalous and consequently yielding a high number of false negatives.

| Mean | Median | Std. Dev. | MAD |
|------|--------|-----------|------|
| 6.59 | 5.45   | 3.08      | 1.52 |

Table 1: Comparison of Mean vs. Median and Standard Deviation vs. Median Absolute Deviation

We addressed the above by replacing the mean and standard deviation used in ESD with more robust statistical measures during the calculation of the test statistic (refer to Equation 5); in particular, we use the median and MAD, as these metrics exhibit a higher breakdown point (discussed earlier in Section 2.2).

Table 1 lists the aforementioned metrics for the time series in Figure 8. The anomalies induce a small difference between the mean and median ($\approx$ 1.2%), however the standard deviation is $> 2\times$ the median absolute deviation. This results in S-ESD detecting only 1.11% of the data as "anomalous",

whereas S-H-ESD correctly detects 29.68% of the input time series as anomalous – contrast the two graphs shown in Figure 8b.

Note that using the median and MAD requires sorting the data and consequently the run time of S-H-ESD is higher than that of S-ESD. Therefore, in cases where the time series under consideration is large but with a relatively low anomaly count, it is advisable to use S-ESD. A detailed performance comparison between the two approaches is presented in Section 4.3.

# 4. EVALUATION

In this section we outline our methodology for evaluating the proposed techniques, discuss the deployment in production, and last but not the least, present results to demonstrate the efficacy of the proposed techniques.

## 4.1 Methodology

The efficacy of S-ESD and S-H-ESD was evaluated using a wide corpus of time series data obtained from production. The time series corresponded to both low-level *system metrics* and higher-level *core drivers* (business metrics). For example, but not limited to, the following metrics were used:

❏ System Metrics

  ▍ CPU utilization

  ▍ Heap usage

  ▍ Time spent in GC (garbage collection)

  ▍ Disk writes

❏ Application Metrics

  ▍ Request rate

  ▍ Latency

❏ Core Drivers

  ▍ Tweets per minute (TPM)

  ▍ Retweets per minute (RTPM)

  ▍ Unique Photos Per Minute (UPPM)

More than 20 data sets were used for evaluation. The system metrics ranged from two-week long periods to four-week long periods, with hour granularity. The core drivers metrics were all four-week periods, with minute granularity.

## 4.2 Production

Increasingly, machine-generated BigData is being used to drive performance and efficiency of data centers/cloud computing platforms. BigData is often characterized by *volume* and *velocity* [43, 44]. Given the multitude of services in our service-oriented-architecture (SOA), and the fact that each service monitors a large set of metrics, it is imperative to *automatically* detect anomalies[2] in the time series of each metric. We have deployed the proposed techniques for automatic detection of anomalies in production data for a wide set of services.

One can also set a threshold to refine the set of anomalies detected (using the proposed techniques) based on the specific requirements of the service owner. For example, it is

---

[2]A manual approach would be prohibitive from a cost perspective and would also be error-prone.

possible for capacity engineers to set a threshold such that only anomalies greater than the specified threshold are reported.

Based on extensive experimentation and analysis, S-H-ESD with $\alpha = 0.05$ (95% confidence) was selected for detecting anomalies in the metrics mentioned in the previous subsection. For each metric, S-H-ESD is run over a time series containing the last 14 days worth of data and an e-mail report is sent out *if* one or more anomalies were detected the previous day. The anomalies and time series are plotted using an in-house data visualization and analytics framework called Chiffchaff. Chiffchaff uses the ggplot2 plotting environment in R to produce graphs similar to the graphs shown in this paper. Additionally, CSVs with the metric, timestamps and magnitude of any anomalies detected is also attached to the email report.

## 4.3 Efficacy

In this section we detail the efficacy of S-ESD and S-H-ESD using the metrics mentioned earlier in this section. In particular, the following – *precision, recall*, and *F-measure* – described previously in subsection 2.3, are reported.

### 4.3.1 Perspectives

The performance of S-ESD and S-H-ESD was investigated from three different perspectives –

(a) Capacity engineering (CapEng), (b) User behavior (UB) and (c) Supervised learning (Inj), wherein anomalies were injected in the input time series to obtain labeled data.

In the first two cases, service owners set a threshold that was then used to categorize the detected anomalies into true positives (TP) and false positives (FP).

Specifically,

❏ The *CapEng* perspective was motivated by the fact that in capacity planning, a primary goal is to effectively scale the system resources to handle the normal operating levels of the traffic, e.g., the expected daily maximum, while maintaining enough headroom to absorb anomalies in input traffic.

  Thus, the objective in the current context is to detect anomalies that have a magnitude greater than a pre-specified threshold (which is in turn determined via load testing).

❏ From a user behavior (UB) perspective, the local intra-day anomalies serve as potential signals of change in user behavior. To this end, the service owners set the threshold on the residual component of the time series of interest.

  Note that setting a threshold as mentioned above is intrinsically directed toward detection of only positive anomalies. In other words, anomalies in the right tail of the underlying data distribution are detected.

