Online Video Data Analytics

Benjamin Le
Jefferson Lai
Pierce Vollucci
Wenxuan Cai
Yaohui Ye
George Necula, Ed.
Don Wroblewski, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-70
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-70.html

May 13, 2015

Acknowledgement

We would like to thank George Necula for advising us throughout the entirety of this Capstone project. We would also like to thank Jibin Zhan from Conviva for introducing the problem space to us and Pat McDonough from Databricks for providing extensive technical support throughout our use of Databricks.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering and Computer Sciences

Data Science and Systems

Online Video Data Analytics

Benjamin Le

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: ____________________________ Date _____________

Print Name/Department: George Necula/Electrical Engineering and Computer Science

2. Faculty Committee Member #2:

Signature: ____________________________ Date _____________

Print Name/Department: Don Wroblewski/Fung Institute for Engineering Leadership

Abstract

This capstone project report covers the research and development of Smart Anomaly Detection and Subscriber Analysis in the domain of Online Video Data Analytics. In the co-written portions of this document, we discuss the projected commercialization success of our products by analyzing worldwide trends in online video, presenting a competitive business strategy, and describing several approaches towards the management of our intellectual property. In the individually written portion of this document, we discuss and evaluate two algorithms used to detect anomalies in seasonal time series of service quality metrics, Autoregression and Seasonal Hybrid Extreme Student Deviate.

Contents

*Co-written with Jefferson Lai, Pierce Vollucci, Wenxuan Cai, and Yaohui Ye

I. Introduction

This report documents the Online Video Data Analytics capstone project completed in the course of the Data Science and Systems focus of the Master of Engineering degree at UC Berkeley. Through the collective efforts of Benjamin Le, Jefferson Lai, Pierce qVollucci, Wenxuan Cai, and Yaohui Ye, our team has not only characterized the need for effective data analysis tools in the domain of online video data, but has also developed analysis tools which attempt to address this need. As we will describe in detail in our Individual Technical Contributions, our work has produced many important findings and we have made significant strides towards a complete implementation of these tools. However, at the time of the writing of this report, additional work is required before our tools can be considered complete. That being said, our substantial progress has allowed us to form a very clear vision of what our finished tools will look like and how they will perform. Our vision leads us to believe that, once finished, our tools can be of great potential value to entities within the online data analytics industry. In order to understand how best to cultivate this value, we have extended our vision to depict tools to marketable products and we have evaluated the potential for our team to establish a business offering these products. In doing so, we have performed extensive research of the current market and industry which our potential business would be entering. The remainder of this report presents our findings and is divided into seven sections. First we introduce our industry partner Conviva in the Our Partner section. Second, we present the objective of our work and the motivation behind the resulting products in the Products and Value section. Third, we introduce and describe the dataset leveraged by our products in the Our Dataset section. Fourth, our team characterizes our industry as well as our competitive strategy in the Trends, Market, and Industry section. Fifth, in our Intellectual Property section, we describe how we plan to protect the value of our work. Sixth, the Individual Technical Contributions section of this report details our specific contributions toward the goals of our project. Finally, the Conclusion section contains a retrospective analysis of the significance of this work and provides an outlook on the potential for continuation of our work in future endeavors.

II. Our Partner

This project is sponsored by Conviva, a leading online video quality analytics provider. Conviva works with video content providers, device manufacturers, and developers of video player libraries to gather video quality metrics from content consumers. Through our partnership with Conviva, we have access to an anonymized portion of their online video quality metric dataset for the development of our products. We also have access to Conviva engineers for collaboration purposes who provide domain knowledge and on site support. For the purpose of the business analysis forthcoming, the entity, “we”, will refer to our capstone team as a separate entity from Conviva. Furthermore, we consider Conviva to be a close partner to our capstone team on whom we can rely for continuous access to their dataset.

III. Products and Value

A vast and painfully prevalent gap exists between the amount of data being generated around the world and the global tech industry’s ability to utilize it. According to IBM, “every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone” (“Bringing Big Data”). While the already monumental quantity of data continues to grow, scientists and engineers alike are just beginning to tap into the power of this data. This is not to say that data does not already pervade nearly every imaginable aspect of life today; it does. Large amounts of data crunching and predictive analysis go on behind the scenes of numerous activities, from returning search queries, to recommending movies or restaurants, to predicting when and where the next earthquake will occur. However, there remains a massive body of questions and problems in both academia and industry that researchers have been unable to use data to answer. One domain in which better utilization of data could yield tremendous benefit is that of online media. Our team aims to serve this niche by building tools that address two critical challenges of online video data analysis: accurate real-time anomaly detection on large scale data and subscriber churn analysis.

Online video providers struggle to consistently serve a TV-like experience with high quality video free of buffering interruptions. Many factors within the "delivery ecosystem" affect the throughput of a video stream and, ultimately, the end user's viewing experience (Ganjam et.al 8). These factors include "multiple encoder formats and profiles, CDNs, ISPs, devices, and a plethora of streaming protocols and video players" (Ganjam et al. 8). An automatic anomaly detection and alert system is necessary in order to both inform a video content provider when their customers are experiencing low quality and, among the many possible factors, diagnose the primary cause of the problem. For example, if all customers experiencing frequent buffering belong to a certain ISP, then the alert system should flag that ISP as the root of the problem. The challenge that plagues many current solutions, however, is related to the aforementioned growth in the amount of collected data. While it is easy to detect when and why predictable measurements misbehave at small scale, it is hard to do so with high accuracy at large scale, across a range of system environments. To meet this challenge, our team has developed our Smart Anomaly Detection system to detect when and why truly anomalous and interesting behavior occurs in measured data. Such a system would greatly help content providers improve both their operational performance and efficiency. This value will be passed down to the viewers who benefit from higher service quality.

A second problem for subscription-based online video content providers is the ability to retain their subscribers. While the problem of diagnosing and eventually reducing subscriber churn has existed as long as the subscription service model, only recently has the tech industry developed the capacity and means to use big data to do so (Keaveny). Furthermore, largely due to the fact that online video hosting and distribution is a relatively new service, nearly all previous works in the area have focused on other domains such as telecommunications or television service subscriptions (Keaveny;

Verbeke 2357-2358). Our team’s Subscriber Analysis toolset aims to fulfill this unmet need by developing predictive models of viewer engagement and churn based on viewing activity and service quality data. Being able to predict churners and identify characteristic predictors from the data allows companies to focus on addressing the problems most critical to their viewers, thereby reducing churn rates. As proven by Zeithaml, there is a real, high cost associated with subscriber churn (Zeithaml). Thus by aiding in the reduction of churn, our Subscriber Analysis product can both help content providers increase revenues and result in higher overall satisfaction for those who purchase online video subscriptions.

For the reasons described above, our team is confident that our Smart Anomaly Detection and Subscriber Analysis products are important and valuable to both content providers and their customers, the viewers.

IV. Our Dataset

Conviva provided 4.5 months of session summary data from a single anonymous content provider for our research and development. 73,368,052 rows of session summaries are in this dataset. Each session summary represents a single instance of a viewer requesting a video object. In addition to service quality data, the type of device used by the viewer, the approximate location of the viewer, and metadata about the video content being accessed are collected into 45 columns. Subscription and demographic information about a viewer beyond their location are not available within this dataset. Fields that might otherwise help identify the anonymous content provider such as video content metadata were also anonymized by Conviva prior to data transfer to protect their customer.

Although the data was preformatted by Conviva before being transferred to our capstone team, we identified two important challenges implicitly encoded within this data through exploratory data analysis and follow-up communication with Conviva engineers. First, several of the fields in the session summaries are not as reliable as we initially believed. For example, fields such as `season` and `episodeName` are often empty. Second, our initial dataset included data generated by an artificial “viewer” that was used by Conviva for testing purposes and exhibited very strange, abnormal behavior. This was very important to keep in mind as we developed and evaluated our tools based on this data.

For our Smart Anomaly Detection product, Conviva informed us of the two most important metrics in assessing QoS that they wished to detect anomalies for. First is the number of attempts in watching online video over a time period. Low number of attempts indicates that users may be unable to access the content due to a datacenter failure. A high number of attempts signals the presence of a viral video. Second is the video start failure (VSF) rate. The VSF rate is the percentage of attempts that have failed to begin properly. VSFs may be caused by bugs in the video player software or by improper encoding/decoding of video content. Unlike attempts, low VSF rate is not a concern for video content providers. However, high VSF rate indicates major issues in the content delivery pipeline. To determine if an attempt has ended in VSF, we look at the `joinTimeMs` and `nrerrorsbeforejoin` columns in the data. The table below provides description about these 2 columns. An attempt ends in VSF if `joinTimeMs < 0 AND nrerrorsbeforejoin > 0`.

Column Name	**DataType**	**Description**
joinTimeMs	int	How long this attempt spent joined with the video stream. If this attempt has not yet joined, then this value will default to -1.
nrerrorsbeforejoin	int	How many fatal errors occurred before video join

For Subscriber Analysis, on the other hand, the nature of the problem is that we cannot know beforehand which fields within the session summary are useful in distinguishing viewers who are likely to churn. At the same time, as a consequence of the first of the challenges mentioned above, the Subscriber Analysis product should not indiscriminately use all fields of the session summary, including both reliable and unreliable fields. Thus, a central component of the work in Subscriber Analysis revolves around selecting a subset of these fields to use to form “features” to be used by the product.

V. Trends and Strategy

Having defined our team’s product and established both how they generate value and for whom they are valuable, we can focus on how we plan to bring these products to market from the standpoint of a new business. Amidst an era of rapid information and especially within the technology-abundant Silicon Valley, bringing such innovations to market requires understanding the market and having a well-formed competitive strategy. In this section, we describe the social and technological trends relevant to our product as well as the market and industry our business would be entering. We then describe the strategy we have developed that would allow our business to be successful in this competitive environment.

Why Us, Why Now

In the past five years, the number of broadband internet connections in the United States has grown from 124 million in 2009 to 306 million in 2014, leading to a compound annual growth rate of 19.8% per year (“Num. of Broadband Conns.”). This growth is indicative of the ever-growing role the Internet plays in daily life. Along with the growth of the Internet, as both a cause and effect, comes the spread of online services. In her article for Forbes, Erika Trautman, CEO of Rapt Media, states that “each year, more and more people are ditching cable and are opting for online services like Netflix and Hulu.”

The emergence of online video services has been so disruptive a shift in video distribution, that it incited a 2012 public hearing concerning public policies from the Senate Committee on Commerce, Science, and Transportation. In the hearing, leaders from technology juggernauts and state senators alike echoed the same viewpoint: online video services are the future of video distribution. Susan D. White, the Vice Chair for Nielsen, a leading global information and measurement company, reported that “the use of video on PCs continues to increase—up 80 percent in the last 4 years…Consumers are saying, unequivocally, that online video will continue to play an increasing role in their media choices” (U.S. Sen. Comm. on Commerce, Sci. & Trans. 9).

Of course, similar to other industries, a business seeking to enter today’s online video industry must meet a myriad of both business and engineering challenges. Unlike many of these industries, however, our industry is well-positioned to easily collect and analyze vast amounts of data to meet these challenges. Out of these conditions, the online video analytics (OVA) industry emerged, helping to translate and transform this data into useful insights that can be directly used by online video providers. A report by Frost & Sullivan summarized the rapid growth in the market:

> Still a largely nascent market, online video analytics (OVA) earned $174.7 million in revenue in 2013. It is projected to reach $472 million in 2020 as it observes a compound annual growth rate (CAGR) of 15.3%....The growth of OVA is largely attributed to the high demand for advanced analytics from online video consumption (Jasani).

Spurred by the massive opportunity in this market, our team has worked with Conviva to identify two of the most significant technical challenges faced by content providers: real-time detection of anomalies in a rapidly changing, unpredictable environment and efficiently reducing subscriber churn.

The challenge of retaining subscribers has existed as long as the subscription-based business model itself. As the competitive landscape of the online video market continues to evolve, the ability to diagnose and mitigate subscriber churn is a crucial component for business success. Sanford C. Bernstein estimated Netflix’s average annual churn rate at 40-50%, which translates to 24-30 million subscribers (Gottfried).

Reducing this churn rate by even a small fraction and keeping the business of these subscribers could mean significant increases in revenue. Just as critical for success is the ability to detect and respond to anomalies or important changes in metrics such as network usage and resource utilization. On July 24, 2007, 18 hours of Netflix downtime corresponded with a 7% plummet in the company's stock (Associated Press).

As previously described, our team provides solutions to these challenges through our Subscriber Analysis and Smart Anomaly Detection products. We believe that while these solutions, which use a combination of statistical and machine learning techniques, are powerful, our primary value and competitive advantage lies in our use of the unique dataset available to us through our partnership with Conviva. In the following sections, we discuss in detail how we plan to establish ourselves within the industry. In particular, we describe how we will position ourselves towards our buyers and suppliers as well as how we will respond to potential new entrants and existing competitors to the market.

Buyers and Suppliers

One of the most important components of a successful business strategy is a deep and accurate understanding the different players involved in the industry. In particular, an effective strategy must define the industry's buyers, to whom businesses sell their product, and its suppliers, from whom businesses purchase resources. In this section, we provide an overview of important entities related to our industry and present an analysis of our buyers and suppliers.

Potential customers for the global online video analytics market include content providers, who own video content, and service providers, who facilitate the sharing of user-generated video content (Jasani; Smith). Among the content providers are companies such as HBO, CCTV, and Disney, who all bring a variety of original video content to market every year. These businesses serve a huge user base and are able to accumulate large amounts of subscriber data. HBO alone was reported to have over 30 million users at the beginning of 2014 (Lawler). This abundance of data presents massive potential for improving these companies' product quality and, correspondingly, market share. Our Subscriber Analysis product can realize some of this potential by helping understand the experience and behavior of their users. Furthermore, with our Smart Anomaly Detection product, content providers can be made aware when significant changes occur in viewer behavior, system performance, or both. These tools can lead to a more valuable product, as seen from the content provider's viewers. While service providers such as Twitch or Vimeo differ from content providers in that they tend to offer free services, the success of these companies is still highly dependent on retaining a large number of active users. Thus, we target service providers in much the same way as we target content providers. Overall, we find that content and service providers, as buyers, are at an advantage in terms of business leverage over us, as sellers. This is primarily due to low switching costs, which arise from the fact that other businesses such as Akamai and Ooyala offer products for processing video data similar to ours (Roettgers). Because buyers ultimately make the choice choosing where to send their data on which both Smart Anomaly Detection and Subscriber Analysis depend, it can be difficult to deter customers from switching to our competitors. However, as we describe later in this paper, our unique approach towards churn analysis may differentiate us from our competitors and decrease buyer leverage over us.

On the other end of the supply chain, we also must consider who our suppliers will be and what type of business relationship we will have with them. Because our product exists exclusively as software, we require computing power and data storage capacity. Both of these can be obtained through the purchase of cloud services. Fortunately, the current trends indicate that cloud services are becoming commoditized, with many vendors such as Amazon, IBM, Google and Microsoft offering very similar products (Hanley). Though our buyers benefitted from low switching costs between us and our competitors, we face even lower switching costs between our suppliers. This is because while there is a considerable amount of effort involved with integrating a monitoring or analytical system with a new set of data, migrating the services between the machines which host them is almost trivial, involving only a transfer the data and minor machine configuration. In addition to cloud services, to a certain extent, we are dependent on device manufacturers and developers of video player libraries. We require them to provide an Application Program Interface (API) which we can use to gather online video analytics data from users. Fortunately, prior relationships with these device manufacturers and developers have been established through our partner Conviva. Conviva can help us open APIs for new devices and video players to maintain the flow of data required for our products.

