2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Wiki-watchdog: Anomaly Detection in Wikipedia Through a Distributional Lens

Chrisil Arackaparambil
Dept. of Computer Science, Dartmouth College
cja@cs.dartmouth.edu

Guanhua Yan
Information Sciences, Los Alamos National Lab
ghyan@lanl.gov

***Abstract*—Wikipedia has become a standard source of reference online, and many people (some unknowingly) now trust this corpus of knowledge as an authority to fulfil their information requirements. In doing so they task the human contributors of Wikipedia with maintaining the accuracy of articles, a job that these contributors have been performing admirably. We study the problem of monitoring the Wikipedia corpus with the goal of *automated, online* anomaly detection.**

We present Wiki-watchdog, an efficient *distribution-based* methodology that monitors distributions of revision activity for changes. We show that using our methods it is possible to detect the activity of bots, flash events, and outages, as they occur. Our methods are proposed to support the monitoring of the contributors. They are useful to speed-up anomaly detection, and identify events that are hard to detect manually. We show the efficacy and the low false-positive rate of our methods by experiments on the revision history of Wikipedia. Our results show that distribution-based anomaly detection has a higher detection rate than traditional methods based on either volume or entropy alone. Unlike previous work on anomaly detection in information networks that worked with a static network graph, our methods consider the network *as it evolves* and monitors properties of the network for changes. Although our methodology is developed and evaluated on Wikipedia, we believe it is an effective generic anomaly detection framework in its own right.

I. Introduction

Wikipedia is a popular source of information content online. As an online encyclopedia, its content has been generated by its community of contributors which is open to anyone wishing to join. This novel methodology of open and collaborative information gathering and organization has led to some impressive results. As of August 2010, there were 3.4 million articles (we will also refer to them as pages) in the English language Wikipedia, and 16.7 million articles when considering all languages [5]. These articles span a wide and eclectic mix of topics ranging from the most popular (such as sports and celebrities) to the most obscure (e.g., List of Middle-earth inns, Permian Tetrapods etc.). In our experience, more people have begun relying on Wikipedia as their first source of information, and are even trusting it to be accurate. As such, this corpus of information wields great power of influence, and many commercial entities are interested in leveraging this power to their advantage. Wikipedia pages commonly appear as the first result in many search queries.

Due to the open nature of Wikipedia, its contributors are additionally tasked with the job of ensuring the relevance and accuracy of articles. Furthermore, the maintainers of the network are charged with ensuring the continued availability and integrity of the network, and providing interesting features to its readers using the knowledge corpus. While the contributors have been doing an admirable job of monitoring and correcting articles, an obvious open question is to design methods to automate some of those processes. In fact, there has been an entire workshop track recently [1] devoted to developing machine learning approaches for detecting vandalism attempts in Wikipedia. The focus of this workshop, however, was limited to classifying a given input text of an article revision as vandalism or not, after appropriate training of the classifier.

In this work, we present Wiki-watchdog, an efficient distribution-based methodology for anomaly detection in Wikipedia. It is intended to support the monitoring of the contributors and maintainers of the network. It is novel in that it is a holistic distribution-based methodology. That is, the goal of our methods is to monitor different aspects of distributions (or histograms) of the revision timeseries of Wikipedia for significant changes. Previous work on anomaly detection (e.g., [14] [12] [10]) considered individual distribution based-metrics like volume, entropy, KL-divergence etc., to compare distributions (KL-divergence is not strictly a metric; we use term metric loosely). Each of these metrics captures a few aspects of the distribution(s) in question to varying degrees. Wiki-watchdog generalizes this approach and employs several efficient distribution-based metrics (and allows for more) that are collectively more powerful for anomaly detection. Each metric is efficient for monitoring massive volumes of timeseries data by applying appropriate data stream algorithms available. Experiments on the timeseries demonstrated that Wiki-watchdog has a higher detection rate than monitoring just entropy or volume of distributions. The rate of false positives is generally considered to be a problem with anomaly detection methods. In our experiments, Wiki-watchdog exhibits a very low rate of false positives—over the course of five years only two of the 64 anomalies flagged, were false positives.

Unlike previous such work that consider (offline) anomaly detection in static network graphs (e.g., [6]), our methods can be implemented efficiently online, and take into account properties of the network that evolve over time. By monitoring distributions of revisions/page and revisions/contributor as they change, we identify three kinds of anomalies: the activity of bots, flash events, and outages.

Wikipedia bots: Wikipedia provides a programming interface for the development of bots—programs automating certain tasks that are tedious to perform manually. For instance, a bot may pull demographic data from a government database and plug it into pages for towns and cities. Another example is the correction of certain frequent typographical errors. Detecting pronounced bot activity is important to the maintainers of the corpus as it can warn them in the case of rogue bots (one of the bots revealed by our analysis had been banned from Wikipedia [2] due to the nature of its activity)

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.86

and of bots triggered by accident (many bot developers provide a means to disable the bot when this happens).

Flash events: Our methods revealed a number of flash events—events resulting in intense editing activity on a small set of pages, usually due to a particular newsworthy event. Some well-known flash events identified are Hurricane Katrina and the Virginia Tech Massacre, by the editing activity that ensued due to these events. Detection of such events is useful in early knowledge of news events as they develop, much like what is provided by the trending topics feature of Twitter. This kind of detection is also useful for identifying pages being abused, for example, due to edit wars [19].

Outages: By our methodology, we were also able to detect instances of outages, either to all of Wikipedia, or to certain functionality within it. The capability of detecting different kinds of outages is important to the maintainers of Wikipedia because, in certain cases the outage is not immediately obvious. For instance, one of the outages discovered in our experiments only affected the functionality of bots.

Our dataset: To view the evolution of Wikipedia, and measure evolving properties for our experiments, we construct a dataset of timeseries of updates to Wikipedia (i.e., sorted by timestamp). Each update consists of the time of the update, the name of the updated page, the contributor responsible for the update, and the internal Wikipedia links that were modified. The nature and size of the dataset makes its construction a non-trivial process; we describe it briefly in Section II.