❏ Lastly, the *Inj* perspective is aimed to assess the efficacy of the proposed techniques in the presence of ground-truth (or labeled data). Given the volume, velocity, and real-time nature of production data, it is not practically feasible to obtain labeled anomalies. To alleviate this limitation, we first fit a smooth spline (B-spline) curve to a time series (obtained from production) to derive a time series which had the same characteristics – trend and seasonality – as of the original time series. Subsequently, positive anomalies were

| One-Tail (Capacity) | Alpha = 0.05 | | | | | | Alpha = 0.001 | | | | | | # of Observations |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **S-ESD** | | | **S-H-ESD** | | | **S-ESD** | | | **S-H-ESD** | | | |
| Dataset # | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | |
| System 1 | 1.00 | 0.05 | 0.09 | 0.26 | 1.00 | 0.41 | 1.00 | 0.02 | 0.05 | 0.28 | 1.00 | 0.44 | 337 |
| System 2 | 1.00 | 0.04 | 0.07 | 0.95 | 0.87 | 0.91 | 1.00 | 0.00 | 0.01 | 0.95 | 0.81 | 0.88 | 721 |
| System 3 | 1.00 | 0.14 | 0.25 | 0.98 | 0.93 | 0.95 | 1.00 | 0.01 | 0.01 | 1.00 | 0.77 | 0.87 | 601 |
| System 4 | 1.00 | 0.57 | 0.73 | 0.71 | 0.99 | 0.82 | 1.00 | 0.43 | 0.60 | 0.84 | 0.95 | 0.89 | 913 |
| System 5 | 0.98 | 1.00 | 0.99 | 0.98 | 1.00 | 0.99 | 0.98 | 1.00 | 0.99 | 0.98 | 1.00 | 0.99 | 337 |
| System 6 | 0.75 | 0.28 | 0.41 | 0.65 | 0.34 | 0.45 | 0.80 | 0.13 | 0.22 | 0.75 | 0.28 | 0.41 | 721 |
| **Average** | 0.96 | 0.35 | 0.42 | 0.75 | 0.85 | 0.76 | 0.96 | 0.26 | 0.31 | 0.80 | 0.80 | 0.75 | |
| Core Driver 1 | 0.99 | 0.48 | 0.65 | 0.83 | 0.78 | 0.80 | 1.00 | 0.40 | 0.57 | 0.95 | 0.61 | 0.74 | 43195 |
| Core Driver 2 | 1.00 | 0.03 | 0.06 | 0.96 | 0.25 | 0.39 | 1.00 | 0.01 | 0.02 | 1.00 | 0.05 | 0.10 | 43200 |
| Core Driver 3 | 0.08 | 1.00 | 0.14 | 0.05 | 1.00 | 0.10 | 0.12 | 1.00 | 0.22 | 0.08 | 1.00 | 0.14 | 43200 |
| Core Driver 4 | 0.34 | 0.20 | 0.25 | 0.20 | 0.30 | 0.24 | 0.54 | 0.18 | 0.27 | 0.29 | 0.22 | 0.25 | 43200 |
| Core Driver 5 | 0.36 | 0.98 | 0.52 | 0.19 | 0.98 | 0.32 | 0.53 | 0.98 | 0.69 | 0.33 | 0.98 | 0.49 | 43200 |
| Core Driver 6 | 0.08 | 0.69 | 0.14 | 0.06 | 0.69 | 0.11 | 0.12 | 0.69 | 0.20 | 0.08 | 0.69 | 0.14 | 43200 |
| Core Driver 7 | 0.27 | 0.97 | 0.42 | 0.20 | 0.97 | 0.33 | 0.33 | 0.97 | 0.49 | 0.25 | 0.97 | 0.40 | 43200 |
| Core Driver 8 | 0.66 | 0.95 | 0.78 | 0.58 | 0.99 | 0.73 | 0.80 | 0.63 | 0.71 | 0.67 | 0.95 | 0.78 | 43200 |
| Core Driver 9 | 0.80 | 0.54 | 0.65 | 0.58 | 0.63 | 0.60 | 0.91 | 0.53 | 0.67 | 0.72 | 0.56 | 0.63 | 43178 |
| Core Driver 10 | 0.82 | 0.20 | 0.32 | 0.67 | 0.30 | 0.42 | 1.00 | 0.08 | 0.14 | 0.81 | 0.20 | 0.32 | 43200 |
| **Average** | 0.54 | 0.60 | 0.39 | 0.43 | 0.69 | 0.40 | 0.63 | 0.55 | 0.40 | 0.52 | 0.62 | 0.40 | |