As Porter argued, strategic positioning requires performing activities either differently or more efficiently than rivals (“Five Competitive Forces” 11). Our partnership with Conviva affords us a large quantity of high quality data for our algorithms to utilize, giving us a slight advantage compared to other services. In order to maintain and build upon this advantage, however, we must focus on developing our products to utilize this data and yield results in a superior manner. Thus, it is clear that our ability to differentiate from competing products and outperform them is key to our business strategy and the following sections describe how we can do so.

New Entrants

“Know yourself and know your enemy, and you will never be defeated” (Sun Tzu 18). This proverb can be applied to almost any competitive situation, from warfare to marketing. Interpreting this teaching in the context of business strategy, we identify that understanding the rivalry among existing and potential competitors is essential to a lasting competitive advantage. This interpretation fits well within the framework of Michael Porter’s five competitive forces. We now examine new entrants through the incumbent advantages and barriers to entry that work to keep this force as a low threat to both of our products. Porter recognized seven incumbent advantages (“Five Competitive Forces” 4-6). The first is supply side economies of scale in which established incumbents have tremendous strength. The code behind a given analysis program is a fixed cost which scales well with an increased number of users, thus reducing the marginal cost of the code with each customer. The servers that receive and process the various users' data are linear, but scale with the number of customers acquired. The real advantage comes from the exponential power of the data supplied by these same customers, a theme we have come back to repeatedly in this paper. As the breadth and quantity of data increase with the combined user base of our customers, our algorithms become increasingly powerful and allow the incumbent product to outperform new entrants. This leads into our second advantage, demand side benefits of scale. As the authority in the field of providing content providers with analytics, incumbents can encourage customer demand by using their data on content quality improvements to provide hard evidence of the bottom line improvement new users can expect. "Increasingly powerful predictive analytics tools will unlock business insights [and drive revenue]" (Kahn 5). Demonstrating that our tools provide access to increases in revenue is key to nurturing demand.

Switching from an incumbent's service provides another barrier to entry, customer switching costs. While switching from one online service to another is not prohibitively expensive considering the benefits offered, the most impacting loss is in the past data the incumbent analysis provider's algorithms had of user's performance. "As we increase the training set size L we train on more and more patterns so the test error declines" (Cortes et al. 241). Via additional training examples, the incumbent's algorithm would consistently outperform the new entrant as the new entrant slowly acquires a pool of data comparable to that of the incumbent.

Just as it does not appear expensive for a customer to switch, it appears feasible for new entrant to join due to minimal physical capital requirements. With Platform as a Service (PaaS) providers, a new entrant merely needs a codified algorithm and a client or two to get started. Still, it is again the data that proves key to providing value to our customers. Importantly, new entrants cannot attain this data until they acquire clients, a classic catch-22 which serves as an inhibiting capital requirement for new entrants.

The global reach of our data partner, Conviva, provides both a size independent advantage as well as an unequal access to potential distribution channels in that it allows for direct international sales in the form of immediate integration of our tools with the systems of our partner's customers. The last relevant advantage as discussed by Porter, concerns restrictive government policy. Privacy concerns do arise when personal data is used, however there are standards for anonymization to be employed when using such data (Iyengar). While governments do allow the use of such data, it has to be acquired by legal means, which means a new entrant is restricted in its means of gathering new data for its algorithms. Thus, after a thorough analysis of the potential new entrants of our industry, the incumbents' advantages suggest that the threat of new entrants is a relatively weak force in our industry.

Existing Rivals

Another category of threats that a successful business strategy must address is that of existing rivals. As Porter described, the degree to which rivalry drives down an industry's profit potential depends firstly on the intensity with which companies compete and secondly on the basis on which they compete ("Five Competitive Forces" 10). We analyze these two parts for each of our products separately.

As machine learning grows in popularity, research into anomaly detection and other analyses of time series data is receiving greater attention both in academia and in industry. A survey of anomaly detection techniques shows a variety of techniques applied in a diverse range of domains (Chandola). Our strategy must take into account the threat of commercialization of technologies into industry competitors. For example, in 1994 Dipankar Dasgupta used a negative selection mechanism of the immune system to develop a "novelty" detection algorithm (Dasgupta). In addition to these potential competitors, there already exist several important industrial competitors working on anomaly detection. In January, 2015, Twitter open-sourced *AnomalyDetection*, a software package that automatically detects anomalies in big data in a practical and robust way (Kejariwal). Our Smart Anomaly Detection product is comparable to products from industry competitors such as Twitter; it is able to integrate with various sources of data, perform real-time processing, and incorporate smart thresholding with alerts. Although our competitors may try to research and develop a superior anomaly detection algorithm, we believe that our superior quantity and quality of data provided by Conviva gives us an edge over our competitors. Thus, we characterize competitive risk for Smart Anomaly Detection as weak. To a large extent, the competitors of Subscriber Analysis include the content providers themselves. Netflix spends $150 million on improving content recommendation each year, with the justification that improving recommendations and subscriber retention by even a small amount can lead to significant increases in revenue (Roettgers). These content providers have the advantage that they have complete access and control over the data they collect. If most companies were able to build an effective churn predictor in-house, the industry would be in trouble. However, we are confident that the quality of our Subscriber Analysis product will overwhelmingly convince content providers facing the classic “buy versus build” question, that building a product of similar quality would demand significantly more resources than simply purchasing from us (Cohn). This confidence is further supported by Porter in the context of the tradeoffs of strategic positioning (“What Is Strategy?” 4-11). In addition to content providers, there also exist competitors such as Akamai and Ooyala, who offer standalone analysis products to content and service providers. These competitors tend to focus on the monitoring and visualization of the data. In contrast, Subscriber Analysis focuses on performing the actual analysis to identify the characteristics and causes of subscriber churn.

Still, our most important advantage over these competitors remains our ability to perform in-depth churn analyses based on the abstraction of session summaries, which consist of a unique combination of metrics exclusively related to service quality. To the best of our knowledge, this is unique to previous and existing works in subscriber churn analyses. Our research has shown that the most prominent existing analysis approaches all incorporate a significant amount of information, often involving direct customer surveys or other self-reported data. Because service quality data is abundant and easy to obtain compared with demographic data, our Subscriber Analysis product can appear extremely appealing to potential customers. This easy to collect and consistent subset of video consumption data means our product has the potential to scale much better than existing approaches which require highly detailed, case-specific, and hard to obtain datasets. However, we cannot guarantee that this algorithmic advantage be sustained as our competitors continue their own research and development. Thus, we conclude that threat of competition to Subscriber Analysis is moderate.

Substitutes

The final element of our marketing strategy concerns the threat of new substitutes. Porter defined substitutes as products that serve the same purpose as the product in question but through different means (“Five Competitive Forces” 11). We first discuss potential substitutes for our Smart Anomaly Detection product.

The gold standard for most alert systems is human monitoring. Analogous to firms hiring security monitors to watch over buildings, video content providers can hire administrators to keep watch over network health. A more automated substitute is achieved through simple thresholding, in which hardcoded thresholds for metrics such as the rate of video failures trigger an alarm when exceeded. Content providers can also utilize third party network performance management software from leaders like CA, Inc. This type of software alerts IT departments of potential performance degradation within the companies' internal networks (CA Inc. 4). Similarly, content providers can pursue avenues besides Subscriber Analysis to reduce churn rates. Examples include utilizing feedback surveys and consulting expert market analysts. Feedback from unsubscribers is an extremely popular source of insight into why customers choose to leave and can go a long way in improving the product and reducing churn rate. These often take the form of questionnaires conducted on the company’s website or through email. In addition, content providers commonly devote many resources towards consulting individuals or even entire departments with the goal of identifying marketing approaches or market segments that generate lower churn rates.

Porter classified a substitute as a high threat when the substitute offers superior price/performance (“Five Competitive Forces” 12). With this in mind, we found that the overall threat of substitutes for Smart Anomaly Detection product is low. In contrast to human monitoring, our product offers a superior value proposition to our buyer. According to Ganjam et. al, many factors, including "multiple encoder formats and profiles, CDNs, ISPs, devices, and a plethora of streaming protocols and video players," affect the end user’s viewing experience (Ganjam 8). The complexity of this delivery ecosystem requires equally complex monitoring with filters to isolate a specific ISP, for example, and to determine if its behavior is anomalous. Such large scale monitoring does not scale efficiently when using just human monitoring. Similarly, simple thresholding poses little threat as a substitute because fine tuning proper thresholds over multiple data streams is difficult and time consuming. Many false positives and negatives still occur, despite such fine tuning (Numenta 11). Network performance management software, on the other hand, poses a considerable threat to us. However, while they are excellent at detecting problems within a content provider’s internal network, they alone cannot increase the quality of service. Xi Liu et al. argue that an optimal viewing experience requires a coordinated video control plane with a “global view of client and network conditions” (Liu 1). Fortunately, thanks to our partnership with Conviva, we have the data necessary to obtain this global view.

Just as with Smart Anomaly Detection, the threat of substitutes for Subscriber Analysis is also low. Although feedback surveys are direct and easy to implement, there are several inherent issues associated with them. Perhaps most prominently, any analysis that uses this data format must make a large number of assumptions in order to deal with uncontrollable factors such as non-response bias and self-report bias (Keaveny). Expert opinion, whether gathered from a department with the company or through external consult, is the traditional and most common approach towards combating subscriber churn. This method, while very effective, tends to be extremely expensive. Still, as demonstrated by Mcgovern’s Virgin Mobile case study, expert opinion can lead to identifying the right market segment, lower churn rates, and ultimately a successful business (McGovern 9).

To mitigate the threat of substitutes, Porter suggests offering “better value through new features or wider product accessibility” (“Five Competitive Forces” 16). For Smart Anomaly Detection, there are several avenues to pursue to provide a better value proposition to our buyers. For example, we can develop more accurate predictors with additional data from Conviva and explore new machine learning algorithms. For Subscriber Analysis, the threat of substitutes continues to be low because, unlike the examples given above, our product can perform effective analyses and generate valuable insights in an automated, efficient fashion. Data obtained through direct customer surveys, while potentially cheap, come bundled numerous disclaimers and can lead to a certain stigma from the subscriber’s perspective. Furthermore, although data obtained through surveys, such as demographic information, might be more helpful in characterizing churners, by focusing on providing churn analysis based only on service quality data, our Subscriber Analysis product has at least one significant advantage. Service quality data from content consumers can be more easily gathered compared to data such as demographic information. Consequently, our product can be more appealing and accessible to content providers, especially those who do not have access to, or would like to avoid the cost of obtaining, personal data about their users. We also point out that both Subscriber Analysis and the substitutes such as those described above can be used in combination with each other. In such a case, our Subscriber Analysis product becomes even more appealing. This is because it can use the data from customer feedback to yield further improved performance. Our product would also make tasks such as identifying appropriate market segments much easier and cheaper to accomplish for content providers.

Strategy Summary

In summary, there are several social and technological trends which make now the right time for commercializing our Subscriber Analysis and Smart Anomaly Detection products. The most prominent among these are the rapid growth in internet connectivity and the spread of online services. In order to evaluate how well positioned we are to capitalize on the opportunity created by these trends, we developed a business strategy through competitive industry and market analysis from several different perspectives. From the perspective of buyers and suppliers, though we find that buyer power is significant, over time we expect to differentiate ourselves from our competitors by leveraging both the superior size of our dataset and our more efficient overall use of the data. We find that supplier power is low for our industry because the only significant resource we require is available through cloud services, an industry in which we have high buyer power and which is quickly becoming commoditized. From the perspective of rivals, the threat of new entrants is low due in large part to the superior quantity and quality of our data as well as the benefits of scale we would stand to benefit from as incumbents. Similarly, while existing competitors do present a threat, we find that our use of superior data and unique approach gives us a significant competitive advantage over them. Finally, we see a weak threat from the perspective of substitutes because we offer superior value at a cheaper price to our customers that only improves in combination with other techniques. Taken together, our evaluations lead us to believe that there is significant potential for a sustained competitive advantage over competitors, and that now is an opportune time to pursue it.

VI. Intellectual Property

Equally important to a team’s ability to build a valuable product and bring it to market is its ability to protect that value. In this section, we explain how we, as a business pursuing the strategy above to bring Subscriber Analysis and Smart Anomaly Detection to market, intend to sustain and protect the value of our work.

The traditional method for protecting the value of a new technology or innovation is obtaining a legal statement regarding ownership of intellectual property, IP, in the form of a patent. Indeed, patents have performed well enough to remain a primary mechanism for IP protection in the US for more than 200 years (Fisher). Unfortunately, when it comes to software, the rules and regulations regarding patents become dangerously ambiguous. The recent influx of lawsuits involving software patents has been attributed to the issuance of patents that are unclear, overly broad, or both (Bessen). Despite software patent laws being an active and controversial topic, these discussions have simply left more questions unanswered. The *Alice Corporation v. CLS Bank* Supreme Court case in 2013 is oft cited as the first source of information about software patentability, and even this case has been criticized for the court's vagueness (*Alice Corporation v. CLS Bank*). As noted by patent attorney and founder of IPWatchDog.com Gene Quinn, a definitive line should be drawn by the courts: a patent describing only an abstract idea, without specific implementation details, is invalid and cannot be acted upon (Quinn).

Thus, faced with the question of patentability, our team must examine the novelty of our Subscriber Analysis and Smart Anomaly Detection products. The goals of Subscriber Analysis and Smart Anomaly Detection are to diagnose the causes of subscriber churn and intelligently detect important changes in measured data respectively. Because these goals are rather broad, there exist a number of existing implementations, both old and new, with similar objectives. As a team considering patentability, we look towards the novelty of our specific approach and implementation. In the course of this introspection, we note that our implementation amalgamates open source machine learning libraries such as SciKit-Learn, published research from both industry and academia, programming tools such as those offered by Databricks, and finally the unique data afforded to us through our partnership with Conviva. With this in mind, we conclude that current patenting processes are flexible enough such that by defining our implementations at an extremely fine granularity, we would likely be able to obtain a patent on our software. However, we strongly believe that there exist several significant and compelling reasons against attempting to obtain a patent for our work. In this section, we elaborate on these reasons and describe an alternative method for protecting our IP which better suits our situation and business goals.

There is an abundance of existing anomaly detection patents of which we must be wary. Several of these patents are held by some of the largest companies in the technology sector, including Amazon and IBM. For example, *Detecting anomalies in Time Series Data*, owned by Amazon, states that it covers “The detected one anomaly, the assigned magnitude, and the correlated at least one external event are reported to a client device” (U.S. Patent 8,949,677). One patent owned by IBM, *Detecting anomalies in real-time in multiple time series data with automated thresholding*, states that in the submitted algorithm, a “comparison score” is calculated by comparing “the first series of [observed] normalized values” with “the second series of [predicted] normalized values” (U.S. Patent 8,924,333). In observance of these patents, we must be wary of litigation, especially when it concerns large technology companies. Recently, many companies in the tech industry, both small and large, have come under fire with a disproportionate number of patent infringement lawsuits (Byrd and Howard 8). Some optimists argue that most companies need not worry, because large technology companies are likely filing patents defensively. However, these companies are often the ones who play prosecutor in these patent infringement cases as well. For example, IBM, a holder of one of these anomaly detection patents, has a history of suing startups prior to their initial public offerings (Etherington). More recently, Twitter settled a patent infringement lawsuit with IBM by purchasing 900 of IBM’s patents (Etherington). In a calculated move by IBM, Twitter felt pressured to settle to protect their stock price in preparation for their IPO. Thus, we must be extremely careful in how we choose to protect our intellectual property. If this means filing a patent, then we must be prepared to use it defensively. This is likely to require a very large amount of financial resources. As we do not currently have these resources to spare and cannot guarantee that the protection offered would be long lasting or enforceable, we seek an alternative to patenting.