Other applications: The methods we provide here can find applications in other centralized information and social networks as well, e.g., Facebook and Twitter. Our methods can be used for mining trends and events in streams generated in those networks. For example, analysis of the stream of addition and deletion of links to the network graph can provide insight into evolving trends of user centrality and connectivity. And applying our methods to streams of message postings can reveal properties of user relationships. The availability of data stream algorithms makes it feasible for the operators of these networks to implement our methods scalably in the face of the massive update streams produced.

Other related work

The survey by Chandola *et al.* [11] presents a classification of anomaly detection methods by application and technique. Our techniques do not neatly fall into any specific class therein. Distribution-based metrics previously used individually for anomaly detection include volume [10], entropy [15] [14], conditional entropy [9], and KL-divergence [12].

Leskovec [16] studied time-evolving properties of networks, developed models governing that evolution, and also suggested anomaly detection in evolving networks. Almeida *et al.* [7] study the evolution of the Wikipedia network with the goal of understanding behavior of users, article revisions, and processes behind them. Sun *et al.* [18] provide methods to detect community structures and their evolution in networks.

The rest of the paper is organized as follows. We describe our dataset in Section II, and the distributions and metrics we use in Section III. We present our anomaly detection methods in Section IV, and experimental results in Section V.

II. Dataset Extraction

We now describe the dataset used in our experiments and the process we use to extract it. Our dataset is extracted from the English-language Wikipedia dump [4] dated January 30, 2010 made available by the Wikimedia Foundation. The dump is a 32GB compressed XML file that decompresses to 6TB. It contains the entire text of all 223 million revisions on the 3 million pages in Wikipedia, starting with the first revision on January 16, 2001. In our dataset we consider only pages in the Main "namespace" of Wikipedia. Other examples of namespaces include the User namespace containing pages about contributors, and the Talk namespace containing pages with user discussions about revisions to other pages. Revisions in the dump are grouped by page, but neither the page entries in the dump nor the revision entries within a page entry are sorted in any particular order. As our goal in this work is to study the properties of Wikipedia as they evolve in time, it is necessary to have a dataset supporting such analysis. Our dataset was extracted from the dump using a 28-node cluster to distribute computational load. It consists of a sequence of revisions sorted by timestamp. An example of a revision is:

```
1256478934 <t>Random_phase_approximation</t>
<c><username>Bodinio</username>
<add>[Keith_Brueckner]</add><del>[Brueckner]</del>
```

The Unix timestamp for this particular revision is 1256478934. The page on "Random phase approximation" was revised by the contributor "Bodinio". As a result, a link to the page "Brueckner" was deleted and a link to the page "Keith Brueckner" was added. The record of the changes to the links is made in our dataset construction to enable future research.

III. Distributional Analysis

In this section we present the motivation for distributional analysis. We also present the distributions we consider for analysis, and the metrics we use to analyze those distributions.

Since the input to our anomaly detection methods is a stream of updates to Wikipedia, our methods must be able to monitor this stream efficiently to produce their results. Each update in the stream contains the name of the page updated, and the contributor responsible. Two natural distributions (histograms) that can be considered from this input stream are: (1) the distribution over the set of pages, of the number of revisions per page; and (2) the distribution over the set of contributors, of the number of revisions per contributor. Then, we can break down our timeline into a series of windows (say, each window being a day), consider the above distributions in each window, and track how they change over the windows.

A. Why Distributional Analysis?

In the context of Wikipedia (and social and information networks in general), distributional analysis has two advantages. First, the contributors of Wikipedia monitor only those pages they are interested in. Distributional analysis looks at Wikipedia as a whole, and provides a multidimensional view of revision activity from several different perspectives. This can reveal anomalies that cannot be observed by individual

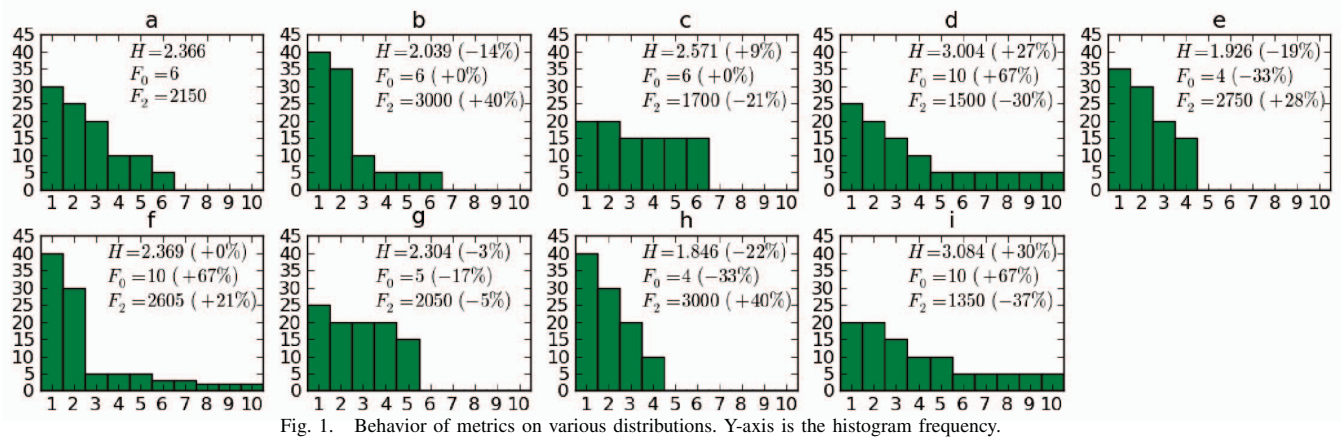

Fig. 1. Behavior of metrics on various distributions. Y-axis is the histogram frequency.

contributors. Second, it is not feasible for even the maintainers of the Wikipedia corpus to manually comb through the stream of updates looking for abnormal patterns. In this light, monitoring of distributional properties offers a good aggregate view of the stream of updates. Entropy is one distributional metric that has previously been shown to be useful [14] [17] for monitoring network traffic for detecting DoS attacks, flash crowds, and port scanning. In this work we show more generally that distributional analysis, using several distributional metrics to capture different aspects of the distribution, is a more effective method for anomaly detection.

