Outage Detection via Real-time Social Stream Analysis: Leveraging the Power of Online Complaints

Eriq Augustine
Cal Poly State University
San Luis Obispo, CA
Department of Computer Science
eaugusti@calpoly.edu

Cailin Cushing
Cal Poly State University
San Luis Obispo, CA
Department of Computer Science
ccushing@calpoly.edu

Alex Dekhtyar
Cal Poly State University
San Luis Obispo, CA
Department of Computer Science
dekhtyar@calpoly.edu

Kevin McEntee
Netflix Inc.
Los Gatos, CA
kmcentee@netflix.com

Kimberly Paterson
Cal Poly State University
San Luis Obispo, CA
Department of Computer Science
klpaters@calpoly.edu

Matt Tognetti
Cal Poly State University
San Luis Obispo, CA
Department of Computer Science
mtognett@calpoly.edu

ABSTRACT

Over the past couple of years, Netflix has significantly expanded its online streaming offerings, which now encompass multiple delivery platforms and thousands of titles available for instant view. This paper documents the design and development of an outage detection system for the online services provided by Netflix. Unlike other internal quality-control measures used at Netflix, this system uses only publicly available information: the tweets, or Twitter posts, that mention the word “Netflix,” and has been developed and deployed externally, on servers independent of the Netflix infrastructure. This paper discussed the system and provides assessment of the accuracy of its real-time detection and alert mechanisms.

General Terms

Algorithms, Design, Experimentation, Measurement

Keywords

Outage detection, social stream media analysis

1. INTRODUCTION

With more than 25 million members worldwide, Netflix, Inc. (Nasdaq: NFLX) is the world's leading Internet subscription service for enjoying movies and TV shows. It streams videos to over 450 Internet-connected devices. For Netflix customers streaming service availability is key to a satisfying experience. The ability to detect and react to service disruptions quickly is a critical business need and worthy of significant research and development efforts.

Netflix has traditionally used three means of detecting service disruptions. Each is described below with their accompanying defects.

Internal monitoring systems. Netflix has internal systems deployed to monitor their streaming service. These systems are designed to monitor key basic health metrics such as server CPU utilization and disk utilization as well as top level service metrics such as stream starts per second. The problem with the system is that it shares a common infrastructure with the streaming service itself. As a result, there have been streaming service disruptions that have also disrupted the internal monitoring efforts. In such situations, the internal monitoring system offers no value in restoring streaming service functionality.

External synthetic transaction monitoring systems. Netflix contracts the use of 3rd party synthetic transaction monitoring systems to monitor streaming service availability from outside of the Netflix streaming service infrastructure. While this service avoids the shared infrastructure weakness of the internal monitoring system, it does have a key defect of its own: it only monitors the transactions that Netflix designates it to monitor, and its impact is limited to only a handful of transactions deemed important enough to be repeatedly tested through synthetic monitoring. Since Netflix services are constantly and rapidly evolving, transactions come and go on a weekly basis. However, the external mechanism is not evolving at the same pace. As a result, there are thousands of possible transactions for Netflix customers that are not monitored by this external mechanism.

Customer Service. Netflix customers experiencing a service disruption will make phone calls to Netflix Customer service to complain. This is a poor means of alerting a technical response team because it is too slow. The time it takes for a customer to become frustrated enough to pick up the phone, plus the time it takes for the customer service team to determine there is a real service disruption, can range from 15 to 60 minutes.

While Netflix customers may be slow to pick up the phone and contact customer service about service interruptions, a certain vocal percentage of them have found another outlet for their complaints: social streaming media. Some Netflix customers tend to post to their Facebook walls or fire off tweets, Twitter posts, describing the movies they are currently streaming via a Netflix service. More importantly, these customers also write about their negative experiences with the services, such as their inability to stream movies due to a (possible) service outage.

In response to the weaknesses of the three existing monitoring systems at Netflix, the technical teams have been manually using the search mechanism of Twitter to search for terms like“Netflix down”to extend their ability to monitor the service as well as determine if reparative efforts during a service disruption were having an impact. Twitter does not suffer from the weaknesses of the 3 existing mechanisms because it is external to the Netflix infrastructure. Customers also provide fast and continuous feedback; customers will tweet about any and all types of failed transactions, and customers will tweet quickly when a disruption begins, and tweets will continue until the disruption is ended.

Netflix was curious to see if automation applied to this manual Twitter monitoring effort and engaged the research team from California Polytechnic State University to design, develop, and deploy such a system.

This paper describes a service outage detection system SPOONS (Swift Perceptions Of Online Negative Situations) that relies on timely, publicly available information, resides outside Netflix-controlled server space, and produces real-time outage warnings. This system complements the internal QA solutions and provides an independent assessment of the state of Netflix streaming services, and, perhaps more importantly, the mood of Netflix customers.

The rest of the paper is organized as follows: Section 2 describes research on the use of online streams for information mining and sentiment analysis; Section 4 describes the requirements for and the architecture of SPOONS and discusses its main components; Section 5 describes an evaluation of the feasibility of this project. Section 6 discusses anomalous event and outage detection methods implemented in SPOONS. Section 7 discusses their accuracy in predicting service outages based the past year of data. Future research and development directions are discussed in Section 8.

2. RELATED WORK

Twitter is an online social networking service that only allows its users to post 140 characters of text. These posts are called tweets. According to Twitter Blog, as of March 14th 2011, Twitter users were posting approximately one billion tweets per week.[14] These relatively small and concise messages are a data mining dream. Many research groups are now developing systems that parse, categorize, or analyze sets of tweets to abstract meaning from the patterns in this cloud of data. Some examples of uses that have been found for this data are tracking disease outbreaks[1], modeling earthquakes[7], and predicting stock prices[4]. Some common methods used to extract patterns are keyword searches, machine learning, sentiment analysis, and time series analysis.[6] The rest of this section describes other research that uses tweets in systems similar to SPOONS.