The goal of our Subscriber Analysis product is to predict the future subscription status of users based on past viewing behavior. Despite our research on existing patents, our team has been unable to find many patents which pose a legal threat to Subscriber

Analysis. Most active patents on video analytics focus on video performance and forecast, such as Blue Kai Inc's *Real time audience forecasting* (US Patent App. 20120047005). In contrast, the patent field of quantization and prediction of subscriber behavior remains largely unexplored. Despite several commercial solutions on the market, there has not been a corresponding number of patents. Thus, Subscriber Analysis does not face the same level of risk of litigation compared to Smart Anomaly Detection. However, there are a handful of patents in other domains that we need to be wary of. *System and method for measuring television audience engagement*, owned by Rentrak corporation, describes a system that measures audience engagement based on the time he or she spends on the program (US Patent 8,904,419). In short, it constructs a viewership regression curve for different video content and measures the average viewing length. For a new video, the algorithm infers the level of viewer engagement based on the video content and the duration the viewer watched. While viewer engagement is a critical component for predicting behavior in Subscriber Analysis, we also incorporate additional data. These include viewing frequency, content type, and video quality. Under such circumstances, we do not see it as necessary to license patents such as the one above for two reasons. First, and perhaps most importantly, we apply churn analysis in the domain of online video, whereas most relevant patents apply to other older domains. Second, our algorithm incorporates a unique set of features corresponding to the data provided by Conviva.

The decision to pursue and rely on a patent in the software is an expensive one in both time and financial resources as well as a risky one due to the tumultuous software patent environment. As such, while we may pursue a patent, it will not be relied upon for our business model. As such, we have two additional IP strategies to investigate, open sourcing and copyrighting.

Open source software is software that can be freely used, changed, and shared (in modified and unmodified form) by anyone, subject to some moderation (Open Source Initiative). Open sourcing has become increasingly popular; both the total amount of open source code and the number of open source projects are growing at an exponential rate (Deshpande, Amit et al). For the purposes of our endeavor, it is not the novelty of our approach but our dataset and partner provided distribution network that distinguishes us. As the algorithms used are already publicly available, open sourcing our code does not cost us anything but provides us the shield of using open source software for our business and the badge having our code publically exposed and subject to peer review. Our business model would entail providing a value-added service company, dedicated to helping customers integrate their existing systems with our anomaly detection library. Through our partnership with Conviva, we have an established distribution network to our potential customers who we can offer immediate integration with Conviva's existing platform. This is a significant advantage as while open source is openly available to all users, they are primarily for experienced users. Users have to perform a significant amount of configuration before they begin using the code, which can pose quite a deterrent. While we will use the open source codebase as a foundation for our service, we will additionally provide full technical support in designing a customized solution that meets the customer's needs. By pivoting towards this direction, we add additional monetary value to the product that we can sell and bridge the technical gap for unexperienced users, relying on a SAAS implementation style for our business model instead of on a patent.

Copyright for software provides another IP Strategy option. While debate continues to surround software patents, copyrights are heavily applied in software. As expressed by Forbes's Tim Worstall, "there's no doubt that code is copyright anyway. It's a specific expression of an idea and so is copyright." There are several differences in the protection offered by copyrights compared to that of patents. While a patent may expose a very specific invention or process to the public and protect for 20 years, a copyright offers much broader protection while still providing the threat of lawsuit for enforcement. The copyright lasts 90 years past the death of the author and offers statutory damages (Copyright.gov). In addition, the scope of what it encompasses proves more relevant to our endeavor. "Multiple aspects of software can qualify for copyright protection: the source code, the compiled code, the visual layout, the documentation, possibly even the aggregation of menu commands" (Goldman). By protecting the numerous aspects of our project, copyright provides us adequate security. Besides the advantages of the protection offered, the process is affordable and efficient. Copyright is automatic as soon as a work is completed, though to file for statutory damages, one must formally register for a fee of less than $100 and an application turnaround time of under a year (Copyright.gov). In addition, even prior to completion of the work, we can preregister with a detailed explanation of the work in progress.

All IP strategies come with risks and copyright is no different. While pursuing a strategy of trade secrets would make our code more private, we would risk losing our protection should the secret be compromised. Also, as a general security principle in the computer science field, only the bare minimum should be relied upon to be kept secret to minimize risk of loss. However, completely publicizing our code for our copyright can be equally dangerous as the competition could copy our code with only slight rewrites. To remedy this, we can limit access to the raw code and only publish the required first and last 25 pages of code needed to attain a copyright on the entire work. In addition to this measure, it is our unique dataset that is the source of our code's advantage over our competitors, and this is already protected by our partner, Conviva, in its aggregated form as a trade secret,

We believe that the novelty of our code and the application of our techniques to our unique dataset would allow us to obtain a software patent. However, while a patent may be most effective at reducing our risk of litigation, we look to alternatives due to the current complexity of filing a software patent and the immense amount of financial resources required to do so. Our research has led us to two very appealing alternatives: open sourcing and copyrighting. For the reasons stated above, we believe that while each of these alternatives have their own risks, their respective merits make them more appropriate for our use than patenting. Moving forward, we plan to employ open sourcing, as we expect that building a large, open community of support will encourage adoption and most benefit our products.

VII. Technical Contributions

1. Overview

Introduction

As previously described in the Product and Values subsection, online video providers must execute two activities to successfully emerge against their competition. One, they must reduce subscriber churn and two, they must provide consistent high quality of service (QoS). Fortunately through online video analytics, solutions are available to mitigate both problems.

Our Solution

Our capstone team developed two tools to assist online video providers in solving their subscriber churn and QoS problems. First is our Smart Anomaly Detection product, an automated alert system for detecting anomalies in service quality metrics. Second is our Subscriber Analysis product, a set of tools that quantify and predict future user engagement.

Subscriber Analysis: Less engaged users who may be on the verge of unsubscribing can be identified with our Subscriber Analysis product. Video content providers then analyze this subset of users to model the behavior of churners. They can then use this information to act preemptively in preserving the customer relationship and reduce churn.

Smart Anomaly Detection: When problems occur within the video delivery ecosystem, it is important for video content providers to mitigate any downtime and restore QoS as quickly as possible. The total downtime in fixing a problem is equal to the time to detect the problem plus the time to diagnose the problem and implement a fix. Through our

Smart Anomaly Detection alerts, video content providers can reduce the detection time and possibly, the diagnosis time. In turn, this increases the customer's average QoS.

Task Breakdown

With two products, our capstone team naturally divided into two sub teams. Jefferson Lai and Yaohui Ye joined the Subscriber Analysis team while Pierce Vollucci, Wenxuan Cai, and I joined the Smart Anomaly Detection team. For more detailed tasks completed by the Subscriber Analysis team, one can refer to the technical papers written by Jefferson Lai and Yaohui Ye.

Pierce Vollucci, with his strong background in statistics, ensured that our sub team's methods were statistically sound through verifying any goodness of fit required from our data (e.g. normality). Pierce Vollucci was also responsible for developing tools to validate the results of our anomaly detectors. In addition, Pierce Vollucci was the primary driver in developing a spike detection algorithm using moving averages for static metrics.

Wenxuan Cai worked in conjunction with Pierce Vollucci on the spike detection algorithm by modeling the process generating the data with a Weibull distribution. He also researched and developed methods for pruning anomalies from our dataset so that outliers were not used during training.

I was responsible for the program used to aggregate and extract information from the dataset in our anomaly detectors. I was also the primary driver in developing an autoregressive model used to detect anomalies in seasonal metrics. Furthermore, I wrote a Python implementation of Twitter’s Seasonal Hybrid Extreme Student Deviate (Seasonal ESD) method for anomaly detection in our dataset (Vallis et al. 3). Pierce Vollucci and Wenxuan Cai used my implementation of those library functions to quickly wrangle the dataset into a usable form for development. My autoregressive model was also integrated into the algorithm suite used by our Smart Anomaly Detection system.

The remainder of this paper is organized as follows. Section 2 provides background on alternative existing anomaly detection systems from Etsy, Twitter, and Numenta. Section 3 describes my methods of discovering, wrangling, and profiling the team's data. Section 4 describes how to model the data using autoregression and Seasonal ESD and reports the results of these models in classifying anomalous data points.

2. Related Works

During our foray into anomaly detection for online video quality metrics, we discovered similar existing anomaly detection systems. Etsy and Twitter provide open sourced software for anomaly detection, though they are designed for IT operation metrics. Grok, an anomaly detection service from Numenta, is advertised as IT analytics for the Amazon cloud. Though all systems share a similar goal, they are all different algorithmically.

Etsy: Skyline by Etsy is a real-time anomaly detection system for monitoring of hundreds of thousands of metrics without manual thresholding ("Skyline" section Introduction). Because Skyline is designed to handle time series metrics with different characteristics from each other, one algorithm cannot model the entire anomalous pattern space. Thus, anomalies are predicted by taking a vote across an ensemble of anomaly detection algorithms ("Analyzer" section Algorithms). The percentage of votes needed to reach consensus can be manually adjusted by the user to tune Skyline's sensitivity to anomalies. Most of Skyline's base anomaly detection algorithms are simple three-sigma based. For example, Skyline tests if the current measurement is three standard deviations away from the exponentially weighted moving average or three standard deviations away from the lifetime average ("Algorithms.py" line 108). However, users are highly encouraged to extend the ensemble with additional algorithms specific to their use case to improve performance. We agree with Skyline's viewpoint that a consensus algorithm is most ideal for anomaly detection. In future work, one can implement a consensus algorithm by including our spike detection algorithm and our autoregressive algorithm in the ensemble.. Rather than strictly counting votes however, we believe one should also consider the confidence level of classification. Algorithms that vote with higher confidence should weigh more and vice versa.

Twitter: On January 6, 2015, Twitter announced AnomalyDetection, “an open source R package that automatically detects anomalies . . . in big data in a practical and robust way” (Kejariwal). This package is used throughout the company in monitoring both system and application metrics. To classify anomalies, Twitter invented the Seasonal ESD algorithm which combines the techniques of time series decomposition, robust statistics, and ESD together (Kejariwal). Through this combination, Twitter achieved an algorithm that exhibited low false positive rates with fast run time performance (Vallis et al. 4-5). We build upon Twitter’s work by writing a Python implementation of the Seasonal ESD algorithm. We evaluate its performance in detecting anomalies in video quality metrics and compare it to the autoregressive algorithm. A more detailed explanation of Seasonal ESD will be presented in Section IV of this paper.

Numenta: Grok by Numenta is an anomaly detection system that monitors system metrics in the Amazon Web Service environment. We briefly summarize the technical details behind the Grok anomaly detection system. We encourage readers who are interested in learning more about Grok to review Numenta’s white papers, “Hierarchical Temporal Memory” and “The Science of Anomaly Detection”. Hierarchical Temporal Memory (HTM) algorithms, a deep learning algorithm similar to Neural Nets, form the core of the Grok anomaly detection system. Inspired by research in neuroscience, HTM attempts to model how neurons in the neocortex interact with each other in response to new inputs and changes between inputs (“Hierarchical Temporal Memory” 28). The neocortex is simulated by a two dimensional array of cells. Each cell can be in one of three states: active, inactive, or predicted (“Hierarchical Temporal Memory” 19). Activated cells represent the neocortex’s response to the spatial and temporal patterns of the current input value. Activated columns are chosen based on the spatial encoding of the input value while rows are chosen based on values seen in previous time steps (“Hierarchical Temporal Memory” 28-31). Cells in a predicted state represent the expected response of the neocortex to the input value at the next time step. As the simulated neocortex observes more data from the monitored time series, it better learns common patterns within the data and encodes each pattern with a distinct combination of activated cells. The neocortex is then better able to recognize current behavior in the time series and is more accurate at predicting future values ("The Science of Anomaly Detection" 4). With accurate predictions, Grok can detect anomalies by looking for values with large hamming distances between the neocortex's predicted response and the actual response. While our work and Numenta's share the same goal in detecting anomalies, we use statistical rather than deep learning methods. Future work in comparing the performance between the two can be done to better understand the implications of favoring one method over the other.

3. Data Discovery, Wrangling, and Profiling

Similar to most data analysis problems, we perform 5 high level tasks: **discovering** data for analysis, **wrangling** data into a workable format, **profiling** the data through exploratory data analysis, **modeling** the data for prediction, and **reporting** results of the analysis (Kandel et al. 5). Discovery, wrangling, and profiling will be discussed in this section while modeling and reporting in the following section.

Data Discovery

Fortunately, Conviva provided a preformatted dataset for developing our Smart Anomaly Detection product. For more information about the Conviva dataset, we direct the reader above to the Our Dataset subsection of the paper. As mentioned previously, Conviva advised us to concentrate on performing anomaly detection on the attempts and video start failure (VSF) rate metrics.

Data Wrangling

To train anomaly detectors, we need to generate time series from the data for attempts and VSF. The first transformation we apply is a one to one mapping of a session summary row to the schema defined in the table below. By mapping the dataset to this new schema, we significantly reduce the amount of data processed during training. The initial dataset contains approximately 45 columns, many irrelevant to video quality. By reducing this to a schema with only six columns, the size of the working set shrinks greatly.

Column Name	**DataType**	**Description**
sessionstartepoch	int	Start of Session in Unix time
isVSF	boolean	Whether or not this session ended in VSF
asn	int	The autonomous system number of the Internet Service Provider (ISP) where the attempt originated from
a_objectid	string	The anonymized id of the video content being accessed.
tag_a_playerversion	string	The anonymized version of the video player used by the viewer
tag_a_playervendor	string	The anonymized vendor of the video player used by the viewer

After reducing the size of the working set, we apply a second transformation to aggregate the session summary rows into time series data with 10 minute time windows. The schema of our data after this second transformation is defined below.

Column Name	**DataType**	**Description**
unixtimestamp	int	Start of the 10 minute time window in Unix time
attempts	int	The number of attempts that were initiated in the 10 minute time window
vsf	float	The percentage of attempts that ended in video start failure in the 10 minute time window

We chose 10 minute time windows because lower intervals introduced more noise in the time series while higher intervals would cause us to detect anomalies at a pace much further away than real-time. Aggregation is done globally on the entire working set by counting the number of attempts in each time window and calculating the percentage of attempts that end in VSF. To build time series data for subset groups of the data, we can filter by any combination of ASNs, object ids, player versions, and player vendors before the aggregation step. However, performing anomaly detection over subset groups of the data is outside the scope of this paper. For analysis on anomaly detection over subset groups of the data for the VSF metric, we direct the reader to Pierce Vollucci's technical paper.

Data Profiling

For our exploratory data analysis, we graph each of the time series to better understand their characteristics. We generate graphs using the Python `matplotlib` package. The number of attempts is a seasonal metric. Figure 1 showcases the run-sequence plot for attempts. A wave pattern is present in the data with consistent peaks in attempts during the awake hours of the day. There is also weekly seasonality in the data with higher peaks on weekends than weekdays. An anomaly occurs on June 16, 2014 with a peak of 33000 attempts.

