PRODUCT REVIEW CREDIBILITY ANALYSIS

Anusha Prabakaran

A Capstone Project Report
submitted in partial fulfillment of the
requirements of the degree of

Master of Science in Computer Science & Software Engineering

University of Washington

Year 2018

Project Committee:

Dr. Min Chen, Committee Chair
Dr. Erika Parsons, Committee Member
Dr. Yang Peng, Committee Member

Abstract

Product reviews on online shopping sites have become the vital source of customers' opinions. These reviews have a significant impact on purchasing decisions and product rankings on popular e-commerce websites. Unfortunately, for the desire of profit or fame, fraudsters (spammers) write deceptive reviews (spam reviews) appreciating or deprecating a product. These reviews mislead potential customers and negatively affect the revenue of many genuine organizations. This fact has raised the need for an effective method to detect the fake reviews and spammers. Drawing from the literature, there are many types of spam detection methods that help to provide reliable resources to customers and businesses. Yet, these methods have drawbacks, like, the supervised approaches have imbalanced data, rating based filtering systems and linguistic approaches are impaired by shrewd spammers, and synthetic datasets do not match the real world scenarios. However, existing research does not use a combination of methods to detect the spam reviews. The aim of this project is to develop a practical end-to-end system that uses a set of three methods: detection of duplicate reviews, detection of anomaly in review count and rating distribution, and detection of incentivized reviews to analyze the Amazon review data and generate a score. This score indicates the credibility of the reviews of a product. The proposed system could facilitate businesses to identify and constrain on vendors and spammers engaging in these dishonest practices. This system could also aid in data mining and online spam filtering systems to filter the product reviews and refine the product rankings. These three methodologies complement each other and identified the spam products with greater accuracy by using a statistical credibility scoring system, without requiring significant computational resources, rather than using a single method.

Contents

List of Tables

List of Figures

1 Introduction

"This gem of a speaker will make you the LIFE of the party. It gets really loud without distortion. When you raise the volume with the remote, the LEDs in the front blink in ascending order to indicate the level of volume. Bluetooth works like a charm, and it pairs quickly and painlessly. Fills my entire office with sound good sound, easy to hook-up to tv and bluetooth connection to iphone or ipad. Very impressed this works well with my amazon echo. All the necessary inputs are provided, and it sounds great."

- Tony Pardo, Amazon.com

On reading the above review about a speaker, it is difficult to distinguish whether it is from a real customer or a spam review. The main challenge lies in identifying the features that contribute to suspiciousness.

Sellers say the flood of inauthentic reviews makes it harder for them to compete legitimately and can crush profits. "It's devastating, devastating," said Mark Caldeira, owner of the baby-products company Mayapple Baby. He said his product rankings have plummeted in the past year and a half, attributing it to competitors using paid reviews. "We just can't keep up."

- The Washington Post, April 2018 [9].

Fake reviews adversely affect not only the customers but also a lot of genuine small businesses.

In this research, we intend to analyze reviews of products by using a combination of certain features and estimate the credibility of the reviews.

1.1 Background

With the near ubiquitous access to the internet and growing popularity of online shopping, there is a rapid increase in the percentage of retail shopping through online stores. Unlike the traditional retail models, many of the online stores operate as marketplaces where the goods are sold by third-party sellers. The purchase decisions at these online stores are significantly based on the product reviews, which contain information of the consumer opinions on the product. The reviews are useful for both the buyers and the sellers. For the sellers, the reviews help establish their trustworthiness and the quality of the goods they sell.

The reviews should be made by real consumers who reveal their honest experiences of a product. However, due to the drive for profit or winning over the competitors, some unscrupulous professionals attempt to bias the product reviews by spamming opinions and writing fictitious reviews that are intentionally written to look authentic to deceive the consumers. Such reviews are called as spam reviews and the fraudsters are called as spammers. The ease of posting reviews has set a way for such spammers to do deceptive opinion spamming. Spammers may increase the value by writing unjust positive reviews or devalue by writing spiteful negative reviews for the targeted competitor's product [22].

1.2 Problem Description

Most of the online products have a huge number of reviews. Though a few consumer sites [47] have consolidated tips and clues to manually spot spam reviews, for products with many reviews, to manually check and distinguish the spam opinions from the real reviews is practically impossible. Several high-profile cases have been reported [37] [38] [57] and spammers have transparently admitted having been paid to write fake reviews in media investigations [27] [60]. Many businesses, in order to increase their sales, have rewarded positive reviews with promotions and coupons.

Opinion spamming is a growing concern among the public consumers and the businesses [42] [44]. They are harmful not only to potential buyers but also for business owners. Therefore, spam detection and opinion mining techniques are needed to assist online businesses to analyze the posted reviews on products, to detect and filter spam reviews and to provide truthful reviews to customers [53]. However, research in spam detection is still not adequate and many problems are not solved yet. Hence, a new system with advanced abilities to detect spam reviews are needed. These systems should be practical to use and easy to deploy.

1.3 Project Objective

To develop a robust end-to-end working system that is practical, scalable and usable, which builds on the recent state-of-the-art methodologies and generates a report of the credibility of the reviews for the given product.

The objective of this project is to develop a complete system for analyzing the product reviews of a large online marketplace Amazon.com and generates review credibility rating for any particular product. The system uses a set of three methods: duplicate review detection, detection of anomaly in review count and rating distribution over time, and incentivized review detection to analyze the review data of a product and uses the output of these methods to generate a cumulative score. This score is indicative of the credibility of the reviews of a product. The system also needs to provide a high-level explanation of how and why this recommendation was made to visually educate the curious user.

The system builds on existing research concepts that are based on duplicate detection [22], review count and rating distribution analysis [19]. Research papers in this area are about the theoretical approaches for detecting fake reviews. They typically use small datasets to showcase the effectiveness of their proposed methodology. They do not discuss the incorporation of these methodologies for a practical system. In this research, we have developed a practical end-to-end system which works seamlessly on large datasets and generates user consumable

reports. The additional novel methodology the system uses is to detect the incentivized reviews. The incentivized review detection is based on text analysis. The system developed should be reusable for future work like building a website on this application and also allow additional modules to plug into this framework with limited effort.

1.4 Summary of Existing Research

In the past few years, many researchers have shown great interest in identifying the truthfulness of the reviews. Opinion spam and trustworthiness of online reviews were initially studied in [23]. Since then, there are numerous studies exploring the different dimensions of detecting the spam reviews. At a top level, approaches can be categorized as content similarity analysis [23] [15], reviewer's behavior analysis [41] [14] [39] [24] which uses the temporal footprints of the user, time bursts review ratio [66], rating deviation based filtering systems [33] [3], linguistic approaches [13] [44] [45] that study the language patterns and psycholinguistic clues in the review text, distributional analysis [13] [14], graph-based methods [2] [12] [30] [64] that leverage the relationship between products, users, and reviews. There are various studies to detect individual spammers [2] [12] [33] [39] [64] and group spammers [40] [67].

[24] built their own Amazon review dataset by crawling Amazon.com and compared the product features mentioned in a review with other reviews to detect the duplicates and the near duplicates. The main technique for spam detection has been supervised learning, where these duplicates and near duplicates have been assumed as spam reviews in training the model. The accuracy of supervised learning based algorithms depends on high quality labeled gold-standard dataset.

Unfortunately, due to the lack of labeled dataset, existing works relied on two main approaches for labeling the data. First, using the human annotators to judge and label the reviews [24] [40]. A study by [45] shows that fake reviews are not easily identified by human

readers. Features like reviewer's ID, rating and helpfulness could be easily manipulated to appear as authentic opinion. Another approach was to use crowdsource deceptive reviews using Amazon Mechanical Turk (MTurk). It is affordable and easy to get large-scale deceptive gold-standard reviews from turkers. Despite the benefit, it is uncertain that these turkers represent the general spammers as real-world situation are diverse [45] [44].

Most of the methods need certain features that are particular to the dataset. Content similarity analysis could be applied to any type of dataset but needs expensive computations. Numerous efforts in spam detection to provide reliable reviews to the consumers and the operators show the complexity of this problem. This research uses these duplicates and near duplicates as one of the methods to detect the spam reviews.

Knowing the importance and the value of the user-generated content which could be exploited in numerous ways, a few reviews curating companies like Yelp [69] have developed their own review analysis and filtering systems to identify and remove the spam reviews. Though this increases the cost of spamming and alleviates the negative impression, the filtering mechanisms that are in use are not adequate to prevent resourceful spammers from review spamming. These spammers continuously adapt their methodologies to circumvent the filtering mechanisms that are added. One such recent adaptation is incentivized reviews, wherein the sellers offer deep discounts to get more favorable reviews than the competitors. These deep discounts cause the customers to overlook the defects of the product and give favorable reviews because of their perceived value. These reviews are no longer useful to customers who have to purchase the product at normal prices.

1.5 Challenges

Large online shopping place like Amazon.com has a huge quantity of review text for spam detection analysis. The use of Amazon review dataset poses three key challenges:

(a) The main challenge with such real-life dataset is that it does not have ground truth, that is, the spam and non-spam labels for each review. There is no reliable gold standard to know for certain that a review is spam or not. The highly cited papers on this topic call the absence of the labels the biggest challenge. Through primary literature review, papers use two options: treating the duplicate and the near-duplicate reviews as spam and paying volunteers to write spam reviews. We do not want to manually label the products because of the time consumption, the cost associated with labeling such large dataset, and human judgments would be partly biased by the context features of the reviews.

(b) Literature review illustrates many methods for fake review detection. Whether these methods can be used for a large dataset is unknown. The previous papers work on small datasets and are more proof of concepts rather than an end-to-end system.

(c) The dataset is of large size. It is vital to appropriately identify a subset to verify the correctness of the proposed methodologies. Also, the data processing can be quite time-consuming.

1.6 Contribution and Beneficiaries

The contribution of this paper are:

(a) The system is a practical, maintainable, scalable and deployable software application.

(b) A unique approach that combines multiple methodologies in a practical fashion to overcome the ever-changing techniques employed by spammers. Fake reviews which have escaped one methodology could be identified by another methodology.

(c) A novel methodology for detecting the reviews when the product was given at incentivized price or free.

This research aims to benefit the following:

(a) To facilitate the consumers to shop online more confidently.

(b) To aid the marketplace operators to improve their customer satisfaction.

(c) To verify that the genuine products do not get drowned by the noise generated by the fake products and dishonest sellers.

1.7 Report Outline

The remainder of the paper is structured as follows: Section 2 discusses the related work, explaining the various types of spam and review spam detection techniques. Section 3 describes the system design, the dataset, and the three proposed methods with the algorithms and the techniques used for detection of spam reviews. Section 4 presents the experimental analysis and results along with the evaluation of the model. Section 5 presents a short discussion. Finally, 6 summarizes our conclusions and explores the future work.

2 Related Works

This section presents the related work in the type of spams and online review spam detection techniques. Spamming pervades in any type of information systems like web, e-mail, short message service (SMS), social blogs, and review platforms. Section 2.1 explains the different types of spams. The research in detecting online review spam could be classified into three categories: Detecting Spam Reviews, Detecting Spammers, and Detecting Group Spam. Section 2.2 analyses the work done in the three categories of the online review spam.

2.1 Types of Spam

Web spams or spamdexing are deceptive activities that attempt to increase the ranking of a page in search engines. The web spam detection algorithms use different types of information like content-based methods, link-based methods, and other data such as user behavior, clicks, HTTP sessions [11] [56]. Email spams or junk emails are a cost-effective way to send unsolicited commercial messages and links to phishing websites. For the detection of email spams, knowledge-based engineering approach and a number of machine learning algorithms have been proposed for building models that depend on extracted features [21] [35]. The performance and accuracy of such models depend on the type and the number of features used. SMS spam is widely prevalent in the Middle East and Asia. Many content-based and semi-supervised machine learning models have been developed to filter them [1].

Review spams are fake reviews written by spammers in order manipulate the value of the product sold online. In comparison with the above types of spam, detection of review spam is important as the manual assessment of reviews and distinguishing real opinions from fake reviews is nearly impossible [23]. It is difficult to detect fake reviews as the spammers disguise themselves. Analysis of online reviews has become a popular research topic recently and a few systems have also been developed to detect the fake reviews. Though many research has

been done in this area, there are no pertinent state-of-the-art methods in distinguishing and detecting the spam reviews. Hence, the inherent complexity of this topic shows the need for more focused research and has become an interest for many machine learning and natural language processing researchers.

2.2 Online Spam Review Detection

A comprehensive review of the various approaches in detecting the spam reviews shows that the approaches could be classified into three categories: Detecting Spam Reviews, Detecting Spammers, and Detecting Group Spam [18].

2.2.1 Detecting Spam Review

Detecting the spam review is majorly identified as a binary classification problem – spam vs non-spam or fake vs real. Studies show that humans are unable to consistently identify or detect spam reviews by just reading a set of reviews. So, automatic filtering and detection of spam reviews are a primarily suitable technique for having honest reviews [45]. The lack of a reliable labeled dataset has been posing a major problem for calculating the accuracy for all the research done in this field.

Section 2.2.1.1 to section 2.2.1.6 details the different approaches to detect the spam reviews.

2.2.1.1 *Writing Style*

A psycholinguistic deception (fictitious opinions that have been deliberately written to sound authentic) detection and genre identification using the computational linguistics distribution of parts-of-speech (POS) and text categorization was done by [45] [44]. The writing style detection using character n-grams as features to show the stylistic information and the lexical content was shown by [15]. It is a stylistic classification task, that is, for a specific domain,

the spam and honest reviews will have similar content but will differ in the way opinions are written (style). Similar telltale signs were used by [4]. The drawback of these studies is that the spammers can adapt their linguistic and verbal skills to mimic honest user reviews and escape the detection. In some cases, the spammers write their honest opinions about a product that they purchased for a non-purchased product to spam it. In such scenarios, the context of the reviews will not be efficient to identify spam reviews.