Online Service Outage Detection. Levchenko et al.[5] implemented a system that uses tweets to detect outages in several widely used Web services such as Amazon, Gmail, Google, PayPal, Netflix, Youtube, Facebook, Wikipedia, and Flickr. They describe Twitter users as acting as millions of sensors who have a large breadth and flexibility of in the definition of failure. The detection mechanism employed in this work is fairly straightforward. A collection of tweets that either contain an “X is down” phrase or an “#Xfail” hashtag, where “X” is the name of a service (e.g., “#netflixfail” or “#Netflixfail”for a report of a Netflix outage) is pulled via TwitterAPI. The tweets are split into time frames ordered by the time of origin. Exponential Smoothing is applied to the collected data and the expected values for each time frame are compared to the actual values, after which an effort to filter out false positives is performed.

Levchenko et al. were only able to validate a subset of their detected events because a full precision and recall validation would have required a list of outages during 2009 for every company they were monitoring. So while the events they were able to verify indicate that the system can detect outages, the full effectiveness of their method is still largely unknown. The methods in the SPOONS system are able to be completely validated because Netflix is providing a complete list of the outages that occurred during the evaluation periods. To evaluate the effectiveness of this method in relation to the other methods in the SPOONS system, the “is down” phrase and Exponential Smoothing spike detection will be integrated into the SPOONS system as the keyword volume method and the trend monitor.

Classification. Hong et al.[3] focused on finding a topic model for Twitter users. They found that they could increase their tweet classification accuracy by using the discovered topics as features in tweet classification.

Time Series Sentiment Analysis. Researchers commonly analyze Twitter traffic to determine whether a notable event has occurred or if the overall mood of tweets correlates with something occurring at the moment. Much of the data encoded in tweets contains polar sentiment, and the average sentiment, or mood, of a Twitter stream can be extracted. Sentiment analysis is in the process of being integrated into the SPOONS system, working under the hypothesis that an outage event can lead to a drop in positive sentiment Netflix-related Twitter traffic.

The goal of the sentiment analysis method developed by O'Connor et al.[11] was to find a correlation between the overall sentiment on Twitter and the results of public polling. To determine the sentiment of the tweets, they mapped each tweet's words to a polarity (positive or negative) and then tracked the aggregate sentiment over time. This polling system didn't use highly advanced or complicated methods for sentiment analysis or time series creation, but their methods were effective enough to see the correlation results that they were looking for.

The system created by O'Connor et al. is similar to the SPOONS system because they both use English tweets that contain a topic identifying keyword and they both use lexicon word sentiment determination and weighted averaging.

3. SPOONS REQUIREMENTS

Netflix has provided the following set of key requirements to be met by the SPOONS system:

R1. Structural Independence. The outage detection system shall be structurally independent of both the software and the hardware infrastructure used by Netflix. It shall rely only on information that is publicly available and free for use. This ensures that the outage detection system stays up even when any or all Netflix servers are experiencing downtime.

R2. Use of Amazon Web Services. Netflix is one of the largest customers of Amazon.com's cloud computing service, Amazon Web Services (AWS). AWS allows users to create new cloud machines (instances) on-the-fly in many regions throughout the world. The outage detection system shall be deployed on an AWS server that is operationally independent of other AWS servers used by Netflix. Using a cloud solution allows the outage detection and alert system to be deployable on a global scale.

R3. Real-time. Netflix is a real-time service and any downtime has an immediate impact on customers. To minimize that impact, the outage detection system shall notify Netflix of detected outages as soon as possible.

R4. Accurate Outage Detection. The number of non-outage situations that raise an alert shall be minimized. While a small number of false positives detected in real-time may be acceptable, the outage detection system shall detect outages and generate alerts with as high precision as possible.

R5. Comprehensive Outage Detection. Not all Netflix outages will generate a signal on Twitter. These may be allowed to go unnoticed by the outage detection system (as the system will have no basis for detecting them), but any outage that causes a signal on Twitter shall be detected.

R6. User-Friendly Online UI. The outage detection and alert system shall have an easy-to-use, informative, online UI which shall provide Netflix employees with real-time information and historic data about the state of Netflix according to Twitter. The information provided shall include: times of outages; times of other anomalous events; current and recent Netflix-related Twitter traffic trends; and samples of Netflix-related tweets.

4. SYSTEM OVERVIEW

The architecture of the SPOONS system is shown in Figure 1. The gatherers collect, via the Twitter API, Netflixrelated tweets as they are published and store them in a database for further processing and analysis. Once gathered, the raw tweets are run through the analysis methods which are a combination of preprocessors, a counter, predictors, and monitors. Each of these methods contributes one or more outage detection procedures. The alert system uses the results from the analysis methods to determine if and when outage alerts should be generated and notifies Netflix engineers via email of the potential outage or service issue. If detailed information about the event is required, the Netflix engineers can access the system's UI through any web connected device. Through the UI, the engineers are able to obtain detailed traffic graphs and outage prediction results, review relevant tweets, and provide additional information about the event.