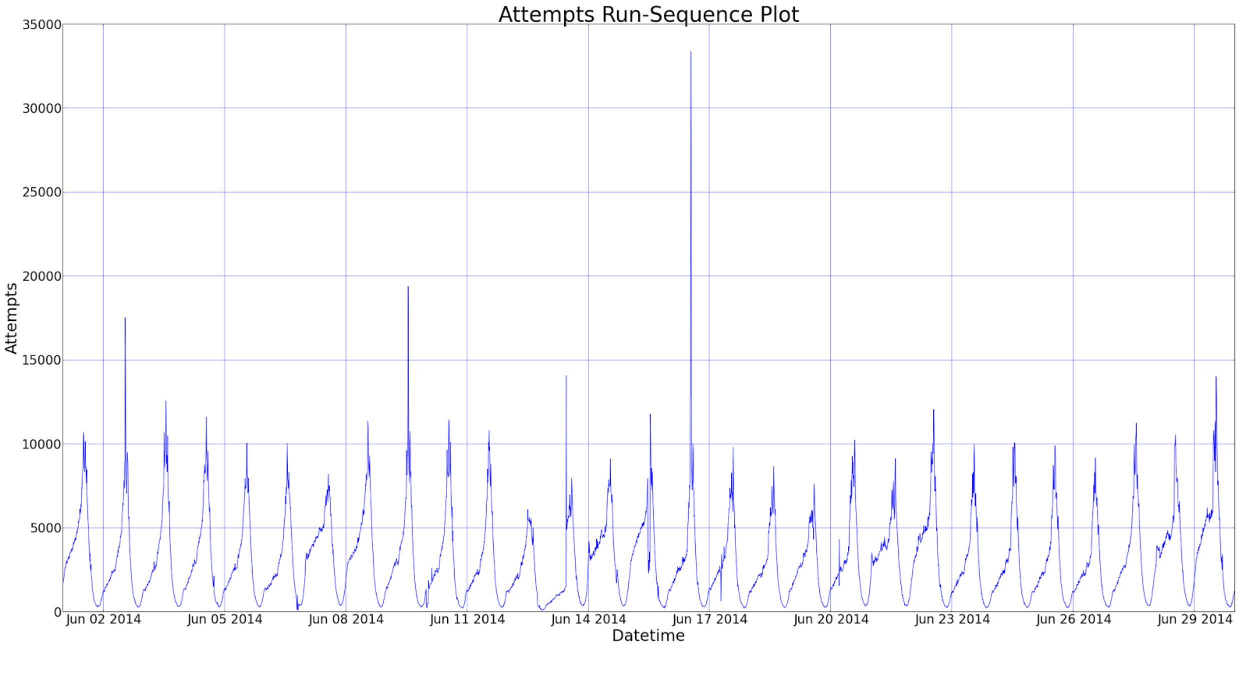

Figure 1: A run-sequence plot for the attempts time series

In contrast, VSF is a static metric. Figure 2 showcases the run-sequence plot for VSF. The average VSF rate throughout the time period is relatively stable, only spiking during anomalies. An anomaly occurs on June 28, 2014 with a VSF rate of 25%.

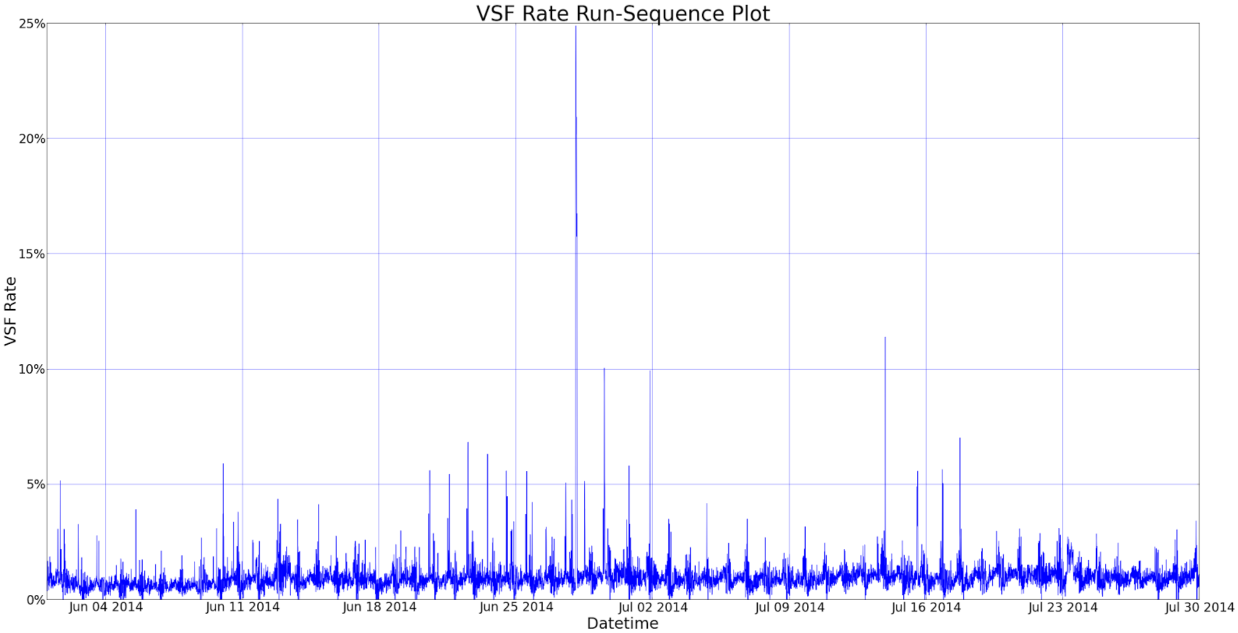

Figure 2: A run-sequence plot for the video start failure rate time series

We also plot the autocorrelation plot for the time series using the Python `statsmodel` package. Autocorrelation is the correlation between values in the time series at the current timestep with prior values. The following is the equation to calculate autocorrelation between a metric X at time t with its time lagged value t - p ("Engineering Statistics Handbook" section. 1.3.3.1).

$$r_p = \frac{Covariance(X_t, X_{t-p})}{Variance(X)} = \frac{\sum_{t=p+1}^{N} (X_t - \bar{X})(X_{t-p} - \bar{X})}{\sum_{t=1}^{N} (X_t - \bar{X})^2}$$

The covariance between $X_t$and $X_{t-p}$is a measure of how these two random variables change together ("Covariance"). Large positive covariance indicates that small and large values of X_t correspond with small and large values of X_{t-p}respectively. Dividing by the variance of the time series normalizes this "similarity" value from -1 to 1, commonly known as the correlation value. Larger absolute values of correlation indicate stronger linear dependence between the two random variables. We call this metric autocorrelation because we are measuring the correlation of a random variable with a time lagged transformation of that same random variable. The autocorrelation plot shows the autocorrelation value for various time lags. Figure 3 showcases the autocorrelation plot for the attempts time series. The blue area of the graph signifies the interval of values where autocorrelation is significantly zero. Autocorrelation values peak significantly with 24 hour periodicity. Furthermore, the daily peaks themselves decrease and then increase in a sinusoidal pattern with 7 day periodicity. This indicates that there is both daily and weekly seasonality in our data.

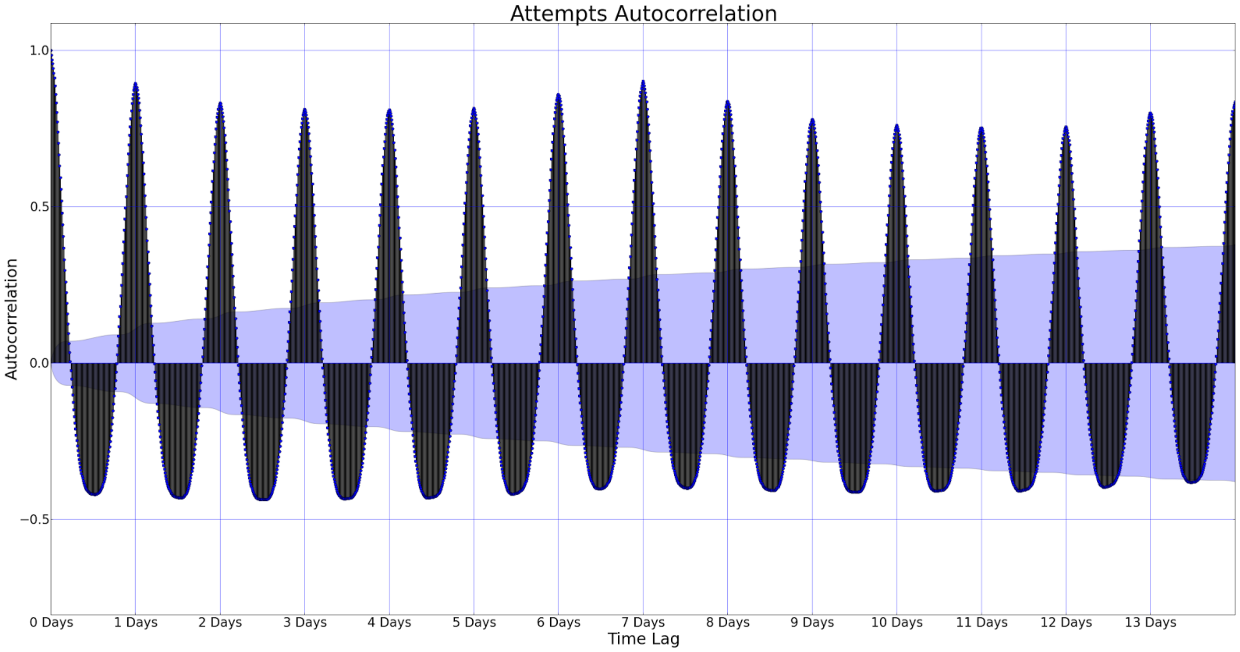

Figure 3: The autocorrelation plot for the attempts time series

Figure 4 showcases the autocorrelation plot for the VSF time series. VSF exhibits much poorer autocorrelation. For most time lags, the autocorrelation value is significantly zero. We conclude that there are very few, if any, regular patterns within the VSF time series.

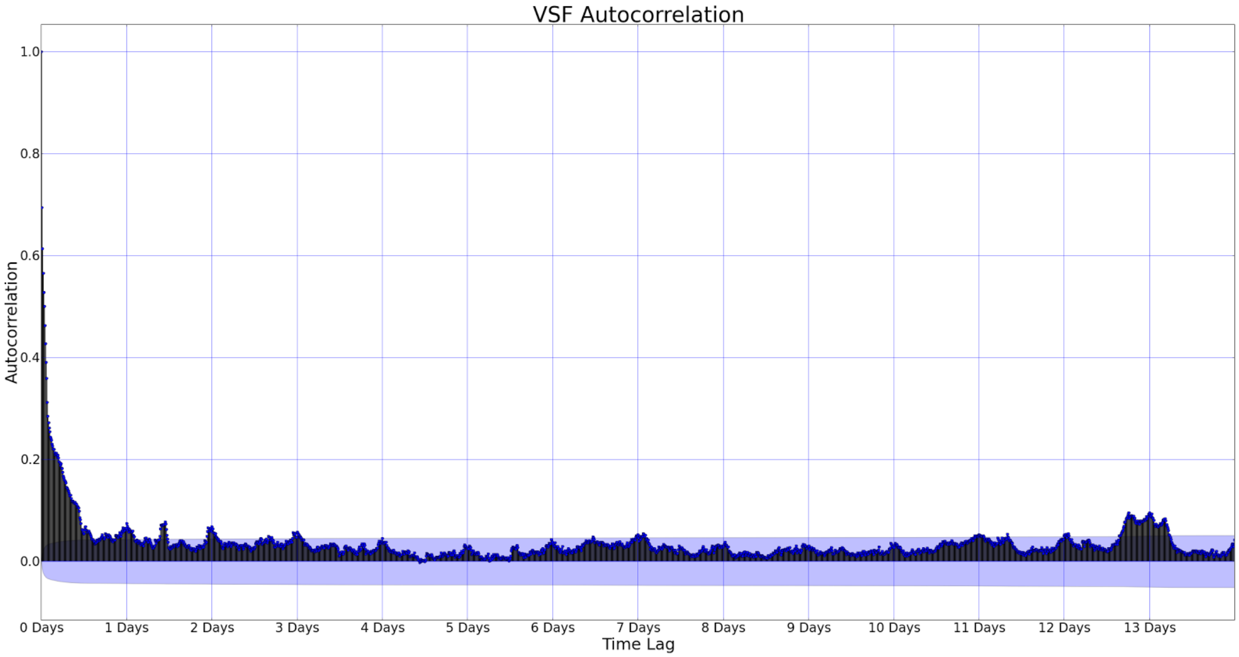

Figure 4: The autocorrelation plot for the video start failure rate time series

Due to the large autocorrelation values for the attempts time series, we predict that we can model this time series with an autoregressive function. Unfortunately, the same cannot be said for the VSF time series. For information on how we perform anomaly detection on the VSF time series, we direct the reader to Pierce Vollucci's and Wenxuan Cai's technical papers. In the remainder of this paper, we focus on detecting anomalies solely for the attempts time series with an autoregressive model and the Seasonal-ESD model.

4. Data Modeling and Results

Autoregressive (AR) Model

As previously stated, we believe that an autoregressive model will fit our time series quite well. Once the model is fitted to the time series, we can then use the model to predict the number of attempts at future timesteps. We can then compare the predicted value with the corresponding observed value. A large discrepancy between the two indicates high probability of an anomaly. The AR(p) model takes the following form

(“Engineering Statistics Handbook” section 6.4.4.4).

$$X_t = c + a_1 X_{t-1} + a_2 X_{t-2} + \dots a_p X_{t-p} + \varepsilon$$

This is essentially a linear regression of the current metric value on *p* prior values. These regression features are known as the AR terms of the model. If seasonality is observed within the data, additional seasonal AR terms can be added by regressing on values one seasonal period in the past. Once AR terms are chosen, least squares fitting techniques is used to estimate the coefficients *a* of the model.

Identifying AR terms: We identify the *p* order of the model by analyzing the partial autocorrelation plot of the time series. Partial autocorrelation between X_t and X_{t-p} is defined as the autocorrelation between the two that is not accounted for by autocorrelation in lags *1* to *p - 1* (“Engineering Statistics Handbook” section 6.4.4.6.3). The following is the equation to calculate the partial autocorrelation between $X_t$and X_{t-p} (“Partial Autocorrelation”).

$$Let\ Y_t = X_t - E(X_t \mid X_{t-1}, X_{t-2}, \dots, X_{t-p-1})$$

$$Y_{t-p} = X_{t-p} - E(X_{t-p} \mid X_{t-1}, X_{t-2}, \dots, X_{t-p-1})$$

$$partial\ r_p = \frac{Covariance(Y_t, Y_{t-p})}{\sqrt{Variance(Y_t)}\sqrt{Variance(Y_{t-p})}}$$

In this equation, $E(X_t \mid X_{t-1}, X_{t-2}, \dots, X_{t-p-1})$ is calculated by first performing linear regression on $X_t$over regressors $X_{t-1}, X_{t-2}, \dots, X_{t-p-1}$. After the best fit coefficients are determined for linear regression, $X_t$can be estimated given $X_{t-1}, X_{t-2}, \dots, X_{t-p-1}$. We calculate$E(X_{t-p} \mid X_{t-1}, X_{t-2}, \dots, X_{t-p-1})$ similarly. By subtracting these estimates from X_t and X_{t-p} respectively, we are essentially removing any effect of linear association between X_t,X_{t-p} and $X_{t-1}, X_{t-2}, \dots, X_{t-p-1}$. As a result, the resulting correlation value from the above equation can be thought of the correlation between X_t and $X_{t-p}$with the effect of correlation due to $X_{t-1}, X_{t-2}, \dots, X_{t-p-1}$ removed.

For example, the correlation between X_t and X_{t-2} due to the forward propagation of autocorrelation between X_t and X_{t-1} will be eliminated through the partial autocorrelation function. Thus, the partial autocorrelation plot gives insight on which time lags are most significant to the observed autocorrelation in the time series ("Engineering Statistics Handbook" section 6.4.4.6.3). Figure 5 showcases the partial autocorrelation plot for the number of attempts. This plot shows that significant partial autocorrelation only occurs for the first time lag and drops dramatically to 0 afterwards. Thus, we choose to model the data with AR(1).

