# Let's see your Digits: Anomalous-State Detection using Benford's Law

Samuel Maurus* Claudia Plant†

## Abstract

Benford's Law explains a curious "naturally-occurring" phenomenon in which the leading digits of numerical data are distributed in a precise fashion. In this paper we begin by showing that system metrics generated by many modern information systems like Twitter, Wikipedia, YouTube and GitHub obey this law. We then propose a novel unsupervised approach called BenFound that exploits this property to detect anomalous system events. BenFound tracks the "Benfordness" of key system metrics, like the follower counts of tweeting Twitter users or the change deltas in Wikipedia page edits. It then applies a novel Benford-conformity test in real-time to identify "non-Benford events". We investigate a variety of such events, showing that they correspond to unnatural and often undesired system interactions like spamming, hashtag-hijacking and denial-of-service attacks. The result is a technically-uncomplicated and effective "red flagging" technique. Although not without its limitations, it is highly efficient and requires neither obscure parameters, text streams nor natural-language processing.

## 1 Introduction

In various domains it is useful to know if the interactions with a running system deviate from expected patterns. A canonical example is financial fraud, where organizations have an interest in exposing any persons attempting to deceive their funding and accounting processes. Other examples include online collaborative ecosystems like Twitter and Wikipedia, which strive to promptly detect and suppress behavior like spamming and attacks which can degrade service quality [17].

On the surface, such manipulative behavior may appear normal and be difficult to detect. A tax cheat may fabricate figures that, although dishonest, lie within expected ranges. Analogously, a spammer may use a team of fake Twitter accounts, each of which have sensible numbers of followers and friends. The manner in which the *absolute values* of such numbers are distributed is often not unusual.

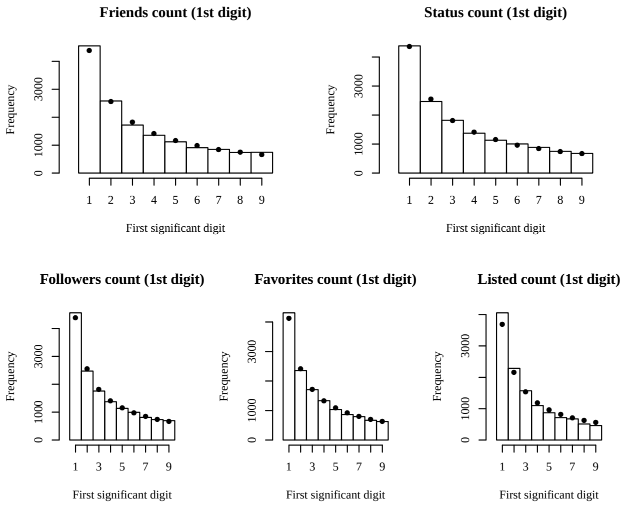


Figure 1: First-digit distributions for the five count-based metrics of 15,000 randomly-chosen Twitter users (friend-, status-, follower-, favorite- and listed-count). The bars represent the measured frequencies, and the filled bullets those predicted by Benford's Law.

The key to the effectiveness of Benford's Law (BL) in such situations lies with its observation that the *leading digits* of many sets of numbers are, by nature, distributed in a surprisingly *non-uniform* way. In contrast, if such numbers are *fabricated*, the leading digits rarely follow that same distribution. For example, the success of BL in financial auditing has shown that the naïve fraudster, in an attempt to contrive believable figures that are varied and "well spread", often fabricates figures with a more or less uniform distribution of leading digits [12]. When doing a Benford analysis, we focus *only* on the leading digits and *purposefully discard the magnitude information.* In this work we propose that monitoring the resulting digit distribution is a novel, robust and elegant mechanism for tracking the integrity of complex systems.

To grasp the basic concept of BL, we share measurements collected from 15,000 randomly-chosen Twitter users. If we consider the number of followers that each user has, we may see that one user has **2**93,845 followers while another has only **3**20. As an exercise in curiosity, how would we expect the distribution to appear when

*Helmholtz Zentrum Munich, Technical University of Munich, samuel.maurus@helmholtz-muenchen.de

†University of Vienna, claudia.plant@univie.ac.at

tallying just the first significant digit (marked in **bold**) of each value? That is, how likely is it that one of our 15,000 users has a follower-count that begins with the digit 9 versus the digit 1? To those unfamiliar with BL, the usual answer is "equally likely".

As Figure 1 shows, however, the distribution is far from uniform. Indeed, it defies intuition: it tells us that finding a follower-count starting with a 1 is *over six times more likely* than finding one starting with a 9. This observation is not unique to this sample. In the same figure we see the distributions for the four other Twitter count-based metrics for each user (friend, listed, status, and favorited counts). All have the same basic form, and we can visually see a strong agreement with the BL predictions in each case. Considering that data from services such as Twitter are considered noisy [2, 7], this strong BL-conformity is a useful property.

In this paper we exploit the BL-conformity of such systems to form an intriguing foundation for describing what it means for systems to be in their "expected" or "natural" state. We then present an approach for the detection of "unexpected" or "unnatural" deviations from that state. Just as Benford's Law can expose fraudulent behavior in financial reports not seen by other tools, we show that the events detected by our framework are not found by state-of-the-art anomaly- and event-detection techniques.

**Contributions:**
**1) We show the Benfordness of key metrics** tracked by Twitter, Wikipedia, YouTube and GitHub (Sections 1, 3 and 4).
**2) We propose a test for Benford's Law conformity** exploiting the law's logarithmic basis and the formal Kolgomorov-Smirnov test (Section 5).
**3) We present BenFound to detect "unnatural" events from streams of *numerical data*.** BenFound does not require a text stream, nor any parameters other than the width $w$ of the sliding window (Section 6). It can be deployed in real-time and has linear run-time complexity in $w$.
**4) We present experiments on synthetic data**, showing that BenFound finds events not visible to state-of-the-art change-point and event-detection techniques that feed on numerical data (Section 9).
**5) We present a number of real-world case-studies** showing how BenFound detects "unnatural" events not found by other techniques (Sections 4, 7, 8).