4.1 Gatherers

[Insert Figure 1]

The gatherer component repeatedly queries the Twitter API for all tweets containing the word "Netflix" or the hashtag"#netflix"(in any capitalization) between the most recently collected tweet and the current time. The tweets are returned in a JSON document which is parsed and saved in the SPOONS database. For each tweet the gatherer saves[1]:

- **twitter id**: a unique identification number from Twitter
- **published time**: the time that the tweet was posted

[1] Some other information, e.g. geographical location of the tweet is also parsed and stored, but has not been used in the actual operation of the system to date.

- **content**: the text that was posted
- **language**: the language that the tweet was written in
- **author**: the username of the account that posted the tweet

The Twitter API will sometimes not return any results for sections of the time period requested. However, the empty sections differ depending on the IP address from which the request was made. This may stem from the *eventual inconsistency* property of Twitter's NoSQL back end[15]. To ensure that the system is gathering all Netflix-related tweets, the gatherer runs on multiple servers (as depicted on Figure 1), each with a separate IP address. The tweets from each server are merged in the SPOONS database using the tweet id to identify and eliminate duplicates.

4.2 Analysis Methods

Analysis methods are processes that analyze tweets and determine if any outage is occurring. They all run in parallel asynchronous unique threads so that each method can detect outages without being blocked by any of the other methods. First the methods aggregate tweet data within a time frame and create a time series. Then they attempt to predict the behavior of normal traffic based on observed day-to-day traffic patterns. Any time when observed traffic differs significantly from the predicted traffic, the method concludes that anomalous traffic is present and logs an event.

To date, two key reasons for anomalous traffic have been determined: (a) Netflix service outages; and (b) media events. Service outages are the events that the system is trying to detect. Media events are time periods when some Netflixrelated news stories (e.g. information about quarterly earnings reports, new product or service announcements, or profiles of key Netflix personnel) appear in press and are mentioned and discussed on Twitter. News story-related tweets often contain bit.ly links to actual news reports or blog posts with opinions and analysis. The analysis methods mark the appropriate event in the online UI, and raise an alarm with Netflix engineers if they conclude that a service outage has been detected.

[Insert Table 1]

4.2.1 Preprocessors

In order to be usable for outage detection purposes, the raw tweets may first need to be run through some preprocessors which prepare them for comprehensive analysis. The output from these preprocessors is used as the input for other processors or counters.

Text Normalization Preprocessor.

The text normalization preprocessor tries to narrow down the number of unique words found in tweets so that other components can more accurately analyze tweets. It uses a combination of the tools described below.

URL Replacement. Detects URLs in bodies of tweets and replaces them with a placeholder.This allows components to quickly identify tweets that contain URL links.

Punctuation/Non-English Character Removal. Removes all punctuation and characters not in the English alphabet. This simplifies word extraction and comparison.

Title Detection. A tweet that says, “watching fighting duel of death”, seems very negative at first until one realizes that “fighting duel of of death” is a movie title. Many tweets mention titles of movies or shows that the tweet author is watching, just watched, or wishes Netflix carried. Words in these titles may carry meanings that can confuse some of the analytical tools. Movie and show titles are identified based on a regularly updated table of titles that are offered through Netflix’s streaming service. When identified, the titles are replaced with a static placeholder recognizable by the other components of the system.

Stopword Removal. Stopwords, or words that carry little or no semantic information, are identified based on a static table of words mapped to levels. Stopwords are assigned levels which allow processes to use different sets of stop words. All words less than 3 character are also automatically considered stop words.

Stemmer. Stemming finds the root of a word. This allows words to be categorized by their roots which decreases the number of unique words being evaluated and emphasizes linguistic patterns. This preprocessor uses Porter's stemmer for the English language [12].

Classification Preprocessor.

The classification preprocessor attempts to determine if the tweet is indicative of a service outage by classifying each tweet into one or more of eight classes. At present, it incorporates five off-the-shelf classifiers from the WEKA machine learning package[2]: Naive Bayes, Bayes Net, J48, K-Nearest Neighbors (KNN), and Sequential Minimal Optimization trained support vector classifiers (SMO). Additionally, a committee which classifies a tweet according to the decisions of the plurality of these five classifiers is implemented. The feature set for all classifiers is the bag-of-words representations of tweets.

The eight classes are:

- Media: relate to a media story about Netflix
- Outage: talk about an outage
- Complaint: complain about Netflix
- Happy: express that Netflix-caused joy
- Neutral: neutrally observe or comment about Netflix
- Watching: contain updates on what the user is currently watching
- Response: respond to another user in a Netflix-related conversation

- Undetermined: all other tweets (usually this category is reserved for Netflix-related tweets in languages other than English)

The classifiers were trained on a small set of 759 tweets which were pulled from from periods of both normal and anomalous traffic. Each tweet in the training set was manually classified by multiple researchers until consensus about the classification was reached. Because the goal is anomalous traffic detection, the training set over-samples the tweets from media, outage, and complaint: categories. Table 1 documents the structure of the training set and shows the number of tweets classified into each of the eight categories. Tweets were allowed to belong to multiple classes because of posts like, “I love netflix! Watching Law and Order online!”, which could be classified as both happy and watching. Before classification, preprocessing is applied to the tweets in order to remove all punctuation and stop words, replace all web links with the meta-word <LINK>, replace all movie and show titles with the meta word <TITLE>, and stem all other words.