2.2.1.2 *Synthetic Dataset*

[45] did their experiments with the synthetically produced dataset on six common online sites in restaurant and travel domain, finding that reviews by first-time reviewers tend to suppress the deception rate. The fake reviews were built using Amazon Mechanical Turk (MTurk). Many researchers have alleviated the problem of lack of labeled dataset by crowdsourcing fake reviews. The downside of this approach is the use of synthetic dataset as they do not reflect the real-world fake reviews. [39] [58] [7] showed that the performance varied while applying the same methods for real-world datasets. The reviews are not written by the same spammers and in the same circumstances which could have visible effects on the style of the reviews. The differences between spam reviews written by Mechanical Turkers was noted by [32], which tend to exhibit parts-of-speech features of imaginative writing, with those from domain experts, which tend to exhibit parts-of-speech features of informative writing.

2.2.1.3 *Content Similarity*

An alternate approach to the synthetic dataset and in contrast to the similarity in the writing style, is that the similarity in the content of the reviews could be used for detecting the spam reviews. The heuristic labeling of duplicates and the near-duplicate reviews were considered as spam reviews [23]. These duplicate reviews were used to train the model using logistic regression and identified the outlier reviews with various product and review centric features.

Although absence of reliable evaluation pleas the accuracy of content similarity into question, a bunch of methods demonstrated that this approach is advantageous [3] [28] [22] [23] [34]. In general, it is believed that spammers write a few numbers of fake reviews and try to use it for assorted products of a brand with various identities.

In this research, duplicate and near-duplicate detection was used as one of the systems to identify the major spam reviews. The drawback is that spammers are able to beat this system by crowdsourcing the fake reviews. The recent investigation by The Washington Post newspaper [9] mentions many Facebook groups where sellers pay few dollars for each fake review.

2.2.1.4 *Review Rating Distribution*

[14] and [20] hypothesize that there are natural distributions of review star ratings, and those businesses or organizations that hire spammers to spawn positive reviews would have distinct star rating distributions from those who do not engage in such deceitful practices. They show that most of the one-time reviewers, who are more likely to be accepted as spammers, leave extreme (one or five) star reviews than users who write multiple reviews.

In this study, the skewed star rating distribution is used as one of the prime heuristics for anomaly detection in the ratings of the product. The one major difference is that, instead of using review ratings from one-time reviewers, this system uses all the product's reviews from all the reviewers for anomaly detection. Moreover, one-time reviewer identification is not enough as there are people with multiple profiles who post fake reviews for many products in return for monetary compensation.

2.2.1.5 *Complimentary Products*

Some of the business or companies offer discounted or complimentary products for their potential customers in exchange for writing a review. Despite these reviews include a dis-

claimer statement declaring that the reviews are reviewer's unbiased opinion, on average, most of such reviews are highly positive and have a higher star rating. [10] and [48] did a study on the Amazon's Vine program (members of this program are provided with free products in exchange for writing a review), found that members had "enrollment-effect", that is, they wrote long, positive reviews in a complex sentence structure with more of personal descriptions of the product. Such effect is likely to extend beyond the Vine Voices to reviewers provided with promotional products from vendors themselves.

In this research, we build a dictionary of incentivized synonyms and use it to identify the reviews which were written when the product was given at a discounted price or for free. The system checks for incentivized reviews where the reviewer acknowledge having gotten the product at discount or free in the review text.

2.2.1.6 *Spatial and Temporal Patterns*

A few studies used the time of review posting distribution to narrow down the number of reviews to be compared to identify the fake reviews (Heydari 2015). They detected the suspicious time intervals with sudden bursts or spikes (anomalies) in the review count. A combination of temporal and spatial patterns was used to detect the spam attack [31]. A longitudinal study along the temporal dimension revealed that most of the fake reviews were written during a certain time of the day and week. The spatial study was done by mapping the IP addresses of the users used in registration to the city level coordinate of the product. For Amazon review dataset, the location of the reviewer is not available for this research to study.

The main downside of the above approach is that there are many products which are seasonally purchased that have high review count at those specific seasons and some products gain popularity over time (linear growth). This system extends this idea by allowing seasonal variations, i.e., holiday products might get lot more reviews in December and January every

year. Seasonal Hybrid ESD (S-H-ESD) is one of the very recent systems that helps detect such anomalies in time series data [19]. This should have high scalability and performance as it was built for cloud scale scenarios.

2.2.2 Detecting Group Spam

The group spam is generated by a group of spammers working together in writing fake reviews to promote or demote a product, maximize the impact and easily control the sentiment of the product. Spammers, who get paid to write, write reviews for several products, which unfortunately gives them away. The number of approaches to detect a group of spammers is limited. The approaches can use purely the word-based textual features, metadata of the reviewers, or a combination of both.

A diverse group spam indicators were detected by using a technique of frequent pattern mining and building a relational model by computing the relationships between various spam indicator values, like time window (short intervals), early time frame (product's first reviews), star rating deviation between members of a group of spammers and other reviews, content similarity between group members, and number of products for which the group is creating spam reviews [41]. Similar item-set mining model and features were used by [26]. The above studies used a combination of word-based and metadata similarity and bargain that labeling group spammers are easier than identifying individual spammers.

A two-step method to detect spammers was done by [68]. First, a graph-based measure to compute network footprint score which detects the statistical distortions caused by spamming activities. Second, a group strainer algorithm which uses the hierarchical clustering algorithm to cluster the groups of spammers.

This project uses the idea of group spamming in a short window to measure the credibility of ratings and the number of reviews. The posting time of the reviews were binned into small time buckets for analysis.

2.2.3 Detecting Spammers

Fake reviews are often written by experienced professionals (spammers) who are paid to write high quality, believable reviews. Opinion spammers seek to alter the perceived quality of a product by creating fraudulent reviews. However, some spam reviews, the star rating of the product is incompatible with the review content. These spammers are aware of the rating deviation based filtering system and try to escape it by moderate ratings but capture the customers by the review content [18]. We alleviate this problem in our approach by taking into account the rating and the content of reviews.

A graph-based networking model – FRAUDEAGLE framework - an unsupervised, general, review graph with three nodes reviewers, products, and reviews was proposed by [2]. It shows the correlation of the nodes among the users and the products using rating behaviors of the reviewers. This framework has the advantage that it functions in an unsupervised fashion without requiring labeled dataset. It exploits the network effect among reviewers and products, unlike the other approaches which use the context of the review or behavioral analysis. It also allows to score users and reviews for fraud detection, group for visualization and sense-making.

Numerous studies have tried different features to improve the performance of classifiers. [29] used sentiment scores, reviewer profile, and product brand as the main attributes to train a semi-supervised ranking algorithm to detect the spammers. Rating deviation was one of the main features used in the detection of spam by [33][54]. The authors performed a rating deviation detection using binomial regression method to detect the anomalous proportion of ratings that manipulate and deviate from the majority opinion. They select a subgroup of suspicious reviewers and perform user evaluation tests for additional analysis.

With the goal of detecting singleton spammers (a person who has written only one review), [66] focused on behaviors of reviewers' using temporal pattern discovery where the

different phases of the product were analyzed. According to the authors, the behavior of the reviewers can be divided into arrival phase (when the spammer is hired or product is purchased), and writing phase (spammer starts writing reviews). The behaviors in were analyzed normal, promotion and spam attack arrival. Accordingly, they found that honest customers write delayed reviews and spammers write instantly after arrival phase.

Similarly, in another approach proposed by [12], the posting time of reviewers was considered as the main features of the five spammer behavioral features, namely ratio of Amazon verified the purchase, rating deviation, content similarity, and burst review ratio used as indicators to detect spammers. Their model has more accurate results than to [66]. The accuracy must have increased with the use of Amazon verified the purchase feature, which limits their approach very specific to Amazon dataset.

The key disadvantage of detecting spammers is scalability. There are a lot of online shoppers and building a profile of each shopper to categorize and detect is not scalable. Hence, this line of research was not pursued in this system. This system uses the anomaly in the rating deviation and the review count over the time of posting for detecting the spam reviews.

3 Methods

In this section, we describe the overall system architecture; the dataset; the three main methodologies: Detection of Duplicate Reviews, Detection of Anomaly in Review Count and Rating Distribution, and Detection of Incentivized Reviews used by the system for generating the credibility score; generation of credibility score; the user interface; and the testing methodology to evaluate and measure the effectiveness and accuracy of the system.

3.1 System Architecture

The system was built on Model-View-Controller (MVC) architectural pattern. The key requirement for this system was low coupling, that is, the ability to develop and execute each of the three methodologies independently because of their varied dependencies and complexities. The other necessities were to separate the generation of the report from the methods, and the ability to demonstrate as a standalone application that could be extended into a web service in the future. Low coupling and the ease of modification for future development are the salient features of MVC architecture. First, the model was developed with the data store and the three methodologies. Then, the view was developed for the report generation. Finally, the controller was developed for actions like selecting the dataset and entering the product ID (asin).Figure 1 shows the system architecture. A detailed explanation of the data and all the methodologies are explained in Section 3.3 of this report.

3.1.1 Model

The model is responsible for the business logic maintaining the data of the system. It responds to the instructions from the controller and requests from the view. The model for this system consists of the data store; the three methodologies: detection of duplicate reviews, detection of anomaly in review count and rating distribution, and detection of incentivized reviews;

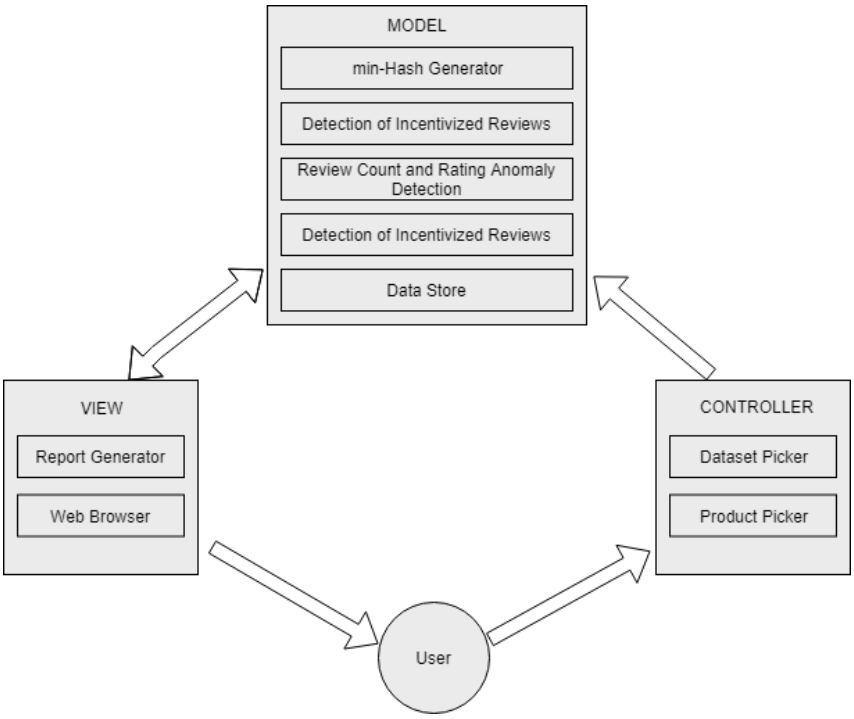

Figure 1: *System Architecture.*

and min-Hash generator used for the detection of duplicate reviews.

3.1.1.1 *Data Store*

It consists of 2 sets of files: the review dataset consisting of reviews and the outputs of min-Hash and duplicate reviews detection. In the current implementation, the data store is the local file system. It is trivial to extend to a cloud-based object store or a remote file share. The methods read the review dataset stored here.

3.1.1.2 *Min-Hash*

It is one of the steps in detecting the duplicate reviews. It reads the input from the dataset stored in the data store and calculates the min-Hash values for all the reviews. It writes the output to a comma-separated values (CSV) file which is stored in the data store.

3.1.1.3 *Detection of Duplicate Reviews*

It reads the output from the min-Hash and builds an inverted index to detect the reviews. The results are stored in the data store in a CSV file.

3.1.1.4 *Detection of Anomaly in Review Count and Rating Distribution*

It reads the input from the dataset and identifies any anomalies in the number of reviews and rating distribution over time.

3.1.1.5 *Detection of Incentivized Reviews*

It reads the input from the dataset and identifies whether the reviews were written when the product was sold at a highly discounted price or for free as in exchange for some honest review. The results are written to a CSV file.

3.1.2 View

The view presents the final report output in the web browser. It consists of report generator and web browser.

3.1.2.1 *Report generator*

The report generator collects all the output from the components and generates an HTML report.

3.1.2.2 *Web browser*

The HTML from the report generator is launched in a web browser.

3.1.3 Controller

The controller responds to the input from the user, validates it and perform interactions on the data model objects.

3.1.3.1 *Dataset Picker*

The user can choose a dataset from the different types of product categories. It interacts with the model and loads all the asin's from the dataset.

3.1.3.2 *Product Picker*

The user inputs the product ID (asin). It interacts with the model and selects the asin from the dataset for further analysis.

Python was the programming language used to build this system. The libraries of Python has many options for data analysis and helped to minimize the development time. Python is also used for server-side programming in many web services. Hence, the software developed as the part of this project could also be extended to be a website. Figure 2 shows the overall data flow in the system.