Table 2: CapEng Perspective: Precision, Recall, and F-measure

| One-Tail (User Behavior) | Alpha = 0.05 | | | | | | Alpha = 0.001 | | | | | | # of Observations |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **S-ESD** | | | **S-H-ESD** | | | **S-ESD** | | | **S-H-ESD** | | | |
| Dataset # | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | |
| System 1 | 1.00 | 0.02 | 0.04 | 0.60 | 1.00 | 0.75 | 1.00 | 0.01 | 0.02 | 0.65 | 1.00 | 0.79 | 337 |
| System 2 | 1.00 | 0.03 | 0.06 | 1.00 | 0.68 | 0.81 | 1.00 | 0.00 | 0.01 | 1.00 | 0.63 | 0.77 | 721 |
| System 3 | 1.00 | 0.12 | 0.22 | 1.00 | 0.82 | 0.90 | 1.00 | 0.00 | 0.01 | 1.00 | 0.67 | 0.80 | 601 |
| System 4 | 1.00 | 0.49 | 0.66 | 0.84 | 1.00 | 0.91 | 1.00 | 0.37 | 0.54 | 1.00 | 0.96 | 0.98 | 913 |
| System 5 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 | 0.99 | 337 |
| System 6 | 1.00 | 0.38 | 0.55 | 1.00 | 0.53 | 0.69 | 1.00 | 0.16 | 0.27 | 1.00 | 0.38 | 0.55 | 721 |
| **Average** | 1.00 | 0.34 | 0.42 | 0.90 | 0.84 | 0.84 | 1.00 | 0.26 | 0.31 | 0.94 | 0.77 | 0.81 | |
| Core Driver 1 | 1.00 | 0.46 | 0.63 | 1.00 | 0.90 | 0.95 | 1.00 | 0.38 | 0.55 | 1.00 | 0.61 | 0.76 | 43195 |
| Core Driver 2 | 1.00 | 0.03 | 0.05 | 1.00 | 0.21 | 0.35 | 1.00 | 0.01 | 0.02 | 1.00 | 0.04 | 0.08 | 43200 |
| Core Driver 3 | 1.00 | 0.27 | 0.43 | 1.00 | 0.40 | 0.57 | 1.00 | 0.18 | 0.30 | 1.00 | 0.28 | 0.43 | 43200 |
| Core Driver 4 | 1.00 | 0.03 | 0.05 | 1.00 | 0.07 | 0.13 | 1.00 | 0.02 | 0.03 | 1.00 | 0.03 | 0.07 | 43200 |
| Core Driver 5 | 1.00 | 0.22 | 0.36 | 1.00 | 0.41 | 0.58 | 1.00 | 0.15 | 0.25 | 1.00 | 0.23 | 0.38 | 43200 |
| Core Driver 6 | 1.00 | 0.39 | 0.56 | 1.00 | 0.52 | 0.68 | 1.00 | 0.27 | 0.42 | 1.00 | 0.39 | 0.56 | 43200 |
| Core Driver 7 | 1.00 | 0.72 | 0.84 | 1.00 | 0.96 | 0.98 | 1.00 | 0.59 | 0.74 | 1.00 | 0.77 | 0.87 | 43200 |
| Core Driver 8 | 1.00 | 0.24 | 0.39 | 1.00 | 0.29 | 0.45 | 1.00 | 0.13 | 0.24 | 1.00 | 0.24 | 0.39 | 43200 |
| Core Driver 9 | 1.00 | 0.35 | 0.52 | 1.00 | 0.56 | 0.72 | 1.00 | 0.30 | 0.46 | 1.00 | 0.40 | 0.58 | 43178 |
| Core Driver 10 | 1.00 | 0.14 | 0.25 | 1.00 | 0.28 | 0.43 | 1.00 | 0.05 | 0.09 | 1.00 | 0.15 | 0.26 | 43200 |
| **Average** | 1.00 | 0.29 | 0.41 | 1.00 | 0.46 | 0.58 | 1.00 | 0.21 | 0.31 | 1.00 | 0.32 | 0.44 | |

Table 3: User Behavior Perspective: Precision, Recall, and F-measure

injected in the derived time series, with varying magnitudes, widths, and frequency. The positions and magnitudes of the injected anomalies were recorded and used to compare against the anomalies detected by both S-ESD and S-H-ESD.

### 4.3.2 Results

We computed Precision, Recall, and F-measure (refer to Section 2.3) to assess the efficacy of S-ESD and S-H-ESD from the three perspectives.

Right-tailed (/positive) anomalies were detected at both 95% and 99.9% confidence levels. The metrics are reported in Tables 2 (CapEng), 3 (UB), and 4 (Inj).

From the tables we note the following: across both system metrics and core drivers, precision increases from about 75% in the CapEng perspective to 100% in the UB perspective for S-ESD, and about 59% to 95% for S-H-ESD. The threshold set in the CapEng perspective results in labeling of the intra-day or off-peak anomalies as false positives, thereby lowering the precision. In contrast, in the UB perspective, the threshold is set for the residual component which removes the seasonality effect. This results in S-H-ESD's higher recall rates, which improve from 47.5%

(S-ESD) to 77% (S-H-ESD) for CapEng and from 31.5% (S-ESD) to 65% (S-H-ESD) for UB.

Comparative analysis of the F-measure reported in Tables 2 and 3 highlights that the F-measure is better in the latter case. This can be attributed, in part, to the fact that in case of the latter the service-owners set the thresholds over the residual component (recall that the time series decomposition removes the trend and seasonal component). This makes anomaly detection independent of the time at which they occurred, e.g., on-peak on a daily max, or off-peak on a daily trough. The low values of F-measure in the CapEng perspective are reasoned at the end of this section.
Table 4 reports the efficacy of S-ESD and S-H-ESD from the Inj (Ground-Truth) perspective. From the table we note that the precision achieved was 100%, meaning that all anomalies detected were *true* anomalies (there were no false positives). Also, the recall was very high, achieving about 96% and 97% (for S-ESD and S-H-ESD respectively) at the 95% confidence level and about 94% and 95% recall at the 99.9% confidence level.

On further analysis we noted that false negatives (anomalies that were not detected) were within the boundaries of

| One-Tail (Injection) | Alpha = 0.05 | | | | | | Alpha = 0.001 | | | | | | # of Observations |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | S-ESD | | | S-H-ESD | | | S-ESD | | | S-H-ESD | | | |
| Dataset | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | Precision | Recall | Fmeasure | |
| mag.75_width5_pct100 | 1 | 0.72 | 0.84 | 1 | 0.79 | 0.88 | 1 | 0.61 | 0.76 | 1 | 0.70 | 0.82 | 43200 |
| mag1.5_width5_pct100 | 1 | 0.96 | 0.98 | 1 | 0.99 | 0.99 | 1 | 0.89 | 0.94 | 1 | 0.94 | 0.97 | 43200 |
| mag3_width5_pct100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 43200 |
| mag3_width10_pct100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 43200 |
| mag3_width25_pct100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 43200 |
| mag3_width50_pct100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 43200 |
| mag3_width100_pct100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 43200 |
| mag6_width5_pct100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 43200 |
| Average | 1.00 | 0.96 | 0.98 | 1.00 | 0.97 | 0.98 | 1.00 | 0.94 | 0.96 | 1.00 | 0.95 | 0.97 | 345600 |