## 2 Preliminaries

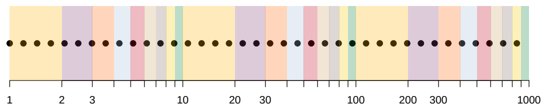
Benford's Law asserts that the probabilities for finding each of the nine possible digits $1, 2, \ldots, 9$ as the first significant digit are *not equal*. It states that the digit 1, for example, occurs more than 30% of the time and the digit 9 less than 5% of the time. More precisely, given a vector $\vec{x} \in \mathbb{R}^n$ of numerical values, BL states that the probability of a randomly-selected element from $\vec{x}$ having the *first* significant digit $d \in \{1, \ldots, 9\}$ is

$$P(D_1 = d) = \log_{10}(d+1) - \log_{10}(d). \qquad (2.1)$$

The bullets (●) in Figure 1 were computed using these probabilities. The general form of BL further specifies the joint probability distribution of *all* the significant digits [5][1].

Although it is clear that not *all* sets of numbers obey the law (a sample of human heights in cm, for example, does not), Benford's original manuscript [3] includes evidence supporting its wide application. Data from diverse sources as baseball statistics, death rates, lists of physical constants and randomly-selected numbers from newspapers were shown to conform. We recommend that the justifiably-skeptical reader perform a simple experiment in this light, such as randomly sampling numerical data from independent web pages.

Many familiar sequences follow BL. These include Fibonacci, the powers of (almost[2]) any integer, and the factorials $(1!, \ldots, n!)$. Such sequences offer an informal starting point for *explaining* BL. It turns out that BL is often (but not always) observed to hold for sets of numbers spanning several orders of magnitude and associated with exponential or multiplicative growth processes. In this light, consider the sequence of numbers generated by the powers of 1.2 (that is, $1.2^1, 1.2^2, 1.2^3, \ldots$) plotted on a logarithmic axis:



It is clear that the values are equally spread out when viewed on a logarithmic scale. It is then obvious that one of these values is more likely to fall into an interval where the first significant digit is one, that is, one of the intervals $[1, 2)$, $[10, 20)$, $[100, 200)$ and so on. We can now understand the BL formula in Equation 2.1 graphically: the probability for any given digit is simply the width of its interval on the logarithmic scale.

With this in mind, let us consider the physiological and psychological reaction to external stimuli that we find in the real world. The growth of the sensation of

[1] Note that the distributions for the second and higher digits quickly approach the uniform.

[2] For base ten it is clear that powers of any integer that is itself a power of ten (e.g. 1000) will not obey the law.

brightness (Fechner's Law), the sense of loudness, the sense of weight, the response of the body to medicine or radiation, and the killing curves under toxins and radiation are all often logarithmic. In economics we often hear about percentage rates of growth. In social media, content can go "viral" with an exponentially growing number of views or downloads over time. As Benford himself elegantly noted: "the analogy is complete, and one is tempted to think that the $1, 2, 3, \ldots$ scale is not the natural scale; but that, invoking the base $e$ of the natural logarithms, *Nature* counts $e^0, e^x, e^{2x}, e^{3x}, \ldots$ and builds and functions accordingly" [3].

Given that BL arises naturally, it should come as no surprise that it is base- and unit-invariant. This means that numbers obeying BL keep obeying BL after we express them in a different base (e.g. octal) or in different units (e.g. feet instead of meters). The mathematical properties of the law are intriguing, and we refer the curious reader to [5] for a formal treatment. Although many facets of BL now rest on solid ground, there remains *no unified approach* that simultaneously explains its appearance in dynamical systems, number theory, statistics and real-world data [4].

In this work we focus on *applications* of the law, so to simplify matters we focus on decimal numbers with no scaling applied. Arguably the most classical application for BL is in fraud-detection. For example, an employee who serially alters the leading digits from 1 to 7 on reimbursement invoices (such that a hotel stay becomes $738.45 instead of $138.45, for example) introduces an artificial and detectable bias to the leading-digits distribution [12].

Our work is based on the idea that different kinds of unnatural events take place in different domains. These events may or may not be malicious in nature. In Twitter, for example, we will see that the lead-up to Father's Day is accompanied by artificial use of the **#FathersDay** hashtag for pushing product sales (spamming and advertising). In such cases we can raise a "red-flag" and investigate the items that violate the digit distribution.

## 3 Further Evidence of Benford's Law Online

Figure 2 shows that, like the Twitter data in our Introduction, the count metrics from YouTube videos and GitHub repositories are close to Benford. For this experiment we collected data from the YouTube[3] and GitHub[4] APIs. We provide the data for reuse and result-reproduction[5].

