DON'T LIE TO ME: INTEGRATING CLIENT-SIDE WEB SCRAPING AND REVIEW BEHAVIOR ANALYSIS TO DETECT FAKE REVIEWS

An Undergraduate Research Scholars Thesis

by

BENJAMIN JOSEPH LEVINSON

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. James Caverlee

May 2019

Major: Computer Science

TABLE OF CONTENTS

ABSTRACT

Don’t Lie To Me: Integrating Client-Side Web Scraping And Review Behavior Analysis To Detect Fake Reviews

Benjamin Levinson
Department of Computer Science & Engineering
Texas A&M University

Research Advisor: Dr. James Caverlee
Department of Computer Science & Engineering
Texas A&M University

User reviews are a widespread across the Internet as an indicator of the quality of a product. However, review systems can be vulnerable to attack. Malicious parties can manipulate the ratings of items by soliciting fake reviews in exchange for small payments. Sellers can use these fake reviews to hurt competitors or to promote their own products, artificially decreasing or increasing the ratings of products by paying for reviews. From previous work that uses crowdsourcing website postings to find fake reviews, we have a trained model that can detect fraudulent reviews using the time and rating features of reviews for a product (Kaghazgaran et. al "TOMCAT", ICWSM’19). This work also provides a web-based demo to validate the reliability of the reviews for a product. We encapsulate this model into a browser application that, when activated on an Amazon product page, crawls the reviews associated with that product, and issues a review manipulation score to the user. We also store the crawled reviews with the intention of building a dataset of reviews over time that can be used for further study into review manipulation and ways to improve review systems. Finally, we have analyzed the behavioral

features of reviews using the dataset provided by previous work (Kaghazgaran et. al "TOMCAT", ICWSM'19) that contains a set of random products and reviews from products known to be targets of manipulation. This analysis has uncovered more possible methods of determining review manipulation on a product by looking at the average number of helpfulness votes that reviews on a product receive, the average title length of a product's reviews, and the average length of a product's reviews. However, we found that the average length of a product's reviews was the feature that was most correlated within our dataset with review manipulation. We expect that future work focusing on the addition of these features will increase the overall effectiveness of the detection model.

ACKNOWLEDGEMENTS

I would like to thank my research advisors, Dr. James Caverlee and Parisa Kaghazgaran, for their guidance, support, and patience during my participation in this research program.

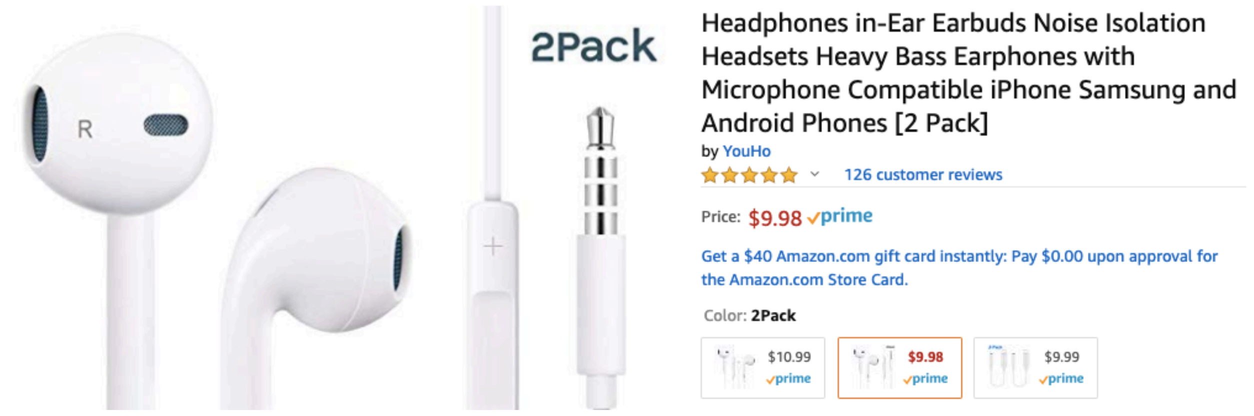
I would also like to thank my family and friends for their interest in my topic and encouragement as I engaged with my research.

CHAPTER I

INTRODUCTION

In an effort to gain user trust, many shopping websites encourage users to post reviews of listed products. The practice of allowing users to post reviews is widespread. For example, IMDB allows users to rate any movie and then displays top-reviewed movies to its users. Other websites like Amazon allow their users to review products that are available for sale. Reviews help to form a trust relationship between users and the websites that feature reviews (Dellarocas 2003). These reviews can be very helpful to new customers who might be interested in purchasing a product but want to know more about the quality of that product before they buy. Correspondingly, the presence of reviews has been found to increase product sales by as much as three times for high-priced items (Askalidis and Malthouse 2016). Online reviews also play a role in increasing awareness of a product, increasing word-of-mouth for the reviewed product (Wenjing et. al 2008). On a website like Amazon, this role is intensified by search algorithms which usually rank higher-rated and more widely-reviewed items highly.

However, in many cases, these review systems can be manipulated to falsely promote a product. Product sellers can promote their items by soliciting fake reviews in exchange for small payments. Sellers can also use these fake reviews to hurt competitors, artificially decreasing the ratings of products in the same category by paying for low-rating reviews. For example, Figure 1 shows a highly rated Amazon product, wired in-ear headphones styled similarly to Apple branded earbuds that appeared on the second page of results for the query "headphones". In Figure 2, we see an example of a suspicious review of this product. Although the review was posted on the product page for these earbuds, the review instead discusses a waterproof, Bluetooth speaker, giving the product the maximum rating of five stars. Suspicious reviews like this one can influence unsuspecting users to purchase low quality products.