Table 4: Anomaly Injection Perspective: Precision, Recall, and F-measure
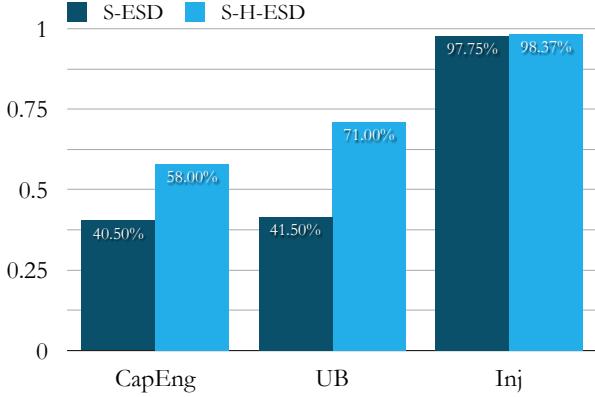


Figure 9: F-measure: CapEng vs. UB vs. Inj (95% Confidence)

the anomalies detected, normally at the tail ends (at the beginning or end of the sustained anomalous behavior).

Lastly, the results reported Table 4 correspond to injection sets containing anomalies with increasing magnitudes from $0.75\sigma$ to $6\sigma$. At $0.75\sigma$, the F-measure of S-ESD and S-H-ESD begins to degrade (0.84 (S-ESD) and 0.88 (S-H-ESD) at the 95% confidence level and 0.76 (S-ESD) 0.82 (S-H-ESD) at the 99.9% confidence level). At $1.5\sigma$, the F-measure was about 0.97 on an average. Injected anomalies with a magnitude of $3\sigma$ or greater achieved an F-measure of 1.00.

Figure 9 summarizes the overall F-measure average for the three perspectives at the 95% confidence level. From the figure we note that in each case S-H-ESD outperformed S-ESD; in particular, the F-measure increased by 17.5%, 29.5% and 0.62% for CapEng, UB, and Inj respectively. This stems from the fact that median and MAD, unlike mean and standard deviation, are robust against a large number of outliers.

The F-measure for the CapEng and UB perspectives is



Figure 10: Illustration of false positives in CapEng perspective

significantly lower than the Inj perspective. This can be ascribed to the following: Capacity planning engineers typically determine the capacity needed to withstand the typical daily peaks (with additional headroom), thus the service-owners tend to set the threshold around the maximum daily peaks. Consequently, anomalies which occur during the off-peak hours of the day, such as in the daily troughs or other intra-day locations, would be marked as false positives (even though they might still be anomalous from S-ESD/S-H-ESD's point of view). This is illustrated in Figure 10 wherein the anomalies (detected using S-H-ESD) above the pre-specified threshold are annotated by ◯ and the rest are annotated by △. The latter, albeit "true" anomalies from a statistical standpoint, are tagged as false positives which adversely impact precision (refer to Equation 9).

## 5. PREVIOUS WORK

In this section, we overview prior work in the context of anomaly detection. A lot of anomaly detection research has been done in various domains such as, but not limited to, statistics, signal processing, finance, econometrics, manufacturing, and networking. For a detailed coverage of the same, the reader is referred to books and survey papers [16, 17, 18, 19].

For better readability, we have partitioned this section into subsections on a per domain basis. As mentioned in a recent survey on anomaly detection [20], anomaly detection is highly contextual in nature. Based on our in-depth literature survey we find that the techniques discussed in the rest of this section cater to different type of data sets (than cloud infrastructure data) and hence are complementary to the techniques proposed in this paper.

**Manufacturing**
Anomaly detection manifests itself in manufacturing in the form of determining if a particular process is in a state of normal and stable behavior. To this end, Statistical Process Control (SPC) was proposed in the early 1920s to monitor and control the reliability of a manufacturing process [45, 46]. *Control charts* are one of the key tools used in SPC.

In essence, the premise of SPC is that a certain amount of variation can occur at any one point of a production chain. The variation is "common" if it is controlled and within normal or expected limits. However, the variation is "assignable" if it is not present in the causal system of the process at all times (i.e., falls outside of the normal limits). Identifying and removing the assignable sources which have impact on the manufacturing process is thus crucial to ensure the expected operation and quality of the manufacturing process.

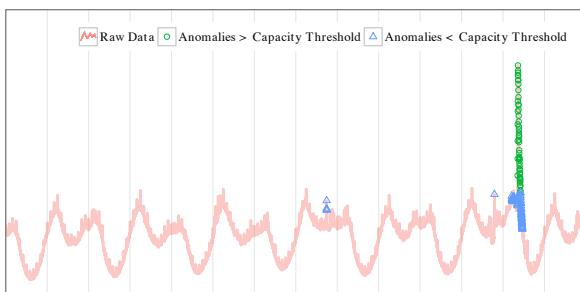A traditional control chart includes points representing

a statistical measurement (such as the mean) of a quality characteristic in samples taken over a period of time. The mean of this characteristic is calculated over all samples, and plotted as the center line. The standard deviation is calculated over all samples, and the upper control limit (UCL) and lower control limit (LCL) defined as the threshold at which the process is considered statistically unlikely to occur (typically set at 3 standard deviations, denoted by $3\sigma$, about the mean/center line). When the process is "in control", 99.73% of all points are within the upper and lower control limits. A signal may be generated when observations fall outside of these control limits, signifying the introduction of some other source of variation outside of the normal expected behavior.