Additionally, the technique has another important advantage: the technique is computationally effective in the analysis of data streams. There has been a lot of activity recently in the development of data stream algorithms for computing the metrics we use in this work.

B. Data stream algorithms

Data stream algorithms research largely began with the seminal paper of Alon *et al.* [8]. Since then the area has received a lot of research attention in line with the explosive growth of information streams on the Internet, e.g., update streams in social networks, traffic through Internet backbone routers, web search queries, database updates etc. The core problem is to estimate useful stream statistics in an online fashion. The challenge is to deal with massive stream sizes, and perform memory efficient computation.

A data stream is defined as a sequence $\sigma = (a_1, a_2, \ldots, a_m)$, where $a_i \in \{1, 2, \ldots, n\}$. Here, the stream has m items in it, and the items in the stream are drawn from a universe of size n. Then, the problem is that we would like memory-efficient compututation of some metric $f(\sigma)$. The metrics f that we will consider will depend only on the frequency distribution of the items in the stream. In particular, f is invariant under permutations to σ. Data stream algorithms research is concerned with the computation of metrics f in a memory efficient manner, without having to store either the entire stream or the entire frequency distribution in memory at any given time: consider that the length of the stream is massive (e.g. packets arriving at a backbone router), and its universe is also massive (e.g. the universe of web search queries). The metrics we consider in this work are presented next, along with a note of some of the algorithms available. The algorithms are probabilistic approximation algorithms; i.e., with high probability these algorithms produce values that are close to their target metric values.

C. Metrics

We now describe the metrics we use in this work, and some of their properties. Following this description, we compare the values of the metrics under different distributions.

- Length (volume) of the stream, m: Although this is a simple distributional metric, it is still able to capture some important anomalies. For instance, when there are outages to the Wikipedia network, the volume of updates may be affected directly. This metric is also useful for normalizing the values of the other metrics, as we will explain in the next section.
- Entropy, $H(\sigma)$: This metric has received a lot of attention in the context of network traffic monitoring for anomalies. Its value is given by $H(\sigma) = -\sum_{1 \le i \le n} p_i \log p_i$, where $p_i = f_i/m$. Entropy is an information-theoretic metric that captures certain aspects of the shape of the distribution. The entropy of a distribution is a maximum of $\log m$ when the frequencies of elements are uniform (but its value depends on the number of elements in the support of the distribution). The introduction of elements of very high frequency into the distribution causes the entropy to decrease, while an increase in the number of elements in the distribution usually results in an increase in entropy. Many anomalies exhibit such effects, as will be seen in our experiments.
- Zeroth frequency moment, $F_0(\sigma)$: The value of this metric is the support size of the distribution, i.e., the number of elements i in the distribution with frequency $f_i > 0$. This metric is useful because it can identify anomalies where new elements appear in (or old elements disappear from) the distribution.
- Second frequency moment, $F_2(\sigma)$: The value of this metric is given by $F_2(\sigma) = \sum_{1 \le i \le n} f_i^2$. This metric enables us to identify when the distribution gets skewed

towards a few elements in the support. This follows from the inequality $(a+b)^2 > a^2 + b^2$, whenever $a, b > 0$.

The frequency moments defined above are generalized as follows: $F_p = \sum_{1 \le i \le n} f_i^p$, for $p \ge 0$. When $p = 0$, we assume that $0^0 = 0$, to get F_0. Note also that $F_1 = m$ (the volume). Data stream algorithms to estimate the frequency moments and entropy are presented in the works of Alon *et al.* [8] and Harvey *et al.* [13] respectively.

D. Behavior of metrics

We now show the behavior of our metrics on distributions with different shapes. Each of the distributions in Figure 1 have the same value of F_1 (the stream length), to ensure that we only examine the effect of distribution shape, and avoid variation in volume. In our detection methodology we normalize the metrics (except F_1 itself) using F_1, to discount the effect of daily variation in F_1 on the metrics.

In Figure 1, Plot (a) shows the normal distribution for this analysis, against which we compare the other plots. Plots (b) and (c) show distributions that are more and less skewed respectively. Plots (d) and (e) show distributions that have a larger and a smaller support respectively. The remaining plots show distributions combining features from Plots (b) through (e). Each plot (except Plot (a)) also shows the values of the metrics and their relative changes (expressed as percentages) when compared to the normal distribution in Plot (a).

We can observe from the metric deviations in Plots (b) through (e) that entropy is able to capture the change in the skew and the support size of the distributions. We also see in those plots that F_2 is able to detect change in the distribution skew, and F_0 is able to detect change in the distribution support size. Now, looking at Plots (f) and (g) we see that although the distribution shapes are significantly different from that in Plot (a), entropy does not register the changes. On the other hand, both F_0 and F_2, however, are able to detect the changes. On comparing Plot (b) and Plot (e) to Plot (a) we find that although the entropy in the two distributions change by about the same amount, that deviation does not indicate *how* the distributions have changed. On the other hand, monitoring F_0 and F_2 tells us the changes in the two distributions are due to the increased skew and decreased support size of the two distributions respectively. To an extent, the same is the case when comparing Plot (c) and Plot (d) to Plot (a).

Again, the entropy in Plot (h) increases significantly due to the combined reinforcing effects of decreased skew and increased support size. But it is only observation of the values of F_0 and F_2 that reveal the reason behind the change in the distribution.

The implication of the above analysis is that it is not sufficient merely to monitor the entropy, and that monitoring other distributional metrics like F_0 and F_2 yields more knowledge of the shape of the distribution and any changes in it. This is useful for increasing the anomaly detection rate, as well as in diagnosis of anomalies when they are found.