Because the methods in SPOONS are mostly concerned with anomalous traffic, there is no reason for the classifiers to distinguish between the five categories (happy, neutral, watching, response, undetermined) forming normal traffic. To further increase accuracy, the classification processor groups the eight tweet classes into three categories:

- Media: contain the media class
- Bad: contain both the outage and complaint classes
- Other/Normal: contain all other classes

4.2.2 Counters

Counters break tweets stored in the SPOONS database - either in their raw form, or in preprocessed form - into time frames based on the publication time of each post. At present, the counters aggregate Netflix-related Twitter traffic into 30 minute time frames with 15 minute shifts. Time frames start on the hour and each hour is covered by four overlapping time frames starting at :00, :15, :30 and :45 minutes respectively. A single day worth of traffic is represented by 96 time frames, with each tweet contributing to two time frames. The overlap allows SPOONS to achieve some built-in smoothing of the traffic, while still maintaining sensitivity to sudden changes in the traffic pattern. Even though the time frames are 30 minutes long, they are triggered to update with the newest batch of tweets from the gatherers every 2 minutes. This means that outages can be detected within about 2 minutes of the outage tweets reaching a spiking level. Counters store their output in the SPOONS database for further use.

4.2.3 Predictors

The key to many of the analysis methods described in this paper and employed in SPOONS is accurate estimation of traffic volume during normal (non-event) time periods. The data calculated by predictors is used by the monitors to detect when the current traffic is anomalous. There are two predictors that are described in this paper: trend and model.

Model Predictor.

The model predictor creates a model of normal traffic that is used to predict future traffic behavior. This model is extracted through time series analysis of volume values. These predictions are then compared to the actual volume values and the standard deviation between the actual and predicted values is computed and maintained.

The model values that represent the traffic for the time frame t + 1 week are calculated by taking the weighted average of the non-outlier values from previous weeks. There are two types of outlying values that are removed: holes and spikes.

Holes, frames with a volume of 0, are caused by a problem with the Twitter API and don't actually reflect a lack of twitter posts about Netflix. When the predictor encounters a hole, the predicted value is set to the current model value and standard deviation is not updated.

Spikes occur when the volume of a frame is more than 2 standard deviations higher than the current model value. Spike values are included in the standard deviation calculation, but are replaced by the current model value for the weighted averaging.

Once outliers have been removed the weighted weekly average is calculated using the formula:

$$\breve{P}(t) = \frac{\sum_{i=1}^{n}(n-i+1)*\breve{V}(t-(i*W))}{\sum_{i=1}^{n}(i)}$$

Here, P(t) is the traffic volume prediction at time t, V(t) is the actual observed traffic at time t, n is the total number of weeks used in the predictor, and W is the ordinal number of the week, with 1 being the earliest week.

The model of normal traffic created by this predictor is used to evaluate the feasibility of detecting Netflix outages through time series analysis of Netflix-related Twitter traffic.

Trend Predictor.

The trend predictor calculates an adjustment for each traffic volume estimate based on previous values.

The single exponential smoothing algorithm[9] is used to construct the smoothed traffic volume prediction S by weighting the most recent previous value A_{t-1} and previous actual values A_{t-n} based on how long ago they occurred and a predetermined smoothing factor α. For $t > 1$ and $0 < \alpha < 1$:

$S_t = \alpha A_{t-1} + (1-\alpha)S_{t-1}$

The most recent previous value A_{t-1} is given a weight of α. Then the remaining weight is split between values before t-1 with the same formula.

The double exponential smoothing algorithm[10] extends the single exponential smoothing algorithm by taking into account the trend of the previous values b_t. For $t > 1$, $0 < \alpha < 1$, and $0 < \beta < 1$;

$S_t = \alpha A_{t-1} + (1-\alpha)(S_{t-1} + b_{t-1})$

$b_t = \beta(S_t - S_{t-1}) + (1-\beta)b_{t-1}$

The smooth and trend values calculated in the exponential smoothing trend predictor are used in the trend monitor to detect sudden spikes in traffic.

4.2.4 Monitors

The goal of the volume monitors is to create a log of events that are being indicated by the volume methods. Each method has one or more monitors. For each time frame, a monitor compares the analysis value to the prediction value and uses the certainty of the prediction and other factors to determine if there is an outage. There are three monitors described in this paper: the ratio monitor, the model monitor, and the trend monitor.

Ratio Monitor.

The ratio monitor uses a specific classifier and takes the ratio of tweets placed into the bad group over the total volume of tweets. When this ratio exceeds a specified value for a set amount of time, an alert is raised. Although it is currently only looking at classifiers, the ratio monitor can be easily abstracted to look at any two streams of data that can produce an intelligible ratio. The ratio is tuned to be between 0.1 and 0.2. [1.]

Trend Monitor.

The trend monitor detects events based on an actual value exceeding the estimated value created by the trend predictor (Section 4.2.3) by more than the allowed threshold using the equation:

$$\textit{actual}\mathrm{Val}_t >= \textit{smooth}\mathrm{Val}_{t-1} + \mathrm{thresholdMultiplier} * \mathrm{trendVal}_{t-1}$$

Where the threshold multiplier can be tuned any number that is a whole number between 1 and 15[2]. This monitor was inspired by the way Levchenko et al.[5] determine outages.

[2] For more information on monitor tuning see Section 7.4

4.3 Alert Generation

At the end of each time frame, each monitor determines whether or not the Netflix-related Twitter traffic during that frame signifies an outage event. If the monitor reaches this conclusion, it issues an alert and contacts Netflix engineers with a brief email specifying the time of the alert and the reasons why the alert was raised. From there, the SPOONS UI extracts this data and plots it on the traffic time line.