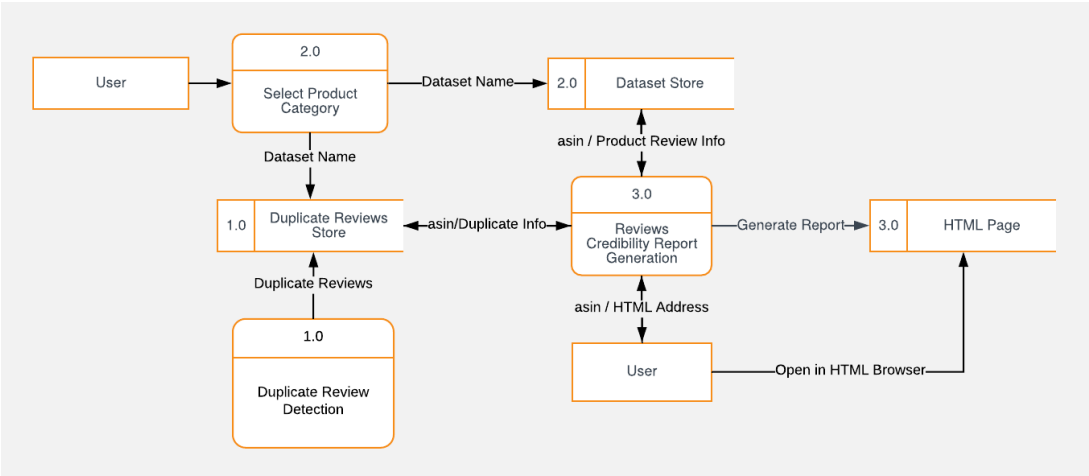

Figure 2: *Overall data flow in the system.*

Figure 3 shows the interactions and detailed data flow between the various components in the system.

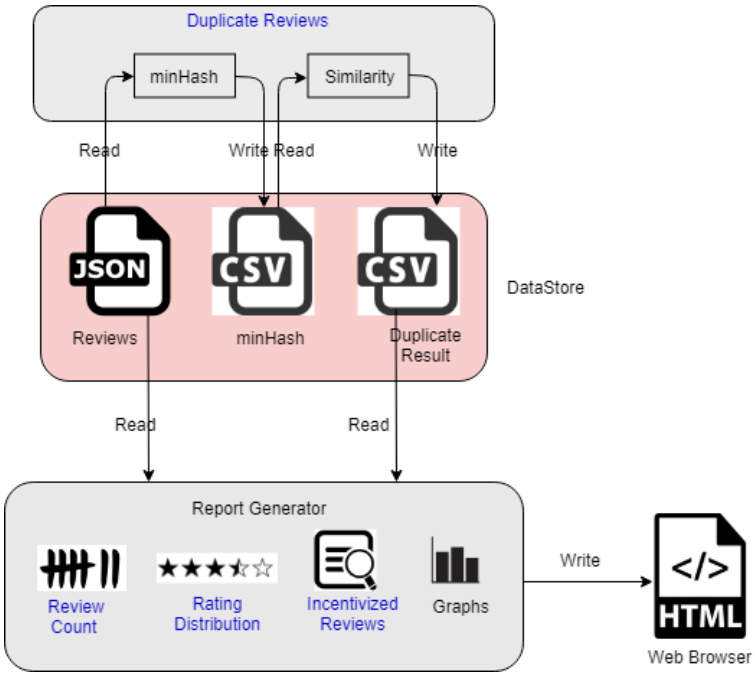

Figure 3: *Interactions and detailed data flow between the various components.*

3.2 Dataset

Amazon product review dataset, used for this research, were collected from the Amazon webpages and were made publicly available [36][16]. The dataset consists of approximately 35 million reviews spanning from May 1996 to July 2014. These reviews were categorized into 24 product categories as shown in Table 1.

It includes reviews, ratings, text, helpfulness votes, and product metadata, like descriptions, category information, price, brand, and image features. The dataset is one review per line in JSON (JavaScript Object Notation) format. Each review in the dataset consists of nine fields as shown in Table 2 . This crawled dataset does not have ground truth labels as to whether the review is a spam or not spam.

Category	Number of Reviews
Books	8,898,041
Electronics	1,689,188
Movies and TV	1,697,533
CDs and Vinyl	1,097,592
Clothing, Shoes and Jewelry	278,677
Home and Kitchen	551,682
Kindle Store	982,619
Sports and Outdoors	296,337
Cell Phones and Accessories	194,439
Health and Personal Care	346,355
Toys and Games	167,597
Video Games	231,780
Tools and Home Improvement	134,476
Beauty	198,502
Apps for Android	752,937
Office Products	53,258
Pet Supplies	157,836
Automotive	20,473
Grocery and Gourmet Food	151,254
Patio, Lawn and Garden	13,272
Baby	160,792
Digital Music	64,706

Table 1: *Product categories and the number of reviews in each category.*

Fields	Explanation
reviewerID	ID of the reviewer
asin	ID of the product
reviewerName	name of the reviewer
helpful	helpfulness rating of the review, e.g. 2/3
reviewText	text of the review
overall	rating of the product
summary	summary of the review
unixReviewTime	time of the review (unix time)
reviewTime	time of the review (raw)

Table 2: *Fields in each review.*

An example of the review is given below:

”reviewerID”: ”A1YJEY40YUW4SE”, ”asin”: ”7806397051”, ”reviewerName”: ”An-

drea", "helpful": [3, 4], "reviewText": "Very oily and creamy. Not at all what I expected... ordered this to try to highlight and contour and it just looked awful!!! Plus, took FOREVER to arrive.", "overall": 1.0, "summary": "Don't waste your money", "unixReviewTime": 1391040000, "reviewTime": "01 30, 2014"

Amazon.com is the most successful and largest e-commerce website with a long history. It is large and covers a very wide range of products. Therefore, it is reasonable to consider this dataset as a representative online retailer site [46].

3.3 The Three Methodologies

The system uses the following three methodologies: Detection of Duplicate reviews, Detection of Anomaly in Review Count and Rating Distribution, and Detection of Incentivized reviews to identify the reviews that hamper the credibility of the review rating. They are explained in Section 3.3.1 to 3.3.3.

3.3.1 Detection of Duplicate Reviews

As the Amazon review dataset does not have labeled spam and non-spam ground truth, which pose the main challenge in this analysis, the earlier studies have used the duplicates and the near duplicate (not exact copy) reviews as spam and the other reviews as non-spam [22]. The authors identified and used three types of duplicates: duplicates from different reviewer IDs on the same product, duplicates from the same reviewer ID on different products; and duplicates from different reviewer IDs. This system uses duplicate reviews as one of the methodologies to score the credibility of the product.

Figure 4 shows the steps used in detecting the duplicate reviews. First, each review is converted into a set of 2-gram shingles, which are formed by combining two consecutive words together. The Jaccard similarity is the ratio of the intersection to the union of these 2-gram shingles of the two reviews. In order to have computational efficiency for a very

large dataset, the following optimizations were done in each step. The shingles are mapped to shingle IDs using the CRC32 hash. min-Hash signatures were calculated for each review using the random hash function which prevents from having to explicitly compute random permutations of all of the shingle IDs. Then, the Jaccard similarities are calculated using the min-Hash signature for each review. They are compared by counting the number of components in which the signatures are equal and divide the number of matching components by the signature length to get a similarity value. The inverted index was used to compute similarity faster. The output displays pairs of reviews with similarity greater than a set threshold. In this case, review pairs with a similarity score of at least 70% were chosen as duplicates [22]. The following sections from 3.3.1.1 to 3.3.1.7 explain each of these steps in detail.

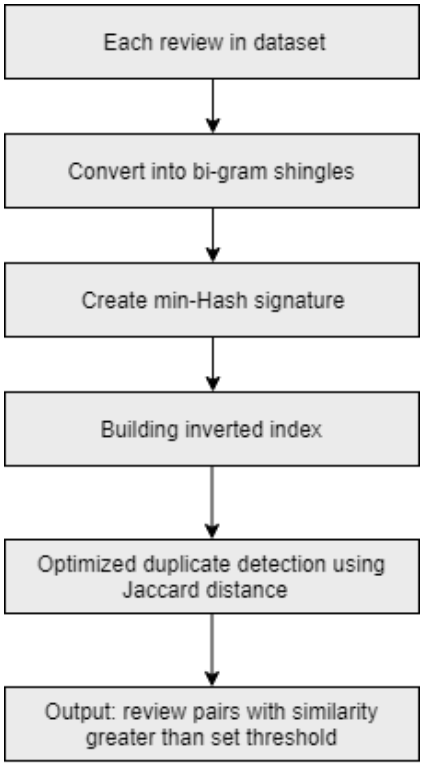

Figure 4: *Overall steps in Detection of Duplicate Reviews.*

3.3.1.1 *Jaccard Similarity*

In general, the similarity is the degree of alikeness between two data objects. In data mining, it is the distance between the object's features. The similarity is measured in the range of 0 to 1. If the similarity is equal or closer to 1, the distance is small and signifies a high degree of similarity between the two objects. If the similarity is equal or closer to 0, the distance is large and signifies a low degree of similarity between the two objects [6]. There are various similarity measures like Euclidean distance, Cosine similarity, Minkowski distance, and Jaccard similarity. For this research, Jaccard similarity was used to calculate the similarity between two texts of the reviews. Other similarity measures are used if the data objects are represented as vectors or points. Jaccard similarity is used for data objects represented as sets (an unordered collection of objects).

Jaccard similarity is defined as the ratio of the cardinality of the intersection of sets over the cardinality of the union of the sets [6]. The resulting score indicates the amount of similarity between the two sets. For any two sets A and B, Jaccard similarity is represented by the formula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

3.3.1.2 *Bi-gram Shingles*

Shingle is a type of tokenizer that constructs n-gram shingles from the given token stream, where n represents the number of contiguous subsequence of tokens to be combined as a single token. In this research, bigram shingles ($n = 2$) are constructed by combining two consecutive words in a review together [23]. Each review in the dataset is represented as a set of bigram shingles. Usage of such bigrams is more meaningful than tokenizing each word as it helps in increasing the relevance between contexts. For example, a review with 10,000 words will have 9,996 bigram shingles.

3.3.1.3 *Substituting Bigram Shingles with Cyclic Redundancy Check (CRC) 32 Hash*

To find the duplicates between two reviews, the most obvious method is to compare each pair of shingles individually. This process is not efficiently scalable for a large dataset. Hence, the shingles are converted into a 32-bit binary sequence by using CRC 32 function [6] [8]. The CRC 32 function converts any variable length string into a hexadecimal value of 32-bit binary sequence. Now, the set for each review is represented as a set of integers instead of substring shingles. Still, the size of the set is large to compute the similarity.

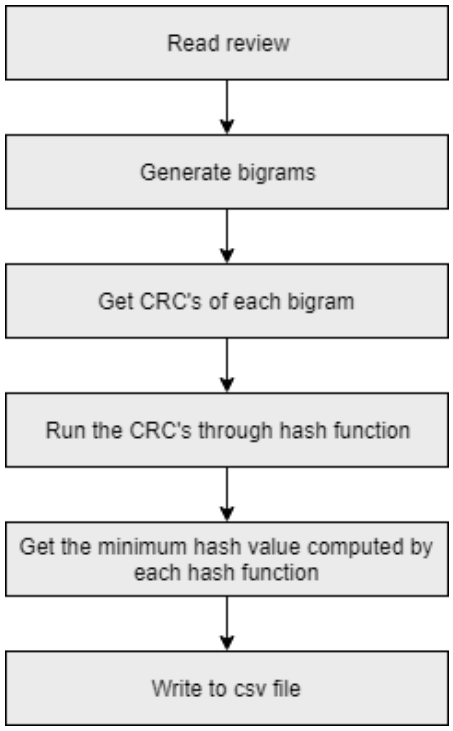

Figure 5: *Steps for min-Hash.*

3.3.1.4 *min-Hash*

Our goal is to have a smaller representation of these large sets called as “signatures” [62]. The signatures of two sets are compared to find the similarity. So, the key property of the signatures is that they should be the good representation of the large set. Signatures for each

set was derived using min-hash scheme with k hash functions, where k is a fixed integer. Figure 5 shows the steps for min-Hash. The value of k was set to 105 for this research. Therefore, 105 numbers of one review are compared with 105 numbers of the other review to calculate the similarity distance. min-Hash signature is used to approximate the Jaccard similarity and aids to scale the computational efficiency for large documents by reducing the number of items in a set to be compared.

3.3.1.5 *Error Calculation*

According to Chernoff Bound [17], the expected error rate for using min-Hash is $O(1/\sqrt{k})$. With an aim to target less than 10% error, this system used $k = 105$ min-Hash signatures. In general, having a $k \approx 100$ leads to a small error probability [63].

3.3.1.6 *Inverted Index*

In the first iteration, min-Hash values of one review were compared with the min-hash values of all other reviews. This process was computationally expensive with the complexity of $O(n^2)$ where n is the number of reviews in the dataset. As an optimization, inverted indexes were built using the min-Hash values. Figure 6 shows the steps for inverted index. This inverted index returned all the products which had a given min-Hash value in $O(logn)$ time. This reduced the time complexity from $O(n^2)$ to $O(nlogn)$.

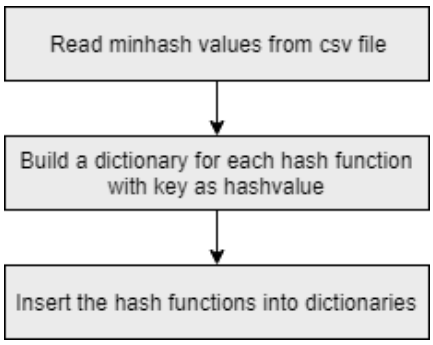

Figure 6: *Steps for inverted index.*

3.3.1.7 *Identifying Duplicates*

In the initial implementation, the efficiency gained by using inverted indexing algorithm was not sufficient for large datasets (example: electronics dataset with more than 1.5 million reviews). On detailed analysis of the runtime performance, we discovered that a large number products had at least one hash matching with the given product. Hence, sorted sets were used for further optimization. Figure 7 shows the steps using sorted set. The key of the sorted set was the product asin and the value was the number of hashes matched. This sorted set was populated while iterating through the inverted index. The sorted set was in the descending order of matches. While iterating the sorted set, for the given product ID, if the Jaccard distance was lesser than the threshold, the loop was terminated as the remaining products are guaranteed to have lesser similarity. This optimization provided significant efficiency speed up.