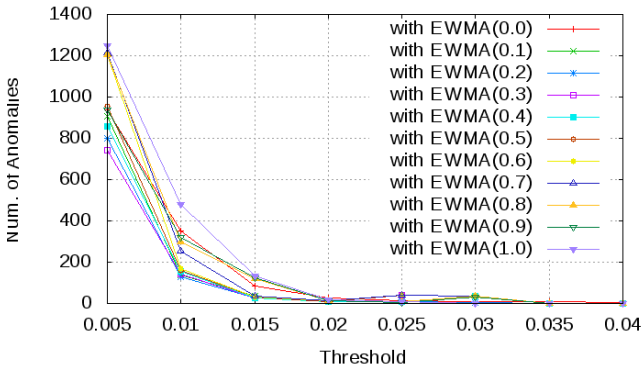

Fig. 2. EWMA(α) anomaly curves for H(page dist.) [Best viewed in color.]

IV. The Wiki-watchdog Detection Algorithm

In this section, we describe our anomaly detection methodology. We divide the input update streams into consecutive windows $W_1, W_2, \ldots$, each of some fixed time interval parameter T. In our experiments, we found that setting T to be 24 hours was most beneficial, since most of our anomalies tend to last from several hours to a few days. Within each interval W_i we compute our metrics on the sub-stream S_i in the interval. Thus, the metric values v_i over the intervals form a timeseries. As mentioned previously, we also normalize the value v_i of each metric to get $\hat{v}_i$ using the sub-stream length $m_i = F_1(S_i)$, to discount the effects of changing length on metric values. For the metrics H, F_0, F_2, we use the normalization factors $\log(m_i), m_i$, and m_i^2 respectively for normalization. These terms are the maximum possible values of the respective metrics for given length m_i. After normalization, each metric has a value $\hat{v}_i$ in $[0, 1]$, which we can then compare to its values from other time windows. We then monitor each such timeseries $\hat{v}_1, \hat{v}_2, \ldots$ for significant deviations, using an Exponential Weighted Moving Average (EWMA) based scheme that can be applied in an online fashion. The EWMA works as a filter to smooth out local fluctuations in the timeseries. This helps in avoiding false positives/negatives due to noise in the timeseries. We compute the EWMA E of the concerned timeseries online, and compare each new value $\hat{v}_i$ of the timeseries against the EWMA value at the time. If the relative deviation $|\hat{v}_i - E|/E$ of the new value from the EWMA exceeds a given threshold parameter τ then we flag the new value as an anomaly.

Detection Algorithm Input—stream S_i in window W_i

1) Obtain the metric value v_i using a data stream algorithm, and normalize to $\hat{v}_i$.
2) If $|\hat{v}_i - E|/E > \tau$, flag window W_i as an anomaly, and ignore $\hat{v}_i$ and the values $\hat{v}_{i+1}, \hat{v}_{i+2}, \hat{v}_{i+3}$ from the three following windows (unless a false positive was determined) when updating E (in the next step).
3) Otherwise, update $E := E + \alpha(\hat{v}_i - E)$.

Parameter α is the EWMA multiplier. It determines how much weight is given to the historical values of the timeseries versus the newer values when computing the EWMA. The EWMA

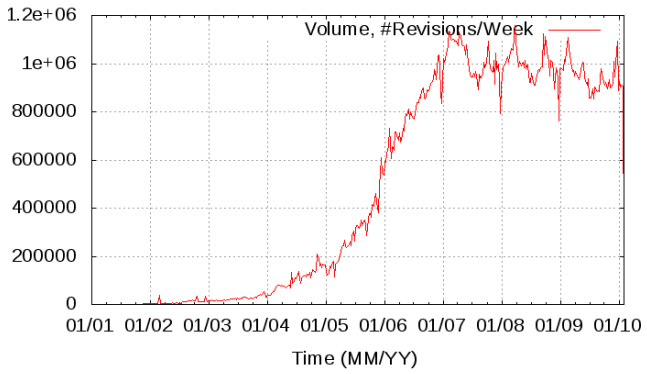

Fig. 3. Volume of weekly revisions

value E is initialized as the average of the metric values from the first few windows. In our experiments we used the values from the first week for initialization. When updating the EWMA, we exclude metric values from anomalous windows, and values from three following windows (unless the anomaly was determined to be a false positive). This is done to avoid the anomaly from destabilizing the EWMA. In a few cases we found that the anomaly spilled somewhat over the boundary of the day, and sometimes continued with a lower intensity over the following days. The metric values in the following windows, however, did not deviate sufficiently enough to be flagged an anomaly. So, to avoid such effects from compromising the stability of the EWMA we ignore those values.

Parameter choices: We now describe how the parameters α and τ were chosen in our experiments. Our first step was to look at the timeseries and determine visually which points appeared to be anomalies (prominent dips or spikes in the plot). This gave us a set of anomalies to work with, and we picked the two parameters to allow for this set of anomalies to be detected. A higher value of α gives more weight to the newly observed value in the EWMA. Given the noisy data we found that setting $\alpha = 0.3$ was a good compromise between detection capability and the smoothing effect of the EWMA. Figure 2 shows the effect of changing the two parameters on the number of anomalies reported by the detection method. For setting the threshold τ we used the the "elbow" in the anomaly detection curve as a reference (e.g., for entropy of the page distribution see Figure 2; we set the threshold to 0.014). The elbow is the point when reducing τ any further results in an exponential increase in the number of anomalies. It is the turning point between the threshold being set too high and too low. We chose a value of τ near the elbow, while also ensuring that our visually identified anomalies were captured.

V. Experimental Analysis

In this section we present the results of our experimental analysis on the Wikipedia dataset described previously. Any anomaly detection scheme requires a stable system to start with, to ensure there is a baseline norm present to compare anomalies against. We found that in the early years of Wikipedia, contributor activity kept increasing before leveling off around 2005. Figure 3 shows the number of edits per week.