[Insert Figure 2]

4.4 User Interface

The ultimate goal of the user interface is to provide an always-accessible platform for quick analysis of outage signals and other anomalous tweet behavior. To this end, the UI was designed with five core principles in mind:

- **High Data Density**: to provide all the data needed to inform the user and fulfill the tasks of the user interface
- **Minimal User Interaction**: to make available all relevant information without requiring the user to explicitly ask for it
- **Coherency**: to convey the information so that it can be quickly understood
- **Accessibility**: to make the information and functionality provided by the user interface should be accessible whenever needed. Initially, this means the user interface should be Web-based. Ultimately, this means the interface should be mobile-friendly as well
- **Performance**: to be responsive and provide an intuitive user experience

These principles directed the design and development of the SPOONS user interface since its inception. As a result, every element in the UI provides some piece of useful quicklyparsable information to the user. Administrative debris (to borrow a term from Tufte [13]) is minimized.

Figure 2 shows a screen shot of the current user interface. The main screen of the UI is composed of three main components: (a) a time series chart, showing any time series data maintained by SPOONS for any time interval; (b) an event log, which specifies when the outages and other detected events occurred; and (c) a sampling of stored tweets from the currently selected time range. Each UI component is briefly discussed below.

4.4.1 Time Series Chart

The time series chart provides an intuitively understandable graphical representation of the information stored in the SPOONS databases, including tweet activity (mainly in the form of relevant tweet volume) and analysis method results. This allows Netflix engineers, who can choose what time series data to display, to quickly scan for anomalous behavior and detect unusual tweet patterns. In addition, any events reported by researchers or Netflix engineers are color-coded by type (blue for media events, red for outages) and overlaid onto the chart to provide additional contextual information. The chart also functions as time range control for the tweet list through a simple click-and-drag gesture users are able to narrow the range from which the tweet list pulls its tweets.

4.4.2 Event Log

At present, SPOONS detects possible service outages and possible media events. The event log presents the user with a detailed look at all events within the currently selected time range. Information available for each event includes type (media, outage, etc.), confirmation-status (e.g. whether Netflix engineers confirmed an outage), duration, start and end times, severity, and any user-supplied explanatory notes. In order to make their location in time apparent to the user, each entry in the log also has a corresponding graphical representation on the time series chart. In addition, the log functions as a medium for event annotation, providing the means for Netflix engineers to report new events, confirm event validity, and supply any relevant notes.

4.4.3 Tweet List

Whenever an event is detected and requires confirmation Netflix engineers want to observe the Netflix-related tweets from the relevant time range that lead to the detection decision. The tweet list provides the means to do so by displaying the tweets used within the system. Initially displaying a random set of tweets from within the currently selected time range, the list can be modified and its range narrowed by all of the other components of the user interface. A click-anddrag on the chart will narrow the range of tweets displayed by the list as well as order them chronologically. Selection of an event within the event log (via double click) will produce a list of randomly selected tweets relevant to that particular event. By using these interactions, users are able to view the tweets that are the most relevant to whatever task or information they are currently interested in.

5. FEASIBILITY

SPOONS began monitoring Netflix-related Twitter traffic on December 17, 2010. In the months that have followed, the system collected and archived over six million tweets. In general, Netflix-related Twitter traffic follows a repeated pattern on a 24-hour cycle. In the absence of anomalous traffic, this cycle can be observed in a straightforward manner. Figure 3 shows an example of a week's worth of normal Netflix-related Twitter traffic.

SPOONS methods operate by predicting an expected normal traffic pattern, then observing the current traffic pattern, and detecting significant deviations between them. The accuracy of the alert generation system is dependent on SPOONS' ability to accurately predict the volume of non-anomalous (normal) traffic at each time frame of the day. If it can do so accurately, then any time Netflix-related Twitter traffic becomes abnormal (the number of Netflix-related tweets increases in a significant way over the expected number), an outage alert can be sent with reasonable confidence. This section looks into the feasibility of detecting Netflix outages through Netlfix-related Twitter posts by evaluating the model predictor to see if it can follow the traffic pattern during normal times and then diverge from the traffic pattern during event times.

[Insert Figure 3 and 4 and Table 2]

The metric used to determine if the predicted time series is diverging from the actual time series for time period t is mean square error (MSE). The four types of time periods that will be compared are:

- **Normal**: times not during an outage or media event
- **Anomalous**: times during an outage or a media event
- **Outage**: times during an outage
- **Media**: times during a media event

Outage times are taken from the list of service outages provided by Netflix. Media times were manually determined by researchers.

Table 2 shows the results of the evaluation run on the tweets in the data set, outage events, and media events between January and November of 2011. It shows that the MSE between expect traffic and outage traffic is about 4 times larger than the MSE between expected traffic and normal traffic which indicates a difference in tweets about Netflix during outage times.

However, it also shows that there is also a large difference between normal traffic and media traffic. This indicates that pure total volume analysis alone will not be able to tell the difference between media and outage traffic. The next section will describe some methods that attempt to reduce the spikes caused by media traffic so that only outage events are detected.

6. OUTAGE DETECTION METHODS

At present, SPOONS estimates outages using two different approaches. In the first set of approaches, the simple volume analysis methods, SPOONS predicts the volume of a specific subset of Netflix-related Twitter traffic and generates an outage alert when this prediction differs significantly from the observed traffic. The second family of detection methods uses classification techniques described in Section 4.2.1 to classify individual tweets, and then raises alerts when the number of tweets from the bad category exceeds expectations.

6.1 Simple Volume Detection

The simple volume detection methods attempt to find outages using only time series analysis of volumes that can be determined by filtering raw tweets based on information that is received directly from the Twitter API. The goal of these methods is to pull a volume measurement for a time frame from the database as soon as that frame ends. This means that the results of these volume measurements will not be blocked by any editors or preprocessors.