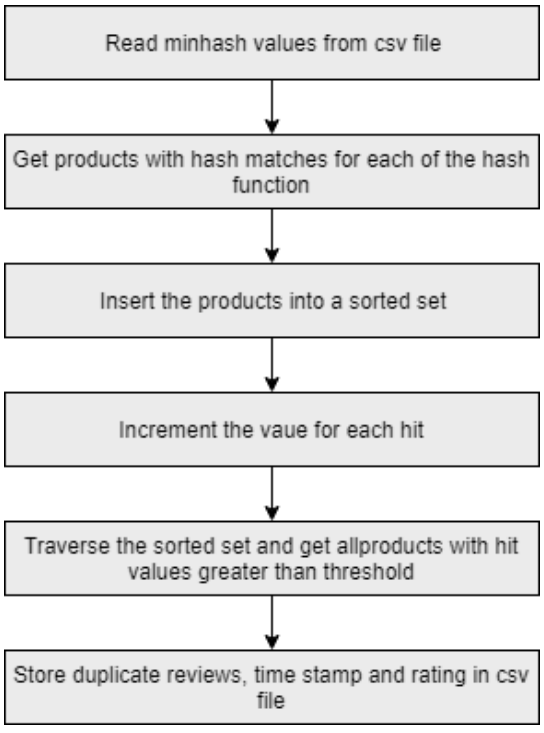

Figure 7: *Identifying duplicates using sorted set.*

3.3.2 Detection of Anomaly in Review Count and Rating Distribution

In a review spam detecting system, identifying the burst patterns of reviewing a product over time is a significant evidence of spammer attacks. Generally, a product is expected to get reviews and ratings progressively over the period and they appear at random time intervals. But, when spammers are hired to write fake reviews, there is a swift increase in the number of reviews in a short interval. The spammers may increase or decrease the rating value of the targeted product in those short time periods and mostly offer extreme ratings. Literature studies show that there is a high correlation between such sudden spikes in the number of reviews and star rating of the product with spammer attacks [66] [33] [3] [14] [20]. Such spikes can be flagged as anomalies in the distribution.

In contrast, such anomalies in reviews and ratings are also applicable for highly seasonal products or due to the unexpected popularity of the product. For example, a sun hat will mostly get its reviews during summer. Hence, detection of anomalies in the presence of seasonality, and an underlying trend is significant [19].

There are two types of anomalies [19]:

(a) Global anomalies: which are the most familiar out of the usual range anomalies.

(b) Local anomalies: which are the underlying trend in the data. For example, in a wave of high activity in the day and low activity in the night, a high activity in the night reports an anomaly.

In this system, Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD) was used to identify the anomalies in review count distribution over time and average review rating over time [19] [25]. S-H-ESD detects both global and local anomalies in the presence of seasonality and growth. This algorithm is computationally fast and efficient for processing large datasets. Figure 8 shows the different examples of anomalies detected by the algorithm and

the types of behaviors accepted as non-anomalies [61]. It detects a sudden increase, abnormal pick, and unusually high activity. It does not detect the linear growth and the linear seasonal growth where the product must have gained its popularity over time. This proves the applicability of this algorithm for this system using Amazon review dataset.

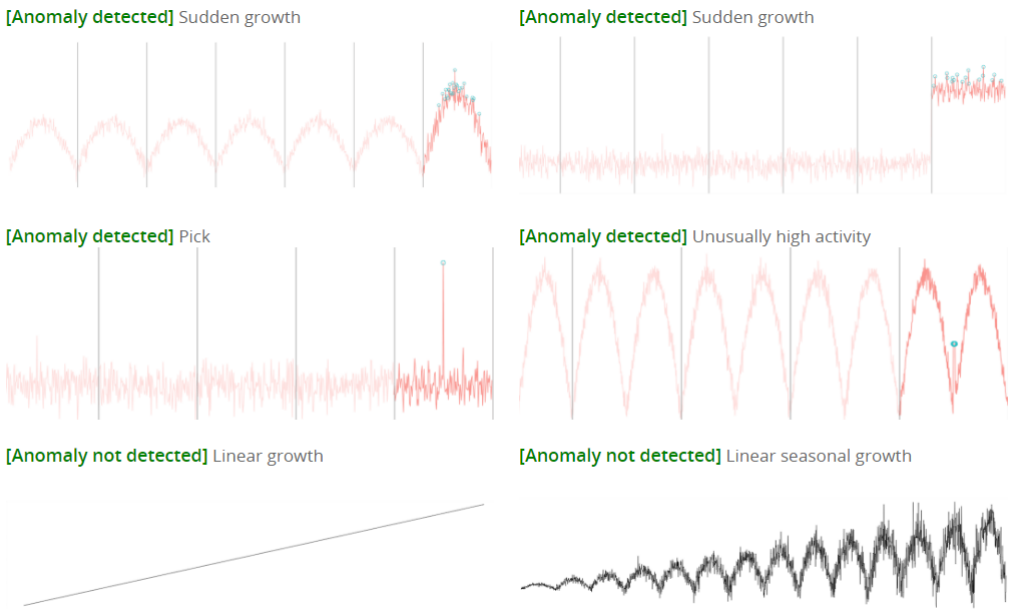

Figure 8: *Graph showing the types of anomalies detected and not detected.*

Figure 9 shows the overall steps in detection of anomaly in review count and rating distribution methodology. First, a time series distribution was built for review count for all the reviews of a product. These reviews were binned into 30-day buckets to generate the time series. Next, using the time series, a pandas dataframe was developed which is a two-dimensional tabular data structure with review count and time series. These dataframes form a seasonal univariate time series which were passed into pyculiarity module which implements the S-H-ESD algorithm [52]. Pyculiarity is a Python port of Twitter's AnomalyDetection R Package [55] [49]. Finally, this module returns a dataframe containing the anomalous timestamps and values. The above methods are repeated for review rating anomaly detection.

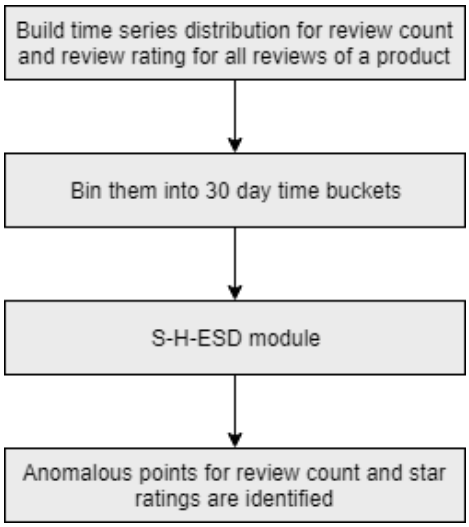

Figure 9: *Overall steps in detection of anomaly in review count and rating distribution.*

3.3.3 Detection of Incentivized Reviews

Amazon.com allowed the sellers to offer products for free or at high discounts in exchange for positive reviews about that product. While most of such reviews have a disclaimer that *"the customer received this product in exchange for an honest review"* stated in the reviews, there still exist many reviews written under this condition without including the disclaimer. The [51] talks about how the incentivized reviews have affected the ratings of the product. Incentivized reviewers, though they claim to be unbiased, tend to give positive and less critical reviews for the products compared with the non-incentivized reviewers. The occurrence of these reviews has transformed the review panels into advertising forum. Detecting such incentivized or biased reviews is often more challenging.

A few examples of incentivized reviews are: *"I got these at no charge in exchange for an honest review; I received this product in exchange for a truthful review and I must say that I am overall satisfied with it; I received this product at a discounted rate in exchange for my fair and honest."*

Figure 10 shows the overall steps in the detection of incentivized reviews. This problem

was addressed by building a collection of synonym phrases for a set of key phrases derived from the above examples using Natural Language Toolkit – WordNet [65] [43]. For the key phrase "honest review", the equivalent phrases this modules search for are "truthful review", genuine review", "genuine feedback", and so on. The dictionary was made of both single and double paired words, for example, "discount" and "no charge". The reviews with these synonyms were identified using the regular expression and the time intervals were also captured for analysis.

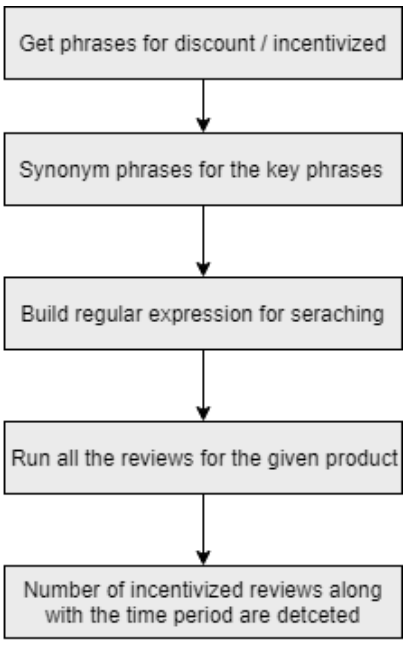

Figure 10: *Overall steps in detection of incentivized reviews.*

WordNet, an open source license, is a large lexical database which resembles the thesaurus [65]. It groups words together based on their meanings and the specific sense of the words into sets of cognitive synonyms (synsets). It was used in this system rather than creating our own dictionary as WordNet is recognized universally [5] for generating the synonyms that are found in close proximity to one another in the network. It labels the semantic relations among words and is used widely in computational linguistics and natural language processing.

3.4 Generation of Credibility Score

A data-driven approach was followed to score the credibility of the product. Depending on the score, a color-coded result is displayed for easy visualization. A scoring scale was created for each product category. Then the product's results were analyzed for credibility with respect to that scoring scale.

3.4.1 Scoring Scale for the Dataset

For each given product category dataset, the total number of reviews were found; and the total number of duplicate reviews, incentivized reviews, rating value anomalies, and review count anomalies were calculated and stored. This provides the ratio of the duplicate reviews, incentivized reviews, and anomalies present in the entire dataset of that product category. Below formulas show the calculation of the scoring scale for duplicate reviews and incentivized reviews for an entire product category dataset.

$$DuplicateReviews = \frac{TotalNumberOfDuplicateReviewsDetected_{(ProductCategory)}}{TotalNumberOfReviews_{(ProductCategory)}} * 100$$

$$IncentivizedReviews = \frac{TotalNumberOfIncentivizedReviewsDetected_{(ProductCategory)}}{TotalNumberOfReviews_{(ProductCategory)}} * 100$$

$$ReviewCountAnomaly = \frac{TotalNumberOfReviewCountAnomaliesDetected_{(ProductCategory)}}{TotalNumberOfProducts_{(ProductCategory)}}$$

$$RatingDistributionAnomaly = \frac{TotalNumberOfRatingAnomaliesDetected_{(ProductCategory)}}{TotalNumberOfProducts_{(ProductCategory)}}$$

3.4.2 Scoring for the Product

For each product analyzed under that product category, the duplicates and the incentivized review ratio were calculated. Below formulas show the calculation of the scoring scale for duplicate reviews and incentivized reviews for a product in the dataset.

$$DuplicateReviews = \frac{TotalNumberOfDuplicateReviewsDetected_{(ProductAsin)}}{TotalNumberOfReviews_{(ProductAsin)}} * 100$$

$$IncentivizedReviews = \frac{TotalNumberOfIncentivizedReviewsDetected_{(ProductAsin)}}{TotalNumberOfReviews_{(ProductAsin)}} * 100$$

If it is less than the total dataset ratio, then the product's reviews could be trusted: else otherwise. The table 3 shows the color scoring scheme for the presence of duplicates and incentivized reviews.

ProductRatio / DatasetRatio	Color Coding	Description
0 to 0.5	Green	Nothing of concern as the ratio is much lower than average
0.5 to 1.1	Orange	Some concern as the ratio is around the average
Greater than 1.1	Red	Highly Concerning as the ratio is above the average

Table 3: *Scoring scheme for duplicate and incentivized reviews.*

The occurrence of an anomaly in the number of reviews and the rating distribution con-

tributes to the suspicion of the credibility of the reviews. The table 4 shows the scoring scheme for the presence of the anomaly.

Number of Anomalies	Color Coding	Description
0	Green	Nothing of concern as no anomalous pattern detected
1	Orange	Some concern as an anomalous pattern is detected
2 and more	Red	Highly Concerning as multiple anomalous patterns are detected

Table 4: *Scoring scheme for number of anomalies.*

For example, if the entire dataset has 1,00,000 reviews, out of which 1000 were duplicates, then the average duplicate review ratio for the dataset is 1%. If a product in that dataset has about 500 reviews of which 10 were found to be duplicates, then the product's duplicate review ratio is 2%. This is twice than the entire dataset's duplicate review ratio, hence it is marked red and product's review credibility score is lowered.

3.4.3 Credibility Score

The final score for the credibility of the product was generated by calculating the average of the above scores. All the methods are given equal weight-age for this calculation. The table 5 shows the final scoring scheme. For example, if the color code for duplicate reviews was green, incentivized reviews was orange and the number of anomalies was green, then the total score is green – no concerns about the reviews of the product. Figure 11 shows three color-coded smileys used in this system.

Color Results from Methods	Color Coding with Emoji	Description
2 or more Green and no Red	Green Smiley	Nothing of concern
2 or more Red	Red Smiley	Highly concerning
For all the other combinations	Orange Smiley	Some concern

Table 5: *Scoring scheme for credibility of the product reviews.*

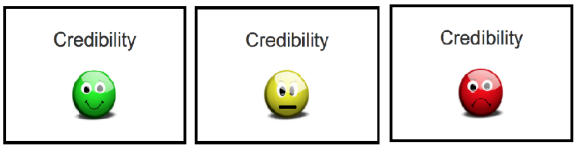

Figure 11: *The three color coded smileys.*

3.5 User Interface Design

The user interface for this system was developed using Tkinter toolkit [59]. It is the Python's most commonly used de-facto standard Graphical User Interface (GUI) package. Tkinter provides various controls used in GUI application. These controls are called as widgets. A number of widgets were used in this system like the frame widget, list box radio button, scrollbar, and progress bar.