Headphones in-Ear Earbuds Noise Isolation Headsets Heavy Bass Earphones with Microphone Compatible iPhone Samsung and Android Phones [2 Pack]

by YouHo

★★★★★ 126 customer reviews

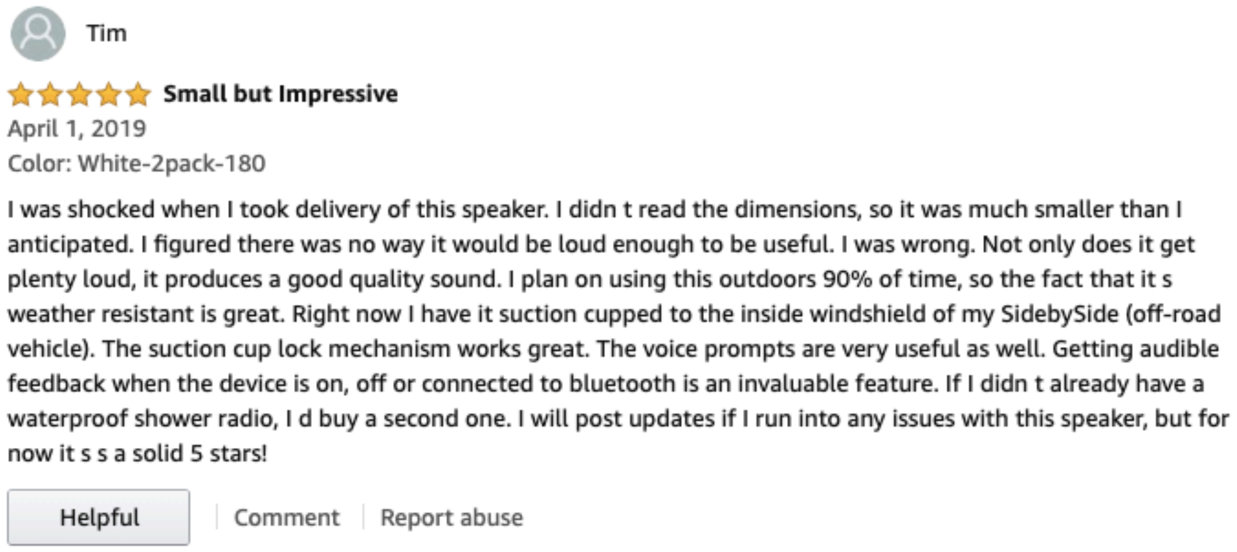
Price: $9.98 ✓prime

Get a $40 Amazon.com gift card instantly: Pay $0.00 upon approval for the Amazon.com Store Card.

Color: 2Pack

$10.99 ✓prime | $9.98 ✓prime | $9.99 ✓prime

Figure 1. Highly-Rated Headphones on Amazon



Tim

★★★★★ **Small but Impressive**

April 1, 2019

Color: White-2pack-180

I was shocked when I took delivery of this speaker. I didn t read the dimensions, so it was much smaller than I anticipated. I figured there was no way it would be loud enough to be useful. I was wrong. Not only does it get plenty loud, it produces a good quality sound. I plan on using this outdoors 90% of time, so the fact that it s weather resistant is great. Right now I have it suction cupped to the inside windshield of my SidebySide (off-road vehicle). The suction cup lock mechanism works great. The voice prompts are very useful as well. Getting audible feedback when the device is on, off or connected to bluetooth is an invaluable feature. If I didn t already have a waterproof shower radio, I d buy a second one. I will post updates if I run into any issues with this speaker, but for now it s s a solid 5 stars!

Figure 2. Suspicious Review Of Headphones

How Product Sellers Solicit Fake Reviews

Many review manipulators have turned to crowdsourcing websites like Microworkers.com to organize and recruit large swaths false reviewers in a short period of time

(Lee et al. 2013, Kaghazgaran et al. 2017). Workers are often incentivized by small monetary payouts or receiving the items that they review for free. The latter reward can have the added side-effect of marking a review as verified as workers may be instructed to purchase the item on Amazon before their review, which paymasters subsequently reimburse. One investigation has found that as verified reviews have become more important to Amazon's search algorithm, some companies have taken to purchasing targeted Amazon items with fake accounts and sending the merchandise to random legitimate addresses, taking a loss on the product but allowing the manipulator to receive verification of their purchase on their review (Goldman and Vogt 2018). As companies have developed methods to fend off attacks by malicious parties on their review systems, attackers have adapted and found new ways to manipulate item pages.

Detecting Review Manipulation

A recent method of identifying fake reviews is to find real postings on crowdsourcing websites requesting fake reviews and then to crawl the products mentioned in these postings and to use the resulting reviews to develop a neural detection model (Kaghazgaran et al. 2019). This research has resulted in a web-based version of the model that crawls a user-inputted product's reviews and then aggregates rating and timing features from those reviews and outputs a determination of whether that product has been a target of manipulation.