	Page			Contrib.				
	H	F_0	F_2	H	F_0	F_2	F_1	Comments
Jan-21-05							+	* False positive
Feb-22-05		+	+	+	+		-	[O] Power failure; no editing
Feb-23-05					+		-	" " "
Feb-24-05					+		-	" " "
Mar-16-05							-	[O] Database outage; editing disabled
Apr-19-05	-		+					[F] Pope Benedict XVI elected
Apr-20-05	-		+					" " "
Apr-21-05			+					" " "
Apr-30-05	+	+		-	-	+	+	[B] Flabot (updating interlanguage links)
May-01-05	+	+		-	-	+		" " "
Jun-07-05							-	[O] Planned outage
Jun-27-05			+		+		-	[O] MediaWiki update; editing blocked
Jul-07-05	-		+					[F] London bombings
Jul-08-05			+					" " "
Jul-16-05	-		+					[F] Harry Potter and the Half-blood Prince released
Jul-20-05			+					[F] (US politics) John Roberts nominated to Supreme Court
Aug-29-05			+					[F] Hurricane Katrina
Dec-25-05							-	[H] Christmas
Feb-05-06	+			-				[B] D6 (adding to Category:Living People)
Feb-13-06							-	[O] Minor downtime
Sep-04-06			+					[F] (Accident) Death of Steve Irwin
Oct-08-06						+		[B] SmackBot
Dec-09-06	+							* False positive
Dec-25-06							-	[H] Christmas
Dec-30-06			+					[F] Execution of Saddam Hussein
Apr-16-07			+					[F] Virginia Tech Massacre
Apr-17-07	-		+					" " "
Apr-18-07			+					" " "
Oct-13-07	+	+		-				[B] CapitolBot (Infobox, towns/cities)
Dec-22-07	+	+		-				[B] DetroiterBot (infobox params, formatting, townships/counties)
Dec-25-07							-	[H] Christmas
Jan-19-08						+		[B] SmackBot
Sep-18-08	+	+		-		+	+	[B] LightBot + Anomebot (demographics) + DinoBot2 (movie rating templates)
Sep-19-08	+	+		-	-	+	+	" " "
Sep-20-08	+	+						" " "
Sep-21-08	+	+						" " "
Oct-05-08				-				[B] LightBot
Nov-03-08						+		[B] LegoBot
Nov-07-08						+		[B] LightBot (date audits)
Nov-08-08						+		" " "
Nov-09-08	+			-		+		" " "
Nov-10-08						+		" " "
Nov-12-08						+		" " "
Nov-22-08	+	+						[B] YoBot (Category adding)
Dec-25-08							-	[H] Christmas
Dec-30-08	+			-		+		[B] LightBot (units/dates/other)
Jan-03-09				-				[B] LightBot
Jan-31-09	+	+		-		+		[B] Cydebot (moving categories)
May-16-09	+	+		-		+		[B] D6 (formatting headline levels, fixing Unicode in templates)
Jun-24-09				-				[B] BOTijo
Aug-07-09						+		[B] Erik9bot
Aug-08-09						+		" " "
Sep-17-09	-	-		+				[O] Bugs with MediaWiki update; bots stopped working
Nov-28-09	+							[B] AnomieBot (editing IPA phonetic symbols) + Full-date unlinking bot
Nov-29-09	+							" " "
Dec-16-09	+	+		-		+	+	[B] SmackBot (date maintenance etc.)
Dec-17-09	+	+		-	-	+	+	" " "
Dec-22-09	+	+		-	-	+	+	[B] SmackBot (delink dates)
Dec-23-09	+	+		-	-	+	+	" " "
Dec-24-09	+	+		-	-			" " "
Dec-25-09	+	+					-	[H] Christmas
Dec-27-09	+	+		-	-	+	+	[B] SmackBot (delink dates)
Dec-28-09	+	+		-			+	" " "

TABLE I
ANOMALIES: [B]=BOT, [F]=FLASH EVENT, [O]=OUTAGE, [H]=HOLIDAY

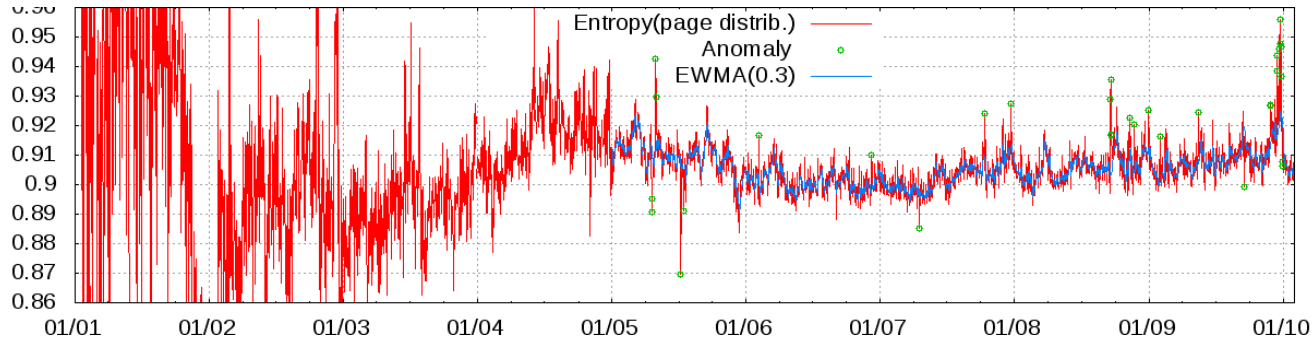

Fig. 4. Timeseries of H(page dist.), when including data prior to 2005. Detection method only applied from 2005. [Best viewed in color.]