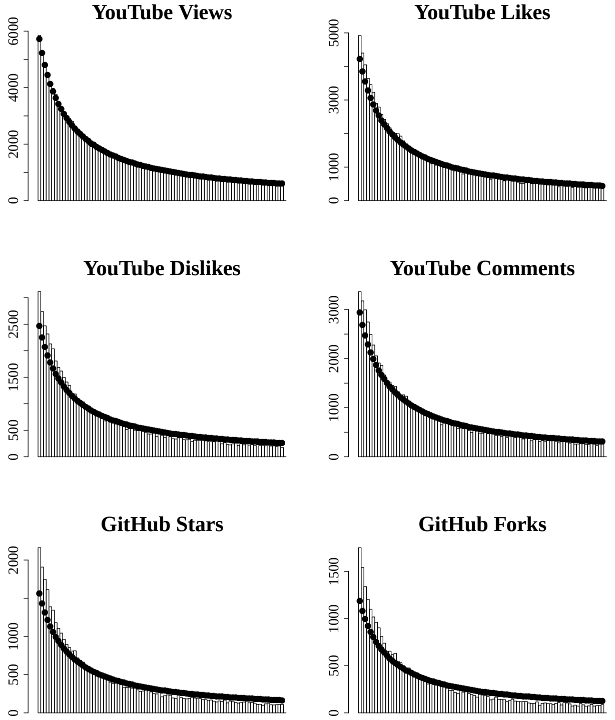


Figure 2: From top left: Leading two-digit histograms for YouTube Views, Likes, Dislikes, Comments, and GitHub Stars and Forks. Each histogram bar on the horizontal axis represents one of the 90 possible leading two digit combinations (10-99).

## 4 Bot-Detection using Benford's Law

In this section we consider the application of Benford's Law in the detection of anomalous behavior in online services in the form of **bots** (where automated computer programs, rather than humans, interact with a system). As a case study we present and share data collected from Wikipedia's real-time Recent Changes (RC) stream[6].

Between July 6 and July 11 2016 we collected $n > 2$ million page-edit events classified by Wikipedia as *non-bot* edits. We recorded the vector $\vec{\Delta} = (\delta_1, \delta_2, \ldots, \delta_n)$, where each $\delta_i$ represents the magnitude in bytes of the corresponding change. Changes ranged from a few bytes to over two megabytes.

Consider in Figure 3 the distribution of the *leading two digits* of this data. From the basic form of the leading-digits distribution it is evident that the data in

[3] developers.google.com/youtube/v3/

[4] developer.github.com/v3/

[5] This manuscript was created using *Knitr*, the engine for dynamic report generation with R. The downloadable source for this document, including all data and embedded R code necessary to generate the results, tables and figures, is available at [todo].

[6] mediawiki.org/wiki/API:Recent_changes_stream

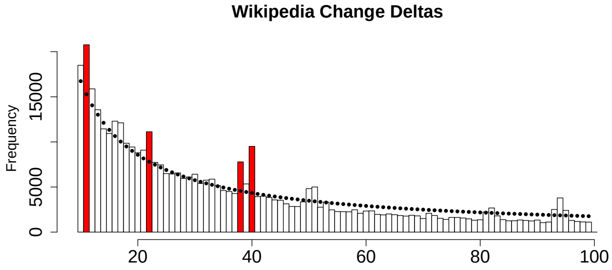


Figure 3: Leading two-digit histogram for Wikipedia page-edit deltas. Red bars are abberations from BL.

question fits the general shape predicted by Benford's Law (again shown by bullets •). However, we see some clear aberrations. For example, noticeable spikes exist for the digit pairs $11, 22, 38$ and $40$. This data is therefore a candidate for "red flagging" and investigation.

Filtering the data to include only the editor comments for the changes with a delta beginning with the digits 40, for example, we quickly notice a significant amount of automation happening, despite the edits being flagged by Wikipedia as *non-bot*. Table 1 contains a sample of the offending comments. We see edits being made by "Addbots" as well as tools like HotCat and Gadget-Merge (automated tools for mass-editing Wikipedia content). The byte counts often *begin* with the digits 40 (despite their full values spanning several orders of magnitude), which manifests as the spike in Figure 3. At the very least, Benford's Law has thus uncovered what appears to be a data-integrity issue in the Wikipedia: many automated transactions are not being suitably labeled.

| **Change comment** | **Bytes** |
|---|---|
| `[[SpiderMum]] [[User:Addbot|Addbot]]` | **40** |
| `Removed using [[Gadget-HotCat|HotCat]]` | **40** |
| `Wbremoveclaims-remove [[P1012,Q902513]]` | **407** |
| `[[MediaWiki:Gadget-Merge]] 0||Q75421` | **4062** |
| `[[OneClickArchiver]]` | **40971** |

Table 1: Examples of automated Wikipedia edits

## 5 Testing Conformity to Benford's Law

We require a mechanism to objectively assess a given numerical sample's conformity to BL. The "state-of-the-art" for such tests was recently surveyed by Nigrini [12]. Broadly speaking, one can partition BL tests into three categories: *single-digit* tests, *all-digits-at-once* tests and tests exploiting the *logarithmic basis* of BL. To evaluate these tests, we consider two geometric progressions (i.e. $a^1, a^2, \ldots, a^n$) with common ratios $a = 1.2$ and $1.8$. From Section 2 we know that each sequence obeys BL exactly in the limit $n \to \infty$.

Table 2 displays the conclusions ("Benford" or "Non-Benford") that each test detailed in [12] makes for our two sequences based on a significance level of 0.05 and sequence length $n = 100$.

| Test | Type | Seq. 1 is | Seq. 2 is | Remarks |
|---|---|---|---|---|
| Z-Stat. | Single | **Benf.** | Non-Benf. | 8 digits deviate significantly for Seq. 2 |
| Chi-Sq. | All | **Benf.** | Non-Benf. | Seq. 2 $\chi^2 = 22.8$ ($9 - 1 = 8$ d.o.f.) |
| Euc. Dist. | All | **Benf.** | Non-Benf. | |
| Hot. $T^2$ | All | **Benf.** | Non-Benf. | |
| JP Sq. | All | **Benf.** | Non-Benf. | |
| MAD | All | **Benf.** | Non-Benf. | Critical values from [12] |
| Mant. Arc | Log | **Benf.** | **Benf.** | |
| K-S Mant. | Log | **Benf.** | **Benf.** | *(Our proposed technique)* |

Table 2: Comparison of BL-conformity tests on two geometric progressions

The tests based on the *discrete digit distribution* ("Single" digit, or "All" at once) fail to label the second sequence as Benford. These techniques are the Z-Statistic, Chi-Square, Euclidean Distance, Hotelling T-square, Joenssen's JP-square and Mean Absolute Deviation (MAD) tests. The conflicting conclusions are troublesome because both sequences are non-random and follow ideal exponential growth. The reason for the conflict is that we selected the second common ratio to maximize the *information-loss during the discretization*. For example, we have a large information-loss when discretizing the sequence value 9.8595 to the "first digit 9" histogram bin. For moderate sample sizes, these discretization errors are enough to change the leading-digit distribution to the extent that these digits-based tests believe that the sequence is non-Benford.