However, because the program is centralized on only a few servers, Amazon interprets the traffic from the review crawler as suspicious and banned the crawlers from accessing the website. In this paper, we solve this problem by creating an application that crawls reviews on the client-side and submits those reviews to a server, effectively making the review crawler a distributed application. We store the crawled reviews in a server-side database in order to build up a large dataset over time. The crawler collects review summaries, body text, user display names, helpfulness votes, and verified statuses. We expect that future research into behavioral characteristics of both benign and manipulated Amazon reviews will benefit from the dataset generated by this improved crawler.

The previous work known as the TOmCAT framework has also resulted in the development of a dataset composed of the reviews of more than 3000 targeted products. In this paper, we will study behavioral characteristics of the reviews in this dataset across each product and compare them to characteristics of a set of random products. We expect that the results of this study will provide a useful guide for the development of more features that can be used to improve the performance of the detection framework.

CHAPTER II

METHODS

In this section, we describe the design of the revised review crawler and the analysis of manipulated and random product reviews.

Browser Extension Design and Implementation

To begin, we outline the original system and our revised design. Our goal was to create a straightforward way for an Amazon user to receive a judgement of whether the reviews for a particular product appeared manipulated according to the model proposed in (Kaghazgaran et al. 2019). The original system comprised four parts—a review crawler to scrape reviews from a particular product, a feature extractor to process those reviews into several key features, a learning model which would issue a judgement based on those features, and a web server to allow for user interaction with the other three parts. However, the review crawlers would periodically be automatically banned by Amazon systems that interpreted crawler traffic as an automated attack.

To solve this issue, we revised the original design as visualized in Figure 3. Instead of a client interacting with a web server that did all of the work of crawling reviews, our new design would distribute the work of the review crawler to a browser extension client that users would install on their computer. By distributing the work of crawling reviews across client computers, we would avoid the problem of only one server making many requests to Amazon to crawl data and being automatically banned. Instead, many clients would make several requests each to crawl product reviews. After activation on a page, the client would scrape that product's reviews and send the data to a server, which would return a judgement of whether that product had been manipulated.

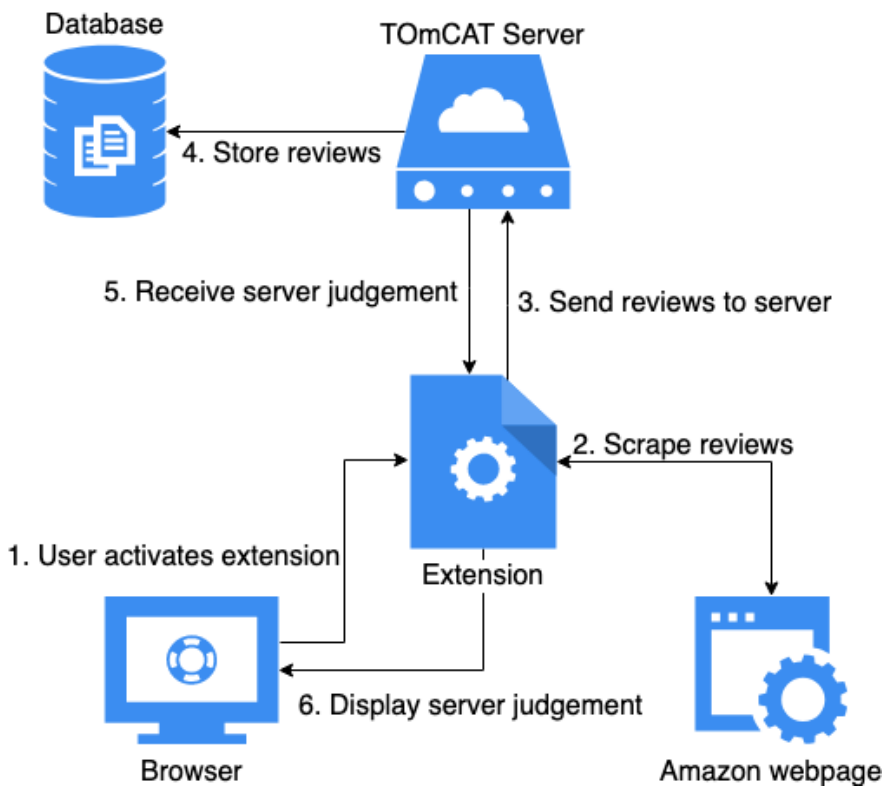

Database
TOmCAT Server
4. Store reviews
5. Receive server judgement
3. Send reviews to server
2. Scrape reviews
1. User activates extension
Extension
6. Display server judgement
Browser
Amazon webpage

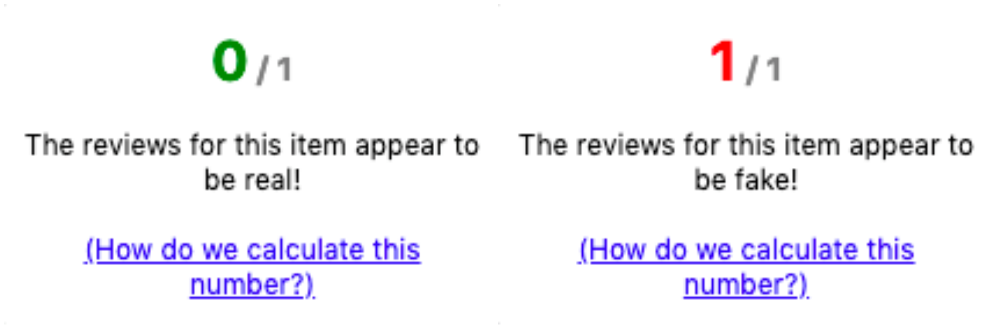
Figure 3. System Design And Workflow

Releasing the software in the form of an extension would allow a public release of the app to be straightforward. Many browsers have online extension stores that enable users to install extensions in very few clicks. Another benefit of using a browser extension as a client is that users could receive judgements on Amazon products without needing navigate to another website and manually input a product ID. Instead, we could display judgements in the same page on which the user had activated the extension. Finally, as more users employed our extension to crawl reviews and the coverage of products crawled increases, our dataset would also grow and become useful to future research involving Amazon reviews.