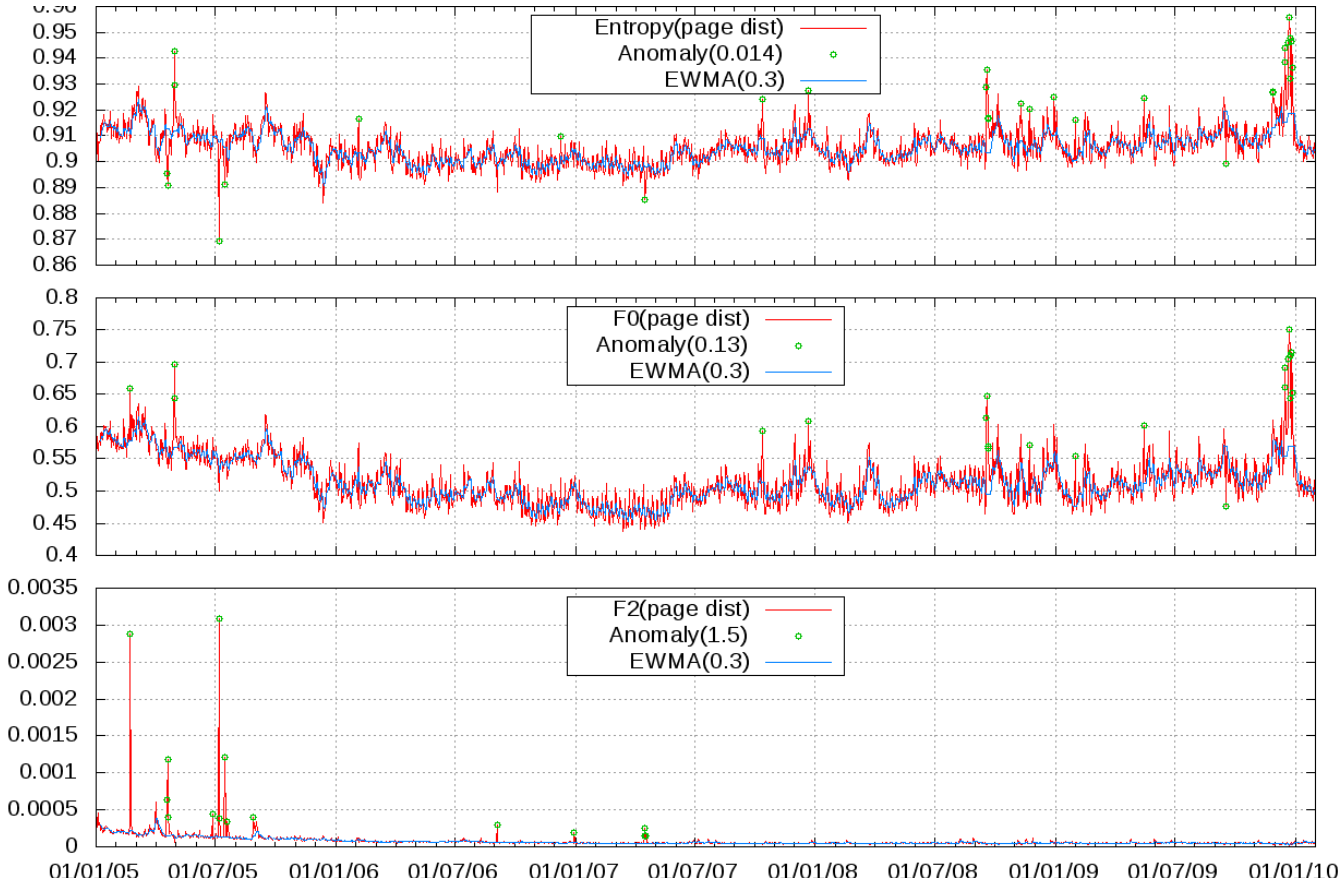

Fig. 5. Anomaly detection with the page distribution timeseries. Y-axis is metric value. [Best viewed in color.]

And Figure 4 gives a comparison of the outcome of the entropy metric of the page distribution, before and after 2005. From these figures we can see the instability of the system prior to 2005. Thus, we apply our detection method only on the timeseries starting 2005. Figures 5 and 6 show the detection method applied on the page distribution and the contributor distribution respectively. Within each figure there are plots for the monitoring of H, F_0, and F_2 over the corresponding distribution. Also, Figure 7 shows the monitoring of the F_1 metric. Note that the value of F_1 in a window is the same whether measured from the page distribution or the user distribution. Each plot shows the metric timeseries (red), EWMA timeseries (blue), and anomalies (green circles) according to our detection methodology. Table I lists out the anomalies found and marked in the plots. A "+" in the table indicates that the anomaly was due to a spike in the timeseries curve (when the metric value deviates higher than the EWMA value), and a "-" in the table indicates a dip in the curve. The last column of the table lists results of our diagnosis of the anomalies found. To determine the causes of an anomaly we used two steps. First, we looked at what effects the anomaly has on the metrics on the distributions. The combination of these effects is usually able to provide a good indication of the properties we will find in the distributions on the day of the anomaly. Second,