6.1.1 Keyword Analysis Method

The keyword volume analysis method calculates the volume of the subset of the English volume data set that contain the phrase “is down”, similar to the IsDown predicate defined by Levchenko et al[5]. Figure 4 is a graph of a traffic pattern created by the keyword volume counter.

An analysis of the tweets in the data set showed that the phase “is down” occurs in 3% of the tweets posted during an outage event, but in less than 1% of the tweets posted during a media event. Therefore in general, the outage events will spike 3 times higher than media events.

6.1.2 Linkless Analysis Method

The linkless volume analysis method calculates the volume of the subset of the English volume set of tweets that do not contain a URL. This method assumes that the presence of a URL indicates that the tweet is about a media story and is not a tweet reporting unknown outages, so it reduces the number spikes in traffic caused by posts about media stories that trigger false positives.

Graph 5 displays an example of a time period where the linkless method only reduced an outage spike by about 20% but reduced a media event practically to the level of normal traffic.

An analysis of the tweets in the data set showed that in the linkless data set, URLs occur in 46% of the tweets posted during a media event, but in only 21% of the tweets posted during an outage event. Therefore in general, this method will reduce media spikes by about half, while only diminishing outage spikes by about a fifth.

[Insert Figure 5 and Tables 3, 4 and 5]

6.1.3 Simple Volume Event Differentiation

This section uses the same evaluation procedure described in Section 5 to show how the simple volume analysis methods increase the differentiation between outage and media traffic. The results of this evaluation are shown in 3. Both keyword volume and linkless volume, decrease the magnitude of difference between media and normal while increasing the difference between outage and normal. Therefore they not only decrease the likelihood of a media event causing a false positive, but also increase the likelihood of an outage event standing out and triggering a true positive. This indicates that the methods' attempts to disclude media tweets from their time series volume counts is effective.

6.2 Classification Volume Detection

The family of classification-based outage detection methods works as follows. Each tweet (either raw text or preprocessed text) is classified using the WEKA[2] classifiers into the nine categories described in Section 4.2.1 The tweets are then combined into the three meta-categories. The outage detector monitors the volume of tweets classified as bad and media and uses the ratio monitor to determine when to generate event alerts (outage for an increase in the volume of bad tweets, news event for increase in the volume of media tweets).

The key to success of the method is the ability of the classifiers to correctly determine the category of the tweet. Table 4 shows how the five WEKA classifiers and the committee perform on classifying the tweets from the training set described in Section 4.2.1 As seen from the tables, the majority of tweets in every category were classified correctly. Although, a significant percentage of bad tweets was not detected. What is more important is that the number of tweets misclassified into the bad category by each classifier has been kept very small (between 3.7% and 8%). This means that whenever a spike in the volume of tweets from the bad category is detected, the vast majority of them are outage reports or other customer complaints.

6.2.1 Classification Volume Event Differentiation

Along with the predictors, the classifiers are also used to estimate the total volume of tweets. The other/normal group is supposed to symbolize normal traffic and background noise. Therefore, during normal time periods, the other/normal class should estimate the total volume. Then in anomalous times, the bad or media groups will become dominant and the other/normal group will poorly estimate the total volume.

The same analysis as described in Section 5 was also run using the volume of tweets each classifier classified as other/ normal as the prediction. Table 5 reports the mean squared errors computed.

The results are similar to Section 5: the MSE for normal periods are at least an order of magnitude less than the MSE for anomalous times.

7. RESULTS

This section reports on the accuracy of the outage predictions using the methods described in the previous section.

7.1 Evaluation Procedure

Detection evaluation is an evaluation of how well detected events correspond to actual outage events reported by Netflix. This evaluation uses the same data set for both configuration and evaluation. So this is an ideal evaluation that determines how well the methods can create a time series that has higher values during an outage time and lower values during normal and media times.

[Insert Tables 6, 7 and 8]

7.2 Definition of the Data Set

The SPOONS database contains the entire collection of tweets that were posted on Twitter during the data time period and contain the word “netflix”. Detection evaluations are configured and evaluated using tweets from the time periods shown in Table 6. The evaluation time period starts after the beginning of the data because the set of reported events from Netflix starts in March. The two months of data before the evaluation period begins are used to allow the predictors to create a smoooth consistent time series that represent normal traffc patterns. Netflix provided a list of 93 outage events that occurred during the evaluation time period, including the events' approximate start and end times. These are the reported events that define the outages that the analysis methods are trying to detect.

7.3 Evaluation Metrics

The accuracy of outage detection is measured using three metrics:

- **Recall**: the percent of the reported events that were caught.
- **Precision**: the percent of the alerts generated by a method that occurred during an outage event. For example, a precision of 0.5 means that for every two alerts raised, one will be correct.
- **F0.5 Measure**: the harmonic mean of precision and recall similar to the F measure; however, F0.5 weights precision greater than recall.

The following definitions are used to calculate the metrics:

- **True Postive**: any intersection between a reported outage range and a detected outage range.
- **False Positive**: any detected outage that has no intersection in the events reported by Netflix.
- **False Negative**: no intersection on an event reported by Netflix is a false negative.

Netflix has specified a precision of about 0.5 to be a usuable amount of false postive noise, so any methods with a precision close to or above 0.5 will be considered for use and then ranked based on their recall.

7.4 Monitor Tuning

Each of the monitors has three parameters which define a parameter configuration. Each method has a unique parameter configuration for each monitor. These parameters are configured by a process that iterates over a large range of values for each parameter and fids the configuration that produces the best F.05 score.

7.5 Analysis

Tables 7 and 8 describe how effective the analysis methods were in detecting outages.