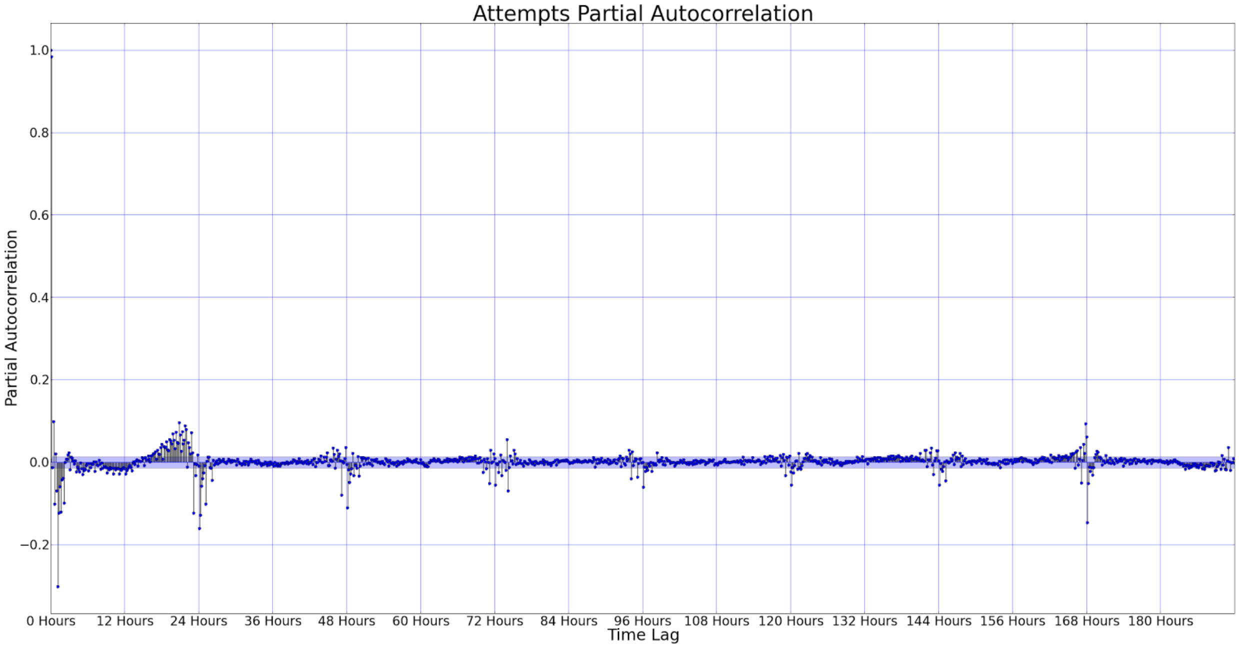

Figure 5: The partial autocorrelation plot for the attempt time series

However, seasonality terms in the regression are still needed. Recall that from the autocorrelation plot in Figure 3, daily and weekly seasonality were observed. Our PACF plot also indicates that there are several small spikes at the 1-day (24 hours) time lag and 1-week (168 hours) time lag where partial autocorrelation is significantly non-zero. Thus, we should add additional daily and weekly terms to regress on. We include one-day, two-day, three-day, one-week, two-week, three-week, and four-week AR terms in the regression. We decided upon three daily and four weekly seasonal terms as opposed to one each such that a prediction will not be significantly thrown off due to an anomaly that occurred one day or one week ago.

Model Estimation: The final chosen model for the attempts time series is described in the equation below. To remind the reader, 1 time lag is equivalent to 10 minutes ago for our use case.

$$\hat{X}_t = c + a_1 X_{t-1\ time\ lag} + a_2 X_{t-1\ day} + a_3 X_{t-2\ days} + a_4 X_{t-3\ days} + a_5 X_{t-1\ week} + a_6 X_{t-2\ weeks} + a_7 X_{t-3\ weeks} + a_8 X_{t-4\ weeks}$$

Next, we build a design matrix for training containing the feature vector of AR terms with the response values. We estimate the coefficients of the AR model with ordinary least squares fitting techniques from the Python `statsmodel` package ("Ordinary Least Squares").

Identifying Anomalies: Generally, we assume that the residuals resulting from the predicted responses of linear regression form an approximate normal distribution. We also assume that the residuals exhibit homoscedasticity (having the same variance throughout the time series). If these two assumptions hold for the residuals from the autoregressive model, z-score tests can be used to identify anomalies. The equation to calculate z-scores from residuals follows (Seo 10).

$$z = \frac{y - \hat{y}}{standard\ deviation}$$

By default, a point is classified as anomalous if the residual has an absolute z-score greater than 3.0. This corresponds to 99.7% confidence in classification. This threshold can be tuned to adjust for sensitivity as needed.

Unfortunately, the residuals do not exhibit normality or homoscedasticity. The Jarque-Bera test rejects the null hypothesis that the residuals are normal. The following is the test statistic calculated by the Jarque-Bera Test where S and K are the sample skew and sample kurtosis of the data respectively ("Jarque Bera Test").

$$S = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^3}{(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2})^3}$$

$$K = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^4}{(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2})^4}$$

$$JB = \frac{N}{6}(S^2 + \frac{(K-3)^2}{4})$$

The sample skew is a measurement of lack of symmetry in the data while the sample kurtosis is a measurement of how smooth the peak of the distribution is. For normal distribution, the skew and kurtosis value are 0 and 3 respectively. Higher values of skew and kurtosis result in a larger JB value and indicate a less likely chance that the data is normal. The Jarque-Bera test rejects the null hypothesis if JB exceeds some critical value dependent on the sample size and p-value threshold. For our residuals, a sample skew of .597 and sample kurtosis of 48.621 are calculated. This results in a JB value of 1399580.820 with a p-value of near zero. There is a near zero chance that our residuals come from a normal distribution.

To observe heteroskedasticity in the data, we analyze the run-sequence plot of the residuals shown in Figure 6. The red lines show the +/- 1 sigma lines calculated from the entire distribution of residuals. From this plot, we can see that there are patterns of higher variance during the middle of the day when a high number of attempts occurs than during the wee hours of the night. Because of the higher variance, larger confidence intervals need to be used during those time periods when determining whether a residual value is anomalous to account for the greater entropy.

To compensate for lack of normality in the residuals, we use a modified version of the z-score test. This test assumes that the lack of normality is due to the presence of anomalies in the observations and that non-anomalous data follows normal distribution.

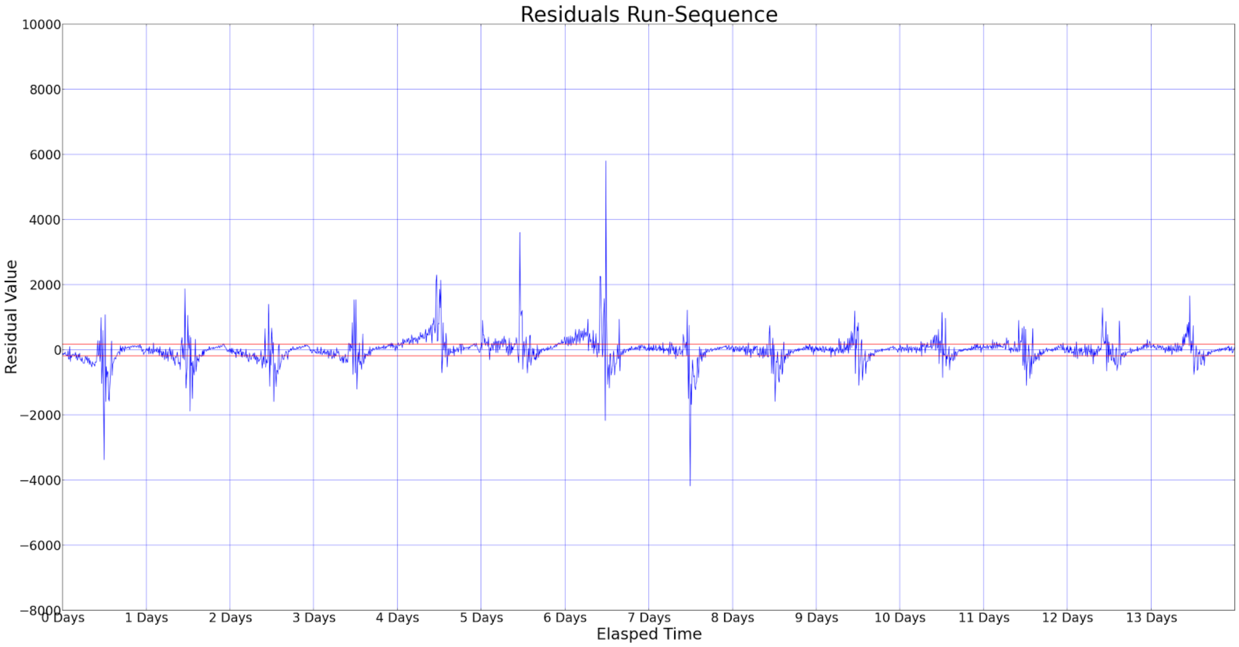

Figure 6: The run-sequence plot for the resulting residuals from the autoregressive model

Therefore, we should use robust statistical methods that are not affected by extreme values to estimate the variance of the normal distribution. In the modified z-score test, the sample standard deviation is replaced by MADe (Seo 12). The following is the equation to calculate MADe.

$$MADe = 1.4826 * Median\ Absolute\ Deviation$$

The magic 1.4826 constant is chosen because for a normal distribution, this causes MADe to be equivalent to the standard deviation. Unlike mean values, medians for parameter estimation are largely unaffected by outliers.

To account for heteroscedasticity, we need to model variance as a function of time. We do so by bucketing all residual values of a one hour time window together (e.g. Tuesday 11-12AM, Friday 10-11PM) and for robustness, calculate their sample MADe value . After this process, we have a variance value for every hour time window for every day of the week to use in the modified z-score test.

Putting everything together, the modified z-score test for anomalies follows.

$$Classify\ as\ anomaly\ if\ z = \frac{y_i - \hat{y}_i}{MADe_{hour\ of\ week\ of\ y_i}} > threshold$$

Autoregressive (AR) Model Results

To generate our predictions, we first train our model on the first day's worth of data and predict attempt values for the second day. We then incorporate the second day's worth of data in our training set and retrain the model before predicting attempt values for the third day. We repeat this process until we reach the last day for predictions. Overall, our AR model does very well in predicting the number of attempts that will occur at future time steps. The root mean squared error (standard error) of our predictions is 366.8, a very low value considering that the number of attempts in a 10 minute time window ranges in the thousands. Furthermore, the R-squared coefficient of determination of our linear regression is around .978. The formula for R^2 is shown below ("Coefficient of determination").

$$R^2 = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y})^2}$$

In the event of perfect regression, the numerator of the fraction will equal to 0 and R^2 equal to 1. Thus, values closer to 1 indicates better fit of the data. We can interpret from our R^2 value of .978 that our regression has accounted for 97.8% of the intrinsic variance in the data and that nearly perfect predictions were achieved. Our predictions pass the eye test as well, our predicted time series very closely matches the wave pattern in the actual time series. However, the goal of this model is not to predict attempts values, rather it is to classify anomalies from the resulting residuals of the regression.

We evaluate the results of our AR model by comparing our classifications with a gold standard set created by Pierce Vollucci. Approximately seven weeks of the attempts metric were manually labeled by observing the run-sequence plot during those weeks and flagging sudden peaks or drops as anomalies. Each data point is labeled with an integer from zero to three. A value of zero signifies no anomaly present while a value of three indicates a very severe anomaly. We consider a data point to be anomalous if it is labeled with a value two or above. 16 anomalies are present in this data set for an average of approximately 2 anomalies a week.

Figure 7 showcases a subset of our manually labeled anomalies over a two week period out of seven weeks. For brevity, we do not include figures of our anomalies for the other labeled weeks. For more information about the gold standard set, we direct the reader to Pierce Vollucci's technical paper.

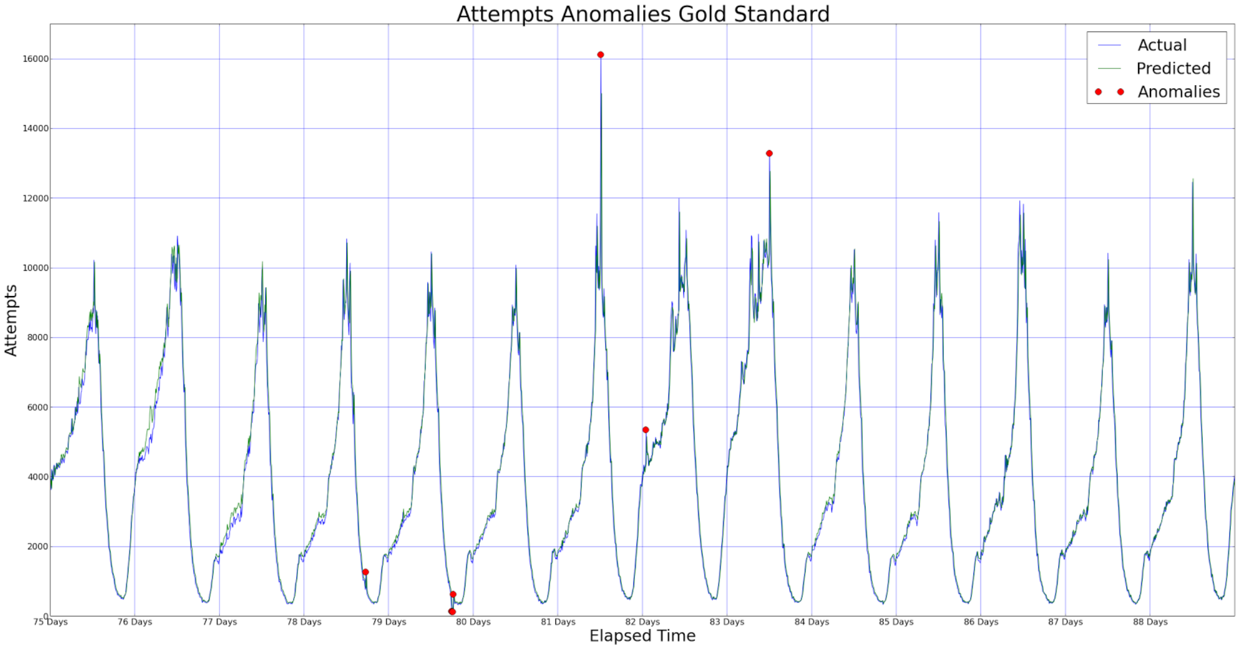

Figure 7: The manually labeled anomalies over a two week period

We first plot the receiver operating characteristic (ROC) curve of our classifier in Figure 8. The ROC plot is generated by calculating the true and false positive rates as the z-score threshold is adjusted with true positive rate on the y-axis and false positive rate on the x-axis (Receiver Operating Characteristic). The formulas to calculate the true and false positive rates are defined below.

$$true\ positive\ rate = \frac{\#\ true\ positive}{\#\ true\ positive + \#\ false\ negative}$$

$$false\ positive\ rate = \frac{\#\ false\ positive}{\#\ false\ positive + \#\ true\ negative}$$

When the z-score threshold is set to a very high value, no residuals will be classified as anomalies and thus, the true and false positive rates will both be 0%. When the z-score threshold is set to 0, all residuals will be classified as anomalies and consequentially, the true and false positive rates will both be 100%. For our classifier to be well performing, we wish for the true positive rate to grow faster than the false positive rate as the z-score threshold is tuned from some large value to 0. This indicates that our classifier is quite good at calculating low z-scores for non-anomalies and high z-scores for anomalies. To evaluate how well our classifier trades off true positive rate for false positive rate as the threshold is adjusted, we calculate the area under the curve (AUC) value. For a perfect classifier, the ROC curve will have a true positive rate of 100%, no matter what the corresponding false positive rate is. In this scenario, the ROC curve will be a horizontal line at $y = 1$ with an AUC of 1. AUC's closer to 1 indicate a better performing classifier.