This systems user interface consists of 3 screens. Figure 12 shows the user interface flow. The first 2 screens get the user input and the third screen displays an HTML report through the default web browser installed on the computer.

Figure 12: *The user interface flow.*

3.5.1 First Page of User Interface

Figure 13 is the screenshot of the first screen of the user interface. Here, the user selects the product category of the dataset that the user wants to load and analyze. All the product categories are mentioned with the number of reviews in parenthesis. The loading time differs depending on the size of the dataset. During the process of loading the dataset, the progress bar keeps the user informed of the amount of loading process completed. Figure 14 shows the progress bar loading the electronics dataset. Once the dataset is loaded, it moves to the

second screen automatically.

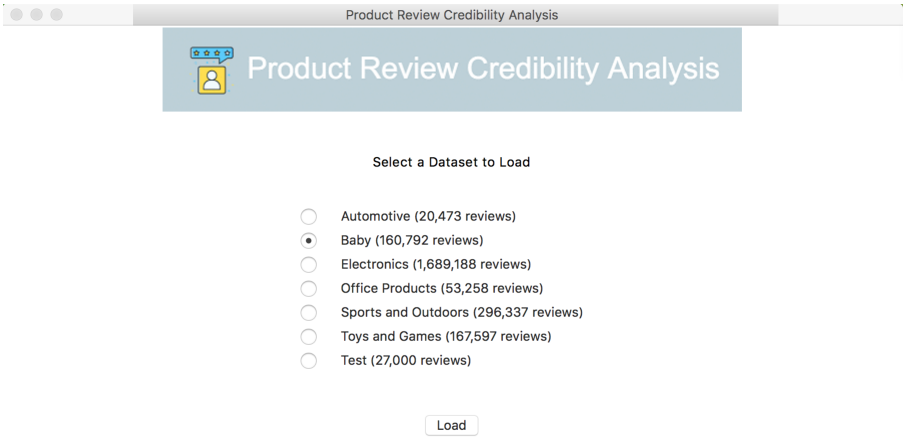

Figure 13: *The first page of user interface.*

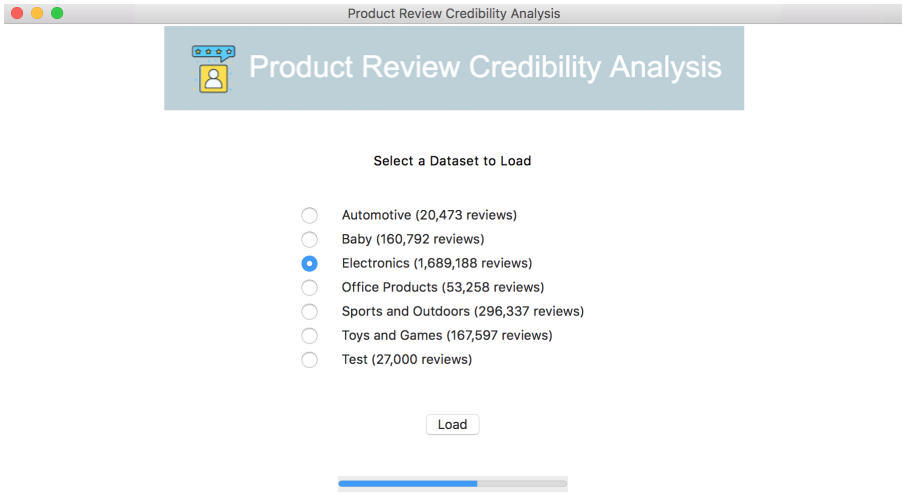

Figure 14: *The progress bar showing the loading progress.*

3.5.2 Second Page of User Interface

Figure 15 is the screenshot of the second screen of the user interface. As seen in the figure, there is a list box with all the product ID's (asin) in that dataset. The asin's are sorted in descending order of the review count. The user can start typing the asin in the text box and the asin's in the list start to get filtered based on the characters entered. Figure 16 shows the filtered asins' in the list according to the asin entered. Once the user has chosen or entered the complete asin, by clicking the generate report button, all the three methodologies are initialized to calculate the values for that product. This produces the HTML code for the report.

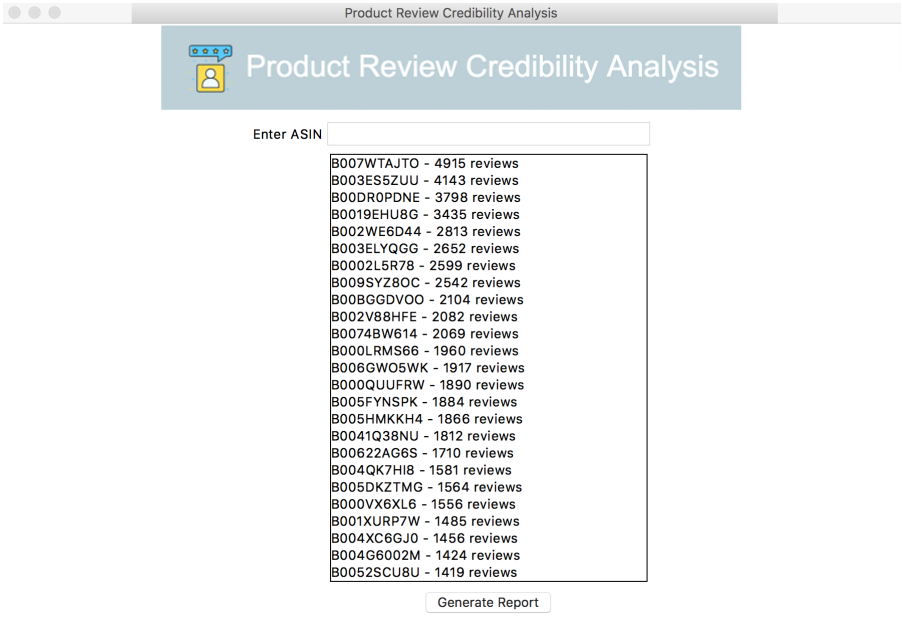

Figure 15: *The second page of user interface.*

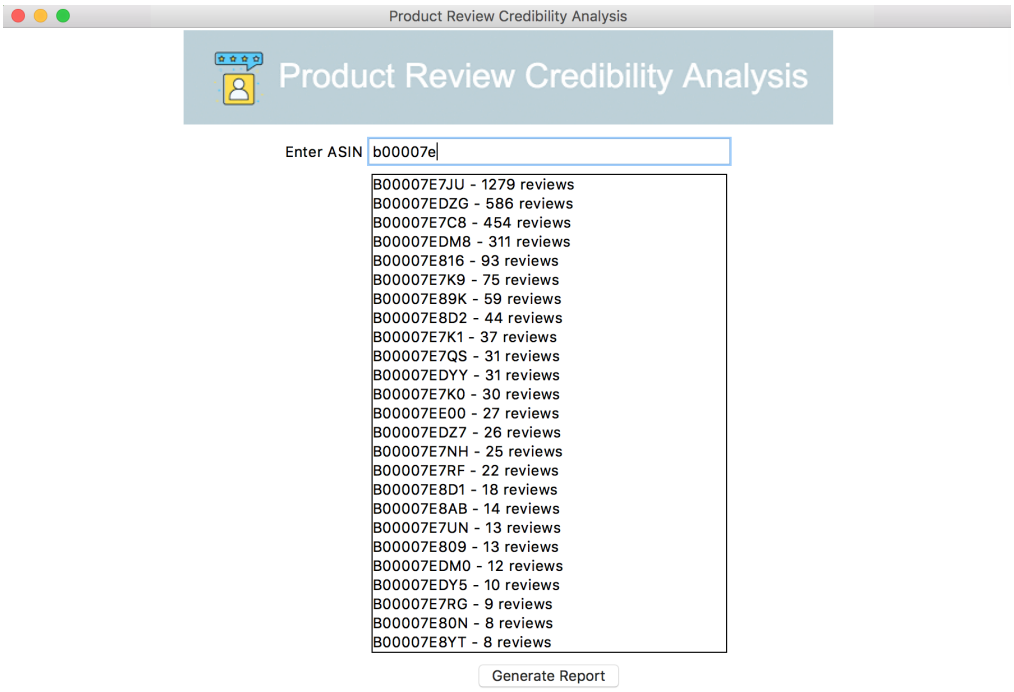

Figure 16: *The filtering of asin.*

3.5.3 Third Page of User Interface

Figure 17 is the screenshot of the third screen of the user interface. The HTML code consists of the following three segments: First, it shows the results and the credibility score. It displays the results of all the methods for the selected product which are color coded in accordance with the general score. The general score for the complete dataset as per the data-driven approach explained in the above Section 3.4 are displayed as a reference in the tabular column. The number of reviews and the time range are also displayed for this product. The credibility score is the final score indicating the trustworthiness of the reviews. Second, the "Open in Amazon" button opens the Amzon.com page for that product. This was done by adding the asin to the string "http://www.amazon.com.dp/". Third, the Matlab plots were converted to HTML by using mpld3 [50]. Mpld3 project provides simple API's to export all the Matlab graphics to HTML code which in turn can be used within the browser. These plots

HTML code is combined with the previous segments and written to an HTML file. Then the browser is invoked to render the HTML page.

Many websites use Python on the server side. In this system, the view is a webpage. The controlled could be made a webpage in future while the model remains the same.

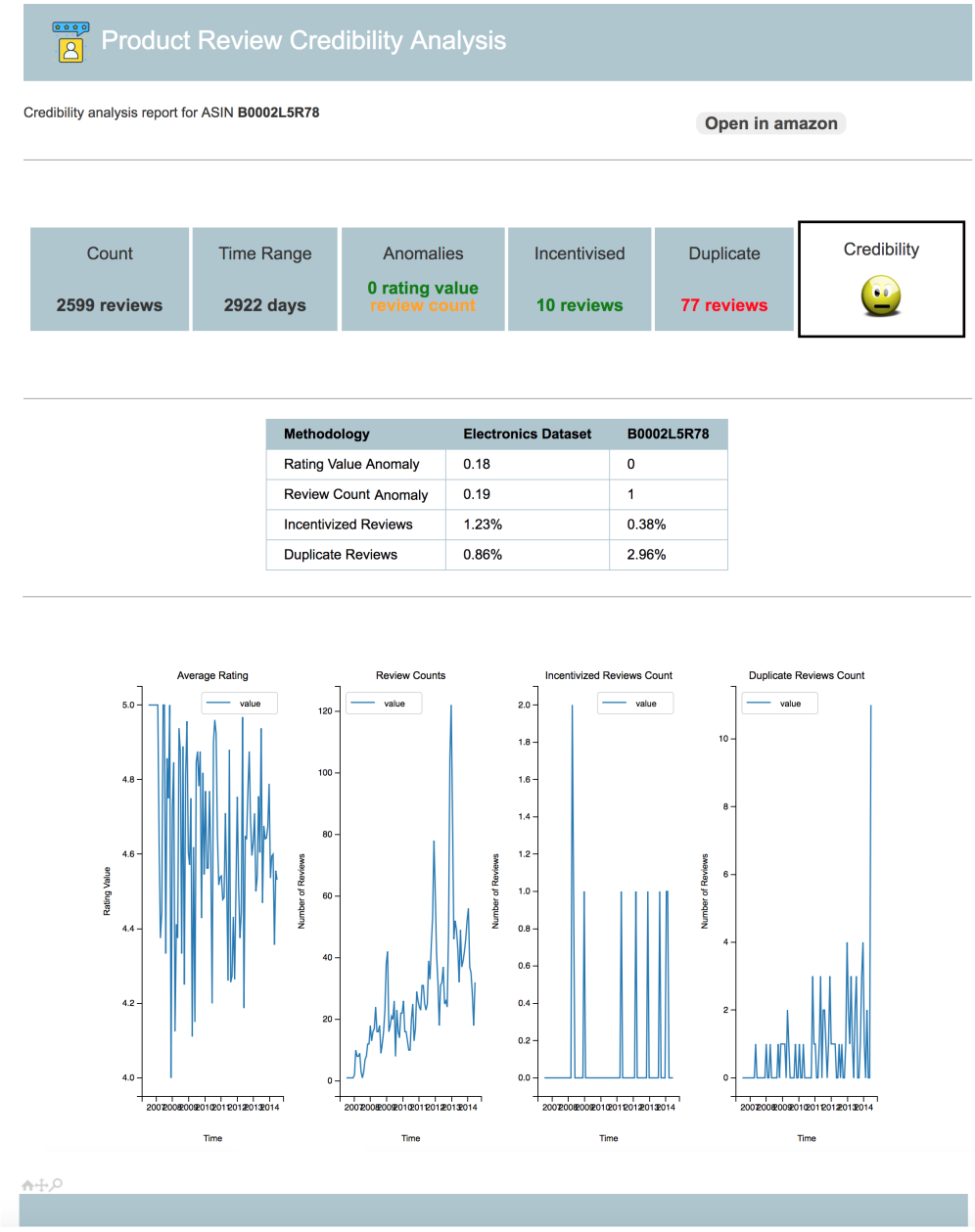

Figure 17: *The third page of user interface.*

3.6 Testing Methodology

Since there is no ground truth for the Amazon review dataset, manual validation of the results was done to check the level of accuracy by performing separate tests for the three methodologies. The results were sampled and analyzed for credibility by human evaluation. The actual accuracy rate was calculated from the sample validation generated by human surveys. The sections 3.6.1 to 3.6.3 explain the testing for the three methodologies used and section 3.6.4 shows the scalability performance testing for this system.