However, in order to store reviews for future retrieval, our system needed one further modification. Originally, the system would store reviews locally on the server in JSON files. File IO is useful as a proof-of-concept method of storing data, but it does not scale well when a server is receiving many requests. Instead, we revised the system to depend on a dedicated database that would store reviews by product ID in addition to the classification of manipulated or not that those reviews received.

We developed several mock-up designs exemplifying how trustworthiness feedback from the extension might be displayed to the user. Initial designs would directly modify the page, marking or removing suspicious reviews. However, we determined that these designs were incompatible with the model, which would only make a trustworthiness judgement over whole sets of reviews associated with an item page, not just a single review. To better align the extension design with this fact, we decided that the model's judgement of that page would be confined to a popup. This would have the added corollary of making the extension more resilient to any changes in the design of Amazon pages. Additionally, instead of the extension always being active and constantly communicating with the server over the course of a shopping trip to Amazon, it would be manually activated by a user button click. This method of activation would reduce the strain on the server of making judgements that the user might not require in the moment while also avoiding scraping reviews unnecessarily.

Since the client application was a browser extension, it was implemented in JavaScript, a very popular language for writing browser extensions. We used the JQuery library to simplify review scraping because the library included several ways of selecting elements from HTML code in addition to the ability to easily issue Ajax calls to Amazon and to our server's API endpoint. Whenever the extension was activated on an Amazon page, the crawler would use information from the product page to generate a list of up to twenty of that product's review pages to scrape. We chose twenty as a maximum number of review's to scrape because each page displayed ten reviews, so twenty pages would allow us to crawl two hundred reviews. Then, the extension would issue that number of Ajax calls in parallel to get those pages' HTML source and pass the source into a scraper function. The scraper would collect review summaries, body text, user display names, helpfulness votes, and verified statuses. Once every page had been scraped the extension would make an Ajax request sending all of this data to the server and wait for a response. Finally, as shown in Figure 4, the extension would display the judgement that the reviews appeared to be either manipulated or benign to the user in a pop-up.


0 / 1

The reviews for this item appear to be real!

(How do we calculate this number?)

1 / 1

The reviews for this item appear to be fake!

(How do we calculate this number?)

Figure 4. Example Judgements Displayed In Extension Pop-Up

Aggregation of Helpful Votes and Textual Features

Our current learning model, known as TOmCAT, issues judgements using rating and timing features aggregated across all of the reviews scraped from a product (Kaghazgaran et. al "TOmCAT", 2019). With our improvements to that system's review crawler, we were motivated to use some of the additional review behavioral attributes that the new crawler could collect to train and enhance the model. To assist with this goal, we developed several new features and analyzed their effectiveness on a dataset of reviews associated with a random selection of products and a dataset of reviews associated with products that are known to have been targets of review manipulation (McCauley et. al; Kaghazgaran et. al "TOmCAT").

Behavioral features

We developed three new behavioral features by which to analyze sets of reviews associated with a product:

Average Helpfulness Votes Per Review (HPR): We expected that fake reviews would be less substantial than real reviews and as a result products with manipulated reviews would have a lower average number of helpful votes per review than random products.

$$HPR(p) = \sum_{i=1}^{n} rating(r_i)/n$$

Average Length of Review Title (LR): The motivation behind this feature was to analyze whether targeted products on average had reviews that followed a different pattern of title length compared to reviews from random products.

$$LR(p) = \sum_{i=1}^{n} titleLength(r_i)/n$$

Average Length of Review (LR): We chose to analyze this feature to give insight into whether reviews from targeted products tended to follow a different length pattern compared to reviews from random products.

$$LR(p) = \sum_{i=1}^{n} reviewLength(r_i)/n$$

CHAPTER III

RESULTS

In this section, we will discuss the results of redesigning our review crawler model and of our analysis of using helpfulness votes and textual features for the purpose of detecting review manipulation. The review crawler, redesigned to be client-side and implemented as a browser extension, successfully avoided the problem found in the previous server-side-only version of the review crawler of being automatically banned from crawling reviews. By distributing crawling traffic across many clients, the application effectively could collect reviews without being banned. Instead of Amazon servers receiving hundreds of repetitive calls to consecutive pages or products from the same IP address, the shift to the client-side meant that, from the perspective of the server, there were only a few instances of atypical traffic arriving from many IPs rather than a great amount of atypical traffic coming from just one IP.

The detection framework issuing judgements from the server was not modified and still operated using the same features found in the original research study. The original work developed a dataset of products known to have been targets of review manipulation on crowdsourcing websites. Further targeted products were identified and incorporated into this dataset by finding users with duplicate reviews across multiple different items (Kaghazgaran et. al "TOMCAT", 19). We have used this dataset again for our comparison of feature distributions.