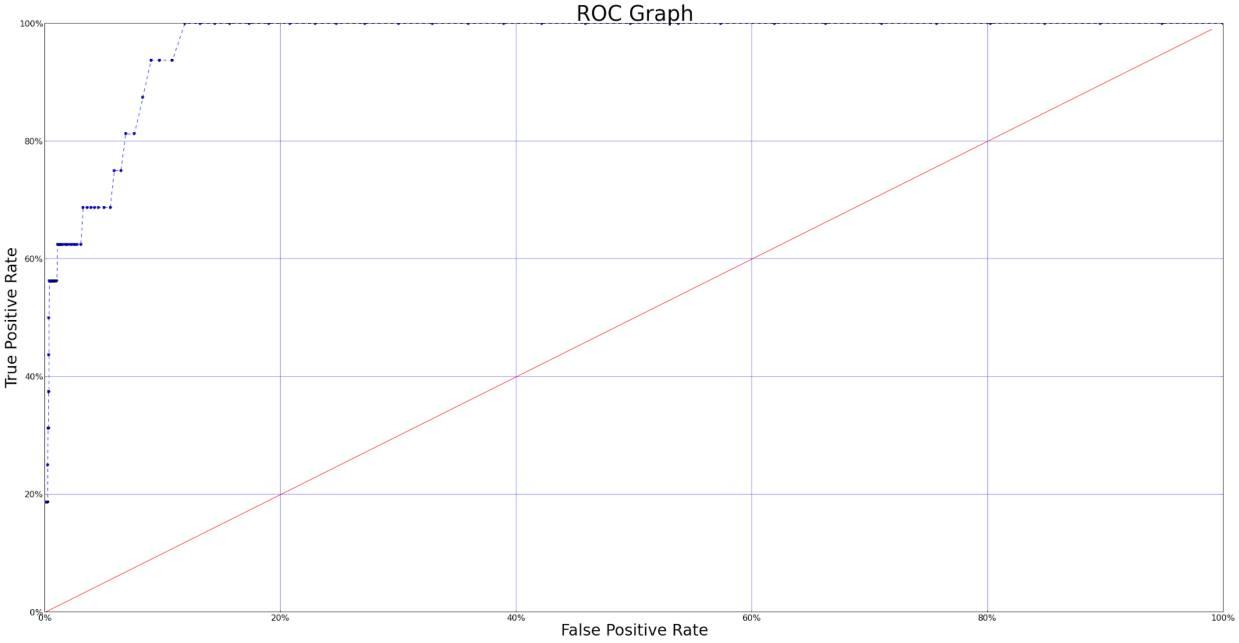

Figure 8: ROC Curve of the AR Model

For a random classifier that randomly assigns z-scores to points, the true and false positive rates will grow linearly as the threshold is adjusted. This is because the number of true positives and number of false positives also grows linearly as threshold is adjusted due to a larger range of z-scores being classified as anomalies. The red line in Figure 8 is the ROC curve of a random classifier. This curve has an AUC of 0.5. Thus, any relatively better performing classifier than random should maintain an ROC curve above the red line with an AUC above 0.5. The ROC curve of our AR model indicates that our classifications are significantly better than random. Our AUC value of .971 further supports this notion.

In addition to the ROC Curve, we score our classifications using the F-measure score with beta equal to 1. The F-measure function gives us a way to combine precision and recall into a single scoring function that we can then more easily maximize for. F-measure is equivalent to the weighted harmonic mean between precision and recall. We tune our z-score threshold value until F-measure is maximized, giving us the threshold that results in the most optimal trade off between precision and recall. The following is the formula for F-measure ("F1 Score").

$$precision \ = \ \frac{\#\ true\ positive}{\#\ true\ positive + \#\ true\ negative}$$

$$recall = true\ positive\ rate = \frac{\#\ true\ positive}{\#\ true\ positive + \#\ false\ negative}$$

$$F(\beta) = (1 + \beta^2) * \frac{precision * recall}{\beta^2 * precision + recall}$$

Using larger values of beta puts more emphasis on optimizing recall over precision and vice versa. Setting beta equal to 1 indicates that precision and recall should be weighed equally. One can adjust beta, depending how much they consider recall to be more important than precision. If tuning for high recall, a human only needs to investigate a subset of flagged anomalies which contains most of the true anomalous set. This is excellent as a substitute for human monitoring where a human needs to investigate every point and decide if it is anomalous. We can filter out most of the points for a human while keeping most of the anomalies to allow for greater efficiency. If tuning for high precision, a human is alerted when anomalies are almost certain. This is ideal if we do not want to frequently disturb a human from other responsibilities, only to find out that a flagged anomaly was a false alarm. This is ideal if manpower to remedy the anomaly is limited because time is only spent to investigate the most anomalous of data points. We tune our z-score threshold to search for a threshold that achieves the highest F-measure score. Figure 9 showcases the F-measure score for various threshold values.

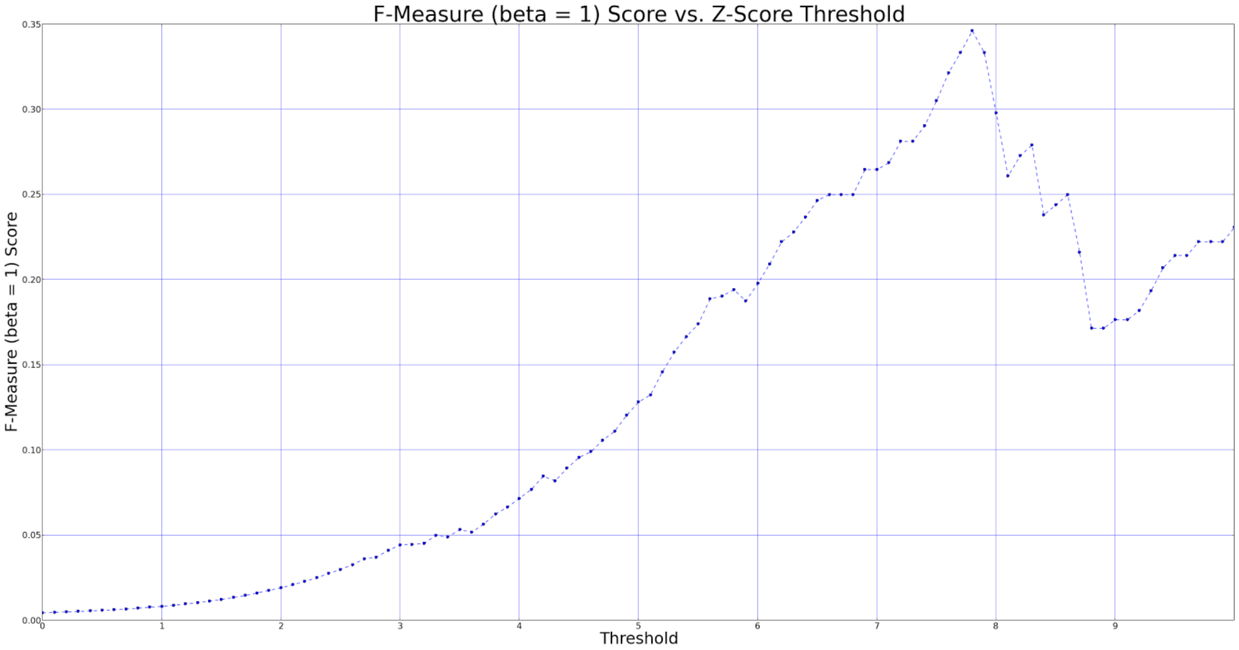

Figure 9: F-Measure values for various z-score threshold values

The highest F-Measure score achieved is .346 with a threshold of 7.8. This threshold results in a precision score of 25% and recall score of 56.25%. For the reader’s interest, we show the resulting confusion matrix for threshold 7.8.

	Predicted Positive	Predicted Negative
Actual Positive	9	7
Actual Negative	27	7163

We plot the flagged anomalies when using a z-threshold value of 7.8 for our AR model in Figure 10 over the same two week period as in Figure 7. From a simple eye test, the flagged anomalies are quite reasonable. Large daily peaks are flagged as anomalous along with sudden drops or rises in the middle of the day.

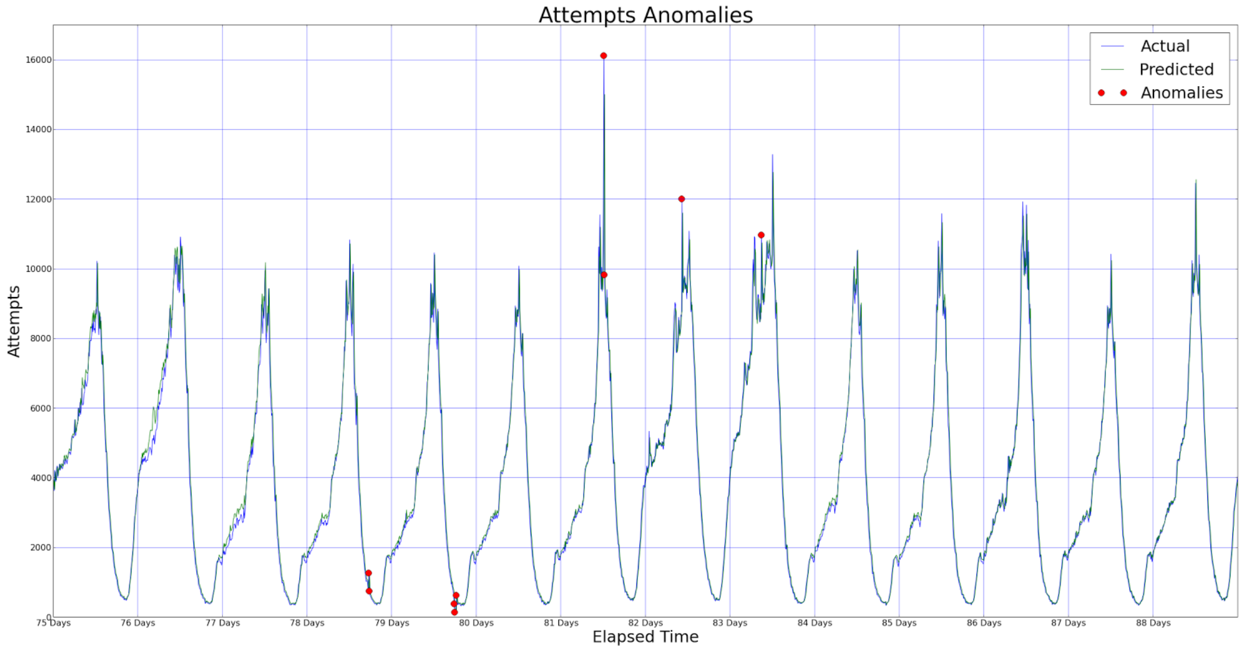

Figure 10: The flagged anomalies (red dots) in our attempts time series found using our AR model w/ threshold 7.8

For comparison, we plot the flagged anomalies when using a z-threshold value of 3.0 for our AR Model in Figure 11. 3-sigma is commonly considered the default threshold to use for outlier tests using z-scores. This threshold results in a precision score of 2.27% and recall score of 93.75%. Although the high recall score is quite promising, the precision score is abysmal. The low precision score is due to the disproportionate number of anomalies that are flagged during the less active hours of the day, most of them false positives. Preliminary investigation shows that the large number of anomalies is due to low variation of residuals during that period of the day because our model is very good at predicting the number of attempts for those hours. More investigation is still needed however to explain the cause of this characteristic and determine if we need to mitigate this behavior by reevaluating how we model the heteroskedastic variance.

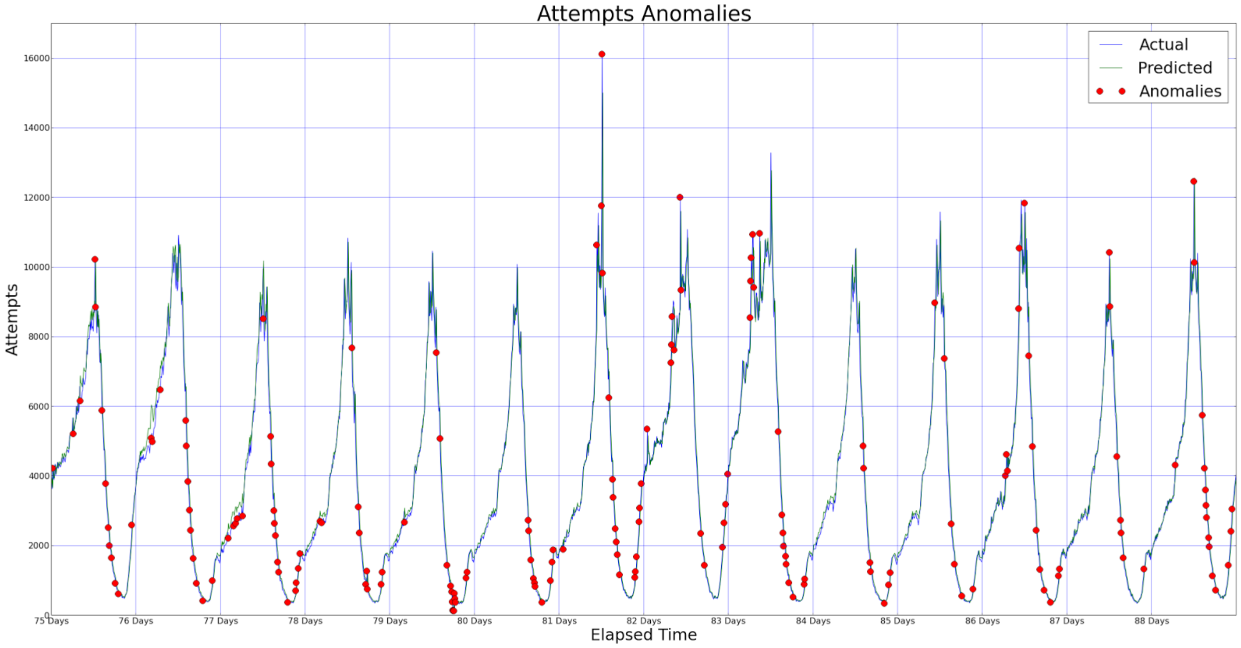

Figure 11: The flagged anomalies (red dots) in our attempts time series found using our AR model w/ threshold 3.0

Overall, although our model is very good at predicting the number of attempts that will occur in the future, translating these predictions into classifications for anomalies is still an open research question. While our classifier is definitely performing better than random, significant improvements need to be done to increase precision and recall. For example, if we are able to improve how we model the heteroskedastic variance in the data in future work, we can possibly mitigate the number of false positives that get flagged during the less active hours of the day for the 3-sigma case. This will allow us to maintain the high recall rate of using a lower threshold while increasing precision significantly.

Seasonal Hybrid Extreme Student Deviate

This algorithm developed by Vallis et al. models the time series as a summation of three components: seasonal, trend, and residual (1). The time series is divided into non-overlapping time windows of at least two weeks and each time window is independently decomposed into the three components (Vallis et al. 3). Below, we provide brief descriptions of how seasonal and trend components are computed and how anomalies are detected from the resulting residuals. For more detail about this algorithm, we direct the reader to Vallis et al.'s paper, "A Novel Technique for Long-Term Anomaly Detection in the Cloud".

Seasonal Decomposition: The Seasonal and Trend decomposition using Loess (STL) algorithm is used to determine the seasonal component of each time window in the time series. STL first takes subsets of values for each seasonal cycle (e.g. all the Monday 2-3PM values and all the Wednesday 10-11AM values) and generates subseries cycles out of them (Cleveland 5). For each cycle, locally weighted regression (loess), is used to fit a smooth regression function to the data (Cleveland 5). The smooth regression function is generated by first fitting a linear function from $t - \Delta$ to $t + \Delta$ on our data, weighting the points with x values closer to t more heavily during fitting (Cleveland 3-4). If Δ is small enough, the fitted linear function will very closely approximate the actual smooth regression function from $t - \Delta$ to $t + \Delta$. We can then piecewise combine linear functions that are each fitted to non-overlapping $t - \Delta$ to $t + \Delta$ time windows to generate the smooth regression function. To calculate weights, commonly the tricube weight function is used.

$$W(u) = max(0,\ (1 - u^3)^3\ for\ u\ \geq\ 0$$

To calculate the weight of a specific point $(t_i,\ y_i)$ relative to t, the following equation is used.

$$w\,((t_i,\ y_i),\ t)\ =\ W(\frac{|t_i - t|}{f(t,\ q)})$$

$$f(t,\ q)\ = |t\ - farthest\ t_i|\ \frac{q}{\#\ data\ points}$$

Parameter q in the above equation is known as the smoothing parameter. As q goes towards infinity, $f(t,\ q)$ will grow towards infinity and all points will have weight 1 towards linear fitting. As q goes towards 0, only points close to t will have any significant weight and consequentially, the smooth function will fit more aggressively with the data.