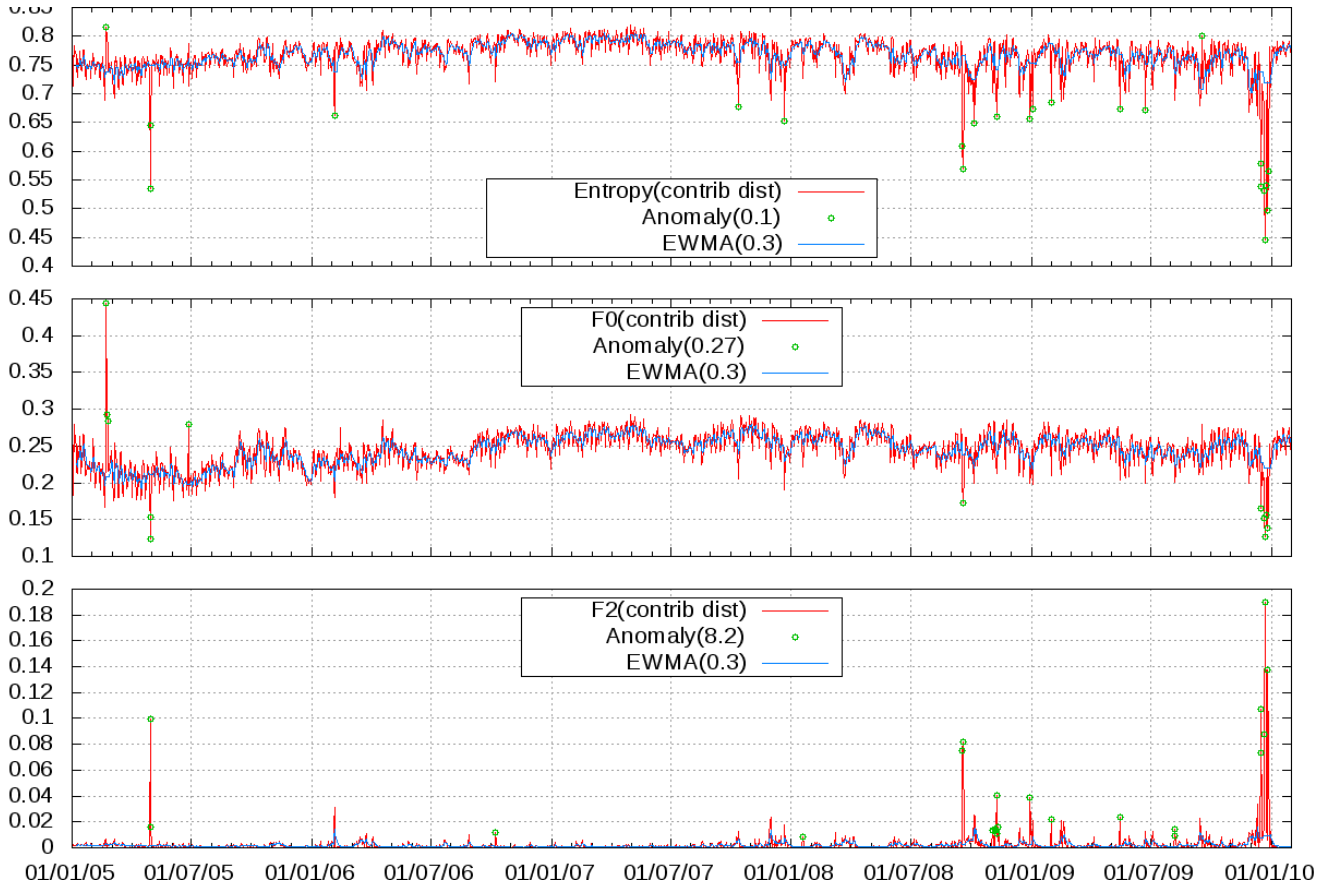

Fig. 6. Anomaly detection with the contributor distribution timeseries. Y-axis is metric value. [Best viewed in color.]

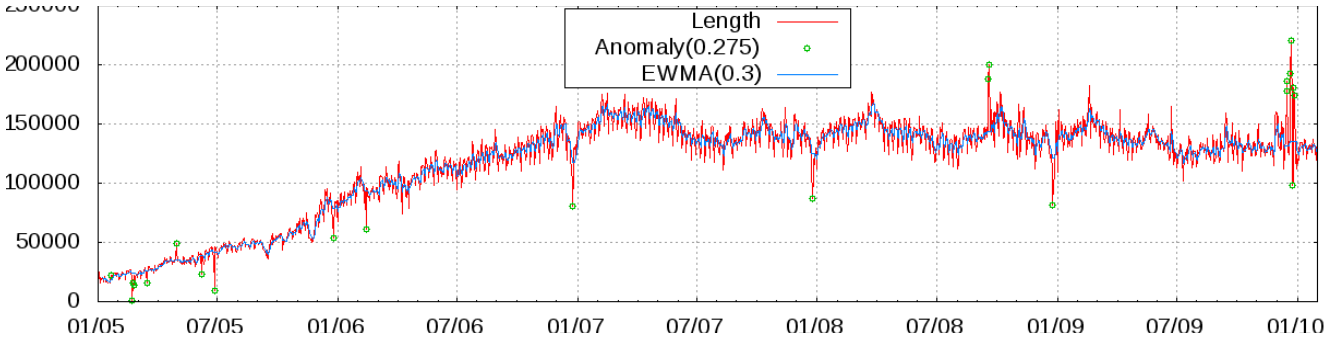

Fig. 7. Anomaly detection with the F_1 (volume) metric timeseries. Y-axis is metric value. [Best viewed in color.]

we extract the histograms on the day of the anomaly and try to look for the properties identified in the first step. Depending on these properties we are then able to find the contributor, page, or other reason for the anomaly. We now describe the anomalies listed in Table I. Basically, we can categorize them into three groups—bots, flash events, and outages. The table comments also indicate the category for each anomaly.

Wikipedia Bots: Bots in Wikipedia perform their activity frequently, searching for appropriate pages to be updated. But sometimes, the activity of one or more bots is more pronounced, meaning that they update a large volume of pages in a relatively short span of time. Such activity reflects in a spike in both H and F_0 of the page distribution, as we would expect. On the other hand, for the contributor distribution we expect the bot activity to result in a dip in H, and a spike in F_2. For some of the bot anomalies, the effect is also observed as a spike in the volume metric. Table I shows 33 anomalies due to 13 bots–Flabot, D6, CapitolBot, etc.

Flash Events: For flash events, the threshold τ serves as a knob for determining the level at which an event is declared as newsworthy. Flash events usually result in a dip in H or a spike in F_2 of the page distribution. We did not observe a significant effect on the contributor distribution.