Table 7 shows the results of the detection evaluation run on each of the simple volume analysis methods. The keyword volume analysis method includes tweets that it identifies as indicating an outage while the linkless volume analysis method excludes tweets that it identifies as not indicating an outage. These results show that the linkless method was more effective than the keyword method at detecting outage events. This is because not all outage events are reported by Twitter users with the phrase ”is down”. However, the keyword method has a higher precision than the linkless method and is currently a more useful method for Netflix engineer to use because when users are posting “is down”, there is almost always an outage occurring.

Table 8 shows that for the classification-based methods, the recall ranges from 21% to 88% and the precision ranges from 38% to 60%. Two classifiers, J48 and SMO detect over 87% of all outages with the precision around 50%, which means that when equipped with these alert generators, SPOONS will require Netflix engineers to check about two reports of potential outages for each real outage: a workload, Netflix engineers consider to be reasonable.

8. CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

This paper has demonstrated that the process Levchenko et al.[5] developed can be applied to this problem to create a useful outage detection method. However, the more advanced classification methods also achieve the neccessary precision threshold while detecting far more of the outages.

Netflix is currently using SPOONS to monitor their 24x7 operation of the streaming service. Netflix engineers monitor the results produced by the different classification methods, the linkless analysis method, and the keyword analysis method.

SPOONS plays a valuable and complementary role within its suite of monitoring systems because it does not suffer from the 3 key weaknesses of the existing monitoring systems. It does not share infrastructure with the systems it is monitoring, it monitors 100% of customer interactions, and it does not suffer from the “latency” of human phone calls as an alert mechanism.

8.2 Future Work

Even though the current version of some of the SPOONS methods have already been deployed at Netflix, additional challenges remain for future development:

Keyword Analysis Improvements.

Matsuo et al.[8] created a procedure for algorithmically determining the keywords in a document.This procedure could be run on a “document” consisting of tweets that were posted during outage times and updated every time Netflix reported offical outage events in the system. This would allow for a larger number of keywords that would dynamically update to new ways of reporting outages over time.

Classification Improvements.

Retraining the classifiers with tweets that occured during a time identified as an outage would allow us to produce a dynamic training set that may be able to keep up with social trends.

Sentiment Analysis.

The development of a sentiment analysis method is already in progress. This method will determine the sentiment of a tweet and then detect outages by looking for significant increases in negative sentiment.

The Nature of an Outage.

Netflix would like SPOONS to include information in the alert email about the nature of an outage, e.g. which hardware platform is experiencing streaming issues.

Malicious Tweet Attacks.

Currently it is possible for a malicious Twitter user to send a large quantity of tweets "reporting and outage" and trigger false positives in the system. The only existing defense against this kind of attack is that Netflix isn't going to announce this monitoring system publically. However, this could possibly be further avoided through the use of author profiling.

9. REFERENCES

[1] A. Culotta. Detecting influenza outbreaks by analyzing twitter messages. In *KDD Workshop on Social Media Analytics,* 2010.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[3] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics,* SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM.

[4] F. Jabr. Using twitter to follow trends beats the stock market. *NewScientist,* (2829), Sept. 2011. http://www.newscientist.com/article/mg21128295.900-using-twitter-to-follow-trends-beats-the-stockmarket.html.

[5] K. Levchenko, B. Meeder, M. Motoyama, S. Savage, and G. M. Voelker. Measuring online service availability using twitter. *In Proc. of the 3rd Workshop on Online Social Networks* (WOSN 2010), 2010.

[6] B. Liu. *Web Data Mining*. Springer, 2007.

[7] Y. Matsu, M. Okazak, and T. Sakak. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW 2010: Proceedings of the 19th World Wide Web Conference*, 2010. http://ymatsuo.com/papers/www2010.pdf.

[8] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. In *Proceedings of the 16th International FLAIRS Conference*, pages 293–296, 2003.

[9] NIST/SEMATECH. 6.4.3.1. single exponential smoothing. e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook/.

[10] NIST/SEMATECH. 6.4.3.2. double exponential smoothing. e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook/.

[11] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *In Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 122–129. AAAI Press, 2010.

[12] M. F. Porter. An algorithm for suffix stripping. In K. Sparck Jones and P. Willett, editors, *Readings in information retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. http://tartarus.org/martin/PorterStemmer.

[13] E. R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, first edition edition, February 1997.

[14] Twitter. #numbers, Mar. 2011. http://blog.twitter.com/2011/03/numbers.html.

[15] K. Weil. NoSQL at twitter, 2010. Presented at Strange Loop conference.

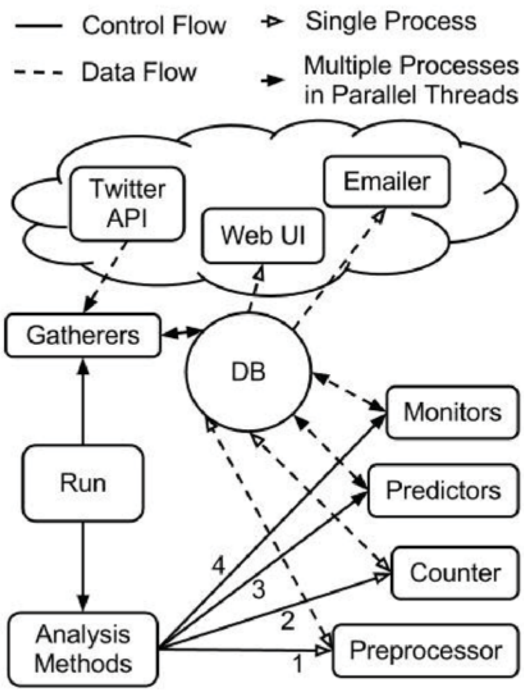

Figure 1: Architecture diagram of SPOONS.