3.6.1 Unit Testing for Detection of Duplicate Reviews

A small test dataset was created by taking 200 reviews from the main dataset. The duplicates and the near-duplicates were synthetically created for unit testing. About 10 random reviews from the test dataset were copied and inserted back into the test dataset as duplicates. About 5 random reviews from the test dataset were slightly modified and inserted back into it to test the near duplicates. The detection of these 15 reviews was manually checked to evaluate the accuracy of the duplicate detection method.

3.6.2 Testing for Anomalies in Review Count and Rating Distribution

The output graphs were visually inspected for bursts in the number of reviews and in the rating distribution. The time period of the unusual decrease or increase in the value were noted and cross-checked with the reviews across the time period in the main dataset.

3.6.3 Test for Detection of Incentivized Reviews

A two-step approach was used to test the incentivized review detection. First, the synonym phrases generated by the WordNet-Natural Language Tool Kit were analyzed to check whether they were corresponding to the context of incentivized or discounted. Second, the

flagged output reviews from the method results were manually checked for the presence of those phrases in the review text.

3.6.4 Scale Testing

Datasets with various sizes were run through the system to check the scalability. The electronics product category dataset has more than 1.5 million reviews which are larger than most of the datasets used in the literature.

4 Results

This chapter presents the evaluation of the methodologies and the generation of the credibility score. The section 4.1 presents the experimental setup. The section 4.2 analyses the correctness of the output from each algorithm and explores the dataset.

4.1 Experimental Setup

To evaluate the proposed methods, 6 product categories from Amazon review dataset, shown in table 6, was used as our experiment data. The table 6 shows the number of reviews and the number of products under each product category.The test machine used was MacBook Pro with 2.7GHz Intel core i5 processor and 8 Giga bytes of RAM. The python code was converted into an OS X application using the py2app package and executed in the test machine.

Product Category	Number of Reviews	Number of Products
Automotive	20,473	1,834
Baby	160,792	7,049
Electronics	1,689,188	63,000
Office Products	53,258	2,419
Sports and Outdoors	296,337	18,356
Toys and Games	167,597	11,923

Table 6: *The 6 product categories used for this experiment.*

4.2 Experimental Results

The correctness and the accuracy of the outputs from each method are presented in section 4.2.1 to 4.2.3. All the manual text evaluation was majorly done on Automotive dataset as it had a smaller number of reviews. The section 4.2.4 examines the efficiency and section 4.2.5 analyzes and concludes the effectiveness of using the combination of methodologies in identifying the credibility score.

4.2.1 Detection of Duplicate Reviews

The output from duplicate detection method was stored in a CSV file separately for each product category. The column headings for this CSV file are product asin, review rating, unixtime, product asin, review rating, unixtime, and similarity score. In order to validate the results of the output, an example of the duplicate detected is presented here. Figure 18 is the screenshot from the output CSV file for the automotive product category.

365	B0014I820Q	4	1368403200	B004QQLR4M	4	1368403200	0.68571429
366	B00155237W	5	1365984000	B00063X7KG	5	1365984000	0.85714286
367	B00155237W	5	1365984000	B002XOXSI2	5	1365984000	0.83809524
368	B0015KROXU	2	1309046400	B00008BKX5	5	1313020800	1

Figure 18: *Screenshot of the output CSV file of Duplicate Detection.*

The asin's were cross referred in the main dataset for the review text. The complete review for the three product asins' are given below.

"reviewerID": "ANKCQ60FES3EZ", "asin": "B00155237W", "reviewerName": "Mack Wu", "helpful": [0, 0], "reviewText": "It is easy to installed. It makes my car look really nice and fun. I love it and recommend it", "overall": 5.0, "summary": "Looks great", "unixReviewTime": 1365984000, "reviewTime": "04 15, 2013"

"reviewerID": "ANKCQ60FES3EZ", "asin": "B00063X7KG", "reviewerName": "Mack Wu", "helpful": [0, 1], "reviewText": "It is easy to apply. It makes my car look really nice and fun. I love it and recommend it", "overall": 5.0, "summary": "Great result", "unixReviewTime": 1365984000, "reviewTime": "04 15, 2013"

"reviewerID": "ANKCQ60FES3EZ", "asin": "B002XOXSI2", "reviewerName": "Mack Wu", "helpful": [1, 1], "reviewText": "It is easy to Use. It makes my car look really nice and fun. I love it and recommend it", "overall": 5.0, "summary": "Great result", "unixReviewTime": 1365984000, "reviewTime": "04 15, 2013"

The product asin was used to find the product description from Amazon.com.

B00155237W : *Cruiser Accessories 76200 Tuf Flat Shield Novelty / License Plate Shield*

B00063X7KG : *Meguiar's G1016 Smooth Surface Clay Kit*

B002XOXSI2 : *Meguiar's G110V2 Professional Dual Action Polisher*

From the above, the reviewer "Mack Wu" has written similar reviews to all the three different types of products and at the same time (reviewTime: 04 15, 2013) which leads to suspicion. This is in line with the observation made by [22].

4.2.2 Detection of Anomaly in Review Count and Rating Distribution

The S-H-ESD algorithm has effectively determined most of the anomalies in the product review distribution and review count over the time period. The graphs are the examples of few products from Electronics dataset. The red circles indicate the anomalies detected by the algorithm. Figures 19 to 22 are the examples of the anomalies detected in rating distribution and review count.

In figure 19, the algorithm has detected an anomaly only in the review count. There is an unusually high number of reviews for the product during the year 2013, while the rating distribution has been between 4 and 5 stars throughout the review time period. The increase in the review count helps to increase the product ranking. This spike in the review count is swift and short-lived which marks it as a suspicion for spam activity.

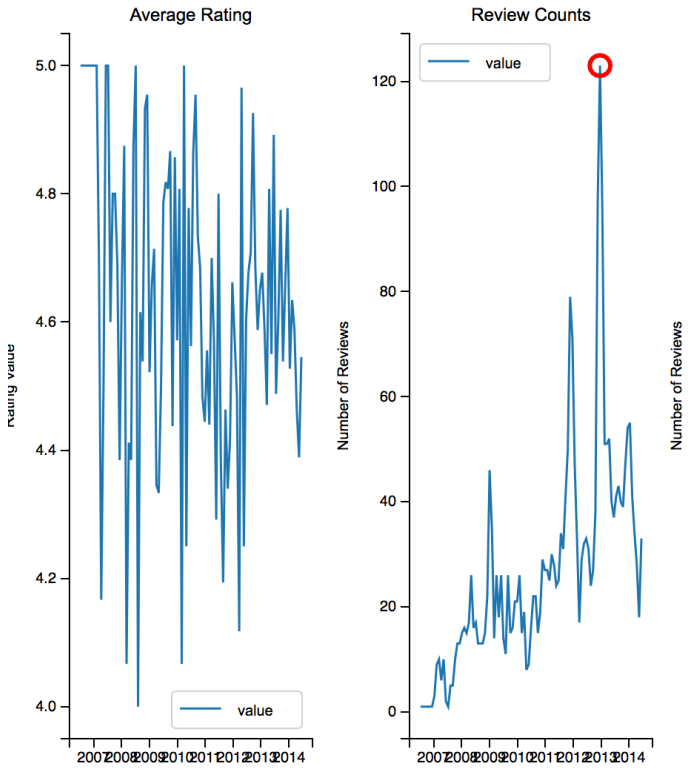

Figure 19: *Example #1 graph showing the anomaly detection.*

In figure 20, the algorithm has detected 3 anomalies in review count and 4 anomalies in rating distribution. The graph illustrates that when the review counts are high, the product had better ratings. In other words, genuine reviews had a lower rating than the potentially fake reviews that were registered during the spikes. This attests that it is significant to look for the anomaly in both the directions.

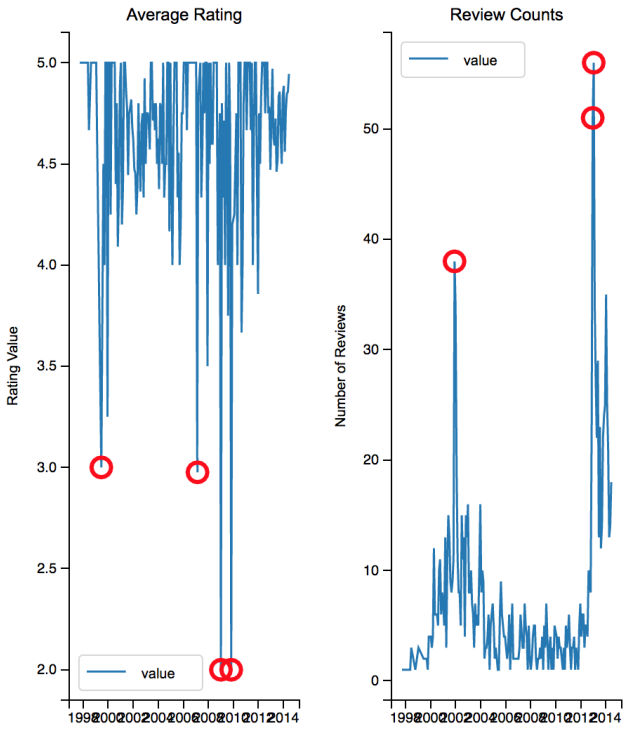

Figure 20: *Example #2 graph showing the anomaly detection.*

In figure 21, the rating values have started low and over time it has increased considerably and there is a sudden spike in the review count. This could be a spam attack where fake reviews with high average rating were written to inflate the overall rating.

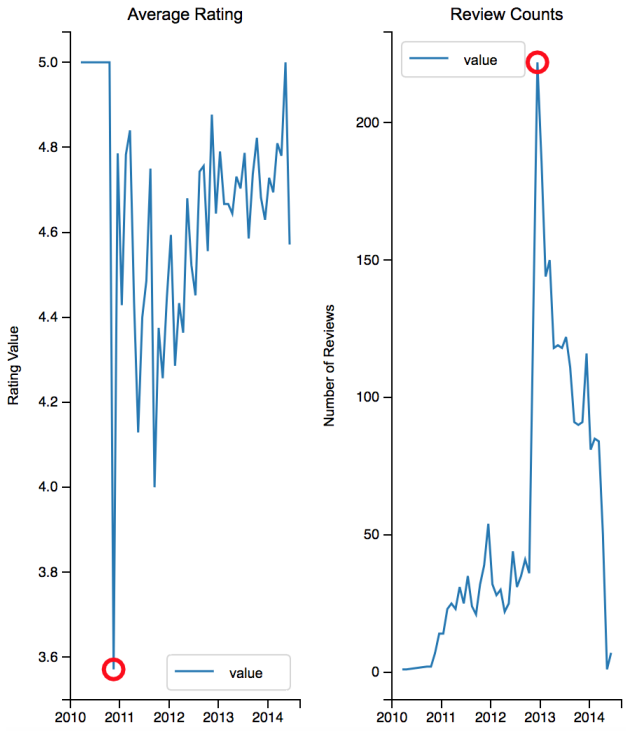

Figure 21: *Example #3 graph showing the anomaly detection.*

In figure 22, the algorithm has not detected any anomaly for the product. The rating distribution and the review counts are almost evenly distributed throughout the time period. It is highly unlikely to be spam as there is a consistent fluctuation and spam attacks are mostly not performed over many years with stable fluttering.

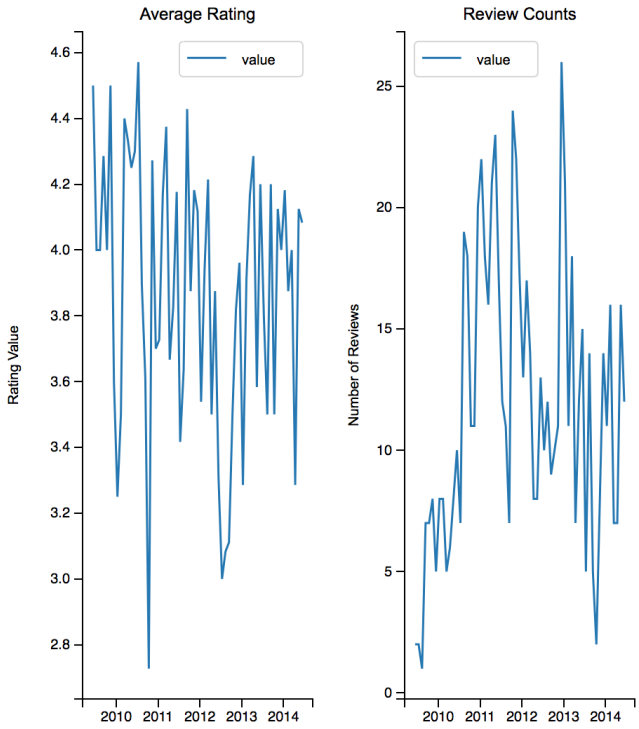

Figure 22: *Example #4 graph showing the anomaly detection.*

4.2.3 Detection of Incentivized Reviews

The main purpose of this detection is to identify whether a product has a majority of the reviews when it was at a discounted price or given free. As mentioned in the literature, these reviews were long and most of them were positive reviews with a high star rating.

Here are a few review examples detected by the algorithm. (For saving space, the long reviews are shortened to highlight the incentivized phrases)

"I've received this as a 'for review' unit but the seller did not ask or even imply that a 'positive' review was expected in exchange.It worked well for ..... I will know EXACTLY what to tell the technicians."

"Following the directions in the accompanying manual, I was able to use ..... this scan tool is a reasonably-priced way to become informed.I was provided a free sample of this device

in exchange for an honest review, which you have just read.