To overcome this weakness, we argue that a BL conformity test should avoid such a discretization. In this light, we can design tests around the *logarithmic basis* of BL, which states that the mantissae[7] of a Benford set are uniformly distributed over $[0, 1)$.

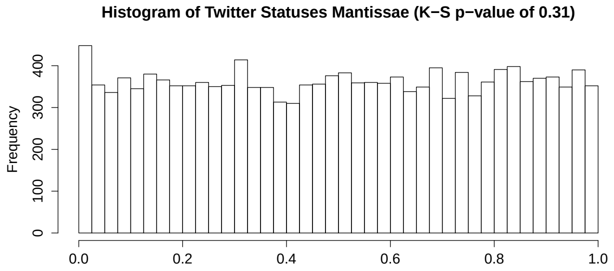
Nigrini [12] also discusses BL tests that exploit this logarithmic basis. The first involves testing for two conditions: if the mantissae have a uniform distribution, then their mean and variance must be $\frac{1}{2}$ and $\frac{1}{12}$ respectively. Nigrini correctly comments that these conditions are insufficient (numerous non-uniform distributions satisfy them) and dismisses this approach.

Another logarithmic-basis test discussed by Nigrini is the recent Mantissa-Arc (Mant. Arc) test [1]. It involves projecting the mantissae around the circumference of the unit circle and testing if the center-of-gravity differs significantly from zero. We see in

[7] The mantissa of a number is the fractional part of its logarithm.

Table 2 that the Mantissa-Arc test is able to correctly diagnose both sequences as Benford. At first glance, this test appears promising, however we soon realize that the center-of-gravity condition is insufficient. Consider, for example, the arbitrary-length sample $10^{1/4}, 10^{3/4}, 10^{1/4}, 10^{3/4}, \ldots, 10^{1/4}, 10^{3/4}$ having only two unique values. These two values are mapped to opposite sides of the unit circle, giving a zero center-of-gravity and incorrectly signaling *perfect* Benford conformity. We therefore suggest to avoid this test.

Why not apply a test which directly compares the empirical cumulative distribution of the mantissae with the cumulative distribution function of the uniform? To the best of our knowledge, this approach has not been suggested. Specifically, we propose applying the formal Kolmogorov-Smirnov (K-S) one-sample test [9] on the mantissae to test whether or not they were sampled from the uniform. Here the null hypothesis is that the mantissae were drawn from the uniform. Adopting the common significance threshold $\alpha = 0.05$, we interpret a $p < \alpha$ as an indicator for non-Benfordness. Consider for example our Twitter status-count mantissae:



The $p$-value corresponding to the application of the K-S test on this sample is 0.31. We hence accept our null hypothesis that the sample was taken from a uniform distribution, and hence that the sample conforms to BL. This same approach correctly diagnoses both our test sequences as Benford (final row in Table 2). We note that the application of the K-S test on a sample involves no obscure parameters – we only need a significance threshold $\alpha$ in order to reach a conclusion for a sample.

## 6 From a Time Series to a Benfordness Signal

Equipped with a tool to measure the Benfordness of a univariate sample, we now turn to the real-time detection of "non-Benford" events. That is, we wish to be alerted when a significant deviation from the "natural" state of a running system occurs.

To achieve this, we track key metrics over time (e.g. follower counts of tweeting users) and apply the K-S Test on a sliding time window. The time window contains a fixed number $w$ of the newest numeric values. For the sample of size $w$ corresponding to each window, we compute the mantissae and apply the K-S test as discussed in Section 5. The resulting $p$-value is the quantity that we then follow over time. We interpret a drop of the $p$-value below a significance threshold $\alpha$ as an indicator for the beginning of "non-Benford" behavior in the system (something unnatural that should be investigated).

**Window Size $w$ and Threshold $\alpha$:** The value for $w$ is user-specified. The K-S test we use for BL-conformity has an asymptotic power of 1. For $w \to \infty$ this implies that there is zero probability of rejecting a true null hypothesis. In practice, however, a large value of $w$ means that we miss out on potentially important dynamics of the system under investigation. At the other extreme, if $w$ is too small, we lack statistical power and thus have a higher false negative rate. As always, $w$ should be chosen considering this trade-off and the application. For our upcoming Twitter case studies (Sections 7 and 8), we will see that a common window size ($w = 2000$) was sufficient.

The value for $\alpha$ can likewise be user-specified, however we will see in real-world case studies (Figure 5) and synthetic experiments (Figure 7) that similar results are obtained for the commonly-chosen threshold levels ($\alpha = 0.01, 0.05$). We use $\alpha = 0.05$ in all our work.

### 6.1 Performance

Assuming a fixed significance threshold $\alpha$ and window size $w$, the K-S critical value can be computed in advance using existing tables for the K-S statistic distribution (for $\alpha = 0.05$ it is $\frac{1.358}{\sqrt{w}}$, for example). Our optimized implementation of BenFound hence works directly with the K-S statistic, avoiding the need to compute $p$-values.