Methodologies for improving the performance of control charts have been investigated since Shewhart's early work. Roberts proposed the geometrical moving average control chart – also known as the EWMA (exponentially weighted moving average) control chart – which weights the most recent samples more highly than older samples [47]. The EWMA chart tends to detect small shifts (1-2 $\sigma$) in the sample mean more efficiently; however, the Shewhart chart tends to detect larger shift (3 $\sigma$) more efficiently. In case the quality characteristic follows a Poisson distribution, alternatives to the EWMA chart have been proposed [48]. Other types of control charts, such as the CUSUM (cumulative sum) chart [49], have been proposed. In [50], Lowry and Montgomery present a review on control charts for multivariate quality control. For further details on SPC and control charts, the reader is referred to the surveys by Woodall and Montgomery [51] and by Stoumbos et al. [52]. Lastly, a recent survey by Tsung et al. presents a comprehensive survey of statistical control methods for multistage manufacturing and service operations [53].

### Finance
Behavioral economics and finance study the causal relationships between social, cognitive, and emotional factors on the economic decisions of individuals and institutions, and their effect on the economic decisions such as market prices and returns.

The interplay of the aforementioned factors often result in, but not limited to, seasonal stock market anomalies. These seasonal anomalies, also referred to as "calendar effects", take many forms. Haugen and Jorion describe the January Effect – stocks, especially small stocks, historically generate abnormally high returns during the month of January – as *"... perhaps the best-known example of anomalous behavior in security markets throughout the world"* [54]. This effect is normally attributed to a theory which states that the investors who hold a disproportionate amount of small stocks sell their stocks for tax reasons at the end of the year (to claim capital loss), and then reinvest at the start of the new year. On the other hand, the January effect is also attributed to the end of year bonuses which are paid in January and used to purchase stocks (thus driving up prices).

In [55], Angelovska reports that early studies on stock market anomalies began in the late 1920's, where Kelly's reports [56] showed the existence of a so-called *"Monday effect"* – US markets have low and negative Monday returns. The Monday effect in the US sock market was actively researched in the 1980s [57] [58] [59]. In [60], Coutts and Hayes showed that the Monday effect exists albeit not as strongly as previous work demonstrated. Wang et al. show that between 1962 and 1993, the Monday effect is strongly evident in the last two weeks of the month, for a wide array of stock indices. For a full literature review on the Monday effect, refer to [61].

Numerous studies support the existence of calendar effects in stock markets, as well as others such as "turn of the month" effects. For instance, Hensel and Ziemba examined S&P 500 returns over a 65 year period from 1928 to 1993, and reported that U.S. large-cap stocks consistently show higher returns at the turn of the month [62]. In contrast, Sullivan et al. argue against the case, stating that there is no statistically significant evidence supporting the claim, and that such periodicities in stock market behavior are the result of data dredging [63]. Subsequently, Hansen et al. attempted to use sound statistical approaches to evaluate the significance of calendar effects, by tightly controlling the testing to avoid data mining biases [64]. In their study of 27 stock indices from 10 countries, calendar effects were found to be significant in most returns series, with the end-of-the-year effect producing the largest anomalies and the most convincing evidence supporting calendar effects in small-cap indices.

### Signal Processing
Techniques from signal processing such as, but not limited to, spectral analysis, have been adopted for anomaly detection. For instance, in [65], Cheng et al. employed spectral analysis to complement existing DoS defense mechanisms that focus on identifying attack traffic, by ruling out normal TCP traffic, thereby reducing false positives.

Similarly, wavelet packets and wavelet decomposition have been used for detecting anomalies in network traffic [66, 67, 68, 69, 70]. Benefits of wavelet-based techniques include the ability to accurately detect anomalies at various frequencies (due to the inherent time-frequency property of decomposing signals into different components at several frequencies), with relatively fast computation.

In [71], Gao et al. proposed a speed optimization for realtime use using sliding windows. Recently, Lu et al. proposed an approach consisting of three components: (1) feature analysis, (2) normal daily traffic modeling based on wavelet approximation and ARX (AutoRegressive with eXogenous), and intrusion decision [72]. An overview of signal processing techniques for network anomaly detection, including PSD (Power Spectral Density) and wavelet-based approaches, is presented in [73].

Additionally, Kalman filtering and Principle Component Analysis (PCA) based approaches have been proposed in the signal processing domain for anomaly detection. In [74], Ndong and Salamatian reported that PCA-based approaches exhibit improved performance when coupled with Karuhen-Loeve expansion (KL); on the other hand, Kalman filtering approaches, when combined with statistical methods such as Gaussian mixture and Hidden Markov models, outperformed the PCA-KL method.

### Network Traffic
With the Internet of Things (IoT) paradigm [75, 76] increasingly becoming ubiquitous[3], there is an increasing concern about security. In a post the FTC said on its website [78]: "At the same time, the data collection and sharing that smart devices and greater connectivity enable pose privacy

---

[3]According to a study by ABI Research [77], it is estimated that 10 billion devices are currently connected to one another by wired or wireless Internet. By the year 2020, that number is expected to exceed 30 billion.

and security risks. Over the years, various anomaly detection techniques have been proposed for detection network intrusion.

For example, in [79], Denning proposed a rule-based technique wherein both network (system) and user data was used to detect different types of abnormal behavior, by comparing audit-trails to different anomalous profiles or models.

In [80], Lazarevic et al. presented a comparative study of several anomaly detection schemes for network intrusion detection.