Analysis of New Features

We conducted our study of helpfulness vote and textual length features by comparing probability distributions of these features across products known to have been targets of review manipulation and across randomly selected products. We have described the datasets in Table 1. We plotted the normalized probability distribution function of each feature, meaning that we also normalized the result by accounting for the average and standard deviations when plotting our graphs.

Table 1. Datasets Description.

Dataset	**# Products**	**# Reviews**
Amazon (McAuley et. al 2015)	3,993	578,604
Targeted products (Kaghazgaran et. al "TOMCAT")	3,529	2,642,552

Average Helpfulness Votes Per Review

Figures 5 and 6 are respectively the resultant graph from applying a probability and a cumulative distribution function to the average helpfulness vote feature of our random and targeted product datasets. From Figure 5, we can see that the distribution of the random and targeted product datasets was fairly similar, but with our manipulated probability line shifted to the left. This shift indicates that targeted products have a higher probability to have reviews that receive fewer helpfulness votes on average than reviews from random products. This is consistent with our expectations that customers will find false reviews less helpful. According to Figure 6, we see that 38.0% of targeted products in our sample set had an average of 0 helpfulness votes per review while only 24.8% of random products had the same average. From this, we can see that targeted products are more likely than random products not to accrue any helpfulness votes at all.

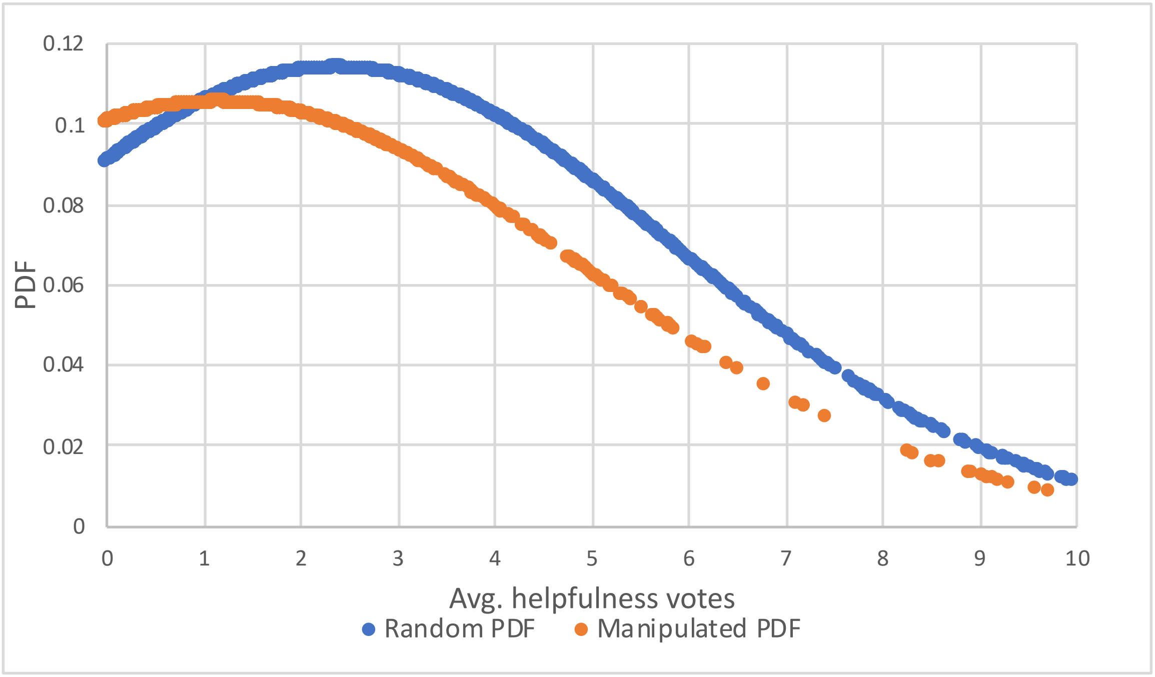


Figure 5. Normal Probability Distribution of Average Number of Helpfulness Votes.