Once smooth functions are fitted to each seasonal cycle, they are combined together to form the seasonal component. In our capstone project, We use Python's `robjects` package to call R's `stl` function for seasonal decomposition. We chose to use four-week time windows for robustness in calculating the seasonal component. By increasing the time window from two to four weeks, we have twice as many data points to estimate the weekly seasonality component with loess. As a result, the impact of any single anomalous metric is reduced when estimating seasonality because of the increased sample size.

Trend Decomposition: Trend is calculated using the piecewise median method. The median of each time window is calculated and combined together to generate a piecewise trend function (Vallis et al. 3). Because metrics do not change much over small time windows, approximating the trend of a time window with a constant is appropriate (Vallis et .al 3). Vallis et al. chose the median method over using STL or Quantile Regression for trend decomposition because the latter two would often overfit to anomalies in the input (2). In contrast, medians are largely unaffected by outliers in the data. In practice on Twitter's production data, piecewise median proved to be an effective estimator of trend for the purpose of anomaly detection (Vallis et al. 5).

Anomaly Detection: After seasonal and trend decomposition, the resulting residuals are calculated by the following formula.

$$For\ time\ window\ X,\ Residuals\ =\ X\ - Seasonal(X)\ -\ Median(X)$$

A modified Generalized Extreme Student Deviate (ESD) test identifies the anomalies within these residuals. Generalized ESD returns up to *r* outliers with significance level *α*. *r* and *α* are user defined parameters. *r* is the maximum number of outliers we would like to return and *α* is the significance level that tunes our sensitivity to outliers. The following test statistic is calculated by Generalized ESD ("Engineering Statistics Handbook" section 1.3.5.17.3).

$$R_i = \frac{max_i|x_i - \bar{x}|}{standard\ deviation}$$

Unfortunately, Generalized ESD is only statistically sound if the residuals are approximately normal ("Engineering Statistics Handbook" section 1.3.5.17.3). Much like the autoregressive residuals, anomalies cause the distribution of residuals to skew slightly towards the extremes. Similarly, to remedy this situation, Vallis et al. replace the sample mean and sample standard deviation with the sample median and MADe respectively (3).

$$R_{robust_i} = \frac{max_i|x_i - median(x)|}{MADe}$$

Because we wish to return up to *r* outliers, *r* R_{robust_i} test statistics are computed from $i = 0,\ 1,\ ...,\ r-1$. During each iteration, R_{robust_i} is calculated using the data point x_i that maximizes $|x_i\ -\ median(x)|$. After calculating R_{robust_i}, data point x_i is removed from the dataset before calculating $R_{robust_{i+1}}$, the test statistic for the next iteration. Thus, each successive R_{robust_i}value is calculated over $n-i$ datapoints. *r* critical values are also computed corresponding with the *r* test statistics. The following equation below is used to calculate the critical values from $i = 0,\ 1,\ ...,\ r-1$ ("Engineering Statistics Handbook" section 1.3.5.17.3). $t_{P,\,N}$ is the value at the *P* percentile point in the cumulative distribution function of a t-distribution with *N* degrees of freedom.

$$\lambda_i = \frac{(n-i-1)t_{p,n-i-2}}{\sqrt{n-i}\sqrt{n-i-2+t^2_{p,n-i-2}}}$$

$$p = 1 - \frac{\alpha}{2(n-i)}$$

Once $R_{robust_i}$and its corresponding λ_i values are calculated for $i = 0,\ 1,\ ...,\ r-1$, we then find the largest *i* such that $R_{robust_i} > \lambda_i$. Let the largest *i* such that $R_{robust_i} > \lambda_i$ be equal to *q*. We then return $q+1$ outliers where each outlier is the corresponding data point x_i used to maximize $|x_i\ -\ median(x)|$ when calculating R_{robust_i} from $i = 0,\ 1,\ ...,\ q$.

Because our time window is four weeks long, we set *r* equal to 672 (24 hours * 28 days) to alert one anomaly per hour on average. We believe that any higher frequency of anomaly alerts will overwhelm the engineers responsible for resolving them. By default, the significance level *α* is set to 0.003. *α* may be adjusted to tune for sensitivity if needed.

Seasonal Hybrid Extreme Student Deviate Results

Performing the Seasonal ESD method on the attempts time series resulted in the following seasonal and trend decompositions showcased in Figure 12 over the same two week period as in Figure 7.

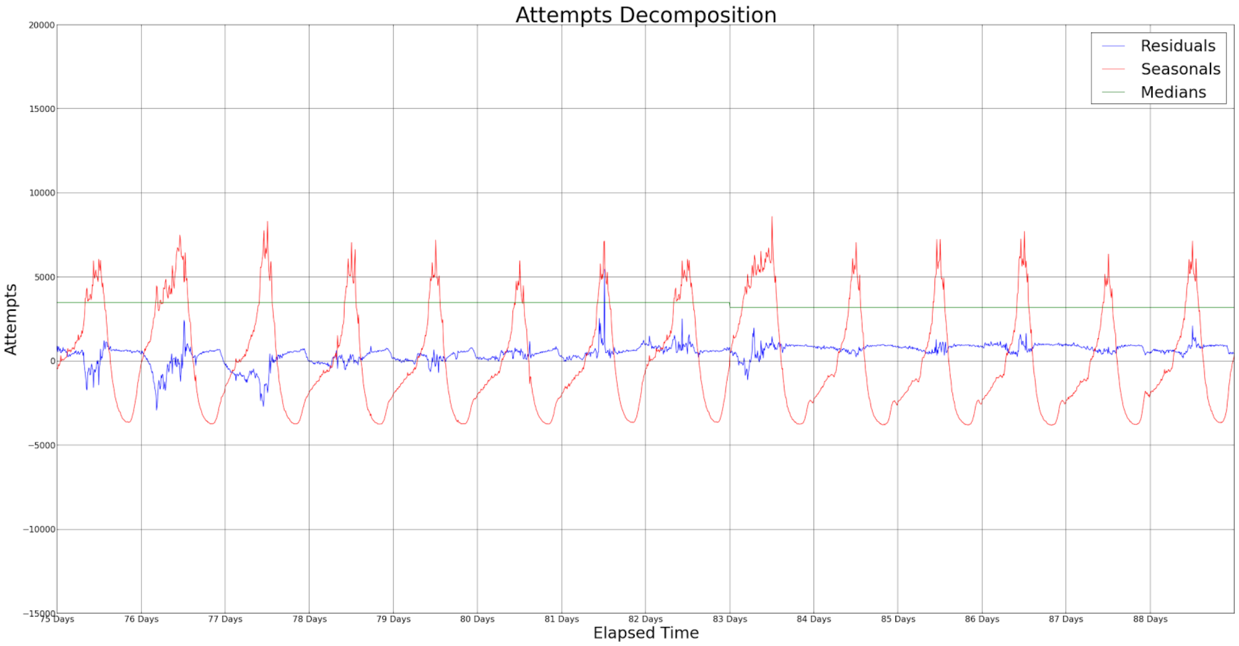

Figure 12: The decomposed attempts time series consisting of the residual (blue), seasonal (red), and trend (green) components

Similar to our evaluation of our AR Model, I first plot the ROC curve of our Seasonal ESD model in Figure 13. Due to how the λ_i critical values are calculated, we are unable to hit 100% true and false positive rates even when setting the significance level at *α* = 1. *α* needs to be set to a value beyond 1 to push the critical values low enough to classify everything as an anomaly, though significance levels above 100% do not entirely make statistical sense.

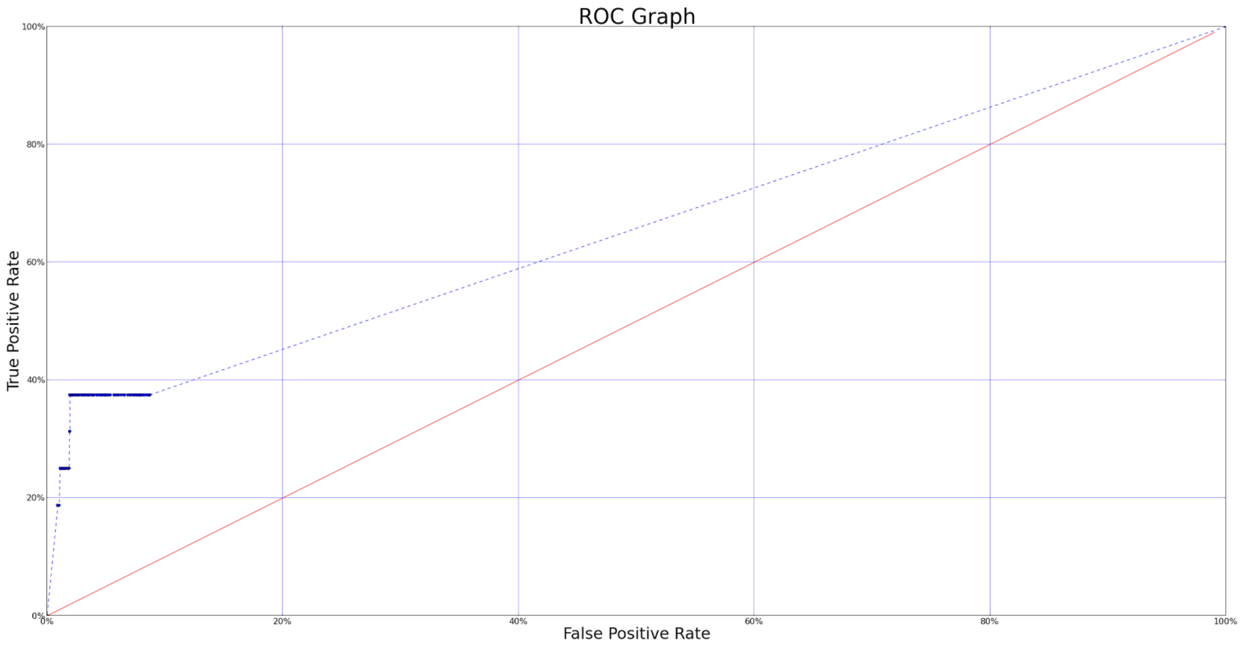

Figure 13: ROC Curve of the Seasonal ESD Model

For the areas where we have points to plot the ROC curve of the Seasonal ESD model as we adjust *α*, the curve is consistently above the red “random classifier” line. The AUC value of .656 indicates that its classifications are better than random, though not by much. Ultimately however, our inability to adjust the threshold for higher true and false positive rates gives us too much incomplete data to make any significant conclusions when analyzing the ROC plot.

We also plot the F-measure score (beta = 1) for various *α* values for Seasonal ESD in Figure 14.

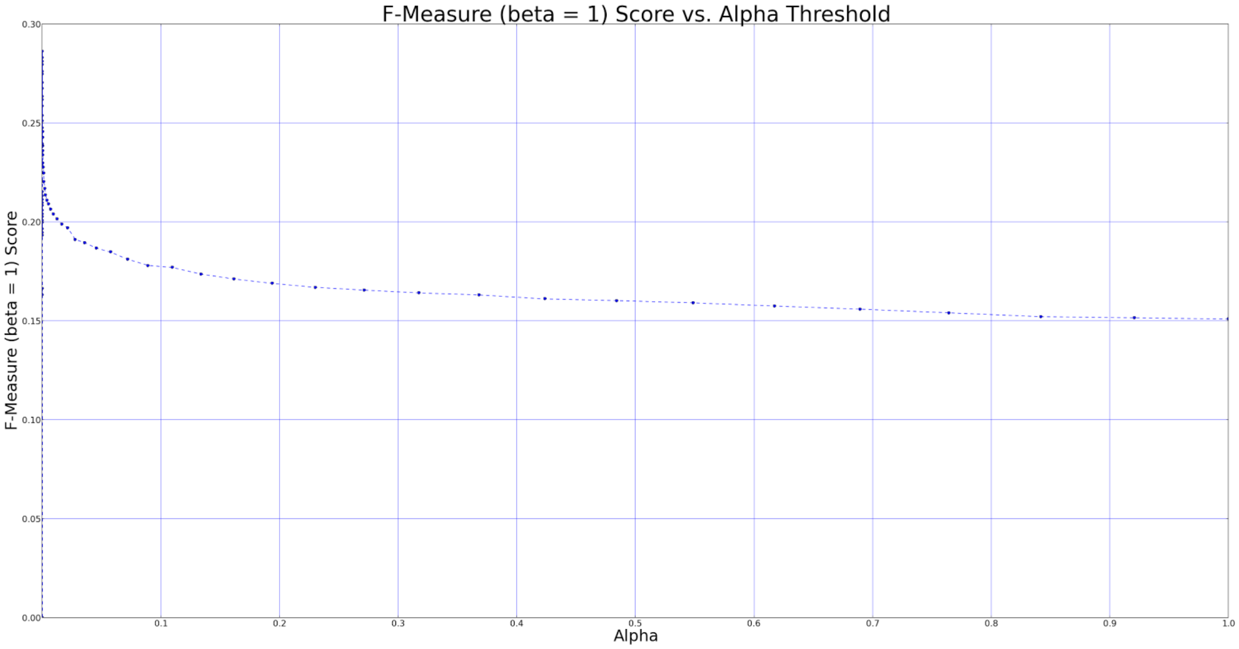

Figure 14: F-Measure values for various α threshold values

The highest F-Measure score achieved is .286 with an *α* value of 6.66e-08. This threshold results in a precision score of 4.14% and recall score of 37.5%. These F-Measure, precision, and recall scores are significantly worse than scores from our AR model. The significance level *α* is also set to a suspiciously low value indicating that only the most anomalous points should be flagged, though our precision score doesn't reflect this. The confusion matrix for the model with *α* value of 6.66e-08 is shown in the table below.

	Predicted Positive	Predicted Negative
Actual Positive	6	10
Actual Negative	139	7051

We plot all the flagged anomalies when using an α value of 6.66e-08 for our Seasonal ESD model in Figure 15. The Seasonal ESD model does not suffer from the same problem as our AR model in having an disproportionate amount of anomalies during the less active hours of the day. However, Seasonal ESD seems to be more sensitive in detecting anomalies in areas where the slopes are slightly less smooth than normal than for sudden peaks and drops. For example, at around day 76, the slight plateau causes the slope to look different from other days which cause anomalies to be flagged.