"I have been using this leather cleaner ..... A top notch leather cleaner - it removed some water stains on our leather couch and added moisture - it worked very well.5 StarsPlease Note I received this product in exchange for an unbiased review"

"I tested Leather Nova against Lexol using ..... If wet, go with Leather Nova. If flat, go with Lexol.I received a complimentary Leather Nova sample and this is my honest review."

"Installed on 2013 Elantra GT (hatchback) with OEM ..... Customer service sent me a relay harness for free (in exchange for good review)..... "

The use of WordNet natural language processing toolkit has helped to detect all the different phrases in context with incentivized and discounted.

In the present-day, most of the businesses offer free shipping. Conversely, in the older reviews, free shipping has attracted customers to purchase a product. The context of "free shipping" mentioned in the reviews has caused identifying those reviews as incentivized reviews. Below is the example of the review which highlights free shipping rather than the discounted product price.

"Shipped in perfect condition and what a great product and a great price... Amazon.com is like finding Coupons all over the world at the terms you choose. I save, time, money and the conveyance of getting it at home. Sometimes receiving it at home is the biggest coupon, saves me time. Next big savings, super saver, Free Shipping!"

"shipping was free with amazon prime"

4.2.4 Efficiency

The system developed performs live analysis for a product after clicking the "Generate Report" button for the methods: detection of anomaly in review count, anomaly in rating distribution, and detection of incentivized reviews. The results are not pre-generated and stored

in a database. All the products were analyzed within a second in real-time even if they had hundreds of reviews. This makes the system usable for online website deployment.

The code base was slightly modified to generate a report (CSV file) for calculating the scale for the data-driven approach. The system was able to generate the report for all the product categories (more than 2 million reviews) within an hour. It is well adapted to large datasets and also could be used for other datasets with little changes to the feature names. A MacBook Pro is able to analyze millions of reviews in an hour, the entire Amazon dataset could be analyzed with limited computing resources.

4.2.5 Credibility Score

The credibility scoring scale was calculated as explained in section 3.4.The percentage scale for duplicate reviews was calculated by the ratio of the total number of duplicate reviews to the total number of reviews in that product category. Similarly, it was calculated for incentivized reviews. The scale for anomalies in the review count was calculated by the ratio of the number of anomalies in the review count to the number of products in the dataset. Similarly, it was calculated for anomalies in rating distribution.

Each product category in the Amazon dataset has its own adaptive scoring scale. For example, it is highly common in certain product categories, like, books where most of the reviews were from the promotional copy rather than the real customer reviews. There is a very high ratio of reviews for promotional copies which are incentivized reviews. The scale is contextual and adapts based on product categories. Hence, any noise in the results was diminished by this adaptive methodology. The table 7 shows the total number of duplicate reviews, incentivized reviews, and anomalies in review count and rating distribution for each product category dataset. The scoring scale for each dataset was generated from this data.

Category	Automotive	Baby	Electronics	Office Products	Sports& Outdoors	Toys& Games
Duplicate Reviews	268	1491	14501	710	3308	3201
Review Count Anomaly	137	1503	11915	168	2471	1083
Rating Value Anomaly	147	1488	11483	164	2623	1006
Incentivized Reviews	178	1070	20760	1030	1599	1363

Table 7: *Total number of detections in each product category.*

The table 8 shows the scoring scale for all the 6 product categories used in this experiment. For example, the calculation of the scoring scale for the automotive dataset is shown below:

$$ScoringScaleForDuplicateReviews = \frac{268}{20473} * 100 = 1.309\%$$

$$ScoringScaleForIncentivizedReviews = \frac{178}{20473} * 100 = 0.869\%$$

$$ScoringScaleForReviewCountAnomaly = \frac{137}{1834} = 0.074$$

$$ScoringScaleForRatingValueAnomaly = \frac{147}{1834} = 0.080$$

Category	Automotive	Baby	Electronics	OfficeProdu	Sports& Outdoors	Toys& Games
Duplicate Reviews	1.31%	0.93%	0.86%	1.33%	1.12%	1.91%
Review Count Anomaly	0.07	0.21	0.19	0.07	0.13	0.08
Rating Value Anomaly	0.08	0.21	0.18	0.06	0.14	0.08
Incentivized Reviews	0.87%	0.67%	1.23%	1.93%	0.54%	0.81%

Table 8: *Credibility scoring scale for the six datasets.*

Figure 23 is an example of the credibility report for the product asin B007WTAJTO in Electronics dataset. It has 4915 reviews in a time period of 774 days. The system has detected 1 anomaly in rating value and 1 anomaly in review count. The presence of an anomaly refers to orange color code as the section 3.4. Hence, the anomalies were represented in orange color. The product has 17 incentivized reviews. The incentivized review scoring for this product is $17/4915 * 100 = 0.345\%$ which are in the range of 0 to 0.5% compared to the incentivized scoring scale of the electronics dataset (1.23%). Hence, the incentivized reviews were represented in green color. The system detected 73 duplicate reviews. The duplicate review scoring for this product is $73/4915 * 100 = 1.485$ which are greater than 1.1% of the duplicate scoring scale of the electronics dataset (0.86%). Hence, the duplicate reviews were represented in red color. Combining all the above, 2 oranges, 1 green, and 1 red, the system illustrates that there is some concern with the reviews of this product and presents an orange smiley as the credibility score.

Figure 24 show an example of a red and green credibility score respectively.

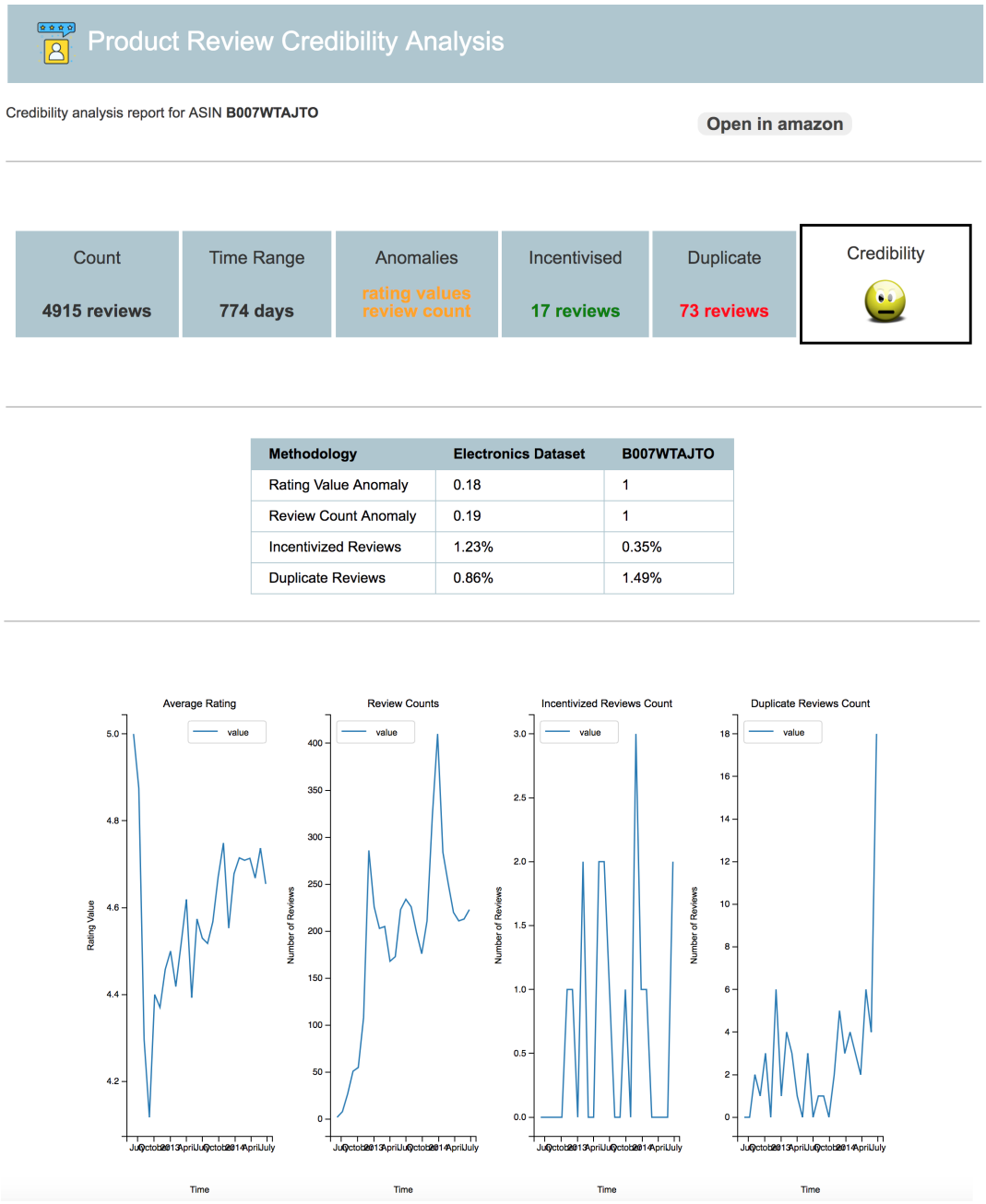

Figure 23: *Example of the final credibility score report for a product.*

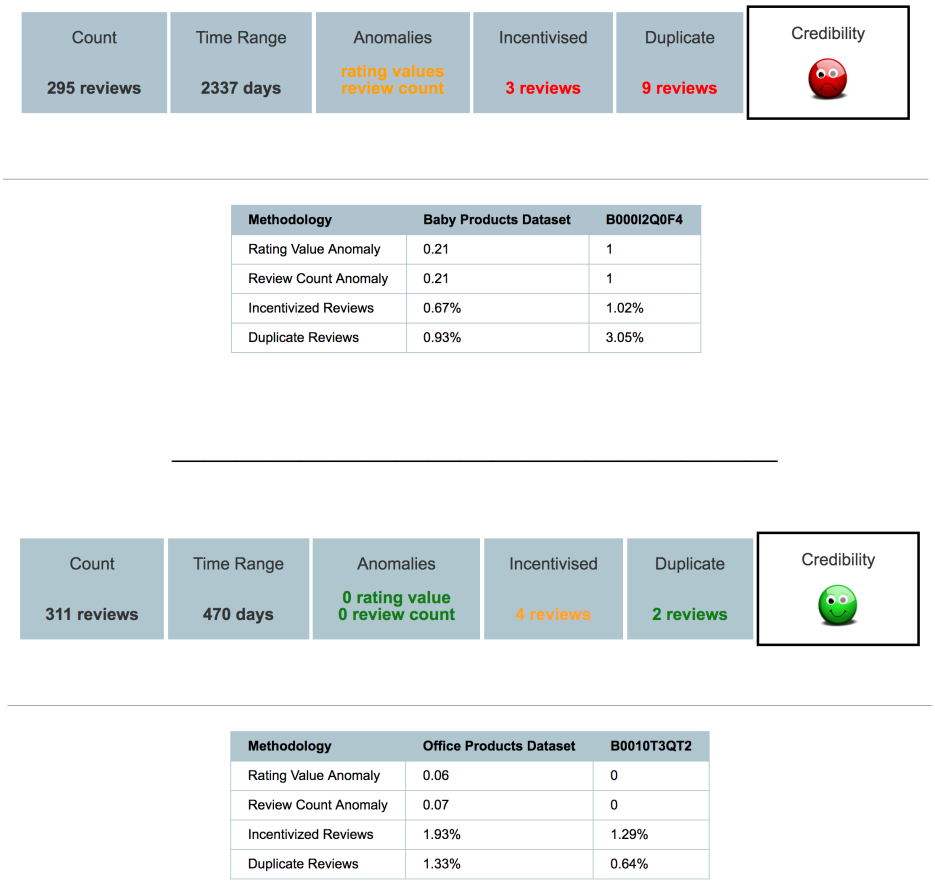

Figure 24: *Red and Green credibility score example.*

The system uses the results from the three methodologies and has combined them to get a better insight into a product rather than depending on a single methodology. Pertaining to detection metrics, as the results illustrate, these metrics are quite significant and well defined in the literature comparing to the variety of other metrics, like, including the length of review, the position of review, brand name frequency and so on. Therefore, incorporating them in fake detection systems would be beneficial. The results show that focusing merely on one metric does not reveal high accurate results compared to the combination of them that enhances the throughput of the detection system significantly. The reviews that have escaped one type of detection method may have been caught by the other detection methods boosts the effectiveness of this system.

5 Discussion

The system developed is a multi-dimensional analysis for detecting the credibility of a product's reviews that is practical and deployable. From the previous section, it shows that 2,387,645 reviews belonging to 104,581 products across 6 product categories were analyzed by this system. This number is much higher than the other projects mentioned in the literature. This system was built upon on multiple ideas from highly cited and widely accepted bodies of research. The objective was to detect the different types of spam reviews which might be missed by using a single detection technique.

The challenge here is that there is no ground truth to validate the results produced by the system. The results section shows the examples of the reviews detected and the implementation of the methods are correct based on the sampling and unit testing. The credibility score of the product for each detection method is based on the following two ratios. First, given the entire product category dataset, the ratio of the detections to the total number of reviews in that dataset was identified. Then, for each product, the ratio of the detections to the number of reviews was identified. For the overall credibility of the given product reviews, the main idea was to identify a fractional structure overlap on which the results from the different methodologies agree. Not all methodologies are in need to agree here. Even if a subset of the methodologies agrees with a high degree of confidence, the credibility score is reduced.

The following observations were made from the results obtained:

(a) Each product category dataset has a wide variation in the number of duplicates, anomalies, and incentivized reviews detected ranging from 0.7% to 2%. Hence, it is essential to have an individual scoring scale for each of the product categories.

(b) Majority of the products were in the green credibility score where all the results were

in green. A few products had red credibility score where all the results were in highly concerning category. Some other products had a combination of green, orange and red results. So, this significant intersection overlap of the results helps the customers to identify the suspicious products with a high confidence.