To calculate the K-S statistic for the current window we would normally need to sort its contents. Using a comparison-based sorting procedure like Heapsort implies a $\mathcal{O}(w \cdot \log w)$ run-time in the worst case for this step. In practice, this sorting only needs to be performed once for the first window. Afterwards, we use standard indexing structures to update our sorted window in $\mathcal{O}(1)$ operations. For each new stream value the calculation of the K-S statistic is hence reduced to $\mathcal{O}(w)$, which is the time to determine the maximum difference between the window's ECDF and the reference uniform distribution.

## 7 Case Study: Hashtag Hijacking

In the context of social media, Hashtag Hijacking occurs when a hashtag is (ab)used for purposes other than intended. We now demonstrate BenFound's ability to detect this phenomenon in real-time.

Consider the hashtag **#FathersDay**. The in-

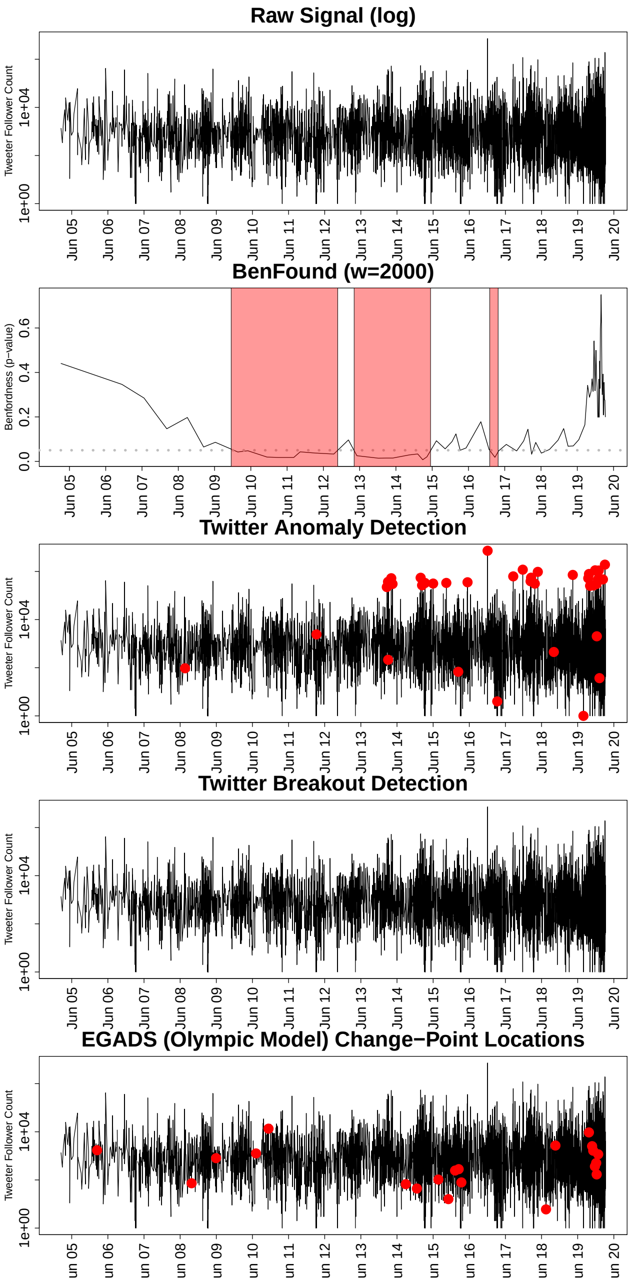


Figure 4: Raw follower counts (top) for users tweeting the hashtag FathersDay. The remaining plots show events, anomalies and changepoints detected by BenFound and other approaches.

tended purpose of this hashtag is to allow Twitter users to celebrate their father. Father's Day occurred in most countries on 19 June 2016. During the time leading up to Father's Day, we listened for and logged all tweets referencing this hashtag. The raw follower counts of the corresponding tweeters are traced in Figure 4. The same Figure shows events, anomalies and changepoints found by BenFound and three state-of-the-art approaches ([10, 8, 18], discussed further in Section 9 and 10). These three approaches consider the *absolute values* of the streamed values (tweeter follower counts).

The BenFound signal shows that the behavior begins Benford, turning non-Benford before Father's Day, and finally regressing to Benford *on* Father's Day. Looking at the data, we quickly see the behavior that the Benfordness signal describes. In the lead-up to Father's Day, a large number of spammers and advertising accounts hijacked the hashtag. The follower-counts of these accounts have a non-Benford (unnatural) digit distribution. This behavior is reflected in the low $p$-value of the Benfordness signal. It first drops below our threshold on 10 June 2016[8], the beginning of the weekend before Father's Day and a logical time for those spammers and advertisers unscrupulously trying to generate profit from the event to begin pushing their content. Example tweets from this time are in Table 3.

On Father's Day itself, spammers and advertisers presumably recognized that time had run out to generate profit from this hashtag. The tweet behavior shifted to the more organic kind of content expected for this hashtag. In Table 3 we find examples of the tweets from 19 June 2016 (users celebrating the efforts of fathers).

The other techniques struggle to extract meaningful information from the raw signal. Two of the techniques find an indigestible number of changepoints/anomalies; the third (Breakout Detection) finds none at all.