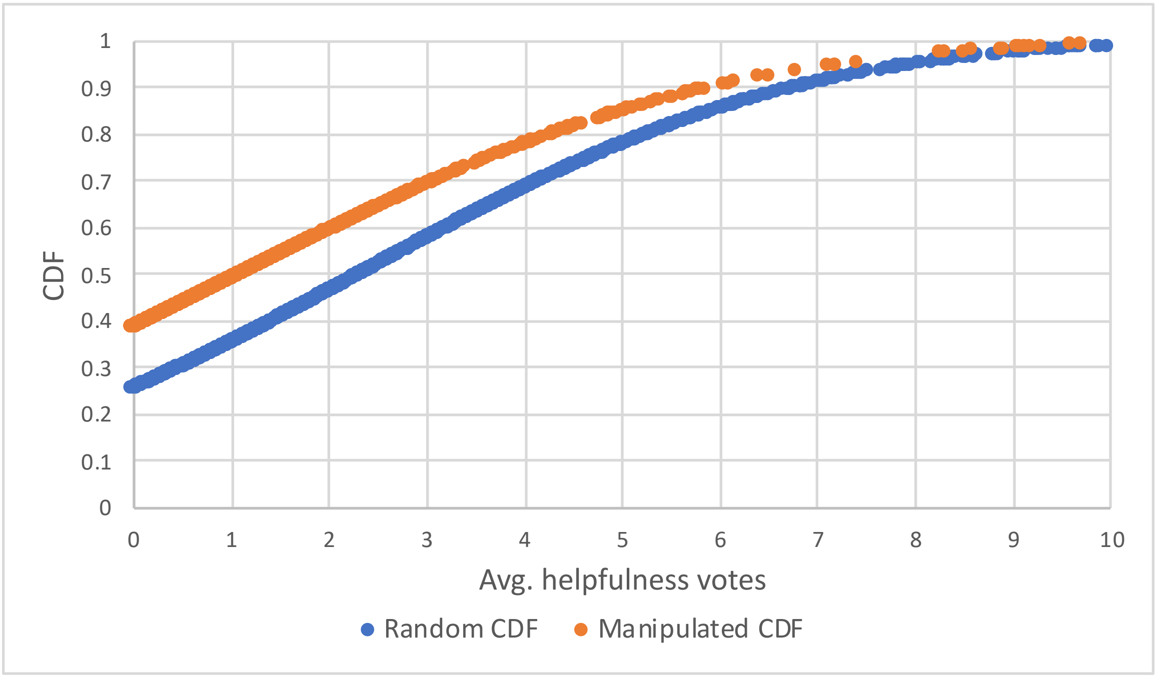


Figure 6. Normal Cumulative Distribution of Average Number of Helpfulness Votes.

Average Review Title Length

We found that the probability distributions of average review title lengths across our target and random product datasets did not vary by large amounts. According to Figures 7 and 8, the probability and cumulative probability distributions for random products are shifted to the left of the manipulated product distributions, signaling that there is a higher likelihood for manipulated products to average longer review titles than those of random products.

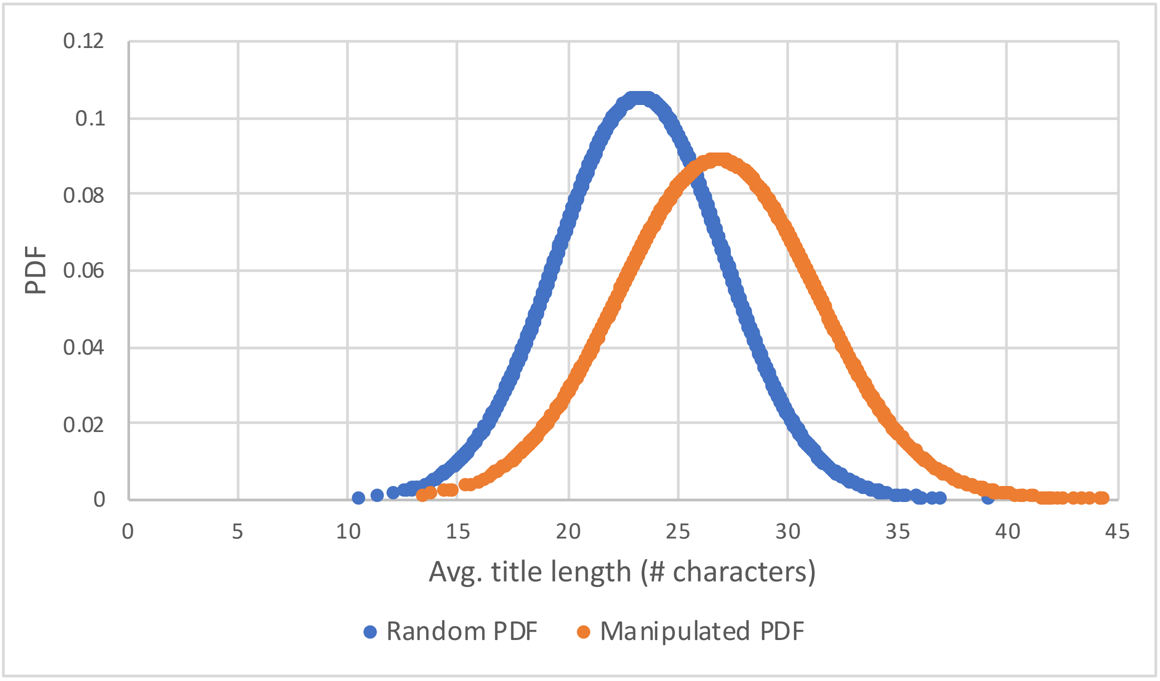


Figure 7. Normal Probability Distribution of Average Review Title Length.

Average Review Length

Figures 9 and 10 describe the probability distributions of the average review length on our two datasets. We found that targeted products in our dataset had reviews that tended to be much shorter than reviews on random products. According to the probability distribution, the most common review length for samples in this category was around 350 characters long, where about 0.2% of reviews were exactly this length. However, since the PDF describes the probability of a review having exactly a certain number of characters across a domain, the cumulative probability distribution provides a more useful description for this feature.