Outages: The outage between February 22–24, 2005 was due to a power failure during which editing on the whole of Wikipedia was cut off. On June 27, 2005 access to Wikipedia

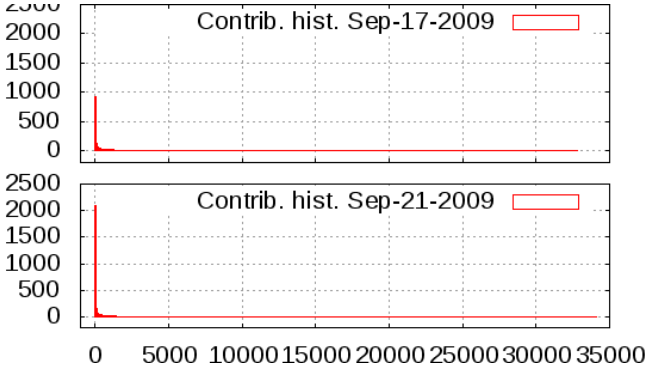

Fig. 8. Sep-17-09 anomaly histogram compared with that a few days later.

was again blocked due to an upgrade to the MediaWiki software that Wikipedia runs on. These anomalies resulted in unpredictable effects on the page and contributor distributions, since there were only a few elements in the histograms (the elements before and after the anomaly from the included windows). This unstable behavior is an indicator of the outage. The dips in the volume metric are the most direct indicator of the anomalies. We also found another unique outage event through our analysis. On September 17, 2009 a bug in an update to the MediaWiki software caused the programming interface for bots to fail. Thus bots that were editing Wikipedia at the time ceased operation. This anomaly was tricky to diagnose. Because it is not a full-scale outage, it does not affect the volume metric. The entropy of the contributor distribution spikes due to the anomaly. On examining the histogram (see Figure 8 for a comparison of the histogram on Sep-17 with the "normal" histogram a few days later), we found that although the support of the distribution decreased slightly (bots disappeared), the distribution became more uniform because of the absence of the bots (who were also the heavy hitters). This combines the effects from Plots (c) and (e) (much like Plot (g)) in Figure 1. For the page distribution, the entropy dips, but it does not seem to indicate why this occurs unless the other metrics and distribution are also considered. F_0 of the page distribution dips significantly. Initially, this was hard to justify; it was not clear why the number of pages edited was smaller. It is only the tail of the distribution with the single-edit pages (due to the bots) that disappeared, which is telling of the anomaly.

We were able to confirm the causes of the outages above by searching the archives of the Wikipedia Signpost [3] that provides news updates on Wikipedia-related events.

Other Anomalies: Besides the anomalies noted above, we were also able to identify the periodic annual lull in revision activity on Christmas day each year. This is clearly reflected in the volume metric (Figure 7).

False Positives: Our distribution-based methodology exhibits a very low rate of false positives. Only two of the 64 anomalies flagged (i.e., 3%) over the course of five years were found to be false positives. The January 25, 2005 false positive can be attributed due to the instability and increasing volume of revisions at the start of our dataset. The second false positive on December 9, 2006 is the result of an unfortunate combination of a small increase in entropy with a small decrease in revision volume giving the impression of a large increase in entropy, due to the normalization.

An important observation to be drawn from Table I is that each of the metrics is significant in our methodology. For each metric, the anomalies flagged are never consistently flagged by another metric.

VI. Conclusions and Open Problems

We developed an efficient, online distribution-based anomaly detection methodology. Our evaluation on our Wikipedia dataset shows that it is possible to detect several kinds of anomalies with a detection rate that is higher than traditional methods, and a low false-positive rate.

One direction for future work on our methodology is to develop a classifier that combines inferences of the different metric-distribution combinations to further automate the diagnosis of anomalies. Another open problem is to study the effects of the various parameters like window length, EWMA multiplier etc. on the quality of detection. The most exciting research direction, in our opinion, is to apply the methodology to other networks, to find what anomalies lie therein.

References

[1] In *1st Intl. Competition on Wikipedia Vandalism Detection, part of 4th Intl. Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse PAN-10*, 2010.
[2] Lightmouse banned. http://en.wikipedia.org/wiki/Wikipedia:Requests_for_arbitration/Date_delinking#Lightmouse_banned.
[3] Signpost. http://en.wikipedia.org/wiki/Wikipedia:Wikipedia_Signpost.
[4] WikiDump. http://en.wikipedia.org/wiki/Wikipedia:Database_download.
[5] Wikipedia Stats. http://stats.wikimedia.org/EN/TablesArticlesTotal.htm.
[6] L Akoglu, M McGlohon, and C Faloutsos. oddball: Spotting anomalies in weighted graphs. In *Advances in Knowledge Discovery and Data Mining, vol. 6119*, Springer LNCS. 2010.
[7] R Almeida, B Mozafari, and J Cho. On the evolution of wikipedia. In *ICWSM'07: Intl. AAAI Conference on Weblogs and Social Media*, 2007.
[8] N Alon, Y Matias, and M Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1), 1999.
[9] C Arackaparambil, S Bratus, J Brody, and A Shubina. Distributed monitoring of conditional entropy for anomaly detection in streams. In *Proc. of IEEE Workshop on Scalable Stream Processing Systems*, 2010.
[10] P Barford, J Kline, D Plonka, and A Ron. A signal analysis of network traffic anomalies. In *Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, 2002.
[11] V Chandola, A Banerjee, and V Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41:15:1–15:58, July 2009.
[12] A D'Alconzo, A Coluccia, and P Romirer-Maierhofer. Distribution-based anomaly detection in 3g mobile networks: from theory to practice. *Int. J. Netw. Manag.*, 20:245–269, September 2010.
[13] N Harvey, J Nelson, and K Onak. Sketching and streaming entropy via approximation theory. In *Proc. of 49th IEEE FOCS*, 2008.
[14] A Lakhina, M Crovella, and C Diot. Mining anomalies using traffic feature distributions. In *Proc. of SIGCOMM*, 2005.
[15] W Lee and D Xiang. Information-theoretic measures for anomaly detection. In *Proc. of IEEE Symposium on Security and Privacy*, 2001.
[16] J Leskovec. *Dynamics of Large Networks*. PhD thesis, CMU, 2008.
[17] G Nychis, V Sekar, D Andersen, H Kim, and H Zhang. An empirical evaluation of entropy-based traffic anomaly detection. In *Proc. of ACM SIGCOMM conference on Internet Measurement (IMC'08)*, 2008.
[18] J Sun, C Faloutsos, S Papadimitriou, and P Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proc. ACM SIGKDD Intl. conference on Knowledge discovery and data mining*, KDD '07, 2007.
[19] F Viégas, M Wattenberg, and K Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proc. of the SIGCHI conference on Human factors in computing systems*, CHI '04, 2004.