**Father's Day lead-up:**
*"Oh Yea! I just entered to #Win LED GlowBowl"*; *"Check out Libbey Ceramic Tiki Mug Blue 7 x 3 16 Ounces"*; *"Enter the #giveaway to #win"*; *"I just entered to #win a $50 giftcard to @cuffdaddy #giveaway"*;
**Father's Day itself:**
*"I love you Dad, Happy Father's Day"*; *"The best thing a man can do for his children is to love their mother"*; *"To all of the champion fathers... Happy #FathersDay!"*; *"Hope you had a great #FathersDay, lads!"*

Table 3: Tweets *leading up to* and *on* Father's Day

## 8 Case Study: Outage Detection

Shortly before the time of data collection, the popular **#PokemonGO** game had been released. We tracked the tweets to this hashtag over a number of days. The Benfordness signal is shown in Figure 5.

The signal shows a sharp drop on July 15 and 16,

[8] We note that times were not measured in an American timezone (rather UTC), hence the reason for the slight shift in the data.

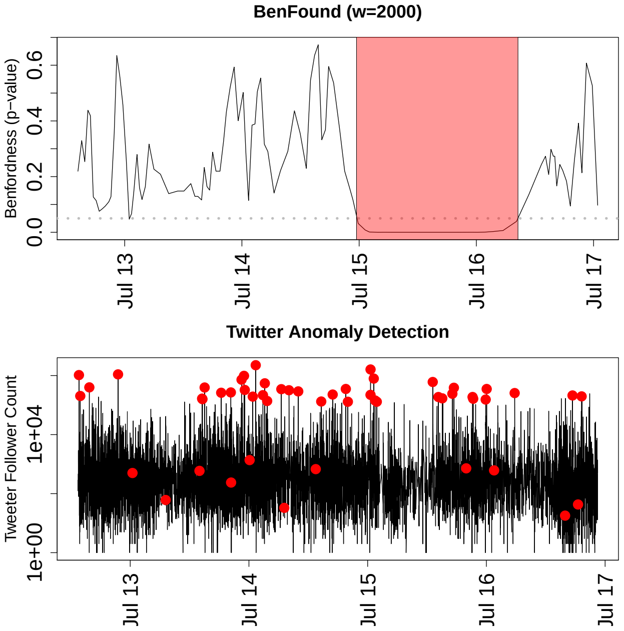


Figure 5: Tracking the Benfordness of follower-counts associated with users tweeting against the #PokemonGO hashtag.

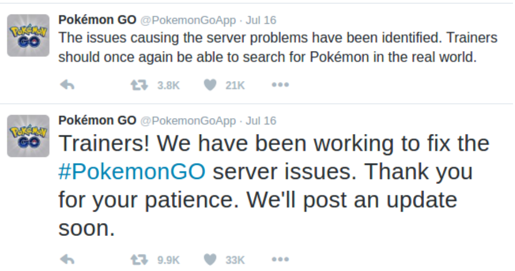
hinting that the "natural" behavior of tweets for this hashtag had changed. An analysis of the data reveals that the the Pokemon GO application suffered severe outages during this time period. The tweets during this time period were largely revolving around these problems (see Table 4) and had a non-Benford digit distribution. The official **@PokemonGoApp** account confirmed the problems on July 16 (Figure 6). As a comparison, the state-of-the-art Twitter Anomaly Detection result is shown in Figure 5, and events/topics detected using two state-of-the-art Twitter text-mining approaches are shown in Table 5.

*SERVER UPDATE: According to @Independent, #PokemonGO servers have been taken down in a DDOS attack.*
*The #PokemonGO servers are down due to a DDOS attack.*
*No word on when they will be back. Stay with us for the latest.*
*Niantic trying to fix the #PokemonGO servers is as successful as Team Rocket trying to kidnap Pikachu.*
*#PokemonGO is down. What do I do now with my time?*

Table 4: Sample of Tweets with the hashtag #PokemonGO on July 15 and 16, 2016

**Top 10 (Twitter NLP [15])**: has in, strict parents, bringing out, **are down**, came over, get off, **go down**, looking for, **been down**, get out
**Top 10 (Twitter Topic Detection (Streaming NMF) [7])**: [argentina, coins, nintendome], [team, argentina, spark], [pokemongonews, argentina, community], [pokecoins, try, need], [argentina, giveaway, must], [try, **servers**, pokecoins], [syrian, hopes, saved], [argentina, coins, lucky], [argentina, need, try], [argentina, **unstable**, lucky]

Table 5: Top 10 events/topics found by two state-of-the-art Twitter text-mining techniques [7, 15].


Pokémon GO @PokemonGoApp · Jul 16
The issues causing the server problems have been identified. Trainers should once again be able to search for Pokémon in the real world.
3.8K 21K

Pokémon GO @PokemonGoApp · Jul 16
Trainers! We have been working to fix the #PokemonGO server issues. Thank you for your patience. We'll post an update soon.
9.9K 33K

Figure 6: Tweets from the verified PokemonGoApp account on July 16, 2016.

## 9 Experiments on Synthetic Data

In this section we use synthetically-generated numerical time-series data to compare our proposed method with state-of-the-art techniques from the field of numerical event- and anomaly-detection. Our generative model is inspired by our observations from online services such as Wikipedia and Twitter. From our introduction we know that normal usage patterns in these services lead to highly-Benford metrics. In contrast, the set of such values coming from "unnatural" usage patterns like bots and hashtag-hijacking is often non-Benford. Our goal is to be alerted when such non-Benford changes occur (denoted a *negative* event), as well as when things return to being Benford (denoted a *positive* event). We make all synthetic models, data and experiments that follow available for reuse[5].

We make the simplifying assumption that one interaction (e.g. page edit in Wikipedia or tweet in Twitter) occurs per unit of time. The $n$ interactions with the system fall at times $t_1, \ldots, t_n$. To evaluate the detection of both *negative* and *positive* events, we divide time into **three intervals**. The first interval spans the range $t_1, \ldots, t_{\lfloor \frac{n}{3} \rfloor}$, the second $t_{\lfloor \frac{n}{3} \rfloor + 1}, \ldots, t_{\lfloor \frac{2n}{3} \rfloor}$ and the third $t_{\lfloor \frac{2n}{3} \rfloor + 1}, \ldots, t_n$. Inside the first and third intervals we generate "natural" (Benford) data. For the second interval we generate non-Benford data. In this way we synthesize a negative event at the transition from the first interval to the second, followed by a positive event at the transition from the second interval to the third.