In [81], Garcia-Teodoro et al. presented an overview of the pros and cons of various approaches for anomaly detection in network intrusion systems such as statistical techniques, knowledge-based techniques (finite state machines, Bayesian networks, expert systems, etc.), and learning based classification of patterns (Markov models, neural networks, fuzzy logic, clustering, et cetera). Recently, Gogoi et al. presented a comprehensive survey on outlier detection for network anomaly detection in [82]; in particular, the authors classified the approaches into three categories: (1) distance-based, (2) density-based, and (3) machine learning or soft-computing based.

The reader is referred to the survey of intrusion detection techniques by Yu for further reading [83].

### Statistics

Anomaly detection has been actively researched for over five decades in the domain of statistics [84, 85, 86, 16, 87]. Recent surveys include the ones from Hodge and Jim [18] and Chandola et al. [88].

The key focus of prior work has been to determine whether a single value is statistically anomalous with respect to an underlying distribution. Work by Markov and Chebyshev provided bounds on the probability of a random value with respect to the expected value of the distribution. The Markov inequality states that for any non-negative random variable X, the following holds true $P(X > \alpha) \leq E[X]/\alpha$, while the more general Chebyshev inequality states that $P(|X - E[X]| > \alpha) \leq Var[X]/\alpha^2$, and shows that values equal to or greater then K standard deviations from the expected value constitute no more then $1/k^2$ of the total distribution.

These bounds can be used as a threshold for determining the "outlierness" of a random value, indicating that a value does not fit the underlying distribution [17]; however, the Markov and Chebyshev inequalities are non-parametric, and create relatively weak bounds that may miss potential outliers in the data [19]. The Chernoff bound and the Hoeffding inequality attempt to create tighter bounds by making assumptions about the underlying distribution. While these tail inequalities provided a closer bounds for testing the outlierness of a data point, their assumptions regarding the distribution make them unsuitable for use when the distribution doesn't follow their underlying assumptions (as in the current context).

Further, the *Box plot* may be applied as a robust means of determining if a data point is anomalous with respect to the underlying distribution [89]. A Box plot divides the data into five groups: the minimum non-anomalous value (*min*), the lower quartile (*Q1*), the median, the upper quartile (*Q3*) and the maximum non-anomalous value (*max*). Data that is 1.5× lower then (*Q1*) or 1.5× greater then (*Q3*) are typically considered anomalous.

## 6. CONCLUSION

In this paper we presented two novel statistical techniques for automatically detecting anomalies in cloud infrastructure data. Although there exists a large body of research in anomaly detection, the seasonal (and trending) nature of cloud infrastructure data limits the application of techniques. To this end, we proposed a method called *Seasonal-ESD* (S-ESD), which combines seasonal decomposition and the Generalized ESD test, for anomaly detection. The second method, *Seasonal-Hybrid-ESD* (S-H-ESD), builds on S-ESD to enable robust anomaly detection when a significant portion (up to 50%) of the underlying data is anomalous. This is achieved by extending the original ESD algorithm with robust statistical measures, median and median absolute deviation (MAD).

The efficacy of both S-ESD and S-H-ESD was evaluated using both core metrics such as Tweets Per Sec (TPS), system metrics such as CPU and heap usage and application metrics. The evaluation was carried out from three different perspectives, viz., capacity engineering (CapEng), user behavior (UB), and supervised learning (Inj). Precision, Recall, and F-measure in each case. Overall, S-H-ESD outperformed S-ESD, with F-Measure increasing by 17.5%, 29.5% and 0.62% for CapEng, UB, and Inj respectively.

In light of the fact that S-H-ESD more computationally expensive than S-ESD (recall that the former requires sorting of the data), it is recommended to use S-ESD in cases where the time series under consideration is large but with a relatively low anomaly count.

As future work, we plan to extend the proposed techniques for detecting anomalies in long time series. The challenge in this regard is that capturing the underlying trend,[4] which in our observation is predominant in the case of long time series, is non-trivial in the presenece of anomalies. To this end, we plan to explore the use of qunatile regression [90] and/or robust regression [91, 92].

## 7. REFERENCES

[1] Federal Government Big Data Rollout. http://www.nsf.gov/news/news_videos.jsp?cntn_id=123607&media_id=72174&org=NSF, 2012.
[2] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation, May 2011.
[3] Big Data Market Size and Vendor Revenues. http://wikibon.org/wiki/v/Big_Data_Market_Size_and_Vendor_Revenues.
[4] New Digital Universe Study Reveals Big Data Gap: Less Than 1% of World's Data is Analyzed; Less Than 20% is Protected. http://www.emc.com/about/news/press/2012/20121211-01.htm.
[5] Web Startups Crumble under Amazon S3 Outage. http://www.theregister.co.uk/2008/02/15/amazon_s3_outage_feb_20%08/.
[6] How Much is the Reputation of Your SaaS Provider Worth? http://cloudsecurity.org/2009/03/13/how-much-is-the-reputation-of-your-saas-provider-worth.
[7] S. Agarwala and K. Schwan. Sysprof: Online distributed behavior diagnosis through fine-grain system monitoring. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, pages 8–8, 2006.
[8] G. Ren, E. Tune, T. Moseley, Y. Shi, S. Rus, and R. Hundt. Google-wide profiling: A continuous profiling infrastructure for data centers. *IEEE Micro*, 30(4):65–79, July 2010.
[9] Mike Y. Chen, Emre Kiciman, Eugene Fratkin, Armando Fox, and Eric Brewer. Pinpoint: Problem determination in large, dynamic internet services. In *Proceedings of the 2002 International Conference on Dependable Systems and Networks*, pages 595–604, 2002.
[10] Anton Babenko, Leonardo Mariani, and Fabrizio Pastore. Ava: automated interpretation of dynamically detected anomalies. In *Proceedings of the eighteenth international symposium on Software testing and analysis*, pages 237–248, 2009.
[11] J. P. Magalh aes and Luis Moura Silva. Root-cause analysis of performance anomalies in web-based applications. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 209–216, TaiChung, Taiwan, 2011.
[12] Hui Kang, Xiaoyun Zhu, and Jennifer L. Wong. Dapa: diagnosing application performance anomalies for virtualized infrastructures. In *Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, pages 8–8, 2012.
[13] Mona Attariyan, Michael Chow, and Jason Flinn. X-ray: automating root-cause diagnosis of performance anomalies in production software. In *Proceedings of the 10th USENIX conference on Operating Systems Design and Implementation*, pages 307–320, Hollywood, CA, 2012.
[14] C. Wang, V. Talwar, K. Schwan, and P. Ranganathan. Online detection of utility cloud anomalies using metric distributions. In *Network Operations and*