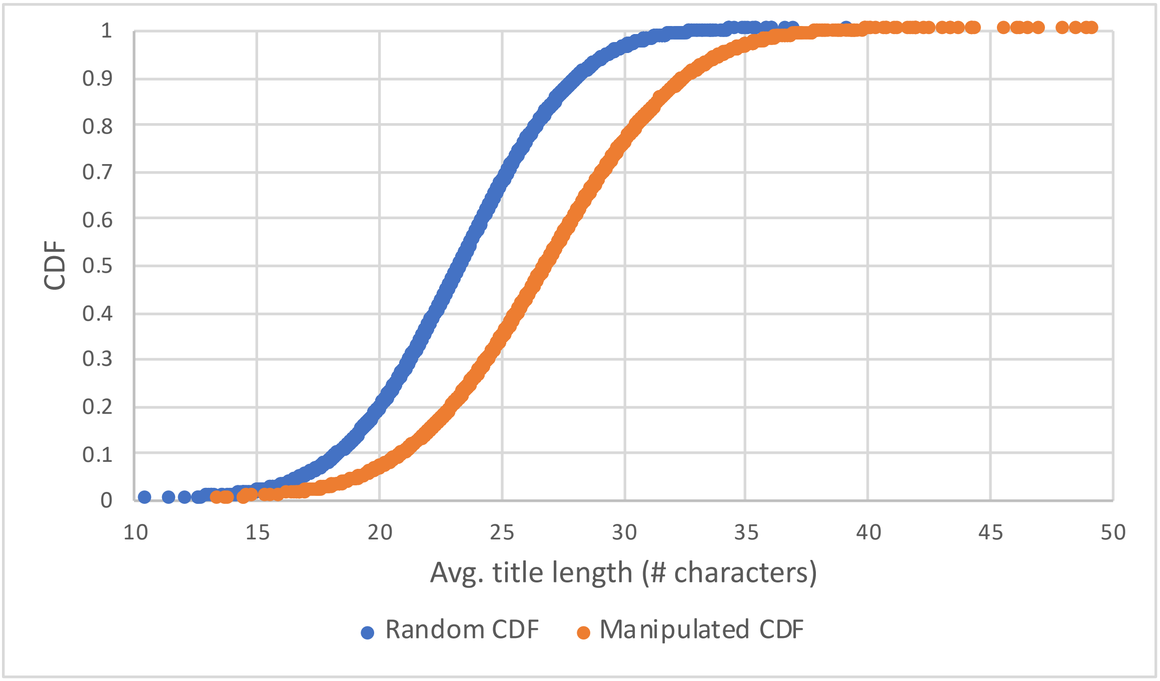


Figure 8. Normal Cumulative Distribution of Average Review Title Length.

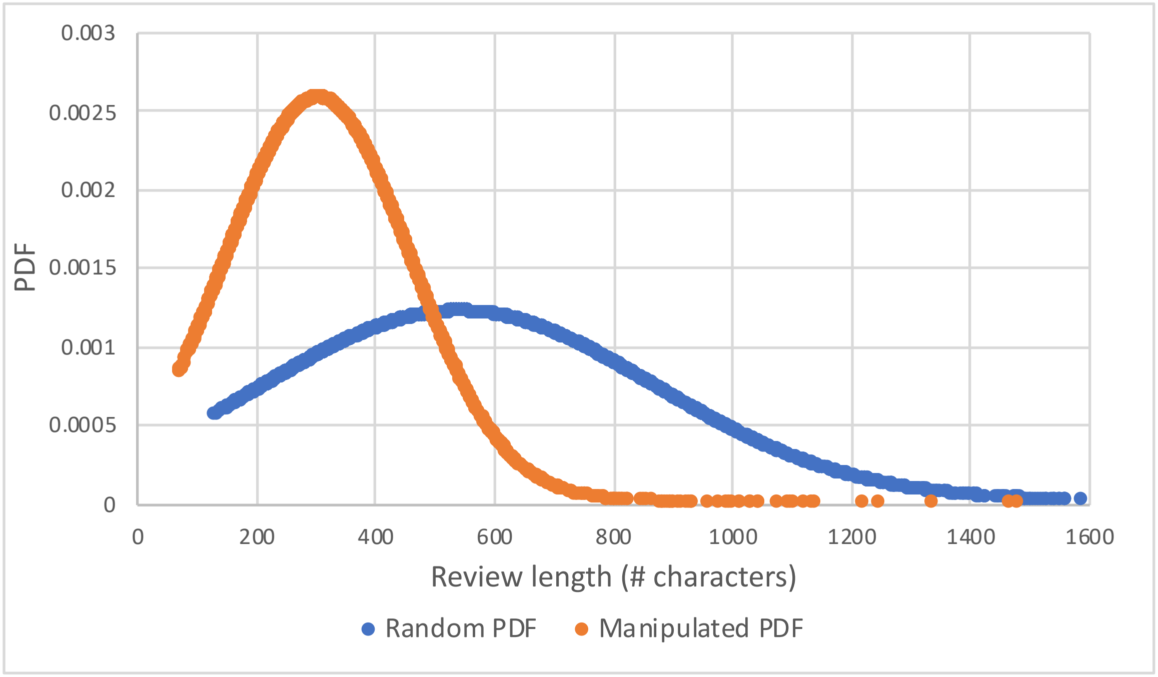


Figure 9. Normal Probability Distribution of Average Review Length.

From Figure 10, we can see that 97.0% of our targeted products had an average review length of less than or equal to 601 characters. On the other hand, only 56.7% of our random product samples had an average review length of less than or equal to 601 characters. This statistic shows that manipulated products tend to have a much shorter average review length compared to random products. A distribution with such a wide variation between our two datasets indicates that this feature will likely be very indicative of whether a product has not been a target of manipulation, especially if that product has reviews that average longer than 601 characters. Consequentially, it seems likely that this would be a useful feature by which to train future models.