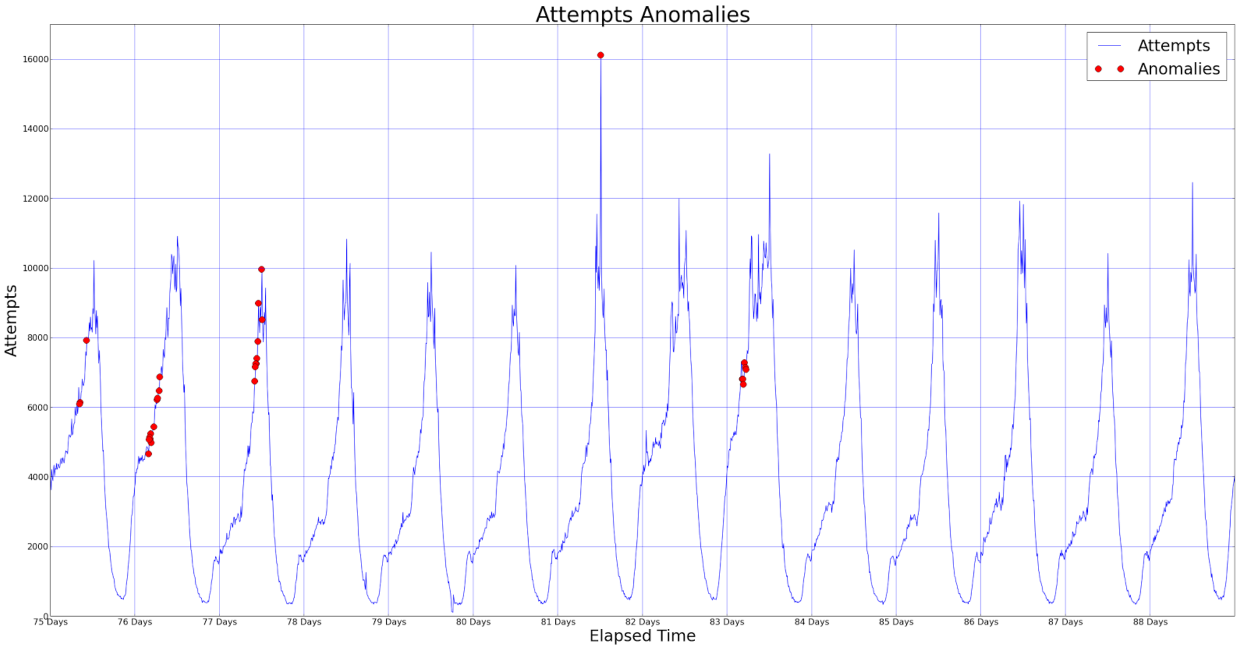

Figure 15: The flagged anomalies (red dots) in our attempts time series found using the Seasonal ESD model w/ α 6.66e-08

Overall, the Seasonal ESD model performs much worse than our AR model when validating with our gold standard set. However, while autoregression cannot be easily extended to time series that do not exhibit autocorrelation and seasonality, Seasonal ESD can. For metrics such as VSF, we can apply the same Seasonal ESD method but with the seasonality trend set to 0. This results in applying Generalized ESD on residuals generated by subtracting the piecewise median trend from the time series, a method similar to spike detection using moving averages. For performance though, our current AR Model is clearly superior than Seasonal ESD, even without any additional modifications to the AR Model that we can do in future work to improve precision and recall.

VIII. Conclusion

Future Work

Additional Validation: Currently, validation is done by comparing our classifications to a single-human labeled gold standard set. Although manual labeling has allowed us to automate our validation, a human unfortunately may not be able to detect anomalies that machines can and can introduce error in their labeling. Although error can be reduced by using an ensemble of labelers to generate the gold standard set, it will still be difficult to determine if a machine has detected an anomaly that a human will have surely missed. Ideally, we would synchronize with Conviva and their operations engineers to rigorously determine which data points are true anomalies or not. Unfortunately, there is a lack of manpower on both sides to perform the rigorous research needed to generate such a set.

Comparison with HTM algorithms: For some data scientists. deep learning defines the new frontier of machine learning algorithms (Metz). Deep learning methods such as neural networks are now applied to solve even the most difficult machine learning problems, including image and speech recognition. With that in mind, we wish to pit our algorithms against the deep learning methods by Numenta. We can use either Grok or a custom implementation built upon Numenta's open sourced libraries for comparison.

Additional Datasets: As described previously, our dataset consists of five months worth of session summaries from one Conviva customer. To further validate the performance of our anomaly detectors, we wish to apply our algorithms to additional datasets from Conviva's other customers.

Additional Metrics: In combination, the sub team has only researched classifying anomalies in the attempts and VSF time series. We can continue to validate the performance of our algorithms by applying the same algorithms to additional service quality metrics such as the rebuffering ratio and the video startup time.

Prototype: At the moment, we only have IPython Notebooks demonstrating the viability of the anomaly detection algorithms. Three types of notebooks are included in our codebase, data loading notebooks to create Hive tables from session summary data located in Amazon S3, library notebooks containing common functions used in our analysis, and sandbox notebooks where we perform the bulk of our anomaly classification and analysis of results. We divide our analysis into four separate notebooks, one for each of the anomaly detection algorithms we codified: autoregression/Seasonal ESD for the attempts metric and MADe/Weibull for the VSF metric. However, no progress has been made towards a prototype that detects and alerts anomalies in real-time with any ensemble of our algorithms. Collaboration between our capstone team and Conviva is required to obtain permission in developing a prototype on Conviva's code stack.

Project Reflection

For Smart Anomaly Detection, we attempted to build a model that could accurately classify anomalous points within the data. Through my contributions, an anomaly detector based upon predicting responses using an autoregressive function proved to be quite effective at classifying anomalies. Furthermore, I developed a Python implementation of the Seasonal ESD algorithm which can be included in our ensemble of anomaly detection algorithms if the team decides to do so. Although we achieved reasonable success with our Smart Anomaly Detection product, we have not been able to achieve the lofty goals we set for ourselves initially. We envisioned working closely with Conviva to build a fully fledged prototype that monitored real-time data streams at scale. Such a project would allow us to demonstrate our engineering and design prowess. Unfortunately, this goal proved to be unreasonable given the amount of man-hours reserved for this capstone project. The team was regulated to researching the viability of anomaly detection algorithms on small datasets by Conviva standards. However, Conviva can still benefit from our research into anomaly detection algorithms that they can then implement themselves.

IX. Acknowledgements

We would like to thank George Necula for advising us throughout the entirety of this Capstone project. We would also like to thank Jibin Zhan from Conviva for introducing the problem space to us and Pat McDonough from Databricks for providing extensive technical support throughout our use of Databricks.

References

Alice Corporation v. CLS Bank. 573 U.S. Supreme Court. 2014. Print.

Associated Press. “Netflix reeling from customer losses, site outage.” *MSNBC*. MSNBC. 24 July 2007. Web. 15 Feb. 2015.

Bessen, James. “The patent troll crisis is really a software patent crisis.” *Washington Post*. The Washington Post. 3 Sept. 2013. Web. 27 Feb. 2015.

Biem, Alain E. “Detecting Anomalies in Real-time in Multiple Time Series Data with Automated Thresholding.” International Business Machines Corporation. US Patent 8,924,333. 30 Dec. 2014.

"Bringing Big Data to the Enterprise." IBM. N.p., n.d. Web. 13 Apr. 2015.

Brundage, Michael L., and Brent Robert Mills. “Detecting Anomalies in Time Series Data”. Amazon Technologies, Inc., assignee. U.S. Patent 8,949,677. 3 Feb. 2015.

Byrd, Owen, and Brian Howard. 2013 Patent Litigation Year in Review. Rep. Menlo Park: Lex Machina, 2014. Print.

CA Inc. “Manage Your Network Infrastructure for Optimal Application Performance.” *CA Technologies*. n.p. n.d. 13 Feb. 2015.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM Computing Surveys (CSUR)* 41.3 (2009): 15.

"Coefficient of Determination." Wikipedia. Wikimedia Foundation, n.d. Web. 23 Apr. 2015.

Cohn, Chuck. “Build vs. Buy: How to Know When You Should Build Custom Software Over Canned Solutions.” *Forbes*. Forbes Magazine, 15 Sep. 2014. Web. 7 Apr. 2015.

Connelly, J.P., L.V. Lita, M. Bigby, and C. Yang. “Real time audience forecasting.” US Patent App. 20120047005. 23 Feb. 2012.

Conviva. "About Us." *Conviva*. n.p., n.d. Web. 28 Feb. 2015.

Cortes, Corinna, Lawrence D. Jackel, and Wan-Ping Chiang. "Limits on learning machine accuracy imposed by data quality." *KDD*. Vol. 95. 1995.

"Covariance." Wikipedia. Wikimedia Foundation, n.d. Web. 21 Apr. 2015.

Cleveland, Robert B., et al. "STL: A seasonal-trend decomposition procedure based on loess." Journal of Official Statistics 6.1 (1990): 3-73.

Dasgupta, Dipankar, and Stephanie Forrest. "Novelty detection in time series data using ideas from immunology." *Proceedings of the international conference on intelligent systems*. 1996.

Deshpande, Amit and Riehle, Dirk. "The total growth of open source." *Open Source Development, Communities and Quality*. Springer US, 2008. 197-209.

"Engineering Statistics Handbook." NIST/SEMATECH E-Handbook of Statistical Methods. NIST, n.d. Web. 14 Mar. 2015.

Etherington, Darrell. "Twitter Acquires Over 900 IBM Patents Following Infringement Claim, Enters Cross-Licensing Agreement." TechCrunch. N.p., 31 Jan. 2014. Web. 25 Feb. 2015.

"F1 Score." Wikipedia. Wikimedia Foundation, n.d. Web. 13 Apr. 2015.

Fisher, William W. “Patent.” *Encyclopaedia Britannica Online*. Encyclopaedia Britannica Inc.

Ganjam, Aditya, et al. "Impact of delivery eco-system variability and diversity on internet video quality." IET Journals 4 (2012): 36-42.

Goldman, Eric. "The Problems With Software Patents (Part 1 of 3)." *Forbes*. Forbes Magazine, 28 Nov. 2012. Web. 01 Mar. 2015.

Gottfriend, Miriam. “Bullish Investors See New Hope for Netflix Profit Stream.” *The Wall Street Journal*. The Wall Street Journal. n.d. Web 14 Feb. 2015.

Hanley Frank, Blair. "Amazon Web Services Dominates Cloud Survey, but Microsoft Azure Gains Traction - GeekWire." *GeekWire*. Geekwire, 18 Feb. 2015. Web. 02 Mar. 2015.

Harvey, Cynthia. "100 Open Source Apps To Replace Everyday Software." *Datamation*. N.p., 21 Jan. 2014. Web. 28 Feb. 2015.

Iyengar, Vijay S. 2002. “Transforming data to satisfy privacy constraints.” *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '02). ACM, New York, NY, USA, 279-288. Web. 12 Feb. 2015.

"Jarque Bera Test." Jarque Bera Test. NIST, n.d. Web. 15 Mar. 2015.

Jasani, Hiral. “Global Online Video Analytics Market.” *Frost & Sullivan*. n.p. 5 Dec. 2014. Web. 12 Feb. 2015.

Kahn, Sarah. "Business Analytics & Enterprise Software Publishing in the US." IBISWorld (2014): 5. Web. 11 Feb. 2015.

Kandel, Sean, et al. "Enterprise data analysis and visualization: An interview study." Visualization and Computer Graphics, IEEE Transactions on 18.12 (2012): 2917-2926.

Keaveney, Susan M. "Customer switching behavior in service industries: An exploratory study." The Journal of Marketing (1995): 71-82.

Kejariwal, Arun. "Introducing Practical and Robust Anomaly Detection in a Time Series." Twitter Engineering Blog. Web. 15 Feb. 2015.

Lawler, Richard. "Netflix Tops 40 Million Customers Total, More Paid US Subscribers than HBO." *Engadget*. N.p., 21 Oct. 2013. Web. 15 Feb. 2015.

Liu, Xi, et al. "A case for a coordinated internet video control plane." Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication. ACM, 2012.

Mcgovern, Gale. Virgin Mobile USA: Pricing for the Very First Time. Case Study. Boston. Harvard Business Publishing, 2003. Print. 9 Jan. 2010.

Metz, Cade. "Facebook's 'Deep Learning' Guru Reveals the Future of AI." Wired.com. Conde Nast Digital, n.d. Web. 16 Mar. 2015.

“Number of Broadband Connections.” *IBISWorld*. IBISWorld. 3. Web. 12 Feb. 2015.

Numenta. “The Science of Anomaly Detection.” *Numenta*. n.p. n.d. 13 Feb. 2015.

Numenta. “Hierarchical Temporal Memory.” *Numenta*. n.p. n.d. 13 Feb. 2015.

Open Source Initiative. "Welcome to The Open Source Initiative." *Open Source Initiative*. N.p., n.d. Web. 28 Feb. 2015.

"Ordinary Least Squares." Ordinary Least Squares. N.p., n.d. Web. 14 Mar. 2015.

"Partial Autocorrelation." Partial Autocorrelation. N.p., n.d. Web. 23 Apr. 2015.

Porter, Michael. "The Five Competitive Forces That Shape Strategy." *Harvard Business Review Case Studies, Articles, Books*. N.p., Jan. 2008. Web. 12 Feb. 2015.

Porter, Michael. "What is Strategy?." *Harvard Business Review Case Studies, Articles, Books*. N.p., Jan. 2008. Web. 12 Feb. 2015.

Quinn, Gene. “A Software Patent Setback: Alice v. CLS Bank.” *IP Watch Dog*. n.p. 9 Jan. 2015. Web. 27 Feb. 2015.

"Receiver Operating Characteristic." Wikipedia. Wikimedia Foundation, n.d. Web. 13 Apr. 2015.

Roettgers, Janko. "Netflix Spends $150 Million on Content Recommendations Every Year." *Gigaom*. N.p., 09 Oct. 2014. Web. 15 Feb. 2015.

Seo, Songwon. A review and comparison of methods for detecting outliers in univariate data sets. Diss. University of Pittsburgh, 2006.

Shelby County v. Holder. 570 U.S. Supreme Court. 2013. Rpt. in Dimensions of Culture 2: Justice. Ed. Jeff Gagnon, Mark Hendrickson, and Michael Parrish. San Diego: University Readers, 2012. 109-112. Print.

Smith, Sarah. "Analysis of the Global Online Video Platforms Market." *-- LONDON, Jan. 5, 2015 /PRNewswire/* --. Reportbuyer, n.d. Web. 02 Mar. 2015.

Stanway, Abe. "Algorithms.py." GitHub. Etsy, n.d. Web. 13 Mar. 2015.

Stanway, Abe. "Analyzer." GitHub. Etsy, n.d. Web. 13 Mar. 2015.

Stanway, Abe. "Skyline." GitHub. Etsy, n.d. Web. 13 Mar. 2015.

Sun Tzu, and James Clavell. *The Art of War*. New York: Delacorte, 1983. Print. 17-18.

Trautman, Erika. “5 Online Video Trends To Look For In 2015.” *Forbes*. Forbes Magazine, 08 Dec. 2014. Web. 14 Feb. 2015.

United States. Cong. Senate. Committee on Commerce, Science, and Transportation. *The Emergence of Online Video : Is It the Future? : Hearing Before the Committee on Commerce, Science, and Transportation*. 112th Cong., 2nd sess. Washington: GPO, 2014. Web. 15 Feb. 2015

Vallis, Owen, Jordan Hochenbaum, and Arun Kejariwal. A Novel for Long-Term Anomaly Detection in the Cloud. Proc. of {6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14). Philadelphia: USENIX Association, 2014. Print.

Verbeke, Wouter, et al. "Building comprehensible customer churn prediction models with advanced rule induction techniques." Expert Systems with Applications 38.3 (2011): 2354-2364.

Vinson, Michael, B. Goerlich, M. Loper, M. Martin, and A. Yazdani. “System and method for measuring television audience engagement.” US Patent. 8,904,419. 26 Sep. 2013.

“What Does Copyright Protect? (FAQ) | U.S. Copyright Office.” *What Does Copyright Protect? (FAQ) | U.S. Copyright Office*. N.p., n.d. Web. 01 Mar. 2015.

Worstall, Tom. "The Supreme Court Should Just Abolish Software Patents In Alice v. CLS Bank." *Forbes*. Forbes Magazine, 29 Mar. 2014. Web. 01 Mar. 2015.

Zeithaml, Valarie A. "Service quality, profitability, and the economic worth of customers: what we know and what we need to learn." Journal of the academy of marketing science 28.1 (2000): 67-85.