---

[4]Capturing the trend is required to minimize the number of false positives.

*Management Symposium (NOMS), 2010 IEEE*, pages 96–103, 2010.

[15] C. Wang, Krishnamurthy Viswanathan, Choudur Lakshminarayan, Vanish Talwar, Wade Satterfield, and Karsten Schwan. Statistical techniques for online anomaly detection in data centers. In *Proceedings of Integrated Network Management*, pages 385–392, 2011.

[16] Douglas M. Hawkins. *Identification of outliers*, volume 11. Chapman and Hall London, 1980.

[17] Vic Barnett and Toby Lewis. *Outliers in statistical data*, volume 3. Wiley New York, 1994.

[18] Victoria J. Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[19] Charu C. Aggarwal. *Outlier analysis*. Springer, 2013.

[20] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009.

[21] Bernard Rosner. On the detection of many outliers. *Technometrics*, 17(2):221–227, 1975.

[22] Bernard Rosner. Percentage points for a generalized ESD many-outlier procedure. *Technometrics*, 25(2):165–172, 1983.

[23] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 2013.

[24] George W Snedecor and William G Cochran. *Statistical methods*. Iowa State University Press, Ames, 1989.

[25] Peter J Huber and Elvezio Ronchetti. *Robust statistics*. Wiley, Hoboken, N.J., 1981.

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

[27] R.A. Fisher. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925.

[28] F. E. Grubbs. Sample criteria for testing outlying observations. *Ann. Math. Statistics*, 21:27–58, 1950.

[29] Frank E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[30] Francisco Augusto Alcaraz Garcia. Tests to identify outliers in data series. *Pontifical Catholic University of Rio de Janeiro, Industrial Engineering Department, Rio de Janeiro, Brazil*, 2012.

[31] Gary L. Tietjen and Roger H. Moore. Some grubbs-type statistics for the detection of several outliers. *Technometrics*, 14(3):583–597, 1972.

[32] Frank R Hampel, Elvezio Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust statistics: the approach based on influence functions*. Wiley, New York, 1986.

[33] Frank Rudolf Hampel. *Contributions to the theory of robust estimation*. University of California, 1968.

[34] David L. Donoho and Peter J. Huber. The notion of breakdown point. *A Festschrift for Erich L. Lehmann*, pages 157–184, 1983.

[35] Peter J. Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.

[36] Yutaka Sasaki. The truth of the f-measure. *Teach Tutor mater*, pages 1–5, 2007.

[37] James M. Lucas and Michael S. Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1):1–12, 1990.

[38] Kevin M. Carter and William W. Streilein. Probabilistic reasoning for streaming anomaly detection. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 377–380, 2012.

[39] S. Ahn and J. A. Fessler. Standard Errors of Mean, Variance, and Standard Deviation Estimators. http://web.eecs.umich.edu/~fessler/papers/files/tr/stderr.pdf, 2003.

[40] Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. STL: a seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[41] A. Stuart, M. Kendall, and J. Keith Ord. *The advanced theory of statistics. Vol. 3: Design and analysis and time-series*. Griffin, 1983.

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

[43] Volume, Velocity, Variety: What You Need to Know About Big Data. http://www.forbes.com/sites/oreillymedia/2012/01/19/volume-velocity-variety-what-you-need-to-know-about-big-data/, 2012.

[44] The Four V's of Big Data. http://dashburst.com/infographic/big-data-volume-variety-velocity/, 2012.

[45] Walter A. Shewhart. Quality control charts. *Bell System Technical Journal*, 5(4):593–603, 1926.

[46] W. A. Shewart. *Economic control of Quality of Manufactured Product*. Van Nostrand Reinhold Co., 1931.

[47] S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 1(3):239–250, 1959.

[48] Douglas C. Montgomery. *Introduction to statistical quality control*. Wiley. com, 2007.

[49] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[50] Cynthia A. Lowry and Douglas C. Montgomery. A review of multivariate control charts. *IIE transactions*, 27(6):800–810, 1995.

[51] William H. Woodall and Douglas C. Montgomery. Research issues and ideas in statistical process control. *Journal of Quality Technology*, 31(4), 1999.

[52] Zachary G. Stoumbos, Marion R. Reynolds Jr, Thomas P. Ryan, and William H. Woodall. The state of statistical process control as we proceed into the 21st century. *Journal of the American Statistical Association*, 95(451):992–998, 2000.