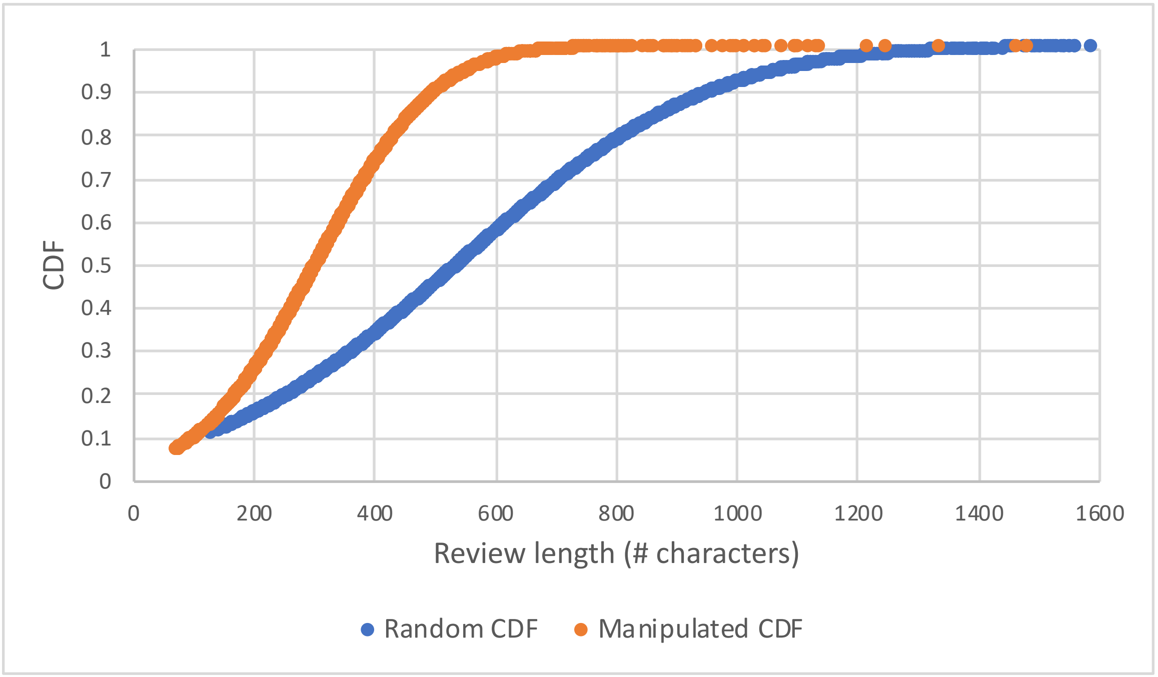


Figure 10. Cumulative Probability Distribution of Average Review Length.

CHAPTER IV

CONCLUSION

In closing, we have proposed and implemented a way to use a client program to develop a distributed review crawler across many systems. By distributing crawling traffic, the review crawler successfully avoids being automatically banned for atypical traffic patterns. Additionally, our proposed design enables the crawler to build a large dataset of reviews over time. We have also proposed three features by which to detect product-targeted review manipulation attacks and analyzed their distributions across a targeted and random product dataset. While we did not find average helpfulness votes and average review title length to be very useful signals of review manipulation, we did find review length to be correlated to review manipulation across our datasets, with shorter reviews being more indicative of manipulation.

Future Work

We look forward to future work that might incorporate our proposed features into an actual model. Additionally, given the success of our review length feature, we suggest that future work focus on more complex textual features like topic modeling and Jaccard similarity between reviews on the same product.

REFERENCES

Askalidis, Georgios, and Edward C. Malthouse. “The Value of Online Customer Reviews.” *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, 7 Sept. 2016.

Duan, Wenjing, et al. “Do Online Reviews Matter? — An Empirical Investigation of Panel Data.” *Decision Support Systems*, vol. 45, no. 4, 2008, pp. 1007–1016., doi:10.1016/j.dss. 2008.04.001.

Chrysanthos N. Dellarocas. 2003. The Digitization of Word-of-Mouth: Promise and Challenges of Online Feedback Mechanisms. *SSRN Electronic Journal* (2003).

Goldman, Alex, and Vogt, P.J.. "The Magic Store." Audio blog post. Reply All. Gimlet, 11 July 2018. Web. 26 Jan. 2019.

Lee, Kyumin, Tamilarasan, Prithivi, AND Caverlee, James. "Crowdturfers, Campaigns, and Social Media: Tracking and Revealing Crowdsourced Manipulation of Social Media" *International AAAI Conference on Web and Social Media* (2013): n. pag. Web. 26 Jan. 2019.

McAuley, J., Targett, T., Shi Q., and Van Den Hengel, A. 2015. Image-Based Recommendations on Styles and Substitutes. *SIGIR*.

Parisa Kaghazgaran, Majid Alfifi and James Caverlee. TOmCAT: Target Oriented Crowd Review Attacks and Countermeasures, *International AAAI Conference on Web and Social Media* (2019).

Parisa Kaghazgaran, James Caverlee, and Anna Squicciarini. 2018. Combating Crowdsourced Review Manipulators: A Neighborhood-Based Approach. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018).

Parisa Kaghazgaran, James Caverlee, and Majid Alfifi. 2017. Behavioral Analysis of Review Fraud: Linking Malicious Crowdsourcing to Amazon and Beyond. *International AAAI Conference on Web and Social Media* (2017).