(c) In addition to the summarized scores, the graphs which display the review count and the rating distribution over time (shown in section 4.2.2) helps the advanced users to visualize the trends, and get a better understanding of discrepancies in reviews of the product.

6 Conclusion and Future Work

In this paper, we presented a complete end-to-end system that uses a combination of three methodologies: detection of duplicate reviews, detection of anomaly in review count and rating distribution, and detection of incentivized reviews to detect the spam reviews and generate a credibility report for the given product. Although there exists much research in this field, the combination of algorithms and developing a practical system for determining the credibility to the best of our knowledge has not been studied in the literature.

The system first identified the duplicates in the reviews using the Jaccard technique. The computational complexity problem for large dataset was solved by using an inverted index for faster comparisons. The anomalies in the review counts and rating distribution over time were detected using S-H-ESD algorithm which allowed seasonal variations and linear growth in popularity. For detecting the incentivized reviews, a dictionary of the related synonyms was created using the WordNet and was compared with the review text. The final credibility score is a data-driven and color-coded result displayed on a web browser. It also displays the analysis report from the methodologies which visually aids the user to gain knowledge about the given product.

The results gathered on Amazon review dataset show the effectiveness of this approach. Some products have credibility concerns flagged by all methods while some others have credibility concerns flagged by only a few methods and a vast majority do not have concerns raised by any of the methods. This shows that all the methods provide useful information, serve as an overlay and aid in the discovery of fake reviews. The system was able to generate the report in real time, within a second for a product in a large dataset.

6.1 Limitations

Similar to the previous approaches in this domain, our approach also suffered from the lack of hundred percent accurately annotated gold-standard dataset. It is an indefinite task to manually identify the spam reviews. Synthetic datasets have given false increment in accuracy and hence was not used in this approach. Another limitation of our work is the use of WordNet for developing synonyms for incentivized review detection which works mainly for the English language. As the internet is getting popular in countries like India and China, natural language processing for local languages is not possible in this system. Finally, nowadays most of the customer reviews include pictures and more research has to be done to identify the fake pictures.

6.2 Threats to validity

The lack of ground truth poses a threat to the correctness of this approach. This system works well for the given dataset and in the context of the current spamming techniques. As the spammers modify their methods of writing fake reviews, this system needs to be improvised in future. For the generalization of this algorithm beyond this dataset, further work is needed to reproduce this case study on other datasets, and it cannot guarantee that the same accuracy will be achieved which concerns the reliability. If the algorithm has written in other sources than Python, it may affect the output accuracy.

6.3 Future Work

To date, the problem of spamming online product reviews is still open to researchers. This paper, however, only represents an initial investigation for the combination of the three algorithms. Much work remains to be done. Every detection approach proposed in the literature suffers from certain drawbacks. In our future work, we will further improve the detection

methods and add new algorithms by studying the impact of key features such as linguistic and relations of spammers, on our approach to enhance its throughput. The system can be developed into a website and deployed. Furthermore, we will update the dataset every month with the new reviews and add them to the database.

For the credibility score, the system uses basic average calculation and gives equal weightage to all the methods. Different statistical approaches could be done to identify a variable weight-age for each method and for each dataset to arrive at altered scores. Moreover, these methodologies require a complete prepared set of reviews to detect the spam ones. We will use Google Image Labeler to create a game and allow the users to label the reviews to create a gold standard dataset. These spam reviews can affect customers and business owners before being detected. Hence, transforming the detection techniques to prediction techniques would be the future direction of this research.

References

[1] Ahmed, I., Ali, R., Guan, D., Lee, Y.-K., Lee, S., and Chung, T. Semi-supervised learning using frequent itemset and ensemble learning for sms classification. *Expert Systems with Applications 42*, 3 (2015), 1065–1073.

[2] Akoglu, L., Chandy, R., and Faloutsos, C. Opinion fraud detection in online reviews by network effects. *ICWSM 13* (2013), 2–11.

[3] Algur, S. P., Patil, A. P., Hiremath, P., and Shivashankar, S. Conceptual level similarity measure based review spam detection. In *Signal and Image Processing (ICSIP), 2010 International Conference on* (2010), IEEE, pp. 416–423.

[4] Banerjee, S., and Chua, A. Y. Applauses in hotel reviews: Genuine or deceptive? In *Science and Information Conference (SAI), 2014* (2014), IEEE, pp. 938–942.

[5] Bird, S., and Loper, E. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (2004), Association for Computational Linguistics, p. 31.

[6] Broder, A. Z. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings* (1997), IEEE, pp. 21–29.

[7] Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., and Al Najada, H. Survey of review spam detection using machine learning techniques. *Journal of Big Data 2*, 1 (2015), 23.

[8] CRC. Crc 32 hash : https://docs. aws. amazon. com/ redshift/ latest/dg/crc32-function.html, url=.

[9] Dwoskin, E., and Timberg, C. *The Washington Post: How merchants use Facebook to flood Amazon with fake reviews*. The Washington Post, 2018.

[10] Fayazi, A., Lee, K., Caverlee, J., and Squicciarini, A. Uncovering crowdsourced manipulation of online reviews. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2015), ACM, pp. 233–242.

[11] Fdez-Glez, J., Ruano-Ordas, D., Méndez, J. R., Fdez-Riverola, F., Laza, R., and Pavón, R. A dynamic model for integrating simple web spam classification techniques. *Expert Systems with Applications 42*, 21 (2015), 7969–7978.

[12] Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., and Ghosh, R. Exploiting burstiness in reviews for review spammer detection. *Icwsm 13* (2013), 175–184.

[13] Feng, S., Banerjee, R., and Choi, Y. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2* (2012), Association for Computational Linguistics, pp. 171–175.

[14] Feng, S., Xing, L., Gogar, A., and Choi, Y. Distributional footprints of deceptive product reviews. *ICWSM 12* (2012), 98–105.

[15] Fusilier, D. H., Montes-y Gómez, M., Rosso, P., and Cabrera, R. G. Detection of opinion spam with character n-grams. In *International Conference on Intelligent Text Processing and Computational Linguistics* (2015), Springer, pp. 285–294.

[16] He, R., and McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web* (2016), International World Wide Web Conferences Steering Committee, pp. 507–517.

[17] Hellman, M., and Raviv, J. Probability of error, equivocation, and the chernoff bound. *IEEE Transactions on Information Theory 16*, 4 (1970), 368–372.

[18] Heydari, A., ali Tavakoli, M., Salim, N., and Heydari, Z. Detection of review spam: A survey. *Expert Systems with Applications 42*, 7 (2015), 3634–3642.

[19] Hochenbaum, J., Vallis, O. S., and Kejariwal, A. Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv:1704.07706* (2017).

[20] Hu, N., Koh, N. S., and Reddy, S. K. Ratings lead you to the product, reviews help you clinch it? the mediating role of online review sentiments on product sales. *Decision support systems 57* (2014), 42–53.

[21] Idris, I., Selamat, A., and Omatu, S. Hybrid email spam detection model with negative selection algorithm and differential evolution. *Engineering Applications of Artificial Intelligence 28* (2014), 97–110.

[22] Jindal, N., and Liu, B. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 1189–1190.

[23] Jindal, N., and Liu, B. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (2008), ACM, pp. 219–230.

[24] Jindal, N., Liu, B., and Lim, E.-P. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (2010), ACM, pp. 1549–1552.

[25] Kejariwal, A. Introducing practical and robust anomaly detection in a time series. *Twitter Engineering Blog. Web 15* (2015).

[26] Kolhe, N., Joshi, M., Jadhav, A., and Abhang, P. Fake reviewer groups' detection system. *Journal of Computer Engineering (IOSR-JCE) 16*, 1 (2014), 06–09.

[27] Kost, A. Woman paid to post five-star google feedback. *ABC7 News* (2012).

[28] Lau, R. Y., Liao, S., Kwok, R. C. W., Xu, K., Xia, Y., and Li, Y. Text mining and probabilistic language modeling for online review spam detecting. *ACM Transactions on Management Information Systems 2*, 4 (2011), 1–30.

[29] Li, F., Huang, M., Yang, Y., and Zhu, X. Learning to identify review spam. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence* (2011), vol. 22, p. 2488.

[30] Li, H., Chen, Z., Liu, B., Wei, X., and Shao, J. Spotting fake reviews via collective positive-unlabeled learning. In *Data Mining (ICDM), 2014 IEEE International Conference on* (2014), IEEE, pp. 899–904.

[31] Li, H., Chen, Z., Mukherjee, A., Liu, B., and Shao, J. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *ICWSM* (2015), pp. 634–637.

[32] Li, J., Ott, M., Cardie, C., and Hovy, E. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2014), vol. 1, pp. 1566–1576.

[33] Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B., and Lauw, H. W. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (2010), ACM, pp. 939–948.

[34] Lin, Y., Zhu, T., Wu, H., Zhang, J., Wang, X., and Zhou, A. Towards online anti-opinion spam: Spotting fake reviews from the review sequence. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on* (2014), IEEE, pp. 261–264.

[35] Ma, W., Tran, D., and Sharma, D. A novel spam email detection system based on negative selection. In *Computer Sciences and Convergence Information Technology, 2009. ICCIT'09. Fourth International Conference on* (2009), IEEE, pp. 987–992.

[36] McAuley, J. Amazon product data : http://jmcauley.ucsd.edu/data/amazon/.

[37] Meyer, D. Fake reviews prompt belkin apology. *CNet News* (2009).

[38] Miller, C. Company settles case of reviews it faked. *New York Times* (2009).

[39] Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., and Ghosh, R. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), ACM, pp. 632–640.

[40] Mukherjee, A., Liu, B., and Glance, N. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 191–200.

[41] Mukherjee, A., Liu, B., Wang, J., Glance, N., and Jindal, N. Detecting group review spam. In *Proceedings of the 20th international conference companion on World wide web* (2011), ACM, pp. 93–94.

[42] Mukherjee, A., Venkataraman, V., Liu, B., and Glance, N. S. What yelp fake review filter might be doing? In *ICWSM* (2013).

[43] NTLK. Natural language toolkit: Wordnet : http://www.nltk.org/_ modules/ nltk /corpus/ reader/ wordnet.html.

[44] Ott, M., Cardie, C., and Hancock, J. Estimating the prevalence of deception in online review communities. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 201–210.

[45] Ott, M., Choi, Y., Cardie, C., and Hancock, J. T. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (2011), Association for Computational Linguistics, pp. 309–319.

[46] Pinch, T., and Kesler, F. How aunt ammy gets her free lunch: A study of the top-thousand customer reviewers at amazon. com. *Unpublished manuscript* (2011).

[47] Popken, B. Ways you can spot fake online reviews. *The Consumerist* (30).

[48] Puranam, D., and Cardie, C. The enrollment effect: A study of amazon's vine program. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media* (2014), pp. 17–27.

[49] Pyculiarity. A python port of twitter's anomalydetection r package : code link : https://pypi.org/project/pyculiarity/.

[50] Python. Python matlab plots in web browser : http://mpld3.github.io/.

[51] ReviewMeta. Reviewmeta : https://reviewmeta.com/blog/analysis-of-7-million-amazon-reviews-customers-who-receive-free-or-discounted-item-much-more-likely-to-write-positive-review/.

[52] S-H-ESD. Anomaly detection with r : code link : https:// github. com/ twitter/ anomalydetection.

[53] Savage, D., Zhang, X., Yu, X., Chou, P., and Wang, Q. Detection of opinion spam based on anomalous rating deviation. *Expert Systems with Applications 42*, 22 (2015), 8650–8657.

[54] Sharma, K., and Lin, K.-I. Review spam detector with rating consistency check. In *Proceedings of the 51st ACM southeast conference* (2013), ACM, p. 34.

[55] Smiller, N. r2py : code link : https://github.com/nicolasmiller/pyculiarity.

[56] Spirin, N., and Han, J. Survey on web spam detection: principles and algorithms. *ACM SIGKDD Explorations Newsletter 13*, 2 (2012), 50–64.

[57] Streitfeld, D. For $2 a star, an online retailer gets 5-star product reviews. *New York Times 26* (2012).

[58] Sun, H., Morales, A., and Yan, X. Synthetic review spamming and defense. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), ACM, pp. 1088–1096.

[59] Tkinter. Graphical user interfaces with tk : https://docs.python.org/3/library/tk.html.

[60] Topping, A. Historian orlando figes agrees to pay damages for fake reviews. *The Guardian 16* (2010).

[61] Twitter. Anomaly detection with twitter graph : https://anomaly.io/anomaly-detection-twitter-r/.

[62] Ullman, J. *Jaccard Similarity - Stanford InfoLab : http://infolab.stanford.edu/ ullman/mmds/ch3.pdf*. 1992, ch. 3.

[63] Vassilvitskii, S. *Dealing with Massive Data: http://www .cs. columbia .edu/ coms 699812/lecture1.pdf*. 2011, pp. 1–30.

[64] Wang, G., Xie, S., Liu, B., and Philip, S. Y. Review graph based online store review spammer detection. In *Data mining (icdm), 2011 ieee 11th international conference on* (2011), IEEE, pp. 1242–1247.

[65] WordNet. Wordnet with ntlk : https://pythonprogramming.net/wordnet-nltk-tutorial/.

[66] Xie, S., Wang, G., Lin, S., and Yu, P. S. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012), ACM, pp. 823–831.

[67] Xu, C., Zhang, J., Chang, K., and Long, C. Uncovering collusive spammers in chinese review websites. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management* (2013), ACM, pp. 979–988.

[68] Ye, J., and Akoglu, L. Discovering opinion spammer groups by network footprints. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2015), Springer, pp. 267–282.

[69] Yelp. The yelp review filter – how it works and how to get your legitimate reviews through : https://vivial.net/blog/how-to-avoid-the-yelp-review-filter-and-get-more-positive-reviews/.