[53] Fugee Tsung, Yanting Li, and Ming Jin. Statistical process control for multistage manufacturing and service operations: a review and some extensions. *International Journal of Services Operations and Informatics*, 3(2):191–204, 2008.

[54] Robert A. Haugen and Philippe Jorion. The january effect: still there after all these years. *Financial Analysts Journal*, pages 27–31, 1996.

[55] Julijana Angelovska. An econometric analysis of market anomaly-day of the week effect on a small emerging market. *International Journal of Academic Research in Accounting, Finance and Management Sciences*, 3(1):314–322, 2013.

[56] Fred C. Kelly. *Why you win or lose: The psychology of speculation*. Courier Dover Publications, 1930.

[57] Kenneth R. French. Stock returns and the weekend effect. *Journal of financial economics*, 8(1):55–69, 1980.

[58] Michael R. Gibbons and Patrick Hess. Day of the week effects and asset returns. *Journal of business*, pages 579–596, 1981.

[59] Richard J. Rogalski. New findings regarding day-of-the-week returns over trading and non-trading periods: A note. *The Journal of Finance*, 39(5):1603–1614, 1984.

[60] J. Andrew Coutts and Peter A. Hayes. The weekend effect, the stock exchange account and the financial times industrial ordinary shares index: 1987-1994. *Applied Financial Economics*, 9(1):67–71, 1999.

[61] Glenn N. Pettengill. A survey of the monday effect literature. *Quarterly Journal of Business and Economics*, pages 3–27, 2003.

[62] Chris R. Hensel and William T. Ziemba. Investment results from exploiting turn-of-the-month effects. *The Journal of Portfolio Management*,

22(3):17–23, 1996.

[63] Ryan Sullivan, Allan Timmermann, and Halbert White. Dangers of data mining: The case of calendar effects in stock returns. *Journal of Econometrics*, 105(1):249–286, 2001.

[64] Peter Hansen, Asger Lunde, and James Nason. Testing the significance of calendar effects. *Federal Reserve Bank of Atlanta Working Paper*, (2005-02), 2005.

[65] Chen-Mou Cheng, H. T. Kung, and Koan-Sin Tan. Use of spectral analysis in defense against DoS attacks. In *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, volume 3, pages 2143–2148, 2002.

[66] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 71–82, 2002.

[67] Vicente Alarcon-Aquino and Javier A. Barria. Anomaly detection in communication networks using wavelets. *IEE Proceedings-Communications*, 148(6):355–362, 2001.

[68] Lan Li and Gyungho Lee. DDoS attack detection and wavelets. *Telecommunication Systems*, 28(3-4):435–451, 2005.

[69] Seong Soo Kim, AL Narasimha Reddy, and Marina Vannucci. Detecting traffic anomalies through aggregate analysis of packet header data. In *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, pages 1047–1059. Springer, 2004.

[70] Anu Ramanathan. *WADeS: A tool for distributed denial of service attack detection*. PhD thesis, Texas A&M University, 2002.

[71] Jun Gao, Guangmin Hu, Xingmiao Yao, and Rocky KC Chang. Anomaly detection of network traffic based on wavelet packet. In *Communications, 2006. APCC'06. Asia-Pacific Conference on*, pages 1–5, 2006.

[72] Wei Lu and Ali A. Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing*, 2009:4, 2009.

[73] Lingsong Zhang. Signal processing methods for network anomaly detection. 2005.

[74] Joseph Ndong and Kavé Salamatian. Signal processing-based anomaly detection techniques: A comparative analysis. In *INTERNET 2011, The Third International Conference on Evolving Internet*, pages 32–39, 2011.

[75] Disruptive Civil Technologies Six Technologies with Potential Impacts on US Interests out to 2025 . http://www.fas.org/irp/nic/disruptive.pdf, 2008.

[76] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, October 2010.

[77] Internet of Everything. https://www.abiresearch.com/research/service/internet-of-everything.

[78] FTC Seeks Input on Privacy and Security Implications of the Internet of Things. http://www.ftc.gov/opa/2013/04/internetthings.shtm.

[79] Dorothy E. Denning. An intrusion-detection model. *Software Engineering, IEEE Transactions on*, (2):222–232, 1987.

[80] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. *Proc. SIAM*, 2003.

[81] Pedro Garcia-Teodoro, J. Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28, 2009.

[82] Prasanta Gogoi, D. K. Bhattacharyya, Bhogeswar Borah, and Jugal K. Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4):570–588, 2011.

[83] Yingbing Yu. A survey of anomaly intrusion detection techniques. *J. Comput. Sci. Coll.*, 28(1):9–17, October 2012.

[84] F. J. Anscombe and Irwin Guttman. Rejection of outliers. *Biometrika*, 2:123–147, 1960.

[85] D. Bernoulli. The most probable choice between several discrepant observations and the formation therefrom of the most likely induction. *Biometrika*, 48:3–18, 1961.

[86] Vic Barnett. The study of outliers: purpose and model. *Applied Statistics*, pages 242–250, 1978.

[87] R. J. Beckman and R. D. Cook. Outlier. . . . . . . . .s. *Technometrics*, 25(2):119–149, 1983.

[88] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[89] Jorma Laurikkala, Martti Juhola, Erna Kentala, N. Lavrac, S. Miksch, and B. Kavsek. Informal identification of outliers in medical data. In *Proceedings of the 5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, pages 20–24, 2000.

[90] R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica*, 46:33—50, 1978.

[91] P. J. Huber. Robust regression: Asymptotics, conjectures and monte carlo. *Annals of Statistics*, 1:799–821, 1973.

[92] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. 2003.