To generate Benford data for **intervals one and three**, we begin with a uniformly-distributed sample $\vec{u} \in [0, 1]^s$. With reference to Section 5, this sample $\vec{u}$

represents our mantissae. Next we generate the sample $\vec{m} \in \{0, 1, \ldots, 10\}^s$ of integers (uniformly sampled with replacement). This sample $\vec{m}$ represents our "magnitude" information. Our final Benford set of values then satisfies $\log_{10}(\vec{b}) = \log_{10}(\vec{u} + \vec{m})$, thus $\vec{b} = (10^{u_1+m_1}, \ldots, 10^{u_s+m_s})$.

To generate *non-Benford* data for **interval two**, we follow [4] and *uniformly* sample in the range $[0, 10^{11}]$.

Figure 7 compares BenFound with 10 different approaches/configurations to event and change-point detection. For this data we are not able to compare with techniques specific for social media (e.g. Twitter) because they rely on the text feed and particular features (hashtags, retweets). BenFound is the only approach able to identify the correct number of changepoints (two), their locations and the fact that they form a single "event". With respect to **runtime**, BenFound's bandwidth is 8900 streamed values per second when using our R prototype with window size $w = 250$. Doubling to $w = 500$ reduces the bandwidth to 7850 streamed values per second. As a real-world reference, we note that around 6000 tweets are tweeted on Twitter every second[9].

## 10 Related Work and Discussion

BL was investigated in the context of online services for the first time in 2015 [13]. The primary contribution was to show that snapshots of randomly-sampled user metrics in *social networks* (Facebook, Twitter, Pinterest) are Benford. The measurements in our introduction support this conclusion. In addition, we have presented results showing that metrics from Wikipedia, YouTube and GitHub are likewise Benford, and focused on the *real-time* detection of BL *deviations* in such systems.

A small set of dedicated BL reference volumes now exist. For theoretical and practical perspectives we respectively refer to [5] and [12]. The latter focuses heavily on BL in forensic accounting, auditing and fraud detection. It treats BL-conformity tests and also compares BL to the well-known Zipf's Law, however it does not treat event-detection. In Section 5 we reevaluate the conformity tests and propose an alternative based on the formal Kolmogorov-Smirnov test.

The topics of event-, anomaly- and changepoint-detection have seen numerous contributions. Focusing firstly on Twitter-specific techniques, we find in [15] and [7] two state-of-the-art approaches for the extraction of event phrases and topics from Twitter. Unlike BenFound, both rely on processing the tweet text stream. Additionally, whilst BenFound only raises a "red flag" when the digit distribution violates BL, these techniques *continuously* yield a set of e.g. top-10 topics.

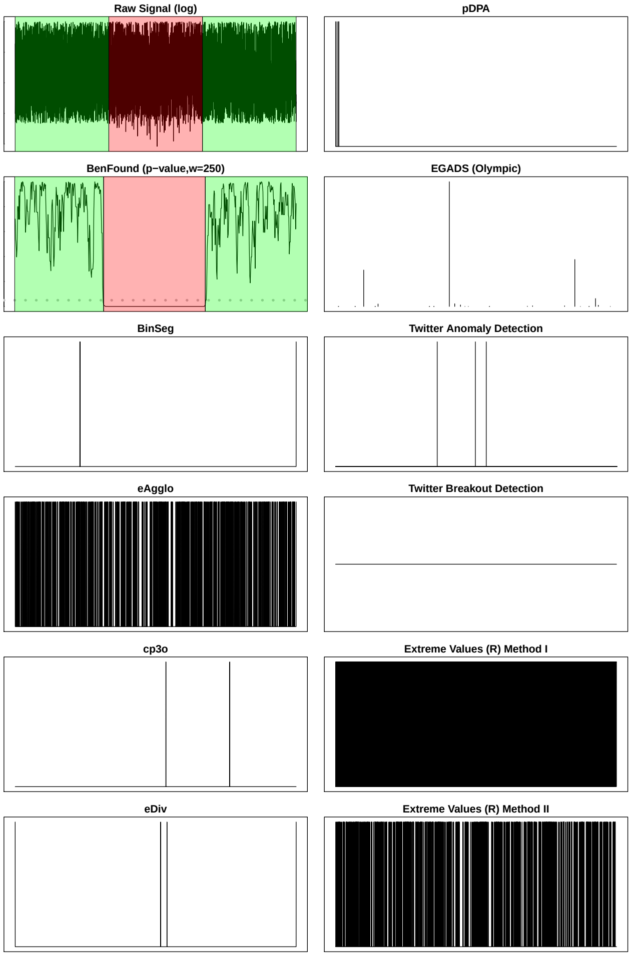


Figure 7: A synthetically-generated time signal (top left) with Benford (green), non-Benford (red) and finally Benford (green) segments. BenFound (row 2, col 1) is the only technique able to correctly detect the change in state and its regression.

As BenFound is not restricted to text and graph structures, it is by no means restricted to social networks. BenFound can be applied to any numerical data stream obeying BL. In searching for comparison techniques we therefore extend our scope to the state-of-the-art in *numerical* anomaly- and changepoint-detection from data-mining and statistics.