Class	# Tweets	Class	# Tweets
`Media`	103	`Neutral`	66
`Outage`	158	`Watching`	135
`Complaint`	146	`Response`	30
`Happy`	147	`Undetermined`	48

Table 1: Overview of the Netflix-related Twitter post training set used to train classifiers in SPOONS.

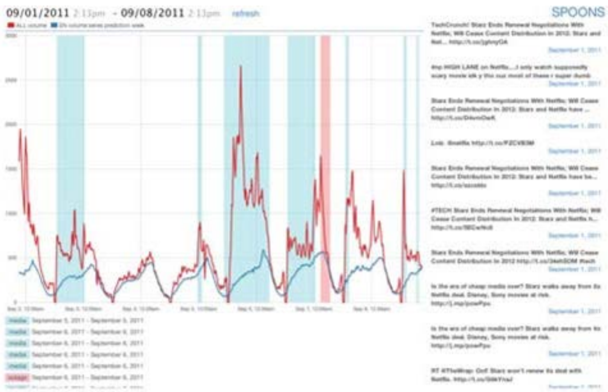

Figure 2: A screenshot of the SPOONS UI depicting multiple time series (predicted traffic vs. actual traffic), media and outage events, and a list of relevant tweets for a given time period.

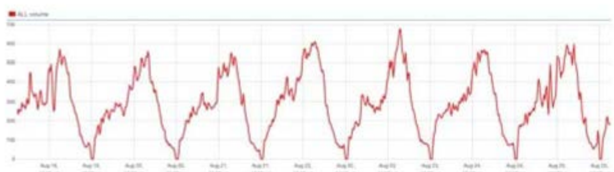

Figure 3: A 7 day period (August 18 - August 25) with no media events or serious outage.

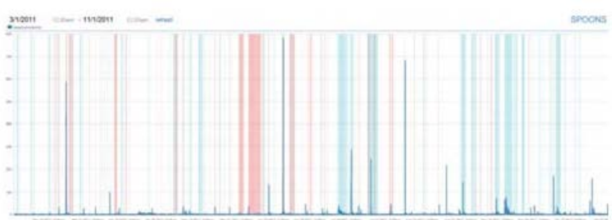

Figure 4: Volume of Netflix-related Twitter traffic containing the phrase "is down" between January and November of 2011.

Normal MSE	Anomalous MSE	Outage MSE	Media MSE
12,043	294,357	45,748	698,436

Table 2: The MSE data for the Netflix-related Twitter traffic prediction.

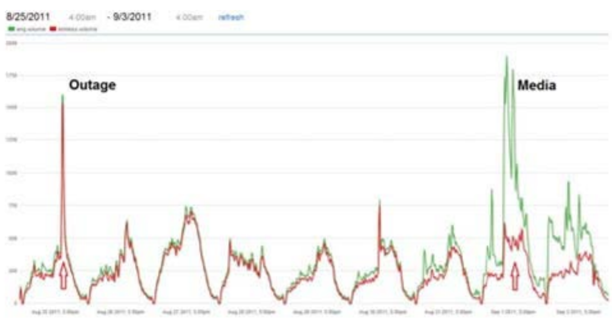

Figure 5: Linkless traffic and total Netflix-related Twitter traffic between August 25, 2011 and September 3, 2011

Time Series	Outage MSE / Normal MSE	Media MSE / Normal MSE
Keyword	83.19	18.80
Linkless	4.33	40.33
Total	3.80	57.99

Table 3: The MSE data for the Netflix-related Twitter traffic prediction.

Bayes Net				J48			
	Media	Bad	Other		Media	Bad	Other
Media	101	0	2	Media	99	1	3
Bad	3	101	200	Bad	3	208	93
Other	22	16	388	Other	22	35	369
KNN				Naive Bayes			
	Media	Bad	Other		Media	Bad	Other
Media	59	0	44	Media	80	0	23
Bad	0	234	67	Bad	1	209	94
Other	20	25	381	Other	10	17	399
SMO				Committee			
	Media	Bad	Other		Media	Bad	Other
Media	98	0	5	Media	99	0	4
Bad	3	234	67	Bad	3	213	88
Other	20	25	381	Other	18	18	390

Table 4: Tweet Classification results for the training set.

Time Series	Outage MSE / Normal MSE	Media MSE / Normal MSE
KNN	25.47	12.56
Naive Bayes	9.97	22.25
Committee	6.26	26.28
SMO	5.73	25.74
Bayes Net	5.62	28.45
J48	5.40	22.23
Total	3.80	57.99

Table 5: The MSE data for the Netflix-related Twitter traffic prediction.

Time Period	Begin	End
Data	January 20, 2011	September 30, 2011
Evaluation	March 14, 2011	September 30, 2011

Table 6: Detection Evaluation Time Periods

Method	Monitor	Precision	Recall	F0.5 Score
Keyword	**Trend**	**0.828**	**0.235**	**0.309**
Linkless	Trend	0.237	0.489	0.361

Table 7: Simple Methods Detection Evaluation

Classifier	Monitor	Precision	Recall	F0.5 Score
J48	**Ratio**	**0.516**	**0.882**	**0.599**
SMO	Ratio	0.485	0.871	0.570
Committee	Ratio	0.600	0.355	0.488
Bayes Net	Ratio	0.500	0.215	0.347
Naive Bayes	Ratio	0.411	0.570	0.453
KNN	Ratio	0.381	0.430	0.396

Table 8: Classification Methods Detection Evaluation