The most recent high-level contribution from the data-mining community is named the Extensible Generic Anomaly Detection System (EGADS in Figures 4 and 7) and stems from work at Yahoo [10]. EGADS is a comprehensive framework that supports anomaly-detection based on a variety of configurable models, features and metrics. Two further approaches come from

[9] internetlivestats.com/twitter-statistics

Twitter [8, 18]. Twitter Anomaly Detection [8], for example, employs time-series decomposition using statistical metrics to detect both global and local anomalies in the presence of seasonality and an underlying trend.

From the recent statistical literature we find in [11] the non-parametric approach eDiv that automatically detects both the number and location of change points. Unlike previous methods which typically focus on detecting a change in mean, variance or kurtosis, eDiv is able to detect any distributional change. From the same work we find eAgglo, the bottom-up variant of eDiv, and another probabilistic pruning method cp3o. These three methods are compared to BenFound in Figure 7. Each is designed for an offline setting and has a quadratic time complexity in the sample size $n$.

The pruned dynamic programming algorithm [14] (pDPA in Figure 7) is an offline approach which aims to infer both the number and positions of the change-points from the data. It uses a functional cost representation of segments in order to prune the search space of the segment neighborhood. It requires a user-specified parameter $k_{max}$ which represents the maximum number of change points to find. Its computational complexity is quadratic in the number of data points $n$.

Finally, we compare to Binary Segmentation [16, 6] (BinSeg in Figure 7). This early approach detects changes in the mean and is commonly used as a baseline.

Importantly, all of these approaches consider the distribution of the *complete* numerical values in question. BenFound, in contrast, focuses on a particular kind of change event: a change in its leading digit distribution. This is achieved by considering BL and purposefully *ignoring the magnitude information* of the temporal measurements. Fundamentally then, BenFound distances itself from the discussed state-of-the-art in that it does not look for changes to the underlying distribution of *complete* values, but rather that of only the *leading digits*. For this reason, BenFound has been able to correctly identify events in synthetic and real data (Sections 7, 8 and 9) that are not found by any of the comparison techniques.

Finally, our technique has **limitations**. Firstly, it is clearly only applicable to numeric data that obeys BL, meaning that it is inappropriate for many applications. Secondly, it requires a manual inspection step to analyze the cause for found events. Finally, it is not immune to false negatives. That is, if parties that artificially manipulate a system are aware of BL, they may tune their interactions such that the system metrics are not significantly affected from a BL perspective.

## 11 Conclusion

Various metrics from online services such as Twitter, Wikipedia, YouTube and GitHub naturally obey BL. When these metrics violate BL in real-time, it is often a sign of significant “unnatural” behavior. In this work we have proposed BenFound, a real-time event-detection approach for “red-flagging” such violations. In the case of online services, the interactions may be anti-social or malicious in nature, like non-permitted bot activity or hashtag-hijacking. We have compared BenFound to state-of-the-art event-detection techniques in controlled settings with synthetic data. Finally, we have deployed our technique to real-world settings and demonstrated practical knowledge discovery in various domains. BenFound is non-parametric, efficient and can be deployed in real-time on numerical data streams that obey BL.

## References

[1] J. Alexander, “Remarks on the use of Benfords Law”, 2009 (available at SSRN 1505147).

[2] É. Antoine, A. Jatowt, S. Wakamiya, Y. Kawai, and T. Akiyama, “Portraying collective spatial attention in twitter”, KDD 2015.

[3] F. Benford, “The law of anomalous numbers”, in *Proceedings of the American Philosophical Society*, vol. 78, 1938, pp. 551–572.

[4] A. Berger, and T.P. Hill. “Benfords Law strikes back: No simple explanation in sight”, in *The Mathematical Intelligencer*, vol. 33, 2011, pp. 85–91.

[5] A. Nigrini, and T.P. Hill, *An Introduction to Benford’s Law*. Princeton University Press, 2015.

[6] A. Edwards and L. Cavalli-Sforza, “A Method for Cluster Analysis”, in *Biometrics*, vol. 21, no. 2, 1965.

[7] K. Hayashi, T. Maehara, M. Toyoda and K. Kawarabayashi, “Real-time topic detection on twitter with topic hijack filtering”, KDD 2015.

[8] NA. James, AK. Kejariwal and DS. Matteson, “Leveraging Cloud Data to Mitigate User Experience from Breaking Bad”, arXiv preprint arXiv:1411.7955, 2014.

[9] A. Kolmogorov. “Sulla determinazione empirica di una legge di distribuzione”. Italian Actuarial Journal, 1933.

[10] N. Laptev, S. Amizadeh, and I. Flint, “Generic and scalable framework for automated time-series anomaly detection”, KDD 2015.

[11] D. Matteson and N. James. “A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data”, in *Journal of the American Statistical Association*, vol. 109, no. 505, 2014, pp.334-345.

[12] M.J. Nigrini, *Benford’s Law: Applic. for Forensic Acc., Auditing and Fraud Det.*. John Wiley & Sons, 2012.

[13] J. Golbeck. “Benford’s Law Applies to Online Social Networks”, in *PloS one*, vol. 10, 2015.

[14] G. Rigaill, “Pruned Dynamic Programming for Optimal Multiple Change-Point Detection”, arXiv:1004.0887.

[15] A. Ritter, Mausam, O. Etzioni, S. Clark, "Open Domain Event Extraction from Twitter", KDD 2012.

[16] A. Scott and M. Knott, "A Cluster Analysis Method for Grouping Means in the Analysis of Variance", in *Biometrics*, vol. 30, no. 3, 1974, pp. 507–512.

[17] Twitter Inc. "Twitter Terms of Service". Online Resource. https://www.twitter.com/tos.

[18] O. Vallis, J. Hochenbaum and A. Kejariwal, "A novel technique for long-term anomaly detection in the cloud", in 6th USENIX (HotCloud 2014).